

Dynamic Token-Pass Transformers for Semantic Segmentation

Yuang Liu^{1*}, Qiang Zhou², Jing Wang², Zhibin Wang², Fan Wang², Jun Wang¹, Wei Zhang^{1†}

¹East China Normal University ²DAMO Academy, Alibaba Group

{frankliu624, zhangwei.thu2011, wongjun}@gmail.com

{jianchong.zq, yunfei.wj, zhibin.waz, fan.w}@alibaba-inc.com

Abstract

Vision transformers (ViT) usually extract features via forwarding all the tokens in the self-attention layers from top to toe. In this paper, we introduce dynamic token-pass vision transformers (DoViT) for semantic segmentation, which can adaptively reduce the inference cost for images with different complexity. DoViT gradually stops partial easy tokens from self-attention calculation and keeps the hard tokens forwarding until meeting the stopping criteria. We employ lightweight auxiliary heads to make the token-pass decision and divide the tokens into keeping/stopping parts. With a token separate calculation, the self-attention layers are speeded up with sparse tokens and still work friendly with hardware. A token reconstruction module is built to collect and reset the grouped tokens to their original position in the sequence, which is necessary to predict correct semantic masks. We conduct extensive experiments on two common semantic segmentation tasks, and demonstrate that our method greatly reduces about 40% ~ 60% FLOPs and the drop of mIoU is within 0.8% for various segmentation transformers. The throughput and inference speed of ViT-L/B are increased to more than 2× on Cityscapes. Code is available at <https://github.com/FLHonker/DoViT-code>.

1. Introduction

Semantic segmentation has been a significant component of autonomous driving [17], image editing [41] and visual scene analysis [6]. As a dense prediction task, it aims to assign each image pixel to a category label. Thanks to the development of deep neural networks, especially vision transformers (ViT) [9], the research for semantic segmentation has achieved great successes at the price of huge computation. Moreover, the transformer-like segmentor,

*This work was done when Yuang Liu was a Research Intern at DAMO Academy, Alibaba. This work was supported by DAMO Academy through DAMO Academy Research Intern Program, and National Natural Science Foundation of China under Grants (No. 62272165).

†Corresponding author.

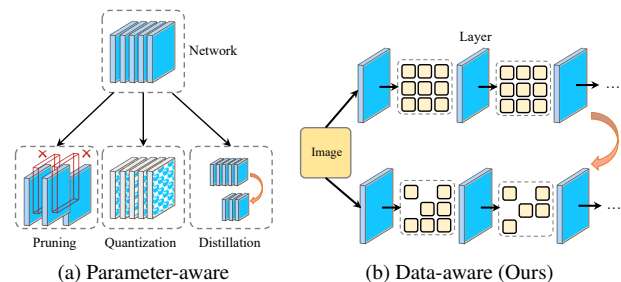


Figure 1. Overview of different algorithms for model acceleration.

e.g., SETR [44], Segmenter [30] and Segformer [34], has overtaken CNN across the board and shows great potential. However, the computational complexity of the transformer architecture makes real-time application of semantic segmentation even more prohibitive. To make these models more suitable to resource-constrained mobile devices, it is urgent to reduce the computation cost and accelerate them.

These years have witnessed the great progress in CNN-type model compression and acceleration brought by parameter-aware approaches. As shown in Figure 1a, the majority of current acceleration approaches can be divided into three categories, including pruning [18, 20], quantization [16, 32] and knowledge distillation [13, 15, 33]. They all focus on reducing the redundant components or parameters of the networks, so we called them parameter-aware acceleration methods. There has been a recent surge of interest to introduce these parameter-aware acceleration methods to transformer-base architectures, both in natural language processing (NLP) [21, 29] and computer vision (CV) [22, 39].

The core of ViTs is the self-attention module, which is naturally different from the convolution operation in CNNs. It works by calculating the relationships among each pair of image patches or tokens, and then capturing the global context of the input image. Benefiting from this nature of self-attention, Rao *et al.* firstly propose DynamicViT [27] that prunes the tokens of less importance and only keeps partial tokens in self-attention for acceleration. A-ViT [35] improves DynamicViT [27] by introducing halting distribution of tokens and requires no extra parameters. Recently,

ATS [10] builds an adaptive token sampler to automatically select the most important tokens. These data-aware acceleration works point out a new direction for model acceleration. But they only pay attention to classification tasks, and do not support dense tasks like semantic segmentation. For classification ViT, actually only the class token is utilized to predict the category of the whole image, and most of the other tokens may be dropped at certain layers. Differently, each token is necessary to semantic segmentation, and all the tokens are required to be utilized by the decoder to predict the categories of pixels. Additionally, the above methods train the dynamic ViT with a fixed token pruning rate for each layer/block, making it unable to make an image-wise trade-off between the input complexity and inference cost.

To this end, we propose a novel dynamic token-pass transformers (DoViT) for semantic segmentation. This is the first attempt of data-aware ViT acceleration on dense prediction tasks. Rather than focusing on the patch redundancy in the input image, we base complexity or learning difficulty of semantic patches/tokens to adaptively determine their computational cost, which makes the backbone achieve an image-wise dynamic inference. It’s difficult to automatically select the more informative or important tokens for self-attention at each layer. Because the pixel classification may fail if any semantic token is not fully learned. The most reliable scheme is to decide whether a token should be preserved or stopped explicitly based on the early prediction results. Therefore, we introduce a semantic early-probe scheme that divides the tokens into two categories, *i.e.*, keeping set and stopping set. The keeping tokens will involve in the next self-attention layers while the stopping tokens will be prevented from the self-attention and directly passed to the decoder via a short path. The keep or stop of each token is determined by the semantic prediction confidence provided by the auxiliary heads. This scheme makes it possible to adjust a fully dynamic inference cost for various input images. We claim that all semantic tokens must be received by the decoder in their original position after a certain level of self-attention operations. A separate self-attention module is presented to optimize the calculation of the keeping/stopping set of tokens where the stopping tokens bring no computation. To restore the order of input tokens, we introduce a token reconstruction module that provides complete and correct feature maps for the decoder and auxiliary heads. Our proposed approach significantly cuts down the inference cost — the FLOPs of SETR on Cityscapes is reduced by 40% ~ 60% within 0.8% mIoU drop, and the throughput and FPS is improved to over $2 \times$ on hardware. In summary, our main contributions are as follows:

- We propose a dynamic token-pass method to reduce the inference cost of vision transformers for semantic segmentation.

