Where Do Tokens Go? Understanding Pruning Behaviors in STEP at High Resolutions

Michal Szczepanski[1†], Martyna Poreba[1*†], Karim Haroun[2]

[1*]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France.
[2]I3S, Université Côte d'Azur, CNRS, Sophia Antipolis, 06900, France.

*Corresponding author(s). E-mail(s): martyna.poreba@cea.fr;
Contributing authors: michal.szczepanski@cea.fr;
karim.haroun@etu.univ-cotedazur.fr;
†These authors contributed equally to this work.

Abstract

Vision Transformers (ViTs) achieve state-of-the-art performance in semantic segmentation but are hindered by high computational and memory costs. To address this, we propose STEP (SuperToken and Early-Pruning), a hybrid token-reduction framework that combines dynamic patch merging and token pruning to enhance efficiency without significantly compromising accuracy. At the core of STEP is dCTS, a lightweight CNN-based policy network that enables flexible merging into superpatches. Encoder blocks integrate also early-exits to remove high-confident supertokens, lowering computational load. We evaluate our method on high-resolution semantic segmentation benchmarks, including images up to $\mathbf{1024 \times 1024}$, and show that when dCTS is applied alone, the token count can be reduced by a factor of 2.5 compared to the standard $\mathbf{16 \times 16}$ pixel patching scheme. This yields a 2.6× reduction in computational cost and a 3.4× increase in throughput when using ViT-Large as the backbone. Applying the full STEP framework further improves efficiency, reaching up to a 4× reduction in computational complexity and a 1.7× gain in inference speed, with a maximum accuracy drop of no more than 2.0%. With the proposed STEP configurations, up to 40% of tokens can be confidently predicted and halted before reaching the final encoder layer.

Keywords: Vision Transformer, Patch, Supertoken, Pruning, Merging, Semantic Segmentation, Computational Complexity, Optimization

1 Introduction

Vision Transformers (ViTs) have demonstrated strong performance in semantic segmentation tasks, primarily thanks to their capacity to capture long-range dependencies. Numerous strategies have been proposed to harness the full potential of ViTs in this context. One line of work focuses on designing transformer architectures specifically tailored for semantic segmentation [1, 2]. For example, SETR [2] views segmentation as a sequence-to-sequence prediction task, while the Pyramid Vision Transformer (PVT) [1] introduces a hierarchical structure to better capture spatial information. Another prevalent approach involves enhancing the transformer-based backbone [3] or modifying the task-specific decoder [4–6]. SegFormer [6] enhances segmentation performance by integrating pyramid features without relying on positional encodings. Segmenter [4] introduces learnable class tokens that interact with the encoder output to generate masks in a data-dependent manner. SegViT [7] pushes the boundaries of self-attention through its attention-to-mask (ATM) module that directly predicts segmentation masks from attention maps. More recently, several works have proposed lightweight or alternative segmentation schemes that leverage the strength of pre-trained ViT backbones. EoMT [8] introduces an encoder-only mask transformer that reuses a frozen ViT backbone and a lightweight mask head, demonstrating that ViTs inherently encode sufficient spatial information for segmentation without complex decoders. CCASeg [9] proposes a convolutional cross-attention decoder that efficiently aggregates multi-scale context with reduced computational overhead. U-MixFormer [10] presents a U-Net–like transformer architecture with mix-attention blocks, achieving competitive performance through efficient feature fusion. S4Former [11] designs a semi-supervised ViT framework with patch-adaptive self-attention, achieving strong results with only partial label supervision.

Despite their strong performance, ViTs still pose significant computational challenges. A primary concern is the quadratic complexity of the self-attention mechanism, which scales poorly with image resolution. As input image size increases, both computational cost and memory consumption grow substantially, hindering the practical deployment of ViTs. Although various efforts have been made to improve their efficiency, achieving a balance between computational complexity, latency, and performance remains difficult, including quantization [12–16] knowledge distillation [17, 18] and pruning. Key studies have demonstrated that these model compression approaches can significantly reduce both model size and computational cost, thereby enhancing the practicality of ViTs in large-scale applications. In this context, we propose SuperToken and Early-Pruning (STEP), a novel token reduction mechanism designed to enhance the efficiency of ViT for semantic segmentation. In contrast to conventional grid-based patch processing, this approach produces superpatches of varying sizes thanks to the proposed dCTS module, allowing the number of tokens to adapt to the complexity of the image content. Furthermore, STEP integrates an early-pruning strategy, in which certain tokens are masked and halted early in the network pipeline, thereby reducing the computational load in subsequent layers. This paper is an extended version of our conference paper [19]. We make several new contributions:

- We conduct extensive experiments on an NVIDIA A100 GPU to evaluate the STEP mechanism integrated into state-of-the-art Transformer backbones (ViT-Large and ViT-Base), using SegViT as the decoder and widely recognized semantic segmentation benchmarks.
- We also demonstrate the potential of our framework on high-resolution images (up to 1024×1024) to assess its scalability for semantic segmentation. The model maintains competitive accuracy while significantly reducing computational complexity and inference time. Notably, to the best of our knowledge, this is the first attempt to evaluate a token pruning strategy in the context of high-resolution semantic segmentation.
- We provide more in-depth analyses, ablation studies, and visualizations.

2 Vision Transformer Pruning: Prior Work

Vision Transformers traditionally partition an image into a uniform grid, treating each patch as an individual token. However, this fixed strategy overlooks the varying importance of different image regions depending on the task. For instance, recognizing fine details may require a high token density, whereas homogeneous areas can be represented with fewer tokens. This raises a key question: is it necessary to process the same number of tokens for each input image? Given the substantial computational cost of ViTs, reducing the number of tokens emerges as a natural and effective way to improve efficiency. When examining existing approaches, pruning techniques can be broadly categorized based on the level at which they operate. Some methods act at the patch-level to reduce redundancy before the input reaches the ViT backbone. Others focus on token-token pruning, eliminating tokens based on similarity or learned importance throughout the transformer layers. Effectively addressing these challenges requires advanced strategies that consider task-specific requirements, reliable token importance metrics, and retraining schemes to compensate for information loss. Importantly, excessive pruning may lead to the removal of critical content, degrading overall model performance. Striking a balance between computational efficiency and accuracy preservation remains a central challenge in token pruning for ViT.

2.1 Patch-level pruning

Patch-level pruning includes the aggregation of neighboring patches into larger, semantically consistent units. Some existing methods rely on learned mechanisms that dynamically predict which patches should be merged, typically using lightweight neural modules. For example, CTS [20] retains the naively sliced square image patches and merges locally the most similar ones. For this purpose, it employs a class-agnostic policy network to predict whether a group of 2×2 neighboring patches belongs to the same class. If so, the patches are merged and represented by a shared token, thereby reducing the overall token count. An alternative idea is to use adaptive resolution or mixed-scale tokenization [21–23]. These approaches dynamically select token sizes or resolutions based on the input image content, while still relying on square-shaped patches. In MSViT [21] a lightweight, four-layer MLP serves as a gating mechanism, making binary decisions on whether a region should be tokenized coarsely (with 32×32

pixel patches) or finely (with 16×16 pixel patches). CF-ViT [22] proposes a coarse-to-fine inference strategy. The model first performs inference on coarse-grained patches. If the confidence is low, only the informative regions identified via global class attention are re-processed at finer granularity. The Quadtree algorithm, integrated into the Quadformer model [23], is combined with a saliency scorer to adaptively partition the image into patches of varying sizes. Regions with higher saliency are represented at higher resolution, while less salient areas are processed at lower resolution. A different strategy for reducing the number of patches is patch pruning, which aims to retain only the most informative patches for the target task. This selective retention can be guided by learned importance measures, enabling the model to focus its computational resources on semantically relevant regions while discarding redundant or background information, as demonstrated by PaPr [24].

