STEP: SuperToken and Early-Pruning for Efficient Semantic Segmentation

Mathilde Proust[1][a], Martyna Poreba[1][b], Michal Szczepanski[1][c] and Karim Haroun[1,2][d]

[1]*Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France*
[2]*Universit´e C^ote d'Azur, Sophia Antipolis, France*

Keywords: Vision Transformer, Token Pruning, Token Merging, Semantic Segmentation, Supertoken, Computational Efficiency, Optimization.

Abstract: Vision Transformers (ViTs) achieve state-of-the-art accuracy in numerous vision tasks, but their heavy computational and memory requirements pose significant challenges. Minimising token-related computations is critical to alleviating this computational burden. This paper introduces a novel SuperToken and Early-Pruning (STEP) approach that combines patch merging along with an early-pruning mechanism to optimize token handling in ViTs for semantic segmentation. The improved patch merging method is developed to effectively address the diverse complexities of images. It features a dynamic and adaptive system, dCTS, which employs a CNN-based policy network to determine the quantity and size of patch groups that share the same supertoken during inference. With a flexible merging strategy, it handles superpatches of varying sizes: 2×2, 4×4, 8×8, and 16×16. Early in the network, high-confidence tokens are discarded and preserved from subsequent processing stages. This hybrid approach reduces both computational and memory requirements without significantly compromising segmentation accuracy. It is shown through experimental results that, on average, 40% of tokens can be predicted from the 16th layer onwards when using ViT-Large as the backbone. Additionally, a reduction of up to 3× in computational complexity is achieved, with a maximum drop in accuracy of 2.5%.

1 INTRODUCTION

Recently, Vision Transformers (ViTs) (Dosovitskiy et al., 2021) have emerged as highly promising alternatives to Convolutional Neural Networks (CNN) models (He et al., 2016)(Sandler et al., 2018). They are pushing the boundaries of existing knowledge in several computer vision tasks, such as classification (Touvron et al., 2021), object detection (Carion et al., 2020) and semantic segmentation (Zhang et al., 2022)(Zheng et al., 2021). ViTs leverage self-attention to capture global contextual relationships, enabling precise understanding of spatial object distributions. Their ability to model long-range dependencies makes them particularly effective for semantic segmentation, especially in complex scenes where accurate boundary delineation is critical.

This advantage positions ViTs as strong con-

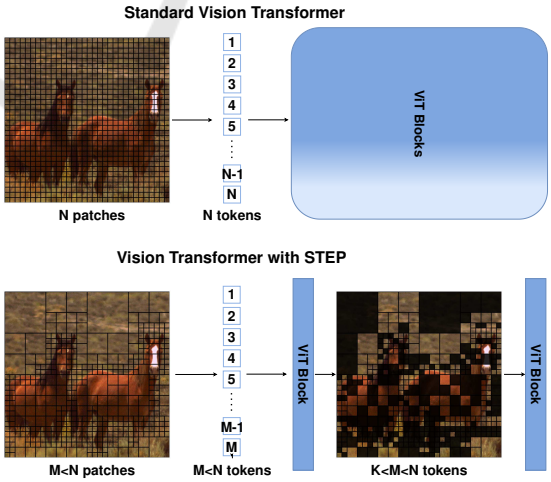

Figure 1: SuperToken and Early-Pruning (STEP) added to Vision Transformer. Semantically similar patches are dynamically merged via our token-sharing policy network to form superpatches. Early-pruned supertokens (black filled) are masked and discarded, and only the remaining tokens are processed in the subsequent layers. STEP boosts efficiency without significant loss in quality.

[a] https://orcid.org/0009-0003-5610-7624
[b] https://orcid.org/0000-0002-5102-7735
[c] https://orcid.org/0009-0000-9061-4396
[d] https://orcid.org/0009-0000-6972-6019

Proust, M., Poreba, M., Szczepanski, M. and Haroun, K.
STEP: SuperToken and Early-Pruning for Efficient Semantic Segmentation.
DOI: 10.5220/0013132800003912
In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 3: VISAPP*, pages 50-61
ISBN: 978-989-758-728-3; ISSN: 2184-4321

tenders in tasks where traditional CNNs often struggle to capture global features. Researchers have explored semantic segmentation using vision transformers in a variety of ways. One approach involves developing custom transformer architectures specifically designed to address the task of semantic segmentation (Wang et al., 2021)(Zheng et al., 2021). Another common approach focuses on enhancing either the transformer-based backbone (Liu et al., 2021)(Wang et al., 2021) or the task-specific decoder (Strudel et al., 2021) (Zhang et al., 2021). Specifically, SegFormer (Xie et al., 2021) enhances the basic architecture by incorporating pyramid features, allowing it to capture multi-scale contexts. Segmenter (Strudel et al., 2021) leverages learnable class tokens in combination with encoder outputs to generate segmentation masks, making the process data-dependent. SegViT (Zhang et al., 2022) advances the study of the self-attention mechanism by introducing an innovative attention-to-mask (ATM) module, which dynamically generates accurate segmentation masks.

Despite their promising performance, ViTs present significant computational challenges. One major issue is the quadratic complexity of the self-attention mechanism, which scales poorly with image resolution. As the size of input images increases, the computational cost and memory requirements rise significantly, making ViTs deployment challenging. Efforts to improve their efficiency still struggle to balance computational complexity, latency, and performance. Reducing complexity or improving inference speed often compromises segmentation accuracy, while optimizing performance can increase computation and slow down inference. Achieving an optimal trade-off between these factors remains a key challenge, particularly in real-time or resource-constrained applications.

In this context, our work introduces SuperToken and Early-Pruning (STEP), a novel token optimization mechanism that dynamically merges semantically similar, neighboring patches into superpatches via a class-agnostic policy network. Unlike traditional grid-based patch processing, this method generates superpatches of different sizes, enabling the number of tokens to adjust according to the complexity of the image content. Additionally, STEP incorporates an early-pruning mechanism where certain tokens are masked and discarded early in the network, further reducing the number of tokens processed in later layers. We can summarize this work's contributions as follows:

- We introduce STEP, a hybrid token optimization mechanism tailored for semantic segmentation that generates superpatches and adapts the token pruning paradigm based on early-pruning strategy.
- We analyze patch merging and token discarding methods, evaluating their impact on latency, computational cost, and accuracy. Our focus is on optimizing these techniques to achieve a balance between efficiency and performance.
- We apply STEP to a mainstream semantic segmentation transformer model (ViT-Large) and conduct extensive experiments on two challenging benchmarks using an A100 GPU. The results show that STEP can reduce computational costs by up to 66% without a significant drop in accuracy.