- We introduce a semantic early-probe scheme to determine the token-pass candidates. The separate self-attention and token reconstruction modules are responsible for sparse token acceleration.
- We conduct extensive experiments on two public segmentation datasets with various ViT models and demonstrate that the proposed method reduces FLOPs significantly with minor drop in mIoU.

2. Related Work

2.1. Semantic Segmentation

Fast development of deep neural network has significantly inspired the exploration of semantic image segmentation. At the very first, FCN [23] achieves pixel-wise image segmentation by removing the final fully-connected layer. As FCN focuses on extracting abstract semantic features, multi-scale feature fusing [1, 25, 28], dilated convolution [2, 3, 38], and spatial pyramid pooling [4, 5, 42] are proposed to induce the network to extract more fine-grained features. To further boost the accuracy of the network, attention mechanisms are introduced to the network [11, 12, 36, 37, 40, 43]. Recently, considering the excellent performance of transformers, many works try to plug it into the semantic segmentation. SETR [44] firstly adopts a transformer based network, ViT, as encoder to extract features, but keep the CNN-based decoder. The Segmenter [30] further expands the transformer architecture to the decoder and designs a pure transformer encoder-decoder segmentation architecture.

2.2. Model Acceleration

Even transformer has brought great improvement for the semantic segmentation, the quadratic number of interactions between tokens increased the computation burdens. To promote the deployment of transformer-based model on edge devices, model acceleration become a popular topic. Traditional parameter-aware model acceleration methods like knowledge distillation [13, 15, 33], quantization [16, 32], and pruning [14, 18, 20] have already been introduced to transformer [21, 22, 24, 29, 39]. Considering the cost of calculating relations among image patches or tokens in transformer, data-aware model acceleration is worth being discussed. Some related works have been studied in image classification, for example, DynamicViT [27] firstly proposes to prune uninformative tokens in a dynamic way by adopting a lightweight prediction module. Then A-ViT [35] improves it by removing the prediction module and halting the computation of tokens by a parameter-free adaptively inference mechanism. EViT [19] reduces computation by progressively discarding or fusing inattentive tokens in Vision Transformers. Recently, ATS [10] builds an adaptive token sampler to automatically select the most important tokens. These methods exhibit great potential in transformer acceleration of classification

tasks, but they are not suitable for the semantic segmentation as each token is meaningful for the pixel-wise prediction.

3. Proposed Approach

3.1. Overview

Figure 2 illustrates the pipeline of our DoViT framework. Consider a transformer-like segmentation network that takes an image $x \in \mathbb{R}^{3 \times H \times W}$ as input to predict a semantic mask $y \in \mathbb{R}^{C \times H \times W}$ (C , H , and W represent number of categories, height, and width respectively):

$$y = \mathcal{D} \odot \mathcal{F}^L \odot \mathcal{F}^{L-1} \odot \mathcal{F}^1 \odot \mathcal{E}(x), \quad (1)$$

where \mathcal{E} , \mathcal{F} and \mathcal{D} are the patch embedding network, backbone layer and segmentation decoder, respectively, L represents the number of layers. The network \mathcal{E} tokenizes the image patches from x into positioned tokens $T \in \mathbb{R}^{N \times E}$, where N and E are the number and dimension of the tokens, respectively. Then the tokens forward with multi-head self-attention (MSA) operations in the backbone layers, which accounts for most of the computation of the entire segmentation network. We select certain layers and execute a semantic early-probe scheme to make a token-pass decision that the hard tokens are kept in the later MSA layers while the easy tokens are stopped from MSA and directly passed to the decoder. To perform an efficient sparse token forwarding in the backbone and decoder network, we introduce a separate self-attention strategy and a token reconstruction module.

3.2. Token-Pass Decision

To progressively lessen the tokens in the ViT backbone, we choose D nonadjacent self-attention layers as ‘‘decision layers’’ that divide the whole L backbone layers into $D + 1$ blocks, *i.e.*, $\{\mathcal{B}^1, \mathcal{B}^2, \dots, \mathcal{B}^{D+1}\}$. The number of layers in the block \mathcal{B}^ℓ is B_ℓ . Different from classification ViT, it’s a challenge to determine the importance of each token at certain layers. Because every token contains specific semantic information and contributes to the segmentation prediction, which indicates that it’s unreasonable to roughly drop the uninformative tokens. We claim that it is crucial to accurately judge whether a token is fully utilized and learned. A straightforward approach is to evaluate the segmentation results with the early tokens. To this end, we propose an early-probe scheme to determine whether a token is kept in calculating or stopped learning. In particular, we insert a lightweight auxiliary segmentation head \mathcal{H}^ℓ following the block \mathcal{B}^ℓ . It’s worth noting that the auxiliary head consisting of one fully-convolutional layer brings tiny extra parameters and computation, which is almost negligible comparing to the whole segmentation network. Assuming that no token-pass decision is applied, and each MSA block works normally with total N tokens input and N tokens output. We reshape the token sequence T^ℓ output by the block \mathcal{B}^ℓ