2.2 Token-level pruning

Token-level pruning typically operates at intermediate layers by removing or merging tokens based on their estimated importance. This usually takes place after one or more Transformer blocks, once sufficient contextual information has been aggregated to make an informed decision about which tokens are less informative or redundant for the downstream task. In contrast to patch-level, token pruning leverages the evolving semantic representations of tokens as they propagate through the network. A key component is the scoring mechanism used to evaluate the importance of each token. These techniques can broadly be categorized into learned and heuristic approaches. Learned token pruning methods [25–31] incorporate trainable modules into the ViT architecture to assess token informativeness. In contrast, heuristic token pruning can be applied to the off-the-shelf ViTs, without further finetuning [32–35]. Regardless of the technique used, the derived score determines which tokens are retained and which can be safely discarded or merged.

2.2.1 Token discarding

Token discarding refers to selectively removing tokens based on predefined importance scores or confidence measures.These methods can typically be divided into hard and soft pruning. In hard pruning [25, 26, 30, 31, 36–38] less important tokens are completely removed based on a predefined importance score. In contrast, soft pruning does not eliminate tokens entirely. Instead, it either aggregates less informative tokens into consolidated representations package token [28, 29, 39], or halts their further processing once they reach a sufficient confidence level [34, 40–43].

DToP [34] and DoViT [41] both adopt the use of dynamic early-exit mechanisms that adaptively prune tokens based on confidence scores computed at intermediate layers, with DoViT adding a reconstruction module for spatial consistency. A-ViT [43] proposes to halt tokens using a cumulative sigmoid-based score derived from token embeddings. Among the methods that focus on generating and consolidating representative tokens, SP-ViT [28] stands out by introducing an attention-based multi-head token selector. This module is inserted at multiple points in the network to rank tokens by importance, consolidate similar ones, and prune the least informative. Similarly, EViT [29] 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 [39] goes further by updating and reintegrating the fused token into the network through a slow-fast evolution mechanism, preserving information more effectively.

Whereas the aforementioned soft pruning maintains spatial structure by preserving compressed token information, hard pruning methods adopt a more aggressive stance by completely removing tokens. ATS [26] prunes tokens by scoring their importance using attention from the classification token and sampling them via inverse transform sampling. It adaptively selects a variable number of tokens per image, is parameter-free, and works with pre-trained models without retraining. CP-ViT [31] dynamically prunes uninformative patches and heads using cumulative attention-based scores computed across layers. AdaViT [30] introduces a lightweight decision network integrated into each Transformer block, jointly optimized with the backbone. At inference time, it outputs binary decisions to selectively retain tokens, activate self-attention heads, or skip entire blocks, enabling dynamic and input-dependent computation. DynamicViT [25] also incorporates lightweight prediction modules at multiple layers to progressively estimate token importance and discard less informative ones. Zero-TPrune [38] applies a two-stage, zero-shot pruning process. It first ranks token importance using attention-based PageRank, then removes redundancy by merging similar tokens. Unlike AdaViT or DynamicViT, it requires no training or architectural modification, aligning more closely with ATS in its plug-and-play nature.

2.3 Token merging

Merging reduces the number of tokens by combining them into more informative, aggregated representations, while preserving key information. This can be done based on criteria like spatial proximity, semantic similarity, or predictive contribution. A common approach involves a hybrid of spatial and feature aggregation: spatial aggregation merges tokens from adjacent regions, while feature aggregation combines tokens with similar semantic representations. DPC-KNN [44] identifies clusters by estimating local token densities and merging those with minimal distance to high density points. TCFormer [45] merges tokens from different locations through progressive clustering, generating new tokens with flexible shapes and sizes. AiluRus[46] reduces token count in ViTs via spatial-aware merging based on Density Peaks Clustering (DPC). Tokens are merged by selecting cluster centers using a score combining feature-space density and spatial distance. Non-center tokens are assigned to their nearest center. A reweighting mechanism adjusts attention to account for merged token groups. Token Pooling [47] employs hard clustering by minimizing intra-cluster distances, using attention from the CLS token to initialize cluster centers. Following each transformer block, it identifies a subset of tokens that best approximates the underlying continuous signal, thereby capturing redundant features. ToMe [33] computes token similarity using cosine similarity between attention keys, then merges the most similar token pairs using a bipartite matching algorithm. The merging is done via a weighted average of their features. ALGM [48] performs token merging in a two-stage process. It first merges locally similar tokens in early layers, then globally merges semantically similar tokens mid-network, using cosine similarity and a ToMe-inspired strategy. LoTM [49] introduces a local constraint by merging only pairs of horizontally adjacent tokens based on cosine similarity. DHTM [50] extends the previous approach by considering all tokens as potential references and selectively merges only the most similar neighboring tokens in each Transformer layer. Unlike prior methods that rely on intermediate ViT features or fixed merging heuristics, DTEM [51] learns a dedicated token embedding solely for merging. This decoupled embedding enables a soft, differentiable merging process during training and efficient hard merging at inference improving both flexibility and performance across tasks.

2.3.1 Hybrid token reduction

Determining whether to discard or merge tokens involves nuanced trade-offs, raising the issue of which strategy yields better performance for a particular task. Recent developments have introduced hybrid approaches that unify token merging and discarding within a single framework to further improve the efficiency of ViT. However, integrating both techniques introduces additional design considerations, particularly in determining when and how to apply each mechanism throughout the network. In this context, LTMP [52]introduces, into every Transformer block, threshold-based masking between MSA and FFN blocks to decide whether to keep, merge, or drop individual tokens. In ToFu [53], the BSM algorithm plays a central role. Given a group of similar tokens, three token reduction strategies are proposed: tokens can either be fused using average merging, merged with MLERP (Norm-Preserving Average), or discarded. Token pruning strategy varies with layer depth: early layers apply discarding, while later layers favor merging. Both LTMP and ToFu adapt token merging from ToMe. PPT [54] is based on the per-layer, per-instance variance of token importance scores. High variance favors pruning, while low variance favors merging. The authors observe that the variance of token importance scores increases with model depth, making token importance more distinguishable in deeper layers. Consequently, token pruning is more effective in deeper layers, while token merging is preferable in shallower layers; a finding that contrasts with the observations from ToFu. DiffRate [55] treats token reduction as a learnable optimization problem, allowing each layer to adjust its compression rate dynamically. Rather than handcrafting which layers should prune or merge tokens, DiffRate treats the compression rates as learnable parameters per layer. These are optimized during training through gradient descent, thanks to a module called the Differentiable Discrete Proxy (DDP). In practice, both token pruning and merging are applied in every transformer layer, but the proportion of each is learned in a differentiable manner. The pruning mechanism in UCC [56] is based on a hybrid importance score that combines both spatial and spectral information. At each Transformer block, tokens with low importance scores are pruned. However, instead of discarding them, UCC merges pruned tokens into the retained ones using a combination of cosine similarity and frequency-aware weighting, thereby maintaining the contextual integrity of the input. PM-ViT [57] proposes layer-wise compression strategy. This approach uses a learnable merge matrix to fuse less important tokens into aggregated representations and a reconstruct matrix to restore token dimensions after the transformer block. Token importance is estimated during training through a gradient-weighted attention scoring mechanism, which avoids extra computation at inference time. Tokens are categorized into three groups: high-importance tokens are preserved, medium-importance tokens are merged, and low-importance tokens are pruned. Shortcut connections are used to reintroduce pruned tokens, ensuring minimal information loss.