2 RELATED WORK

Traditionally, ViTs create vision patches by dividing an image into a uniform, fixed grid, with each grid cell treated as a token. However, not all regions of an image are equally crucial for specific tasks. For example, detailed analysis of facial features may require many tokens for accurate representation, while broader regions such as the sky or large uniform surfaces may need only a few tokens. This leads to the question: is it really necessary to process that many tokens at every layer? Given the high computational demands of vision transformers, reducing the number of tokens is a straightforward way to lower computation costs. There are many techniques that can be used to accelerate the inference speed of the ViT model, including quantization (Lin et al., 2022), (Yuan et al., 2022), (Li and Gu, 2023), distillation (Wu et al., 2022), (Yang et al., 2024), and pruning (merging or discarding). Key studies have shown that such reduction techniques can lead to substantial decreases in model size and computational cost, making ViTs more feasible for deployment on resource-constrained devices or in large-scale applications. Pruning involves, based on certain criteria, definitely discarding or merging tokens. However, identifying which tokens are less important or similar can be complex and vary across different layers, as ViTs may focus on different regions at each layer. Addressing these challenges requires advanced pruning techniques. Factors like task-specific requirements, token importance metrics, and retraining strategies are crucial for ensuring the effectiveness of token pruning methods. Additionally, aggressive pruning might lead to the loss of critical information, resulting in degraded model performance. Balancing the intensity of pruning with performance preservation is crucial. Another challenge is the dynamic nature of token importance,

which can vary across different images or tasks, necessitating adaptive pruning strategies that can dynamically adjust based on the input.

2.1 Token Discarding

There are different pruning approaches that involve discarding tokens such as 1) heuristic token reduction like importance scoring-based pruning, which removes tokens based on their relevance assessed through attention weights or entropy; 2) learned pruning, which trains the model to identify less important tokens using auxiliary networks; and 3) gradual pruning, which prunes tokens incrementally across layers to balance efficiency and accuracy. Learned token reduction (Michel et al., 2019), (Kong et al., 2022), (Meng et al., 2022), (Song et al., 2022), (Hao and Jianxin, 2023) typically requires training auxiliary models to rank the importance of tokens in the input data, which is often viewed as a drawback. In contrast, several works have proposed heuristic token reduction that can be applied to the off-the-shelf ViTs without further finetuning. For instance, ATS (Fayyaz et al., 2022) serves as a plug-and-play module that samples tokens based on their similarity to other tokens in the attention map. However, a limitation of this method is its reliance on the class token (*cls*), which might not be applicable or present in dense prediction tasks such as segmentation or object detection.

Many token removal strategies are mainly tailored for image classification, based on the idea that eliminating uninformative tokens, such as backgrounds, has minimal impact on recognition performance. This is effective because classification tasks focus on global features for single-class predictions. In this vein, A-ViT (Yin et al., 2022) computes halting probability score to recognize tokens to be discarted and in this manner perform dense compute only on the active tokens deemed informative. CP-ViT (Song et al., 2022) defines the cumulative score to dynamically locate the informative patches and heads across the ViT model, according to their maximum value in attention probability. DynamicViT (Rao et al., 2021) add an extra learnable neural network to remove redundant tokens progressively and dynamically. The proposed prediction module estimates token's importance score with a MLP (Vaswani et al., 2017). AdaViT (Meng et al., 2022) integrates a jointly optimized lightweight decision network into every transformer block of the ViT backbone to derive inference strategies. Its purpose is to determine which patches to retain, which self-attention heads to activate, and which transformer blocks to bypass for each image.

A slightly different way of proceeding is soft pruning. Instead of discarding less informative tokens completely, they are integrated into a consolidated package token. For instance, SP-ViT (Kong et al., 2022) proposes an attention-based multi-head token selector, which is inserted multiple times throughout the model, to rank, consolidate and prune tokens based on their importance scores. Similarly, EViT (Liang et al., 2022) focuses on the progressive selection of informative tokens during training. It masks and fuses regions that represent the inattentive tokens to expedite computations. The attentiveness value is chosen as a criterion to identify the *top-k* attentive tokens and fuse the rest. Evo-ViT (Xu et al., 2022) proposes an unstructured instance-wise token selection and a slow-fast token updating module. Informative tokens and placeholder tokens are determined by the evolved global class attention. Both informative and non-informative tokens are then updated in different manners. Unlike the previous discarding methods, this makes it possible to maintain the complete spatial structure and information.

Token pruning can improve computational efficiency, but it has notable drawbacks. Primarily, it risks information loss, which may decrease accuracy. The variability in the number of tokens across different inputs complicates batched inference, leading to inefficient resource utilization and increased overhead in managing token counts. Additionally, the process often requires extra training to determine which tokens to retain, adding complexity and prolonging overall training time. To address this issue, zero-shot token pruning methods, such as Zero-TPrune (Wang et al., 2023), can prune large architectures at negligible cost, seamlessly switch between pruning configurations, and efficiently tune hyperparameters. However, it is important to note that excessive pruning may lead to overfitting, causing the model to become overly specialized to the training data.

2.2 Token Merging

Merging combines tokens to create fewer, more comprehensive ones, effectively reducing the overall token count while retaining essential information. Tokens can be merged based on various criteria such as spatial proximity, semantic similarity, or their contribution to the final predictions. One approach to merging is spatial aggregation, which combines tokens representing nearby regions. Another method is feature aggregation, where tokens with similar features or activations are merged. To tackle these challenges, various concepts can be employed, including traditional pooling operations, convolutional integration

within transformers, attention pooling, grid-based token merging, and cluster-based merging. Combining tokens in a way that retains the original information content can be challenging. Merging tokens risks losing fine-grained information encoded by individual tokens, which can affect the model's ability to capture subtle details. Poor aggregation can result in significant information loss, leading to degraded model performance.