to a deep feature map $\mathbf{f} \in \mathbb{R}^{E \times h \times w}$, where E is both the dimension of the tokens and channels of the feature map, h and w represents the width and height of the feature map, $N = hw$. We feed the feature map \mathbf{f}^ℓ into the ℓ -th auxiliary head \mathcal{H}^ℓ to obtain a probability map $p^\ell \in \mathbb{R}^{C \times h \times w}$ with the softmax operation:

$$p^\ell = \text{softmax}(\mathcal{H}^\ell(\mathbf{f}^\ell)). \quad (2)$$

The scalar $p_{c,i,j}$ represents the probability of the token $T_{i,j}$ belonging to the c -th semantic category, and the maximum value $q_{i,j}$ of $\{p_{c,i,j}\}_{c=1}^C$ represents the confidence of label prediction in terms of $T_{i,j}$. With these insights, we can allocate a prediction confidence to each token $T_{i,j}$ and obtain a score map

$$q_{i,j}^\ell = \max\{p_{c,i,j}^\ell | c \in \{1, 2, \dots, C\}\}. \quad (3)$$

Generally, the tokens with high prediction confidence are uninformative and easy to segment, while the tokens with low prediction confidence are complex and hard to learn. So we can utilize the confidence score to determine a token’s pass. To align with the original shape of token sequence $T \in \mathbb{R}^{N \times E}$, we reshape the confidence map $q^\ell \in \mathbb{R}^{h \times w}$ to $q^\ell \in \mathbb{R}^N$. We define a binary decision mask $M^\ell \in \{0, 1\}^N$ to indicate whether to keep or stop each token T_n^ℓ at the ℓ -th block,

$$\tilde{M}_n^\ell = \begin{cases} 0, & \text{if } q_n^\ell > \xi \\ 1, & \text{else} \end{cases}, \quad (4)$$

where $0 \leq \xi \leq 1$ is a threshold parameter. The number of sparsely-keeping tokens is gradually decreased block-by-block. But the decision mask \tilde{M}^ℓ is calculated based on the hypothesis that the head \mathcal{H}^ℓ receives a complete feature map without token reduction. So we need to ignore the tokens that are stopped at the previous blocks by updating \tilde{M}^ℓ with

$$M^\ell = \tilde{M}^\ell \odot M^{\ell-1} = \prod_{i=0}^{\ell} \tilde{M}^i, \quad (5)$$

in which \odot is the Hadamard product, and $M^{\ell-1}$ is the real decision mask from the last block, $M^0 = \tilde{M}^0 = \mathbf{1}$.

In this way, we have selected the keeping tokens at each block, they will involve in the MSA blocks until meeting the stopping criteria. With the early-probe scheme, the token-pass is dynamic adaptively in terms of the patch complexity, rather than subject to a fixed keeping/stopping ratio at each stage. It’s necessary to collect the stopping tokens to build a complete feature map for the auxiliary heads and segmentation decoder. To achieve an efficient self-attention with the keeping/stopping tokens, we design a separate token forwarding algorithm.

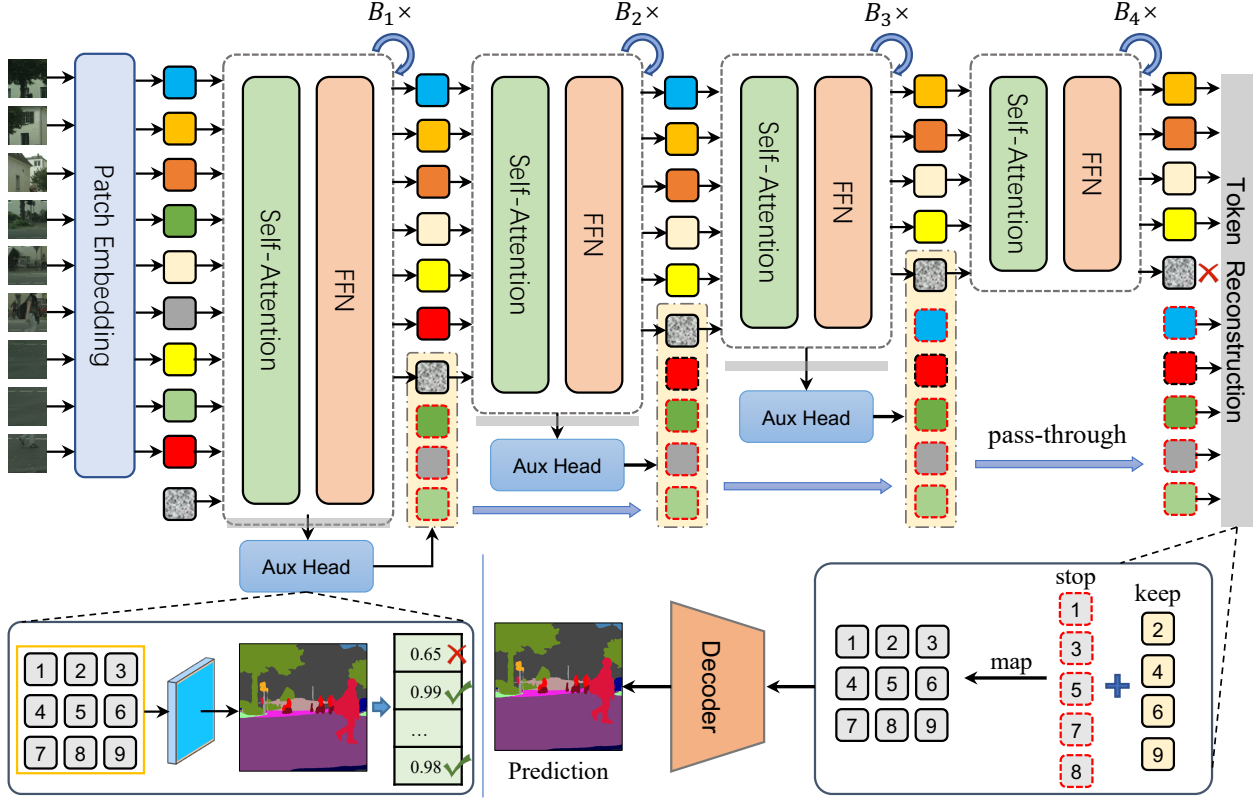


Figure 2. The overview of our dynamic token-pass transformers for semantic segmentation. The token-pass decision depends on the auxiliary (Aux) heads inserted between the transformer blocks. With the keeping and stopping token sets, the separate self-attention module can capture the more informative context with less computation, which represents the core of our dynamic token-pass algorithm. The token reconstruction module is responsible for converting the sparse tokens sets to a structured feature map with ordered tokens.