2.4 Token Reduction for Dense Tasks

Most existing token reduction techniques have been primarily evaluated on image classification or generative tasks such as diffusion, with their applicability to dense prediction tasks remaining relatively underexplored. Token discarding methods, in particular, often involve the permanent removal of tokens that are deemed uninformative for the final prediction. This is feasible in classification settings due to the architectural design of ViTs, where the output is derived solely from the class token, which is always retained. However, in dense prediction tasks such as semantic segmentation, this strategy is not viable, as accurate pixel-wise predictions require preserving information from all spatial tokens. Consequently, more nuanced token reduction strategies like token merging or soft pruning are necessary to maintain spatial fidelity while reducing computational overhead. Consequently, only a few of the aforementioned token reduction methods are suitable for dense prediction tasks. Several approaches specifically developed for segmentation leverage merging-based mechanisms, whether at the patch [20, 21] or token-level [48]. ALGM extends ToMe's global merging with segmentation-aware local merging and adaptive control, making it effective for dense prediction tasks. STViT [58] and Ailurus [46] have been validated across various dense prediction tasks, including object detection, instance segmentation, and semantic segmentation. TCFormer [45, 59] is presented as a general-purpose method applicable to various vision tasks, including object detection and semantic segmentation. However, its main limitation lies in the quadratic computational complexity of the KNN-DPC algorithm with respect to the number of tokens, which hampers its efficiency at high input resolutions. Among token-level pruning strategies, soft pruning is generally preferred, as it allows for more flexible token selection and gradual reduction without hard elimination. In particular, approaches that incorporate early stopping mechanisms appear especially well suited. In this direction, methods such as DToP [34], Paumer [40], and DoViT [41] have been specifically designed for semantic segmentation. SViT [42] validates its approach on object detection and instance segmentation benchmarks. In contrast, DynamicViT [25] adopts hard token pruning, and its extended version [60] also proves the method's effectiveness for object detection and instance segmentation. Hybrid token reduction methods, such as ToFu [53], have shown promising results on image generation tasks. PM-ViT [57], on the other hand, demonstrates its approach on image classification and semantic segmentation.

3 Methodology

In this work, we introduce a novel token reduction strategy designed to improve the efficiency of ViTs. Our approach, named STEP (SuperToken and Early-Pruning), integrates two complementary techniques: supertoken generation and early token pruning. A supertoken is a compact representation derived from aggregating multiple spatially adjacent and semantically similar image patches into a single superpatch. The STEP mechanism, integrated into the vanilla ViT architecture, effectively reduces sequence length while preserving the essential spatial and semantic structure of the image, resulting in a more efficient yet accurate segmentation pipeline. This is achieved through dynamic adjustment of patch merging rates and token halting.

3.1 Motivation

Token-level pruning strategies applied deeper in the network often rely on intermediate attention scores or learned token importance, requiring additional computation and training complexity. These methods also typically maintain the full input sequence during early layers. From our point of view, reducing token count at the patch-level provides several practical and architectural benefits compared to token-level pruning. Since patches constitute the input units for the ViT, eliminating redundant ones at this stage directly shortens the input sequence. This leads to immediate reductions in computational cost and memory usage across all subsequent layers. The impact is particularly significant in the case of high-resolution semantic segmentation, where the initial number of tokens can be extremely high. Moreover, patch-level reduction is inherently more interpretable and compatible with pre-trained models.

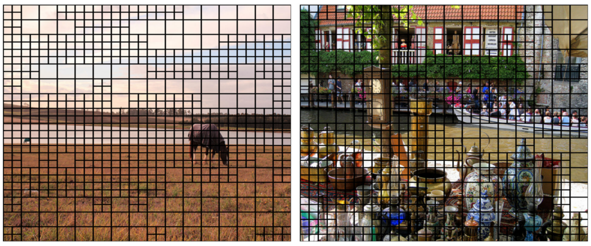


Fig. 1: Example of failures of CTS due to the top-K merging strategy. Left: Too few merges on simple images; Right: Too many merges on complex images.

However, existing approaches have certain limitations. CTS [20], for example, fixes the number (K=103) and size (2 × 2) of superpatches as hyperparameters. This can be problematic for complex images, as it may lead to the merging of patches that should remain separate. Conversely, for images with homogeneous content, the merging rate can often be suboptimal (see Figure 1). MSViT [21] addresses one of the limitations of CTS by dynamically adapting the number of merged patches. Nevertheless, the highest resolution patch remains limited to 2 × 2. Quadformer employs three grouping sizes namely 8 × 8, 4 × 4, and 2 × 2 patches in its mixed-resolution tokenization scheme. A key limitation of this approach is the increased inference-time overhead observed in small models, especially ViT-Small. Although the saliency scorer is lightweight in terms of parameters, its execution is relatively slow compared to the fast inference of compact models. This motivates our focus on content-adaptive patch-level pruning, which dynamically adjusts supertoken resolution and number based on local semantic homogeneity. We retain a regular grid structure after patch merging to simplify positional embedding interpolation and maintain compatibility with standard ViT architectures.

Moreover, we believe, as demonstrated by soft pruning approaches, that tokens vary in difficulty, and that simpler tokens may be predicted earlier, eliminating the need for a complete forward pass through the entire network. Once a sufficient confidence level is reached, their further processing can be halted. In segmentation tasks, this idea becomes even more appealing, since tokens cannot be entirely removed due to the requirement for per-token predictions. We therefore consider this method to be complementary to input sequence length reduction, as it enables a progressive shortening of the set of tokens processed as the network deepens. Such a hybrid approach not only reduces computation but also allows ViTs to better allocate attention and processing power to semantically rich regions, making them more suitable for high-resolution semantic segmentation.

3.2 Overview of the STEP

STEP is a hybrid token-reduction approach that operates on two levels: it first merges patches at the local level, then performs additional token pruning at selected stages of the network (Figure 2). The process begins by dividing the image into a uniform grid of superpatches, following the standard procedure used in vanilla Transformers (in our case, 16×16 pixel patches). Next, a module called dCTS performs token merging based on similarity, resulting in a grid of superpatches with non-uniform sizes. The token-sharing module transforms the created superpatches into supertokens. Superpatches are resized to the standard 16×16 pixel resolution using bilinear interpolation and projected into the embedding space in the same way as regular patches. The latter is performed by applying a linear embedding function f_{embed}, which maps the superpatches into their corresponding token representations:

$$Z = f_{\text{embed}}(P') \tag{1}$$

where P' represents the set of superpatches, and f_{embed} is the linear embedding function that generates the supertokens Z. The transformer-based ViT models process the resulting supertokens and produce the final output through per-token predictions. An early exit strategy is also implemented through an auxiliary decoding head within encoder blocks, which are divided into S stages. This allows tokens that are confidently predicted by the model in the early layers to be halted, thereby reducing overall computational costs without compromising segmentation accuracy. Only the most challenging tokens continue to propagate through the deeper layers of the transformer.