As with discarding-based pruning, the common approach in merging is to process in multiple stages, gradually reducing the sequence length while preserving information. For instance, ToMe (Bolya et al., 2023) and (Bolya and Hoffman, 2023) introduce a novel non-training method which averages similar tokens based on the efficient bipartite matching algorithm. Information aggregation from neighboring token is done using a pooling-like operation that combines the features of multiple tokens into a single representative token. This approach is related to DTM (Zizheng et al., 2022), where tokens based on objects scales and shapes are dynamically merged. In contrast, TokenLearner (Ryoo et al., 2021) uses a relatively small number of tokens, learned through the aggregation of the entire feature map, which is weighted by a dynamic attention map conditioned on the feature map. This sophisticated method for tokenizing the input employs a spatial attention mechanism designed to adaptively identify important regions and generate tokens from them. Token Pooling (Marin et al., 2023) addresses the problem using a cost-efficient clustering-based downsampling operator. Following each transformer block, it identifies a subset of tokens that best approximates the underlying continuous signal, thereby capturing redundant features. For token downsampling, Token Pooling employs K-Means or K-Medoids algorithms, or their weighted versions, WK-Means or WK-Medoids, respectively. TCFormer (Zeng et al., 2022) merges tokens from different locations through progressive clustering, generating new tokens with flexible shapes and sizes. STViT (Huang et al., 2022) proposes a per-token attention mechanism consisting of three processes: aggregating tokens into super tokens via the soft k-means, modeling global dependencies in the super token space, and then upsampling the super tokens. PeToMe (Tran et al., 2024) emphasizes preserving informative tokens through an additional metric called the energy score. This score identifies large clusters of similar tokens as high-energy, making them potential candidates for merging, while smaller, unique, and isolated clusters are considered low-energy and are preserved.

2.3 Hybrid Token Reduction

Choosing between token discarding and merging strategies can be intricate, leading to the question of whether one technique may be more effective than the other for a particular task. In this context, ToFu (Kim et al., 2024) amalgamates the benefits of both token pruning and token merging. In practice, the depth of the layer determines the chosen merging strategy: early layers use pruned merging, while later layers transition to average (or MLERP) merging. DiffRate (Chen et al., 2023) includes both token discarding and merging, and formulates token compression as an optimization problem. LTMP (Bonnaerens and Dambre, 2023) adds merging and discarding components with learned threshold masking modules in each transformer block between the Multi-head Self-Attention (MSA) and MLP components. In the same vein, PPT (Wu et al., 2023) combines token pruning for inattentive tokens and token pooling for attentive tokens. It is achieved via an adaptive token compression module inserted inside the standard transformer block.

2.4 Token Pruning in Dense Tasks

In classification tasks, token pruning methods often permanently remove tokens since they no longer affect the outcome. This is because the classification relies primarily on the class token, which is always retained. In dense prediction task like semantic segmentation, patches cannot be discarded entirely, as information from all patches must be retained to ensure accurate pixel-level predictions. ViTs handle this by processing a large number of tokens, which need to be merged effectively to maintain fine-grained details while reducing computational complexity. Consequently, only two of the previously mentioned token reduction methods are suitable for dense prediction tasks, which require high spatial resolution, detailed information, and precise preservation of contextual relationships. For instance, the work of DynamicViT (Rao et al., 2021) has been extend to more network architectures including hierarchical vision transformers as well as more complex dense prediction tasks like object detection and semantic segmentation (Rao et al., 2023). ToFu (Kim et al., 2024) produce promising results for image generation task. The authors of TCFormer (Zeng et al., 2022) envision the proposed method as general and applicable to a wide range of vision tasks, such as object detection and semantic segmentation. However, the major limitation of TCFormer is that the computational complexity of the KNN-DPC algorithm is quadratic with

respect to the number of tokens, which limits TCFormer's speed when dealing with large input resolutions. Among the methods specially designed for the segmentation task we can cite Content-aware Token Sharing (CTS) (Lu et al., 2023), Dynamic Token Pruning (DToP) (Tang et al., 2023), and SVIT (Liu et al., 2024). CTS proposes a class-agnostic policy network trained separately from ViT to predict if neighbouring image patches contain the same semantic class. DToP utilizes early-pruning for high- confidence tokens, enabling the prediction of simpler tokens to be completed earlier without requiring a full forward pass through the entire network. Recently, SVIT introduced a lightweight, 2-layer MLP to effectively choose tokens to be processed in the transformer block. The pruned tokens are preserved in feature maps and can be reactivated in later layers.

3 METHODOLOGY

In this work, we propose a novel hybrid token reduction mechanism aimed at enhancing the efficiency of ViTs for semantic segmentation tasks. Our method, called STEP (SuperToken and Early-Pruning), combines two advanced techniques: supertoken processing and early-pruning (Figure 2). This integration effectively reduces token redundancy while preserving essential image details. The STEP approach ensures optimal allocation of computational resources by dynamically adapting the merging and discarding processes. The flexibility of this method allows for the preservation of important details in complex images while simplifying the processing of less complicated ones. Following the content-aware patch merging process, the image is split into a grid of superpatches with non-uniform sizes, facilitating dynamic scalability tailored to the image content. Figure 3 shows that regions with higher semantic homogeneity produce larger superpatches, whereas regions with greater complexity lead to smaller superpatches. The token-sharing module then transforms the created superpatches into supertokens. This conversion is performed by applying a linear embedding function f_{embed}, which maps the superpatches into their corresponding token representations:

$$\mathbf{Z} = f_{\text{embed}}(\mathbf{P}')$$

where $\mathbf{P}'$ represents the set of superpatches, and f_{embed} is the linear embedding function that generates supertokens $\mathbf{Z}$. The transformer-based ViT models process the resulting supertokens and produce the final output through per-token predictions.

Additionally, we integrate an early-pruning system, as introduced in DToP. This strategy allows confidently predicted tokens in the early layers to exit the network sooner, thereby lowering overall computational costs without compromising segmentation accuracy. Only the most challenging tokens continue to propagate through the deeper layers of the transformer.

3.1 Content-Aware Patch Merging

The STEP method begins with the class-agnostic policy network, designed to determine which patches can be combined into a superpatch, merging them only when they belong to the same semantic class. This draws inspiration from CTS, which utilizes a lightweight CNN to produce probability scores for each 2×2 patch group. Subsequently, the k superpatches are generated based on the highest-ranked probabilities (Figure 3). We advocate for a merging process that takes into account the complexity of the image. Thus, fixing the number and size of superpatches as a hyperparameter is a limitation of the CTS method. For complex images, this approach may force the merging of patches that should remain separate. Conversely, for simpler images, the number of merged zones could be larger.