3.3. Sparse Token Forwarding

With the decision masks $\{\tilde{M}^\ell\}_\ell^D$, a sparse token sequence can be extracted from the blocks. [27] and [35] both execute the self-attention operation with the sparse tokens via a mask mechanism, where the exiting/stopping tokens still involve in the calculations of query, key and value. To achieve a real token reduction, they design different training and inference phases in which the stopping tokens are inconsistent. The inconsistent tokens input to the decoder could make a bias between the training and inference phases. To this end, we propose a simple yet efficient separate self-attention module that keeps the consistency of dynamic token-pass in two phases.

Separate Self-Attention. The keeping and stopping tokens are processed separately in each block, except the first block \mathcal{B}^1 in which no token reduction is performed. To gather the sparse tokens to a compact and structured representation, we define a function $\mathcal{G}(T, M)$ that selects tokens from $T \in \mathbb{R}^{N \times E}$ with the mask M and combine them to a new token sequence with the size of $|M| \times E$, where $|M|$ is the number of nonzero items. The keeping/stopping token

sequence $\hat{T}^\ell / \check{T}^\ell$ up to the ℓ -th block can be obtained by

$$\hat{T}^\ell = \mathcal{G}(T^\ell, M^\ell), \check{T}^\ell = \mathcal{G}(T^\ell, (1 - M^\ell)). \quad (6)$$

In this way, the keeping/stopping tokens output by the block \mathcal{B}^ℓ has been divided into two parts. In dynamic token-pass inference, only the keeping tokens are considered in the MSA modules of the next block, *i.e.* $\mathcal{B}^{\ell+1}$, which can be formulated as

$$\text{MSA}(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}) = \text{softmax}\left(\frac{\hat{\mathbf{Q}}\hat{\mathbf{K}}^\top}{\sqrt{d}}\right)\hat{\mathbf{V}}, \quad (7)$$

where $\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}} \in \mathbb{R}^{N \times d}$ are the query, key and value embeddings of the keep tokens \hat{T} , d is the dimension of the embeddings. Then, the computational complexity of self-attention is reduced from $\mathcal{O}(N^2 \cdot d)$ to $\mathcal{O}(|M|^2 \cdot d)$, $|M| \leq N$. The stopping tokens \check{T}^ℓ will directly pass the next block $\mathcal{B}^{\ell+1}$ without any calculation. After separate self-attention in block $\mathcal{B}^{\ell+1}$, the keeping and stopping tokens can be combined as a complete sequence $\hat{T}^{\ell+1}$.

Token Reconstruction. However, due to the token sparsification and separation for self-attention, the combined token sequence $\hat{T}^{\ell+1}$ output by block $\mathcal{B}^{\ell+1}$ is out-of-order

and inconsistent with the original image patches. To this end, we introduce a token reconstruction module to locate the tokens in $\hat{T}^{\ell+1}$ to their original position as a new sequence. It can be formulated as $T^{\ell+1} = \mathcal{R}(\hat{T}^{\ell+1}, I^{\ell+1})$, in which \mathcal{R} is a transform function that maps each token to the corresponding position or rank in the sequence, $I^{\ell+1}$ is the map of token indices that can be generated incidentally by \mathcal{G} . Finally, the reconstructed tokens $T^{\ell+1}$ can be fed into the next block. After reshaped as a feature map $f^{\ell+1}$, the auxiliary head $\mathcal{H}^{\ell+1}$ can utilize it to make token-pass decision for the next block, and the decoder can finish the semantic prediction with it. As shown in Figure 2, we add a token reconstruction module before the auxiliary heads and segmentation decoder.

Token Merging. Considering that there may be some useful information provided by the stopping tokens, we merge them as one representative token and aggregate it with the class token before calculating self-attention in the next block. For example, after obtaining a token sequence T^ℓ from the block \mathcal{B}^ℓ via the above algorithms, the class token can be updated by

$$T_0^\ell \leftarrow \frac{1}{2} \left(T_0^\ell + \frac{1}{|S^\ell|} \sum_i^{|S^\ell|} \sum_j^E \hat{T}_{i,j}^\ell \right), \quad (8)$$

where $|S^\ell| = |\mathbf{1} - M^\ell|$ is the total number of stopping tokens. Then the keeping tokens including T_0^ℓ can sequentially involve the self-attention in the next block. Note that we preserve the class token in MSA following the standard in ViT [9], and remove it when building feature maps. In fact, the class token T_0 is a default keeping token in all the blocks, not depends on the token-pass decision.