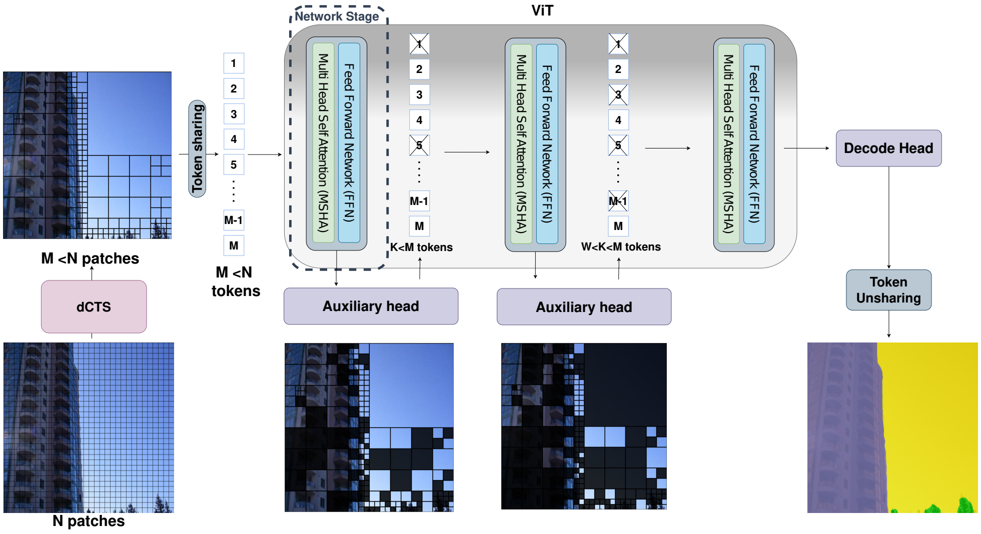


Fig. 2: STEP overview. The image is first divided into fixed-size grid patches. The dCTS policy network then predicts which groups of patches can be merged into superpatches of varying resolutions, which are subsequently transformed into supertokens. Following the DToP approach, the ViT encoder blocks are organized into S stages. This multi-stage structure, equipped with auxiliary decoders, dynamically masks high-confidence tokens (represented as black squares), while the remaining tokens are propagated through the subsequent layers. The final decoder head combines predictions from all stages to generate the final output. Figure inspired from [19].

3.2.1 Semantic-aware patch aggregation

We propose a flexible and content-adaptive strategy, referred to as dynamic CTS (dCTS), inspired by CTS but designed to more effectively address the inherent complexity and variability of image contents. This merging step is guided by a lightweight class-agnostic policy network built upon the EfficientNetLite0 architecture [61], pre-trained on ImageNet-1K [62]. For each image, groups of adjacent patches are considered and a similarity score is computed to assess whether the group likely belongs to a single semantic class. Given any window of n neighboring patches $\mathcal{W} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$, the policy network predicts a similarity score $\mathbf{S}$ as follows:

$$\mathbf{S} = \sigma\left(\mathbf{W}_p^\top(\mathcal{W})\right) \tag{2}$$

where $\mathbf{W}_p$ is the learned weight matrix of the policy network and σ denotes the sigmoid activation function. Fusion is performed using a threshold-based approach: if the similarity score exceeds a predefined threshold τ, the patches are merged into a superpatch:

$$\mathbf{p}^{\text{sp}} = \text{concat}(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n) \tag{3}$$

In practice, the policy network processes the input image after it has been divided into uniform patches. For each group, it produces a similarity score $\mathbf{S}$, a continuous value in the range $[0, 1]$, which is interpreted as the probability that the group is homogeneous. Rather than predicting the exact class, the network leverages this probability and applies a predefined threshold τ to categorize each group into one of two classes: (i) likely belonging to a single semantic class or (ii) likely heterogeneous. This probabilistic interpretation supports a flexible, threshold-based decision mechanism for patch merging. Fusion proceeds in a coarse-to-fine manner, starting with the largest window sizes (16×16 patches) and progressively evaluating smaller windows (8×8, 4×4, and 2×2). This hierarchical order ensures that there are no conflicts between nested groups , *i.e.*, it prevents the merging of a smaller 2×2 patch group that is already part of a larger region deemed mergeable. Figure 3 presents illustrative results of the merging process using our dCTS method. These examples showcase the ability of dCTS to adaptively merge patches in homogeneous regions (e.g., background or sky) while maintaining higher spatial resolution in semantically complex areas such as object boundaries or textured regions.

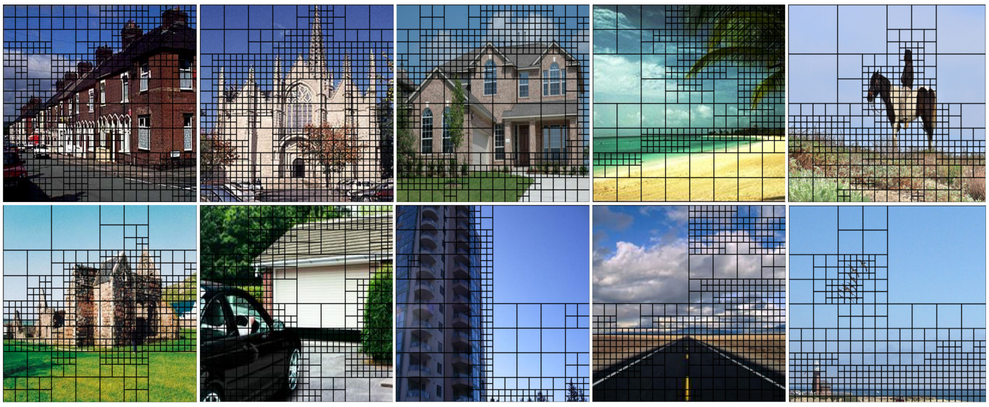


Fig. 3: Resolution-aware splitting on superpatches using dCTS (from 2×2 to 16×16).

3.2.2 Early-pruning mechanism

We incorporate a state-of-the-art early exiting mechanism inspired by DToP into our pipeline. The core principle behind DToP is to identify easy-to-predict tokens at intermediate layers and exclude them from further processing. To achieve this, the model is structured into M sequential stages. After a fixed number of attention blocks, an auxiliary head computes a confidence score $c_i^{(m)}$ for each token $\mathbf{z}_i$. Tokens whose confidence exceeds a predefined threshold τ are considered high confidence and are masked out, that is, removed from subsequent computation. Low-confidence tokens $\mathbf{Z}^{(m+1)}$

continue to propagate through the subsequent encoder layers :

$$\mathbf{Z}^{(m+1)} = \left\{ \mathbf{z}_i \;\middle|\; c_i^{(m)} < \tau \right\} \tag{4}$$

Similar to DToP, our implementation employs auxiliary heads that adopt the attention-to-mask (ATM) module [7]. These heads are architecturally identical to the final decoder head, ensuring consistent behavior across all stages. The effectiveness of such an early-pruning mechanism heavily depends on the proper placement of the auxiliary heads. If placed too early in the network, the model may fail to generate reliable predictions, as the representations are not yet sufficiently informative. Conversely, if the auxiliary heads are positioned too late in the network, most of the computational cost has already been incurred by the time pruning occurs. As a result, the potential savings in inference time and FLOPs are significantly reduced, defeating the main purpose of early exiting. The authors of DToP introduce auxiliary heads at specific layers, namely the 6th and 8th for ViT-Base, and the 8th and 16th for ViT-Large. Although this configuration yields a reasonable trade-off between computational cost and segmentation accuracy, it remains largely empirical and lacks a principled justification. The exploration of pruning positions is limited to a small set of static configurations, and the impact of pruning positions on inference time is not explicitly discussed. We argue that further investigations are needed to establish more generalizable and adaptive guidelines for auxiliary head placement. This includes studying the internal evolution of token difficulty, exploring data- or budget-adaptive strategies, and considering the impact of auxiliary head placement on real-time inference efficiency.