STEP introduces an improved method that more effectively addresses the diverse complexities of images, thereby overcoming the limitations of the original CTS approach. We integrate a dynamic, adaptive system (dCTS) based on the EfficientNetLite0 model (Tan and Le, 2019), pre-trained on ImageNet-1K (Russakovsky et al., 2015). It employs a more adaptable merging strategy, which involves the use of patch windows of varying size, including 2×2, 4×4, 8×8, and 16×16 patches. For any given window of n neighboring patches $\mathcal{W} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$, the class-agnostic policy network predicts a similarity score S for W:

$$\mathbf{S} = \sigma\left(\mathbf{W}_p^{\top}(\mathcal{W})\right)$$

where $\mathbf{W}_p$ is the learned weight matrix of the policy network and σ is the sigmoid activation function. Additionally, we employ a threshold-based system rather than relying on a predetermined number of merges. Our policy network evaluates the patch groups to estimate the likelihood that they belong to the same class. If the similarity score $\mathbf{S} \geq \tau$, where τ is a predefined threshold, the patch window W is concatenated to form a superpatch as follows:

$$\mathbf{p}^{\text{sp}} = \text{concat}(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n)$$

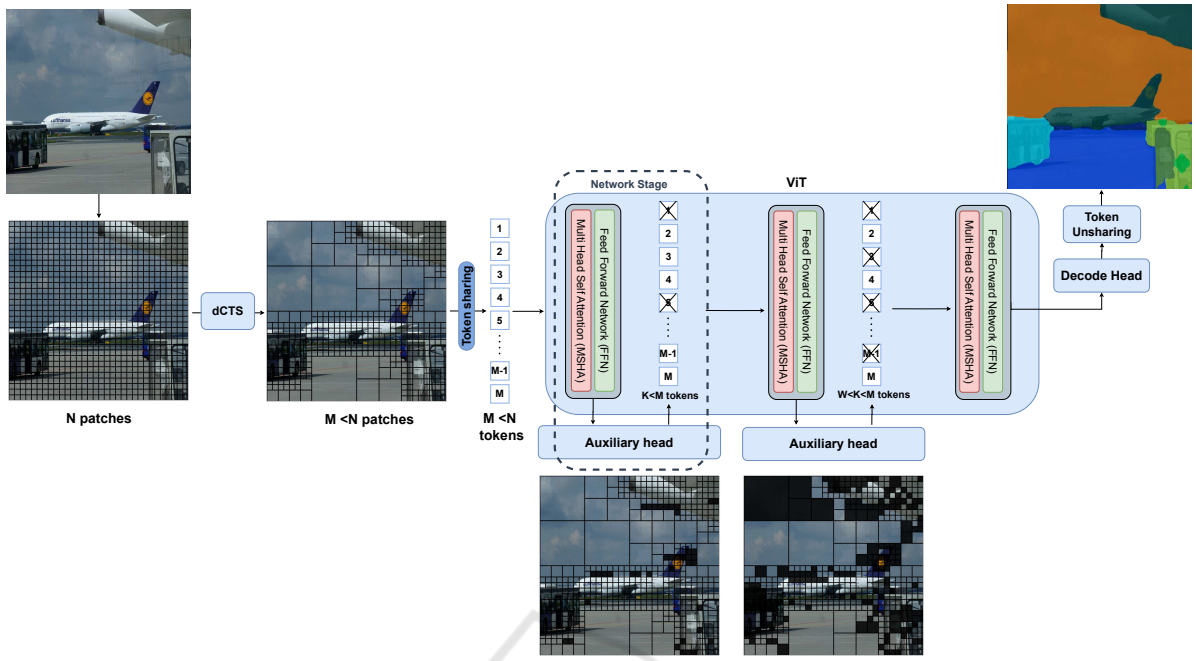

Figure 2: STEP Overview. After dividing the image into patches, our dCTS policy network predicts which groups can form superpatches, which are then transformed into supertokens. Similar to DToP, the ViT model, composed of M attention blocks, is divided into K stages, with built-in auxiliary heads. On this diagram, 3 stages finalize tokens with a high level of certainty. The final decode head combines forecasts from all stages to create the final results.

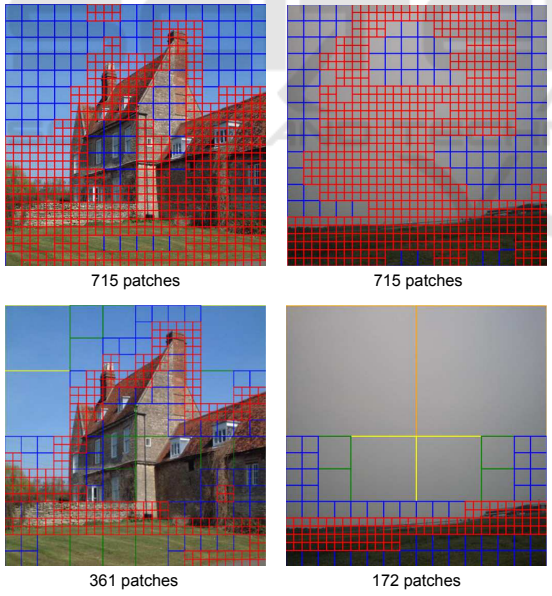

Figure 3: Class-agnostic policy network generates superpaches from neighboring similar patches. The number of tokens to process in the ViT is indicated below each image, relative to the original 1 024 patches. Top: Results obtained after applying the CTS method, with k fixed at 103 as the number of merged groups of 2×2 patches; Bottom: Results after applying our dCTS policy network, allowing dynamic token merging up to groups of 16×16 patches.

3.2 Early Token Pruning

DToP offers an early stopping mechanism that masks and discards high-confident tokens, yet keeps them accessible for later processing stages and final class estimation. As a result, we structure the model into M stages. The model directs tokens to an auxiliary head after a predetermined number of attention block layers and employs a stopping criterion based on its confidence in the predictions. Specifically, at stage M, a confidence score $c_i^{(m)}$ is computed for each token $\mathbf{z}_i$. Tokens with confidence scores greater than a predefined threshold θ are considered high-confidence tokens and are discarded, while the remaining low-confidence tokens continue through the network:

$$\mathbf{Z}^{(m+1)} = \{\mathbf{z}_i \mid c_i^{(m)} < \theta\}$$

where $\mathbf{Z}^{(m+1)}$ represents the set of tokens passed to the next stage.