3.4. Training Pipeline

With the feature maps $\{f^\ell\}_{\ell=1}^{D+1}$ consisting of hierarchically-stopping tokens, the auxiliary heads $\{\mathcal{H}^\ell\}_{\ell=1}^D$ and segmentation decoder \mathcal{D} predict semantic probability maps $\{p^\ell\}_{\ell=1}^D$ and p^s . To train the segmentation network in a supervised manner, the ground truth label map \bar{y} is used to compute the cross-entropy (CE) loss

$$\mathcal{L}_{CE}(p^s, \bar{y}) = \frac{1}{HW} \sum_{i=1}^{HW} \sum_{j=1}^C -\bar{y}_{i,j} \log(p_{i,j}^s), \quad (9)$$

where $\bar{y}_{i,j}$ is the real value (1 or 0) of the j -th class for the i -th pixel, and $p_{i,j}^s$ corresponds to the probability predicted by the segmentation decoder \mathcal{D} . Analogously, the D auxiliary heads are updated with loss function

$$\mathcal{L}_{AH} = \sum_{\ell}^D \mathcal{L}_{CE}(\mathcal{U}(p^\ell), \bar{y}), \quad (10)$$

where $\mathcal{U}(\cdot)$ is the upsampling function.

To alleviate the performance damage caused by dynamic token-pass, we employ a self-distillation framework to train the DoViT-based segmentation network with the corresponding ViT-based network as a teacher. We denote the teacher’s probability map as p^t , then the self-distillation loss is formulated by Kullback-Leibler(KL) divergence:

$$\mathcal{L}_{SD}(p^s, p^t) = \frac{1}{HW} \sum_{i=1}^{HW} \sum_j^C p_{i,j}^s \log \left(\frac{p_{i,j}^s}{p_{i,j}^t} \right). \quad (11)$$

The overall loss function is

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha \mathcal{L}_{AH} + \beta \mathcal{L}_{SD}, \quad (12)$$

where $\alpha, \beta > 0$ are two hyper-parameters to control the relative importance.

4. Experiments

4.1. Experimental Settings

4.1.1 Datasets and Metrics

Cityscapes [8] is a widely used urban scene understanding dataset, with 19 common classes for evaluation. It contains 2,975 fine-annotated images with 1024×2048 pixels for training, 500 for validation, and 1,525 for testing.

ADE20K [45] is one of the most challenging semantic segmentation datasets that contains challenging scenes with 20,210 fine-annotated images with 150 semantic classes in the training set. The validation and test set contain 2,000 and 3,352 images respectively.

Metrics. The numbers of float-point operations (FLOPs) and parameters (Params) are introduced to measure the computational complexity and model size of the segmentation network. We report the throughput and frame-per-second (FPS) to show the inference speed of networks. We employ the common metric of mean Intersection over Union (mIoU), Pixel Accuracy (PA), and mean Pixel Accuracy (mPA) for scene segmentation on all datasets. Note that mIoU is the primary and more persuasive metric for segmentation.

4.1.2 Implementation Details

We implement our DoViT framework with PyTorch [26] on 8 NVIDIA V100 GPUs. We evaluate the proposed method on two popular transformer-like segmentation architectures, *i.e.*, SETR [44] and Segformer [30], using classic ViT [9] and DeiT [31] as backbone. For the base and small transformer backbone, we select the (3,6,9)-th layers as the decision layers. In the total 24 self-attention layers of the ViT-large, the (6,12,18)-th layers are the decision layers. Three single-layer auxiliary heads with 1×1 kernel are inserted following the decision layers. And the original auxiliary heads for SETR [44] can be reused without specific

Network	Backbone	PA (%)	mPA (%)	mIoU (%)	Params (M) ↓	FLOPs (G) ↓
SETR	ViT-L	95.91	85.71	78.10	305.74	2484.27
	DoViT-L (Ours)	95.93	85.27	77.98 (-0.22)	306.54 (+0.8)	1088.80 (-56%)
	ViT-B	95.65	84.22	76.59	87.62	703.28
	DyViT-B/0.9	94.99	79.10	71.60 (-4.99)	91.12 (+3.5)	626.79 (-11%)
	DyViT-B/0.85	94.82	77.44	70.11 (-6.48)	91.12 (+3.5)	581.84 (-17%)
	DoViT-B (Ours)	95.64	83.95	76.40 (-0.19)	88.23 (+0.6)	330.09 (-53%)
	ViT-S	95.39	83.14	74.80	22.57	177.99
	DyViT-S/0.9	94.10	77.52	67.47 (-7.33)	23.62 (+1.1)	158.87 (-11%)
	DoViT-S (Ours)	95.52	83.26	75.13 (+0.33)	22.89 (+0.3)	91.13 (-49%)
SETR	DeiT-B	95.78	85.25	77.41	87.62	703.28
	DoDeiT-B (Ours)	95.62	84.33	76.70 (-0.71)	88.23 (+0.6)	332.73 (-53%)
	DeiT-S	95.47	82.69	75.14	22.57	177.99
	DoDeiT-S (Ours)	95.29	82.58	74.63 (-0.51)	22.89 (+0.3)	83.64 (-53%)
Segmenter	ViT-L	96.07	86.41	79.10	333.82	2705.40
	DoViT-L (Ours)	95.88	85.60	78.47 (-0.63)	334.62 (+0.8)	1257.86 (-54%)
	ViT-B	95.89	85.55	77.83	103.38	826.99
	DoViT-B (Ours)	95.73	84.57	77.40 (-0.43)	103.99 (+0.6)	442.19 (-47%)
	ViT-S	95.68	84.22	76.61	26.47	208.05
	DoViT-S (Ours)	95.65	84.11	76.65 (+0.04)	26.79 (+0.3)	119.50 (-43%)

Table 1. Main results on Cityscapes. Performance comparison of different segmentation models with varying backbones on Cityscapes validation set. The “DoViT” and “DoDeiT” are standard ViT/DeiT backbone with our acceleration method. The suffix “L/B/S” represent the large/base/small transformer, respectively.

modification. All the setups of data augmentation, network training and accuracy evaluation follow the official implementation of SETR [44] and Segmenter [30] in codebase *MMSegmentation* [7]. The co-efficient α and β are set to 1.0 and 0.4 by default, respectively. The confidence threshold $\xi = 0.985$ is optimal for Cityscapes, and $\xi \in [0.96, 0.985]$ is suitable for ADE20K. Without specific instruction, the parameters, FLOPs, throughput and FPS are reported with a 1024×2048 resolution for Cityscapes, and 512×512 randomly cropped images for ADE20K. To test the adaptive inference cost of each image in our DoViT, we randomly sample 100 images from the validation set and report their average FLOPs, throughput and FPS. For parallel training of images with various numbers of sparse keeping tokens, we utilize a distributed environment, where the batch size per GPU is set to 1 for Cityscapes and 2 for ADE20K. If the batch size per GPU is larger than 1, the numbers of keeping tokens of batch images on one GPU are set the same, by striking an average according to the sort of confidence.