4 Experiments

This section presents a detailed evaluation of our STEP mechanism on widely used benchmarks, focusing on both predictive accuracy and computational efficiency. The evaluation begins with a description of the main architectural and hyperparameter choices involved in the design of our STEP method. In particular, we analyze the impact of threshold parameters that control the semantic-aware patch merging via dCTS (Section 4.2). We also investigate the placement and configuration of early-exit branches, with a focus on their number and depth within the transformer architecture (Section 4.3). To comprehensively assess STEP performance and isolate the effect of each component, mean Intersection over Union (mIoU) is used to evaluate segmentation accuracy, while GFLOPs (giga floating-point operations) provide an estimate of the model's computational complexity. GFLOPs are computed using the fvcore package[1], ensuring consistent measurement across all configurations. These metrics are reported for both standard-resolution and high-resolution settings (Section 4.4).

4.1 Experimental Setup

We integrate STEP [7] into the SegViT semantic segmentation framework. All experiments are conducted using MMSegmentation[2][63], an open-source PyTorch-based

[1] https://github.com/facebookresearch/fvcore

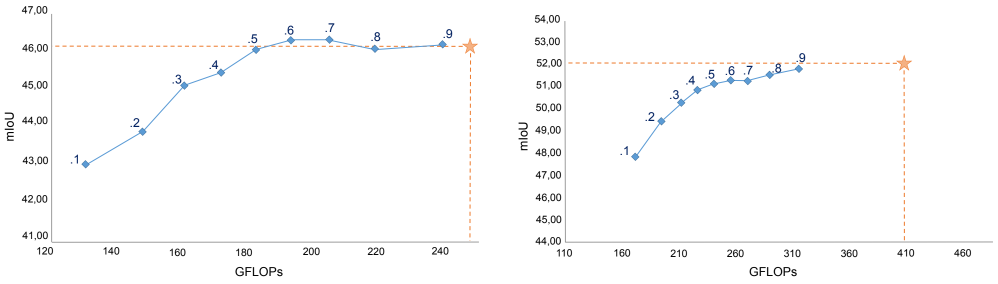
[2] https://github.com/open-mmlab/mmsegmentation

library that facilitates flexible backbone integration. We evaluate our approach using both ViT-Base and ViT-Large models. ViT-Base includes 12 transformer encoder layers, a hidden size of 768, and 12 attention heads, while ViT-Large consists of 24 layers, a 1024-dimensional hidden state, and 16 attention heads. In both cases, input images are first divided into a non-overlapping 16×16 pixel grid of patches. Experiments are conducted on three widely used semantic segmentation benchmarks: COCOStuff10k [64], which includes a wide variety of objects in complex, real-world scenes, ADE20K [65], a comprehensive dataset for scene parsing, and Cityscapes [66], which focuses on urban street scenes with high-quality pixel-level annotations. The standard evaluation is conducted using fixed input resolutions, namely 512×512 in accordance with commonly adopted benchmarking protocols. To evaluate scalability under high-resolution conditions, additional experiments are conducted on the Cityscapes dataset, which offers images with a consistent resolution of 2048×1024. The DToP confidence threshold is set to 0.95 for COCOStuff10k, and 0.9 for ADE20K and Cityscapes. Optimization is performed using AdamW with an initial learning rate of 6e-5, a weight decay of 0.01, and a cosine learning rate schedule. Training follows the standard mmseg configuration. Models are trained for 160K iterations on ADE20K, 80K iterations on COCOStuff10k, and 90K for Cityscapes with a batch size of 4. Data augmentation includes random horizontal flipping, resizing with a scale ratio between 0.5 and 2.0, and random cropping. We acknowledge that the chosen parameters may not be optimal for achieving the highest possible performance (e.g., mIoU). However, our primary objective is not to maximize accuracy, but rather to demonstrate the efficiency gains enabled by our token reduction approach.

4.2 dCTS Under Varying Thresholds

We conduct a series of experiments to determine the optimal merging threshold τ for various superpatch sizes in our dCTS approach. In this process, we assess model performance in terms of mIoU and GFLOPs, using ViT-Large as the backbone and two different datasets, with standard image resolutions typically used for segmentation tasks. This enables us to identify the best trade-off between computational efficiency and segmentation accuracy for each superpatch size. For example, when merging only 2×2 patch groups, we find that setting the threshold to $\tau = 0.4$ achieves the best trade-off between accuracy and computational cost. This configuration leads to a modest accuracy drop of approximately 1%, while reducing computational complexity by at least 30%. This trend is consistently observed across both datasets (see Figure 4).

In our dCTS approach, we apply the same principle by assigning a distinct threshold value τ to each patch group size. Specifically, we set a high threshold of 0.9 for larger patch groups, while lower values are used for smaller ones, starting from $\tau = 0.4$ for the smallest 2×2 groups. This strategy is motivated by the need to prevent errors when forming large superpatches, as incorrect merges at this scale can significantly degrade the quality of the final segmentation. Table 1 summarizes the results obtained for several threshold τ configurations. From this, 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



(a) Cocostuff10k

(b) ADE20k

Fig. 4: Tuning the merging threshold τ hyperparameter influences both segmentation accuracy and computational cost when employing the ViT-Large backbone. The blue curve illustrates the effect of varying τ when merging only 2×2 patch groups. The orange star indicates the performance of CTS with a fixed number of merged patches.

strict on segmentation quality, allowing a potential 1% loss in mIoU, but reducing complexity by 36%.

Table 1: Performance of the dCTS method on the COCOStuff10k dataset using size-dependent merging thresholds τ for different superpatch sizes 2×2, 4×4, 8×8, and 16×16.

		Threshold τ							
Metric	CTS	.6.9.9.9	.6.8.9.9	.4.9.9.9	.4.8.9.9	.4.7.9.9	.4.6.9.9	.4.5.9.9	.4.4.9.9
mIoU	46.1	45.9	46.0	45.3	44.8	43.9	44.1	43.7	43.7
GFLOPs	248	189	181	159	156	153	151	149	147

As shown in Table 2, on high-resolution images, an average of 2988 patches (out of 4096) are merged using dCTS, representing an increase compared to the 412 patches merged with CTS. It can be also observed that the merging of larger neighboring patch groups such as 8 × 8 and 16 × 16 remains relatively rare. This is consistent with the nature of Cityscapes, which mostly contains visually complex scenes with multiple objects and diverse textures, where large homogeneous regions are relatively uncommon. Nonetheless, dCTS achieves on average a 2.5× reduction in the number of patches on high-resolution images. This trend is also confirmed by experiments on other standard-resolution datasets [19], where the merging approach enables up to a 6× token reduction for highly homogeneous content, and up to 3× for more complex scenes, compared to standard fixed-grid slicing.

Table 2: Statistical insights into token pruning via STEP on the Cityscapes dataset for different input resolutions.