Each auxiliary head adopts attention-to-mask module (ATM) (Zhang et al., 2022) as the segmentation head. The core concept of DToP is to identify easy tokens in the intermediate layers and exclude them from further computations by assessing the difficulty level of all tokens. This highlights the importance of strategically placing auxiliary heads. If they are positioned too early in the network, the model may struggle to predict the class of any tokens. The

authors of DToP suggest that dividing the backbone into three stages with token pruning at the 6th and 8th layers for ViT-Base and 8th and 16th for ViT-Large achieves a desirable trade-off between computational cost and segmentation accuracy. However, we question this choice, especially since the inference time was not considered. Furthermore, with this configuration, the reduction in computational complexity for ViT-Large is negligible compared to utilizing just the auxiliary head at the 6th layer. We advocate for additional studies to improve our guidelines on the positioning of auxiliary heads.

4 EXPERIMENTS

We implement STEP within the semantic segmentation framework SegViT (Zhang et al., 2022). All experiments are conducted using MMSegmentation (mmseg)(MMSegmentation Contributors, 2020)[1], an open-source toolbox based on PyTorch, which facilitates easy model customization by enabling the combination of various backbones. We incorporate the ViT-Large model, which features 24 encoder layers, a 1024-dimensional hidden layer, and 16 attention heads. The model processes images by dividing them into 16×16 pixel patches.

We conduct extensive experiments on two widely used semantic segmentation datasets: COCOStuff10k (Caesar et al., 2018), which contains a diverse range of objects in complex, real-world scenes, and ADE20k (Zhou et al., 2017), which is a comprehensive dataset for scene parsing. The image sizes are 512×512 for COCOStuff10k, and 640×640 for ADE20K. For DToP, we set a fixed confidence threshold θ at 0.95 for the COCOStuff10k dataset, and 0.9 for the ADE20K dataset. We employ an AdamW optimizer with an initial learning rate of 6e-5, weight decay of 0.01, and a cosine learning rate schedule. We follow the standard mmseg training settings. We train the models for 160K iterations on ADE20K, 80K on COCOStuff10k, using a batch size of 4. Data augmentation includes random horizontal flipping, random resizing (ratio 0.5 to 2.0), and random cropping. The mean intersection over union (mIoU) evaluates segmentation accuracy, the quantity of floating-point operations in giga FLOPS (GFLOPs) indicates the complexity of the model, whereas frames per second (FPS) measures the throughput on a single NVIDIA A100 GPU. We use the fvcore package[2] to compute GFLOPs for all configurations.

[1]https://github.com/open-mmlab/mmsegmentation
[2]https://github.com/facebookresearch/fvcore

4.1 Ablation for Confidence Threshold

We conduct a series of experiments to identify the optimal thresholds for different superpatch sizes in our dCTS approach. During this process, we evaluate the model's performance in terms of mIoU and GFLOPs. This allowed us to determine the optimal balance between computational efficiency and segmentation accuracy for each superpatch size. For example, when merging only groups of 2×2 patches, Figure 4 clearly shows that a threshold τ of 0.4 for tokens belonging to the same class achieves the best balance between accuracy and computational complexity.

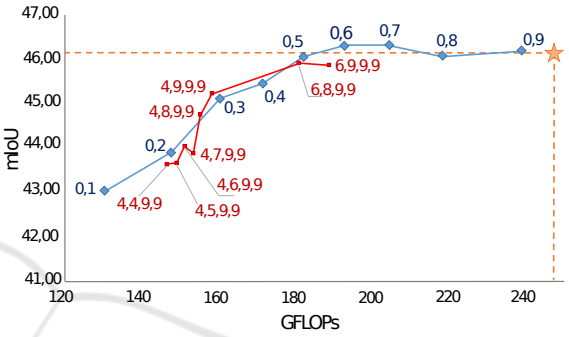

Figure 4: Adjusting the merging threshold hyperparameter affects both accuracy and computational complexity when using the ViT-Large backbone with the COCOStuff10k dataset. The blue curve demonstrates the effect of merging only groups of 2×2 patches, while the red curve depicts the use of varying thresholds that depend on the size of the superpatches. The orange star represents CTS's performance when fixing 103 merged patches of size 2×2.

Table 1 summarizes the results obtained for several threshold configurations. Each patch size is assigned with a unique threshold value τ. We set the threshold probability at 0.9 for larger groups of patches while adjusting its values for smaller groups. This is driven by the need to avoid errors in creating large superpatches, as such mistakes would significantly compromise the quality of the final segmentation. We determine the optimal combination to be τ-4999 or τ-6899 for the 2×2, 4×4, 8×8, and 16×16 superpatch sizes, respectively. Compared to the CTS, the first configuration allows no loss in segmentation accuracy while reducing computational complexity by 27%. The second is less strict on segmentation quality, allowing a potential 1% loss in mIoU, but reducing complexity by 36%.

Through an adaptive approach, we allow for a more personalized merging process tailored to each image. Our dCTS method achieves a significant reduction in the number of tokens compared to CTS. For example, as shown in Figure 3, a complex image can achieve a token reduction by a factor of 2, while

a simpler image can see a reduction by a factor of 4. This decline in tokens explains the decreased computational complexity, allowing for more efficient processing.

Table 1: Experiments on COCOStuff10k dataset. The dCTS method applies thresholds τ according to the size of the superpatches. The values are decimal numbers for the 2×2, 4×4, 8×8, and 16×16 superpatch sizes, respectively.

Threshold τ	mIoU	GFLOPs
CTS (τ not relevant)	46.1	248
.6 .9 .9 .9	45.9	189
.6 .8 .9 .9	46.0	181
.4 .9 .9 .9	45.3	159
.4 .8 .9 .9	44.8	156
.4 .7 .9 .9	43.9	153
.4 .6 .9 .9	44.1	151
.4 .5 .9 .9	43.7	149
.4 .4 .9 .9	43.7	147

4.2 In-Depth Exploration of Pruning Positions

We explore alternative positions for the auxiliary head by dividing the ViT model into two or three stages. For each configuration, we measure the impact on segmentation accuracy, computational complexity (Figure 5), inference time (Figure 6), and the percentage of pruned tokens (Figure 7) to establish the most effective placement strategy.