4.2. Main Results

Cityscapes. One of the most advantages of our DoViT framework is that it can reduce the computational complexity (*i.e.*, FLOPs) of a wide range of transformer-like segmentation networks with a tiny drop of accuracy. Table 1 summarizes the performance and computation comparison between our framework and various state-of-the-art segmentation models. We mainly highlight the mIoU drop and FLOPs

reduction rate in the brackets. With our method, the FLOPs of networks are reduced by 40% ~ 60%, with less than 0.8% mIoU loss. Especially, the mIoU is improved a little rather than reduced for some networks with the DoViT-S backbone, benefiting from the dynamic and sparse token pass. Moreover, we extend the DynamicViT [27] backbone to segmentation architectures, abbreviated as “DyViT/ ρ ” ($0 \leq \rho \leq 1$ is the token ratio) in the table. The implicit token exiting strategy with regularization of keeping ratio, is insufficient for semantic segmentation models, in which the complex context confuses the token selection. As we can see that when the FLOPs are reduced by less than 20%, the SETR drops more than 5% mIoU. We also present the incremental parameters of DoViT and DyViT, relative to the standard ViT. To align with the embedding dimensions of larger transformers, the auxiliary heads with larger input dimensions introduce more extra parameters. But the extra parameters can be negligible comparing to the backbone itself.

ADE20K. To evaluate the effectiveness and efficiency of our approach, we conduct extensive experiments on the ADE20K dataset, as shown in Table 2. We adopt various confidence threshold $\xi \in [0.96, 0.985]$ and compare the trade-offs between the mIoU performance and computational reduction. For SETR with ViT-base, our method can reduce 30% FLOPs without mIoU drop. The ADE20K dataset is very challenging due to the large-scale semantic categories and complex scenes, making the models predict with lower confidence for numerous pixels. Even though leveraging

Network	Backbone	ξ	mIoU (%)	FLOPs (G) ↓
SETR	ViT-B	–	46.37	88.54
	DoViT-B	0.985	46.54 (+0.17)	66.06 (-25%)
	DoViT-B	0.98	46.41 (+0.04)	63.16 (-29%)
	DoViT-B	0.96	45.74 (-0.63)	61.07 (-31%)
	ViT-S	–	42.81	22.82
	DoViT-S	0.985	42.56 (-0.25)	19.85 (-13%)
SETR	DoViT-S	0.98	42.33 (-0.47)	19.53 (-14%)
	DoViT-S	0.96	42.26 (-0.55)	18.15 (-20%)
	DoViT-S	0.96	42.26 (-0.55)	18.15 (-20%)
SETR	DeiT-B	–	43.67	88.54
	DoDeiT-B	0.96	43.24 (-0.43)	60.18 (-32%)
Segmenter	ViT-S	–	46.19	26.57
	DoViT-S	0.96	45.84 (-0.35)	21.75 (-18%)

Table 2. Main results on ADE20K. We apply our method on SETR and Segmenter with different backbones. The mIoU performance and FLOPs reduction are reported on the validation set, when utilizing different thresholds ξ .

a lower threshold can early stop more tokens and reduce more inference cost, it’s difficult to reduce the FLOPs up to 20%, especially for the small networks, *e.g.*, DoViT-S, DoDeiT-S. Generally, the smaller networks achieve a less ratio of FLOPs reduction, caused by the lower confidence at early-probe.

Acceleration Effort. In Figure 3, we compare speedup on one NVIDIA V100 GPU in terms of SETR (ViT-B/DoViT-B) on two datasets respectively. It’s worth noting that limited to the super-resolution, *i.e.*, 1024×2048 pixels, we utilize 512×512 randomly cropped inputs to evaluate the throughput on Cityscapes. Figure 3a illustrates that both the throughput and FPS of the large and base models are improved by over $2\times$, without requiring hardware/library modification. Meanwhile, our method improves the throughput and FPS of ViT-large ($\xi = 0.98$) variants by 27% and 30% on ADE20K (Figure 3b). It is a pity that the throughput of ViT-small can be improved by 23% on ADE20K, but the FPS decreases a little due to the extra computation of auxiliary heads.

4.3. Ablation Study

Effects of different components. To verify the effectiveness of each component in our framework, we conduct

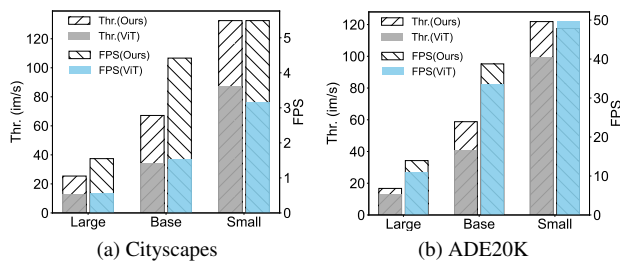


Figure 3. The throughput (Thr.) and FPS improvement of SETR-DoViT-B on Cityscapes

DoViT	Token Merging	Self-Distillation	mIoU
✓			75.37
✓	✓		75.66
✓	✓	✓	76.40

Table 3. Effects of different components in our framework. We provide the results after removing the self-distillation and token merging in terms of SETR-DoViT-B on Cityscapes.