Metric	Input 512 × 512					Input 1024 × 1024				
	Superpatch resolution					Superpatch resolution				
	1 × 1	2 × 2	4 × 4	8 × 8	16 × 16	1 × 1	2 × 2	4 × 4	8 × 8	16 × 16
Mean	640	52	9	0.3	0	1108	452	50	10	0.8
Maximum	816	84	20	3	0	2212	656	32	21	5
Minimum	424	24	0	0	0	632	246	2	0	0

4.3 Prune Smart: Where Should Tokens Exit?

To determine the optimal placement of the auxiliary head, we conduct an ablation study of the early-exit mechanism based on DToP, using standard-resolution images commonly employed in semantic segmentation benchmarks. We choose to partition the large ViT backbone (24 encoder layers) into a maximum of three stages. For each configuration, we evaluate its impact on segmentation accuracy, computational complexity (Figure 5), inference time (Figure 6a), and the percentage of pruned tokens (Figure 6b), in order to identify the most effective positioning strategy. The results clearly demonstrate that the number and placement of auxiliary heads directly impact computational complexity and inference speed. For instance, placing two auxiliary heads at the 8th and 16th layers achieves a 22% reduction in GFLOPs (289 vs. 373), while maintaining segmentation accuracy comparable to the baseline SegViT model, which performs no token pruning. However, this gain in efficiency comes at the expense of throughput, with inference time increasing threefold compared to the unpruned baseline. In contrast, using a single auxiliary head placed deeper in the network (e.g., at the 16th or 18th layer) offers a more favorable trade-off. Although it slows inference, it still provides a significant reduction in computational cost. Figure 6b further shows that with a single pruning head the percentage of pruned tokens increases linearly, reaching around 40% on average. Remarkably, this level of pruning is comparable to what is achieved with two early-exit heads, regardless of their configuration. This suggests that a well-placed single auxiliary head can be nearly as effective as a more complex multi-head setup.

Identifying the optimal pruning configuration is a non-trivial and nuanced process. If the primary goal is to reduce computational complexity, our results indicate that splitting the large model (i.e., ViT-Large) into two stages and inserting auxiliary heads after the 8th and 16th layers yields the most effective token pruning. However, if inference speed is the main concern, a more suitable approach is to use only a single auxiliary head, positioned as early as the 16th layer, which balances token reduction with acceptable latency overhead. To adapt this strategy to smaller 12-layer architectures like ViT-Base, we interpolate our results and identify the 8th layer as the optimal position for deploying a single auxiliary head.

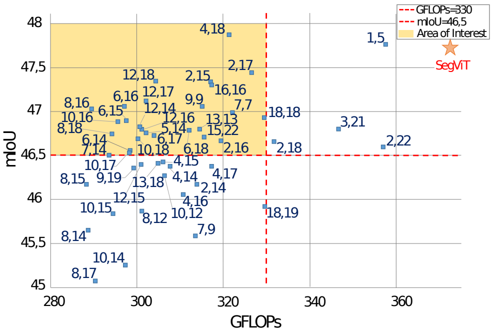
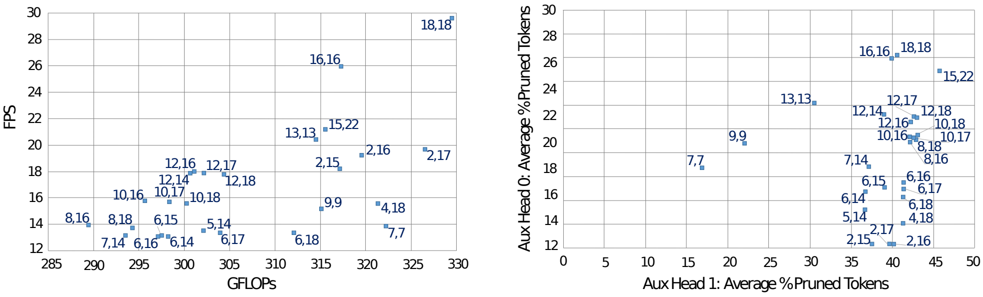


Fig. 5: Pruning head configuration analysis on the COCOStuff10k dataset. The numbered markers indicate the positions of the auxiliary heads, while the star corresponds to the performance of the baseline SegViT model without pruning. The plot illustrates the trade-off between segmentation accuracy (mIoU) and computational complexity (GFLOPs). Configurations within the yellow rectangle are selected for further analysis, as they yield at least a 10% reduction. Figure from [19].



(a) Trade-off between throughput and computational cost.

(b) Average percentage of pruned tokens per configuration.

Fig. 6: Exploration of the pruning head configuration on the COCOStuff10k dataset. Figure from [19].

4.4 Results and Discussion

To enable a fair evaluation, we compare our token reduction mechanism against SegViT in its original form. We also construct its pruned variants by applying state-of-the-art token reduction techniques. Specifically, the CTS method is used for patch merging, followed by soft token pruning using DToP. A combined configuration incorporating both techniques is also evaluated, as it represents a 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. For experiments on ViT-Base, we adopt the configuration proposed by the original DToP, placing auxiliary pruning heads after the 6th and 8th encoder layers. In addition, we evaluate our own strategy by applying a single pruning head after the 8th layer, as a lighter alternative aiming for better inference efficiency.

Table 3 reports the performance of STEP integrated into ViT-L with standard low-resolution inputs. The results indicate that STEP achieves segmentation accuracy comparable to the baseline, with mIoU degradation remaining below 2.5% across configurations. Moreover, it consistently yields a substantial reduction in computational complexity across different datasets. Introducing two auxiliary heads further amplifies this gain, achieving up to a 2.8× reduction in GFLOPs. However, this comes at the cost of significantly lower throughput. To ensure the robustness of our conclusions, we also replicate the experiments using ViT-Base as the backbone. The corresponding results are reported in Table 4. Our STEP method achieves up to a 2.5× reduction in computational cost compared to the SegViT baseline, while incurring an accuracy drop comparable to that observed with ViT-Large. We further assess the effect of patch fusion on performance using our dCTSτ-6899 variant. Notably, it achieves the best trade-off by reaching 48.2 mIoU on ADE20k, which is identical to the baseline, while requiring only 73 GFLOPs and delivering a high inference speed of 98 FPS, nearly twice as fast.

Table 3: Performance evaluation of our STEP mechanism integrated into ViT-Large.

	ADE20k (512 × 512)			COCOStuff10k (512 × 512)		
Method	mIoU$\nearrow$	GFLOPs$\searrow$	FPS$\nearrow$	mIoU$\nearrow$	GFLOPs$\searrow$	FPS$\nearrow$
SegViT	53.0	624	38	46.7	373	44.5
+CTS[1]	52.0	410	41	46.2	251	40
+DToP[1]	52.3	465	6	46.6	290	15
+CTS[1] & DToP[1]	51.2	334	12.5	45.4	210	17
+STEP@[8,16]τ-6899	51.2	224	14	46.0	173	18
+STEP@[8,16]τ-4999	50.8	334	15	45.3	150	20
+STEP@[18]τ-6899	51.7	395	22	46.0	201	30
+STEP@[18]τ-4999	50.4	261	26.5	45.1	177	29

[1] Default configuration from the original paper

The results in Table 5 and Table 6 highlight how our STEP mechanism and the dCTS patch merger effectively handle varying image resolutions. As the resolution increases to 768 × 768 and 1024 × 1024, SegViT suffers a dramatic increase in computational cost and a substantial drop in inference speed. Our STEP configurations maintain a more stable trade-off. When using ViT-Base as the backbone, we observe

Table 4: Performance evaluation of our STEP mechanism integrated with ViT-Base.

	ADE20k (512 × 512)			Cityscapes (512 × 512)		
Method	mIoU$\nearrow$	GFLOPs$\searrow$	FPS$\nearrow$	mIoU$\nearrow$	GFLOPs$\searrow$	FPS$\nearrow$
SegViT	48.3	113	53	67.7	110	70
+CTS[1]	47.8	75	40	67.6	73	56
+DToP[1]	45.8	91	25	68.2	82	24
+CTS[1] & DToP[1]	46.3	62	25	67.5	59.5	27
+dCTS τ-6899	48.2	73	98	67.5	77	66
+STEP@[6,8]τ-6899	46.9	64	24	67.2	61	22
+STEP@[6,8]τ-4999	45.3	50	34	64.3	44	26
+STEP@[8]τ-6899	47.1	68	32	67.4	66	32
+STEP@[8]τ-4999	45.8	53	43	64.2	44	32

[1] Default configuration from the original paper

Table 5: Performance evaluation of our STEP mechanism integrated with ViT-Base.