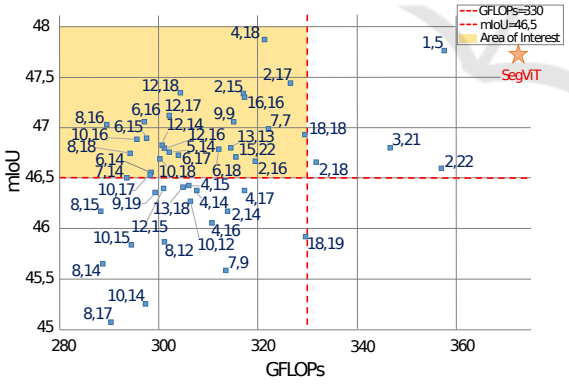

Figure 5: Exploration of the pruning head configuration on the COCOStuff10k dataset. The numbers represent the positions of the auxiliary heads, with the red star marking performance relative to the reference SegViT, where no pruning is applied. The plot compares computational complexity (GFLOPs) to segmentation accuracy (mIoU). The yellow rectangle highlights the configurations selected for further analysis, as they achieve at least a 10% reduction in GFLOPs with a maximum accuracy loss of 1.5%.

The results demonstrate that the number of auxiliary heads impacts computational complexity and

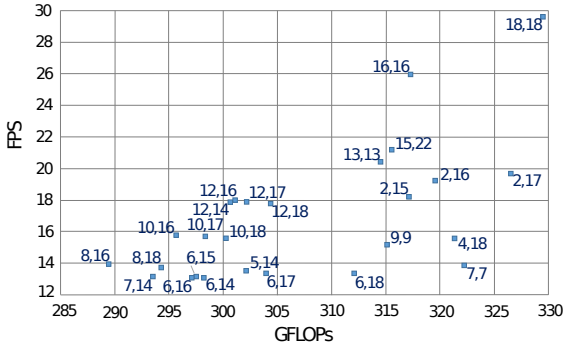

Figure 6: Exploration of the pruning head configuration on the COCOStuff10k dataset. The plot compares throughput (FPS) to computational cost (GFLOPs).

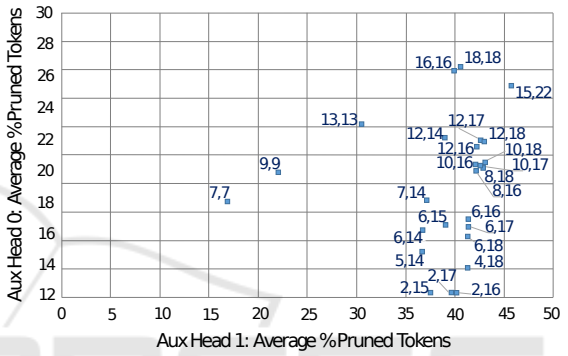

Figure 7: Exploration of the pruning head configuration on the COCOStuff10k dataset. The plot shows the average percentage of pruned tokens achieved.

inference speed. For example, placing the pruning heads at positions 8th and 16th results in a gain of 22% (289 vs. 373) GFLOPs while maintaining segmentation accuracy compared to the SegVit, where no tokens have been pruned. However, adding extra heads also contributes to longer inference times. Placing the pruning heads at the 8th and 16th layers slows down the process by a factor of four, while using a single head at either the 16th or 18th layer results in a twofold increase in inference time. Given this observation, a single auxiliary head presents the best trade-off between reducing complexity and ensuring real-time inference. This level of throughput can only be attained by placing the auxiliary head deeper in the network. As we proceed, the percentage of pruned tokens rises linearly with the use of only one auxiliary head for pruning, reaching an average of approximately 40% after the 16th position. At this stage, the pruning rates attain levels comparable to those achieved with two auxiliary heads, irrespective of their configuration.

Identifying the ideal configuration is not an easy task and is fairly nuanced. If the primary goal is to minimize computational complexity, we suggest di-

viding the network into two stages and positioning the auxiliary heads for pruning after the 8th and 16th layers. Conversely, if inference time is the critical criterion, we recommend using only a single head, positioned as early as the 16th layer.

4.3 Comparative Study

We use SegViT, which performs no merging or token pruning, as a baseline for comparison. We also create its variants by sequentially applying different optimization techniques: first, merging patches using the CTS method, then implementing token pruning with DToP, and finally combining both techniques. The latest is the preliminary version of our STEP mechanism. Throughout this process, we adhered to the baseline configurations and parameters established by the authors. We combine a fixed number of 2×2 patches for CTS, specifically merging 103 patches, and position the auxiliary heads at the 8th and 16th layers for DToP. In our STEP method, we apply the previously described threshold configuration for dCTS. We choose to divide the ViT-Large model into two and three stages, naming them STEP@[18] and STEP@[8,16], respectively. The values in brackets indicate the pruning heads positions.

Figure 8: Distribution of pruned tokens with STEP@[8,16] and segmentation results on COCOStuff10k dataset at each stage of the pruning process highlighting varying image complexities. From left to right: few tokens pruned, an average number pruned, and most token pruned.

Tables 2 and 3 summarize the performance achieved. The results indicate that incorporating STEP into SegViT enables us to maintain a comparable mIoU, with the segmentation accuracy loss not exceeding 2.5%, depending on the chosen configuration. Our STEP yields a notable reduction in GFLOPs; for instance, STEP@[18]τ-4999, achieves a decrease of 58% on ADE20K and 52% on COCOStuff10k. Implementing two auxiliary heads leads to a greater reduction in GFLOPs compared to having a single pruning head after layer 18th. However, in this case, the throughput is twice as slow. In our STEP mechanism, since we also use the configuration [8, 16], which is the same as in the original DToP, we clearly see that our dCTS provides a significant reduction in computational complexity compared to CTS, reducing it by 37% for ADE20K and 28% for COCOStuff10k.

Table 2: Performance evaluation of our STEP mechanism, integrated into ViT-Large, on the ADE20K dataset.

Method	mIoU	GFLOPs	FPS
SegViT	53.0	624	37.7
+CTS	52.0	410	41.1
+DToP	52.3	465	6.3
+CTS&DToP	51.2	334	12.5
+STEP@[8,16]τ-6899	51.2	224	13.9
+STEP@[8,16]τ-4999	50.8	209	14.8
+STEP@[18]τ-6899	51.7	395	21.7
+STEP@[18]τ-4999	50.4	261	26.5

Table 3: Performance evaluation of our STEP mechanism, integrated into ViT-Large, the COCOStuff10k dataset.