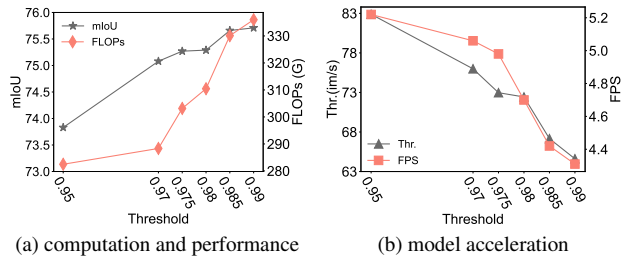


Figure 4. Impact of the confidence threshold ξ to computation (a) and acceleration (b).

ablation analysis on Cityscapes with SETR-DoViT-B, and present the results in Table 3. With the token merging strategy, the mIoU of the DoViT is improved by about 0.3%. With the pixel-wise self-distillation, the gap between the DoViT and ViT segmentation networks can be reduced to 0.2% mIoU.

Impact of the threshold. To investigate the impact of the confidence threshold ξ , we train SETR-DoViT-B on Cityscapes without self-distillation. Figure 4a depicts the trade-off between the performance and computation, varying the threshold from 0.95 to 0.99. It is obvious that with the increase of threshold, mIoU and FLOPs will increase, which is reasonable — more computation brings better performance. In order to balance the computation and performance loss, it is suitable to set $\xi \in [0.985, 0.99]$ for Cityscapes. Thanks to the numerous easy-to-learn patches of cityscapes, the FLOPs can be reduced by 50% when $\xi = 0.99$. In addition, we plot the line of inference speed (throughput and FPS) with threshold, as shown in Figure 4b. Varying the threshold from 0.95 to 0.99, the throughput (Thr.) can be improved from 34.1 (as shown in Figure 3a) to a range of [63, 83], and the inference can be speeded up by at least $2.9\times$, *i.e.*, from 1.52 FPS to over 4.42 FPS.

4.4. Visualization

In Figure 5, we show the qualitative results of two cityscape images. We find that applying DoViT results in progressive reduction of keeping tokens/patches when forwarding block-by-block. Meanwhile, the easy-to-learn patches, such as that cover road, sky and tree, are stopped from self-attention early, while the hard patches consisting of complex

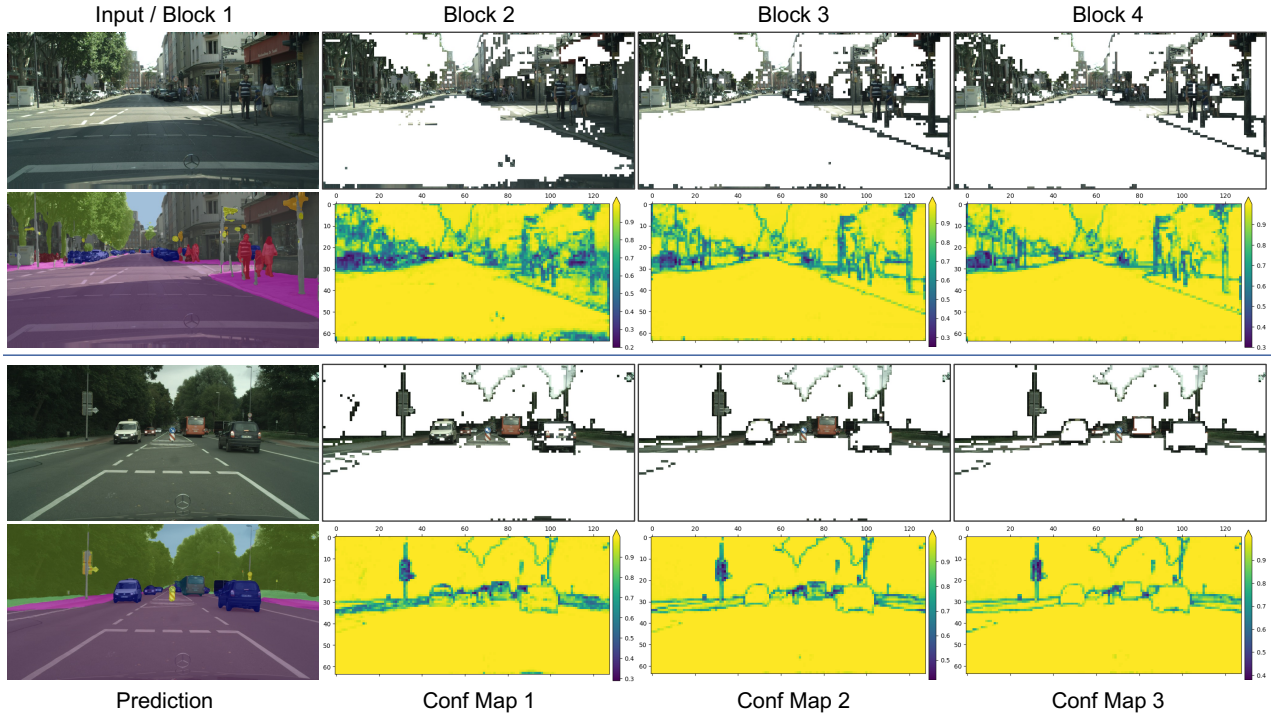


Figure 5. Visualizations of keeping tokens and segmentation results of two images from Cityscapes. The first and third rows depict the corresponding patches of the keeping tokens input to each block, where the white region is corresponding to the stopping tokens. The second and fourth rows show the final prediction results and confidence score map at each block.