	Cityscapes (768 × 768)			Cityscapes (1024 × 1024)		
Method	mIoU$\nearrow$	GFLOPs$\searrow$	FPS$\nearrow$	mIoU$\nearrow$	GFLOPs$\searrow$	FPS$\nearrow$
SegViT	73.7	301	65	75.2	670	24
+CTS[1]	72.9	190	45	74.9	403	46
+DToP[1]	73.5	198	22	75.0	430	16
+CTS[1] & DToP[1]	72.7	135	22	75.0	296	21
+dCTS τ-6899	72.8	182	68	72.7	247	62
+STEP@[6,8]τ-6899	72.6	131	21	72.0	183	25
+STEP@[6,8]τ-4999	69.8	95	22	71.0	149	25
+STEP@[8]τ-6899	69.9	149	28	72.0	199	35
+STEP@[8]τ-4999	69.9	105	22	71.1	163	36

[1] Default configuration from the original paper

that the reduction in FLOPs is most significant compared to the two reference methods, CTS and DToP. However, this does not consistently translate into proportional gains in throughput. In this case, a noticeable drop (up to 4%) in segmentation quality is observed. This suggests that the confidence threshold used in our early-pruning mechanism may need to be re-evaluated to better balance efficiency and accuracy. The dCTS, particularly itsτ-6899 variant, emerges as a strong compromise between accuracy and efficiency. It consistently delivers high mIoU across both backbones and resolutions, with much lower GFLOPs and significantly improved throughput. For example, on ViT-Large at 1024 × 1024, dCTS achieves only a marginal drop in mIoU while requiring just 802 GFLOPs, which is 2.6 times less complex than SegViT, and reaches 41 FPS, surpassing SegViT's 12 FPS by more than a factor of three. This demonstrates the capacity of dCTS to maintain segmentation quality while significantly enhancing inference efficiency in high-resolution and real-time applications. By applying the complete STEP framework, computational complexity can be reduced by as much as 4×. Overall, the higher the input image resolution and the larger the backbone, the more our STEP approach exhibits clear advantages in terms of both efficiency and segmentation performance

Table 6: Performance evaluation of our STEP mechanism integrated with ViT-Large.

	Cityscapes (768 × 768)			Cityscapes (1024 × 1024)		
Method	mIoU$\nearrow$	GFLOPs$\searrow$	FPS$\nearrow$	mIoU$\nearrow$	GFLOPs$\searrow$	FPS$\nearrow$
SegViT	74.4	970	37	75.7	2086	12
CTS[1]	74.5	622	40.5	75.7	1283	20.5
DToP[1]	73.7	589	13	75.4	1176	7.5
+dCTSτ-6899	74.4	598	47	74.9	802	41
+STEP@[8,16]τ-6899	73.6	424	13	73.8	514	13.5
+STEP@[18]τ-6899	74.3	490	23.5	74.5	655	20.5

[1] Default configuration from the original paper

We observe that strategically placing the pruning heads allows for greater reduction in GFLOPs. This is likely due to the fact that, regardless of the configuration, an average 48% of tokens can be halted early in the network for ViT-Large under high-resolution images. Adding STEP to ViT-B when processing standard-resolution images results in an average pruning of 39% of tokens (Table 7). This is a consistent trend, which we also observed in the ablation study (4.3) conducted on COCOStuff10k at the same resolution. Figure 8 and Figure 7 illustrates how tokens are halted across images by each auxiliary head, revealing that many tokens are pruned early in simple scenarios, while they are retained until the final prediction phase in more complex scenes.

Table 7: Token pruning dynamics per auxiliary head in STEP on Cityscapes for various input resolutions.

	Input 512 × 512 ViT Base			Input 1024 × 1024 ViT Large		
	STEP@[6,8]		STEP@[8]	STEP@[8,16]		STEP@[18]
After	Aux1	Aux2	Aux1	Aux1	Aux2	Aux1
Mean	245	269	273	717	885	833
Maximum	400	422	446	1143	1340	1238
Minimum	107	127	101	300	362	247

To better understand why total throughput does not scale linearly with FLOPs savings, we present Figure 9 and Figure 10, which show the per-layer inference time and computational cost with STEP applied to both ViT-Base and ViT-Large under high-resolution. Although fewer tokens are processed in the later layers, identifying and discarding high-confidence tokens introduces bottlenecks. Despite the low computational cost of the auxiliary pruning head based on the ATM module, the masking operations likely cause the observed slowdown due to their irregular control flow, dynamic memory access patterns, and tensor shape variability. Further analyse of computation flow is necessary to verify if all operations are realized on GPU and there is unnecessary memory copy between GPU and CPU which can effectively slow the whole process. Additional overheads, such as memory allocation and synchronization costs, may further diminish the expected performance gains. These findings suggest that optimizing only the number of tokens is insufficient, one must also consider the computational efficiency of the pruning mechanism itself.

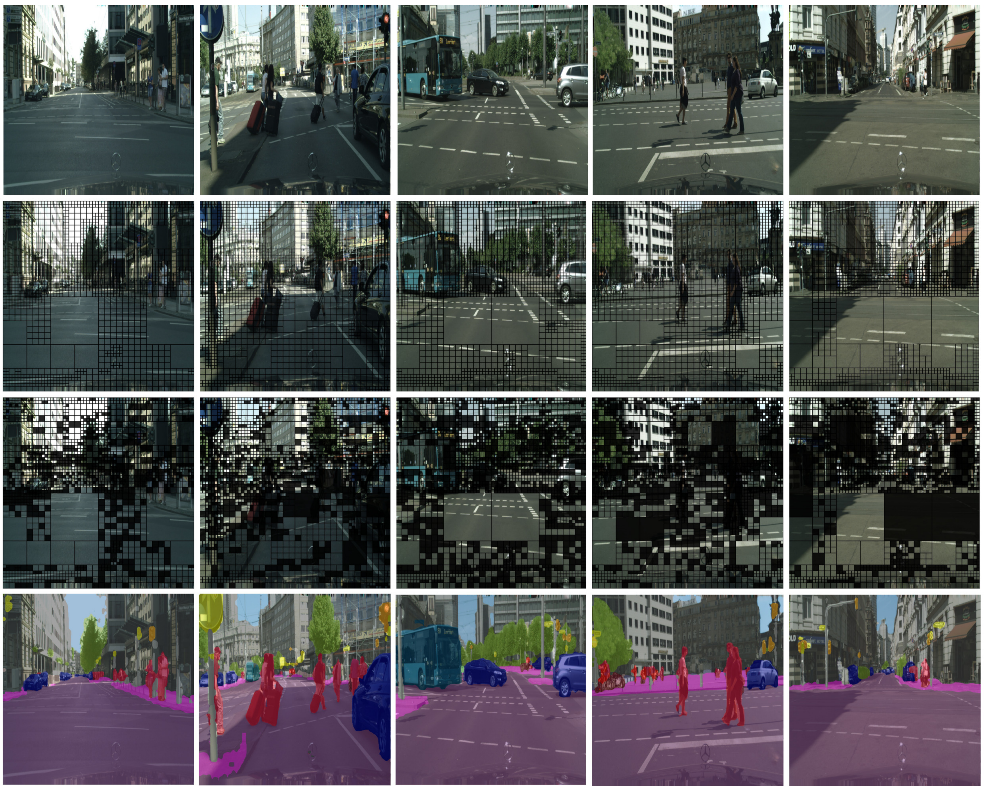


Fig. 7: Distribution of pruned supertokens across the different stages using STEP@[8] on ViT-Base. From top to bottom: input image from the Cityscapes dataset at 1024 × 1024 resolution, generated superpaches via dCTS, pruned tokens marked in black, and final segmentation results.