Method	mIoU	GFLOPs	FPS
SegViT	46.7	373	44.6
+ CTS	46.2	251	40.3
+ DToP	46.6	290	15.0
+CTS& DToP	45.4	210	17.3
+STEP@[8,16]τ-6899	46.0	173	17.9
+STEP@[8,16]τ-4999	45.3	150	20.1
+STEP@[18]τ-6899	46.0	201	30.2
+STEP@[18]τ-4999	45.1	177	29.2

Figure 8 illustrates how tokens are pruned across images by each auxiliary head, revealing that most tokens are pruned early in simple scenarios, while they are retained until the final prediction phase in more complex scenes. Figure 9 and Figure 10 display sample visualizations of predictions with STEP integrated into ViT-Large.

5 CONCLUSION

We presented a novel token reduction method called SuperToken and Early-Pruning (STEP) that enhances token management in ViTs for semantic segmentation by integrating patch merging with an early-pruning

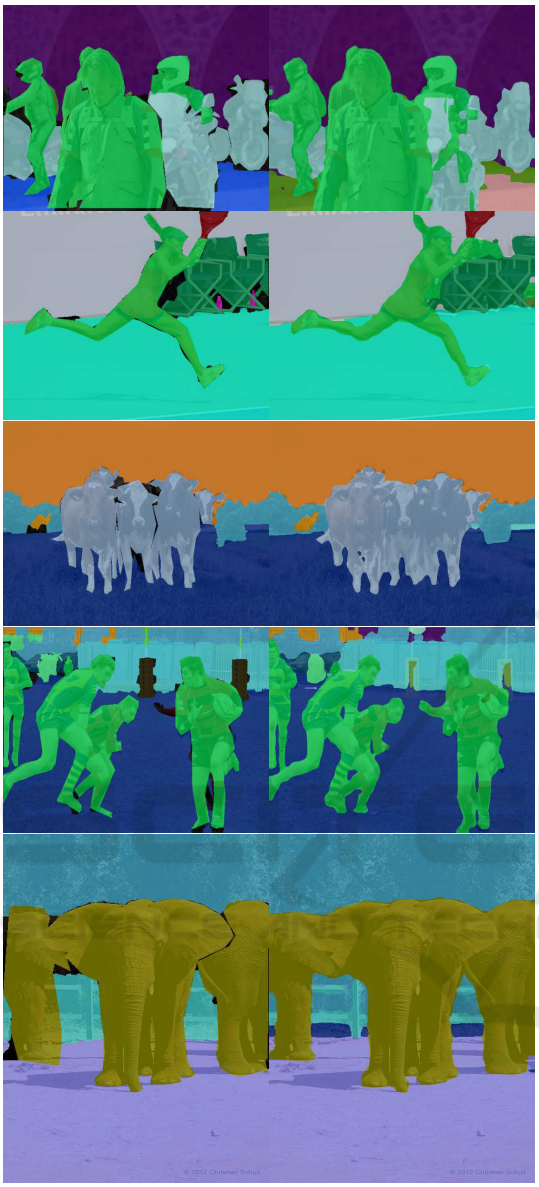

Figure 9: Visualized results on COCOStuff10K STEP@[8,16]τ-4999 added to ViT-Large. Left: Ground truth; Right: Predicted segmentation.

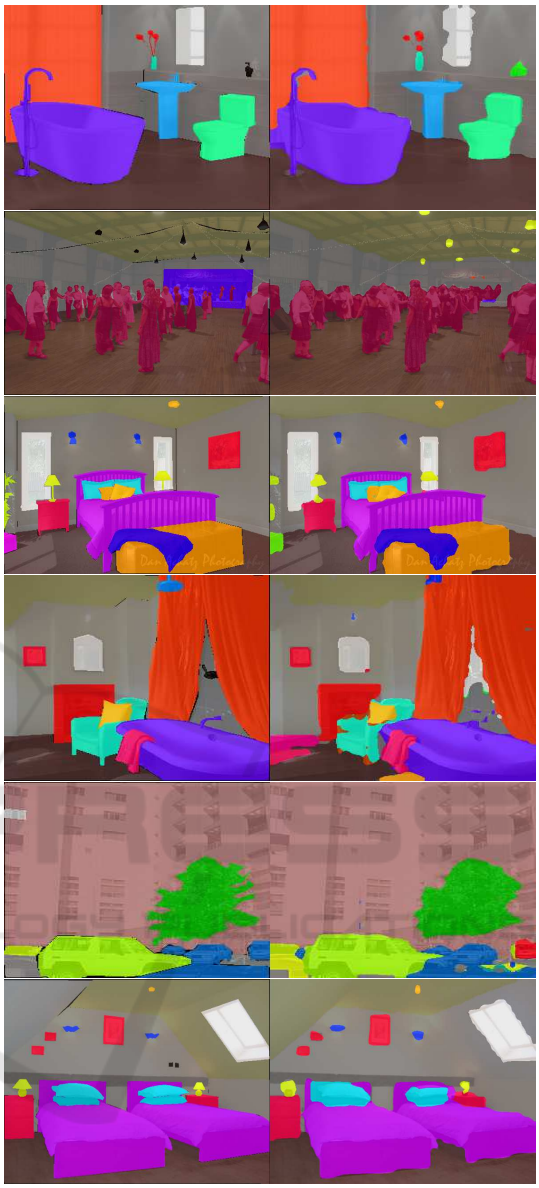

Figure 10: Visualized results on ADE20K with STEP@[8,16]τ-6899 added to ViT-Large. Left: Ground truth; Right: Predicted segmentation.

mechanism. Our approach employs a versatile merging strategy that utilizes superpatches of varying sizes and introduces an improved patch merging technique to effectively handle diverse image complexities. Extensive experimental tests were conducted to establish optimal parameters for creating superpatches. We also explored the advantages of pruning tokens within the network, finding that starting from the 16th layer of ViT-Large, 40% of tokens can be classified with high accuracy. While using auxiliary heads to prune high-confidence tokens reduces computational complexity, it significantly slows down inference. We recommend that future research focus on the examination of auxiliary heads, as pruning tokens using segmentation heads with ATM modules has been shown to be excessively slow.