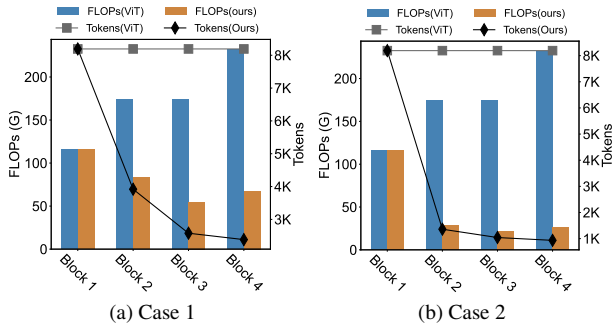


Figure 6. Case study of two images in Cityscapes. We report the number of keeping tokens and FLOPs at each transformer block, with a 1024×2048 input to SETR with ViT-B/DoViT-B. The total number of tokens is 8,192 as the patch size is set to 16.

context, will be kept until the end of the vision transformers. Additionally, we visualize the confidence score maps predicted by the three auxiliary heads. The tokens/patches with higher scores (closer to yellow) are supposed to be removed from the calculation. What’s more interesting is that, though the tokens covering easy patches are removed, some more informative edge parts are preserved, *e.g.*, the outline of the trees and cars. Figure 6 demonstrates the corresponding quantitative information of the two inference cases. The

number of tokens involving in the four ViT blocks are all 8,192. In case 1 (a), over half of the tokens are removed at the second block, and the FLOPs per block also drops accordingly. In case 2 (b), over 80% tokens are stopped at the second block, and only 10% tokens are kept at the last block. These results reflect the efficiency and interpretability of our dynamic token-pass method. The early-probe scheme determines token-pass adaptively, rather than forcibly stopping a fixed ratio of tokens for all images. Thus, the inference costs of different images could be very different.

5. Conclusion

In this work, we explore the segmentation transformer acceleration from a perspective of data-redundancy. We have introduced dynamic token-pass transformers (DoViT) to adaptively adjust the inference cost based on input complexity. DoViT gradually reduces the number of tokens passing self-attention layer and shorts the hierarchical-stopped tokens into a unified decoder. We evaluate the effectiveness of our approach in computation reduction and inference speedup, and discuss some meaningful issues. In the future, we plan to combine our data-aware transformer acceleration method with the parameter-aware model compression approaches and extend it to other dense prediction tasks.

References

- [1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 5, 2015. 2
- [2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 2
- [3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017. 2
- [4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2
- [5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 2
- [6] Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Un-supervised monocular depth estimation with semantic-aware representation. In *CVPR*, pages 2624–2632, 2019. 1
- [7] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. <https://github.com/open-mmlab/mms Segmentation>, 2020. 6
- [8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 5
- [9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 5
- [10] Mohsen Fayyaz, Soroush Abbasi Kouhpayegani, Farnoush Rezaei Jafari, Eric Sommerlade, Hamid Reza Vaezi Joze, Hamed Pirsiavash, and Juergen Gall. ATS: Adaptive token sampling for efficient vision transformers. In *ECCV*, 2022. 2
- [11] Jun Fu, Jing Liu, Jie Jiang, Yong Li, Yongjun Bao, and Hanqing Lu. Scene segmentation with dual relation-aware attention network. *IEEE TNNLS*, 32(6):2547–2560, 2020. 2
- [12] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 2
- [13] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *IJCV*, 129(6):1789–1819, 2021. 1, 2
- [14] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. In *IJCAI*, 2018. 2
- [15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2
- [16] Qing Jin, Linjie Yang, and Zhenyu Liao. Adabits: Neural network quantization with adaptive bit-widths. In *CVPR*, pages 2146–2156, 2020. 1, 2
- [17] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *IEEE intelligent vehicles symposium*, pages 163–168, 2011. 1
- [18] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *ICLR*, 2017. 1, 2
- [19] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *ICLR*, 2022. 2
- [20] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. HRank: Filter pruning using high-rank feature map. In *CVPR*, pages 1529–1538, 2020. 1, 2
- [21] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Haotang Deng, and Ju Qi. FastBERT: a self-distilling bert with adaptive inference time. In *ACL*, 2020. 1, 2
- [22] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. In *NeurIPS*, volume 34, pages 28092–28103, 2021. 1, 2
- [23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2
- [24] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. AdaViT: Adaptive vision transformers for efficient image recognition. In *CVPR*, pages 12309–12318, 2022. 2
- [25] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, pages 1520–1528, 2015. 2
- [26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5
- [27] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, volume 34, pages 13937–13949, 2021. 1, 2, 4, 6
- [28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 2
- [29] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-BERT: Hessian based ultra low precision quantization of bert. In *AAAI*, volume 34, pages 8815–8821, 2020. 1, 2

- [30] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7262–7272, 2021. 1, 2, 5, 6
- [31] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 5
- [32] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: Hardware-aware automated quantization with mixed precision. In *CVPR*, pages 8612–8620, 2019. 1, 2
- [33] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *arXiv preprint arXiv:2004.05937*, 2020. 1, 2
- [34] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, volume 34, pages 12077–12090, 2021. 1
- [35] 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. 1, 2, 4
- [36] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *ECCV*, pages 191–207. Springer, 2020. 2
- [37] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *CVPR*, pages 12416–12425, 2020. 2
- [38] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2
- [39] Hao Yu and Jianxin Wu. A unified pruning framework for vision transformers. *arXiv preprint arXiv:2111.15127*, 2021. 1, 2
- [40] Yuhui Yuan, Lang Huang, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 2
- [41] Jianfu Zhang, Peiming Yang, Wentao Wang, Yan Hong, and Liqing Zhang. Image editing via segmentation guided self-attention network. *IEEE Signal Processing Letters*, 27:1605–1609, 2020. 1
- [42] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 2
- [43] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, pages 267–283, 2018. 2
- [44] 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 *CVPR*, pages 6881–6890, 2021. 1, 2, 5, 6
- [45] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 5