REFERENCES

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., Feichtenhofer, C., and Hoffman, J. (2023). Token merging: Your ViT

but faster. In *International Conference on Learning Representations*.

Bolya, D. and Hoffman, J. (2023). Token merging for fast stable diffusion. *CVPR Workshop on Efficient Deep Learning for Computer Vision*.

Bonnaerens, M. and Dambre, J. (2023). Learned thresholds token merging and pruning for vision transformers. *Transactions on Machine Learning Research*.

Caesar, H., Uijlings, J., and Ferrari, V. (2018). Coco-stuff: Thing and stuff classes in context. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1209–1218, Los Alamitos, CA, USA. IEEE Computer Society.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

Chen, M., Shao, W., Xu, P., Lin, M., Zhang, K., Chao, F., Ji, R., Qiao, Y., and Luo, P. (2023). Diffrate : Differentiable compression rate for efficient vision transformers. *arXiv preprint arXiv:2305.17997*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Fayyaz, M., Abbasi Kouhpayegani, S., Rezaei Jafari, F., Sommerlade, E., Vaezi Joze, H. R., Pirsiavash, H., and Gall, J. (2022). Adaptive token sampling for efficient vision transformers. *European Conference on Computer Vision (ECCV)*.

Hao, Y. and Jianxin, W. (2023). A unified pruning framework for vision transformers. *Science China Information Sciences*, 66:1869–1919.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, H., Zhou, X., Cao, J., He, R., and Tan, T. (2022). Vision transformer with super token sampling. *ArXiv*, abs/2211.11167.

Kim, M., Gao, S., Hsu, Y.-C., Shen, Y., and Jin, H. (2024). Token fusion: Bridging the gap between token pruning and token merging. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1372–1381.

Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., et al. (2022). Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 620–640. Springer.

Li, Z. and Gu, Q. (2023). I-vit: Integer-only quantization for efficient vision transformer inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17065–17075.

Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., and Xie, P. (2022). Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*.

Lin, Y., Zhang, T., Sun, P., Li, Z., and Zhou, S. (2022). Fq-vit: Post-training quantization for fully quantized vision transformer. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1173–1179.

Liu, Y., Gehrig, M., Messikommer, N., Cannici, M., and Scaramuzza, D. (2024). Revisiting token pruning for object detection and instance segmentation. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2646–2656, Los Alamitos, CA, USA. IEEE Computer Society.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022.

Lu, C., de Geus, D., and Dubbelman, G. (2023). Content-aware Token Sharing for Efficient Semantic Segmentation with Vision Transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Marin, D., Chang, J.-H. R., Ranjan, A., Prabhu, A., Rastegari, M., and Tuzel, O. (2023). Token pooling in vision transformers for image classification. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 12–21.

Meng, L., Li, H., Chen, B.-C., Lan, S., Wu, Z., Jiang, Y.-G., and Lim, S.-N. (2022). Adavit: Adaptive vision transformers for efficient image recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12309–12318.

Michel, P., Levy, O., and Neubig, G. (2019). Are sixteen heads really better than one? In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

MMSegmentation Contributors (2020). MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation.

Rao, Y., Liu, Z., Zhao, W., Zhou, J., and Lu, J. (2023). Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10883–10897.

Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., and Hsieh, C.-J. (2021). Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Ryoo, M. S., Piergiovanni, A. J., Arnab, A., Dehghani, M., and Angelova, A. (2021). Tokenlearner: What can

8 learned tokens do for images and videos? *CoRR*, abs/2106.11297.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Song, Z., Xu, Y., He, Z., Jiang, L., Jing, N., and Liang, X. (2022). Cp-vit: Cascade vision transformer pruning via progressive sparsity prediction.

Strudel, R., Garcia, R., Laptev, I., and Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272.

Tan, M. and Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, abs/1905.11946.

Tang, Q., Zhang, B., Liu, J., Liu, F., and Liu, Y. (2023). Dynamic token pruning in plain vision transformers for semantic segmentation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 777–786, Los Alamitos, CA, USA. IEEE Computer Society.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR.

Tran, H.-C., Nguyen, D., Nguyen, D., Nguyen, T., Lê, N., Xie, P., Sonntag, D., Zou, J., Nguyen, B., and Niepert, M. (2024). Accelerating transformers with spectrum-preserving token merging.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, H., Dedhia, B., and Jha, N. K. (2023). Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. *arXiv preprint arXiv:2305.17328*.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578.

Wu, K., Zhang, J., Peng, H., Liu, M., Xiao, B., Fu, J., and Yuan, L. (2022). Tinyvit: Fast pretraining distillation for small vision transformers. In *European conference on computer vision (ECCV)*.

Wu, X., Zeng, F., Wang, X., Wang, Y., and Chen, X. (2023). Ppt: Token pruning and pooling for efficient vision transformers. *arXiv preprint arXiv:2310.01812*.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. (2021). Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090.

Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., Zhang, L., Xu, C., and Sun, X. (2022). Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2964–2972.

Yang, Z., Li, Z., Zeng, A., Li, Z., Yuan, C., and Li, Y. (2024). ViTKD: Feature-based knowledge distillation for vision transformers.

Yin, H., Vahdat, A., Alvarez, J., Mallya, A., Kautz, J., and Molchanov, P. (2022). A-ViT: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yuan, Z., Xue, C., Chen, Y., Wu, Q., and Sun, G. (2022). Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, page 191–207, Berlin, Heidelberg. Springer-Verlag.

Zeng, W., Jin, S., Liu, W., Qian, C., Luo, P., Ouyang, W., and Wang, X. (2022). Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11101–11111.

Zhang, B., Tian, Z., Tang, Q., Chu, X., Wei, X., Shen, C., and Liu, Y. (2022). Segvit: Semantic segmentation with plain vision transformers. *NeurIPS*.

Zhang, W., Pang, J., Chen, K., and Loy, C. C. (2021). K-net: Towards unified image segmentation. *Advances in Neural Information Processing Systems*, 34:10326–10338.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., et al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. (2017). Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zizheng, P., Bohan, Z., Haoyu, H., Jing, L., and Jianfei, C. (2022). Less is more: Pay less attention in vision transformers. In *The Thirty-Sixth AAAI Conference on Artificial Intelligence*, pages 2035–2043.