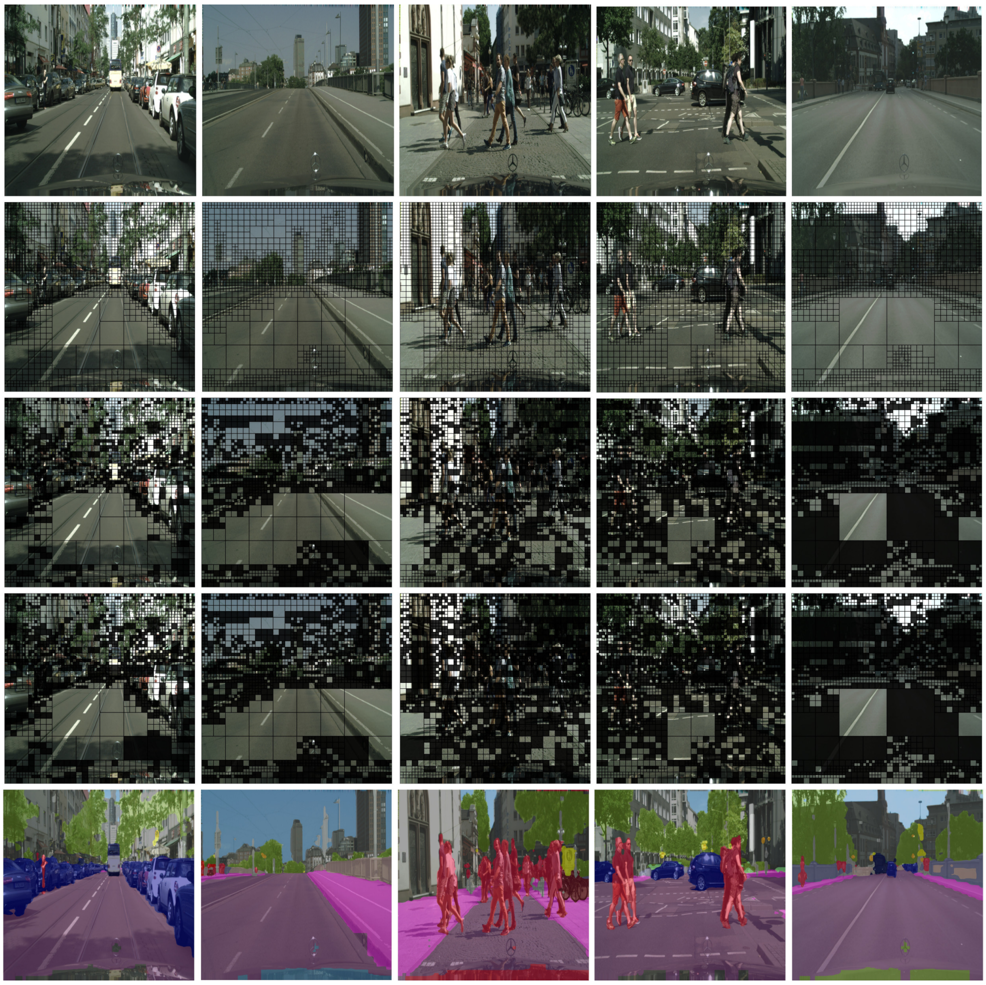
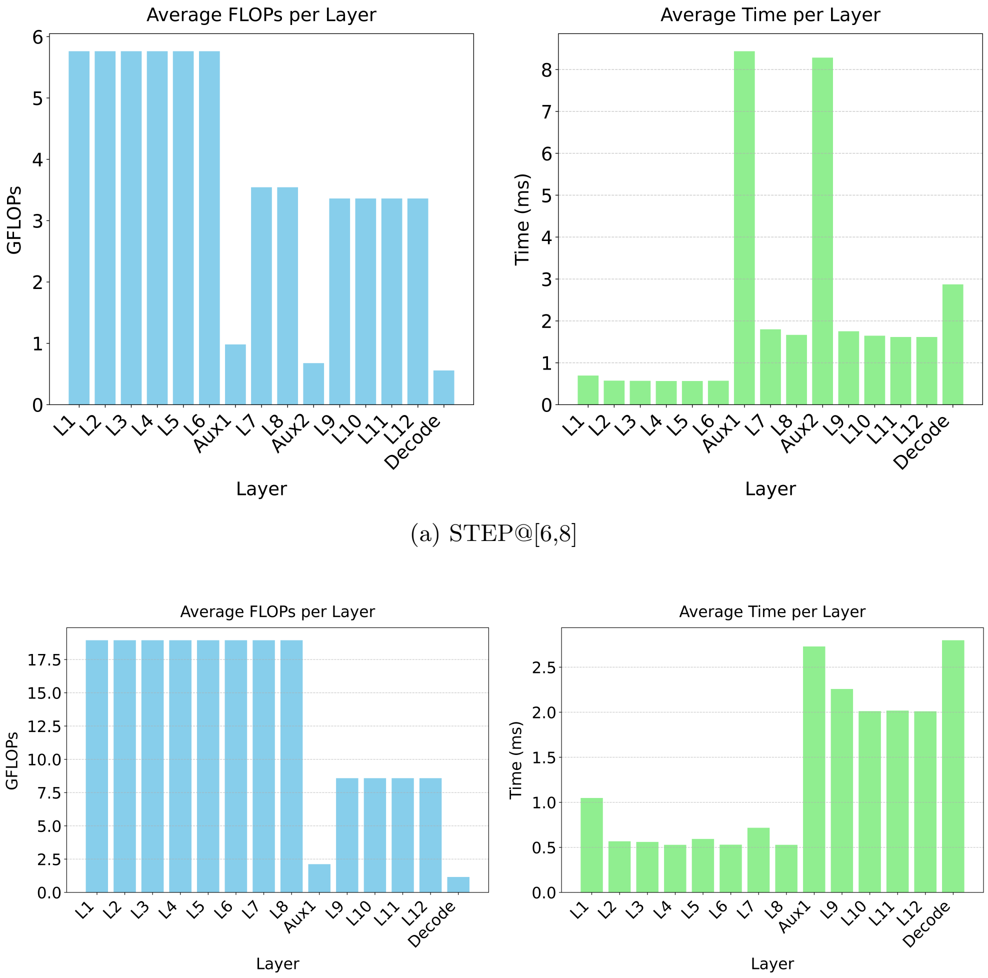


Fig. 8: Distribution of pruned supertokens across the different stages using STEP@[8,16] on ViT-Large. From top to bottom: input image from the Cityscapes dataset at 1024 × 1024 resolution, generated superpaches via dCTS, pruned tokens marked in black for auxiliary heads 1 and 2, and final segmentation results.

5 Conclusion

We introduced a novel token reduction method, SuperToken and Early-Pruning (STEP), designed to improve token efficiency in ViTs for semantic segmentation. STEP combines adaptive patch merging with an early-pruning mechanism. At the core of this method lies an enhanced patch-level merging technique, referred to as dCTS, which employs a flexible strategy to form square-shaped superpatches of varying sizes, allowing the model to better capture the spatial complexity of image content. Additionally, we investigated the benefits of early-pruning tokens via DToP within the

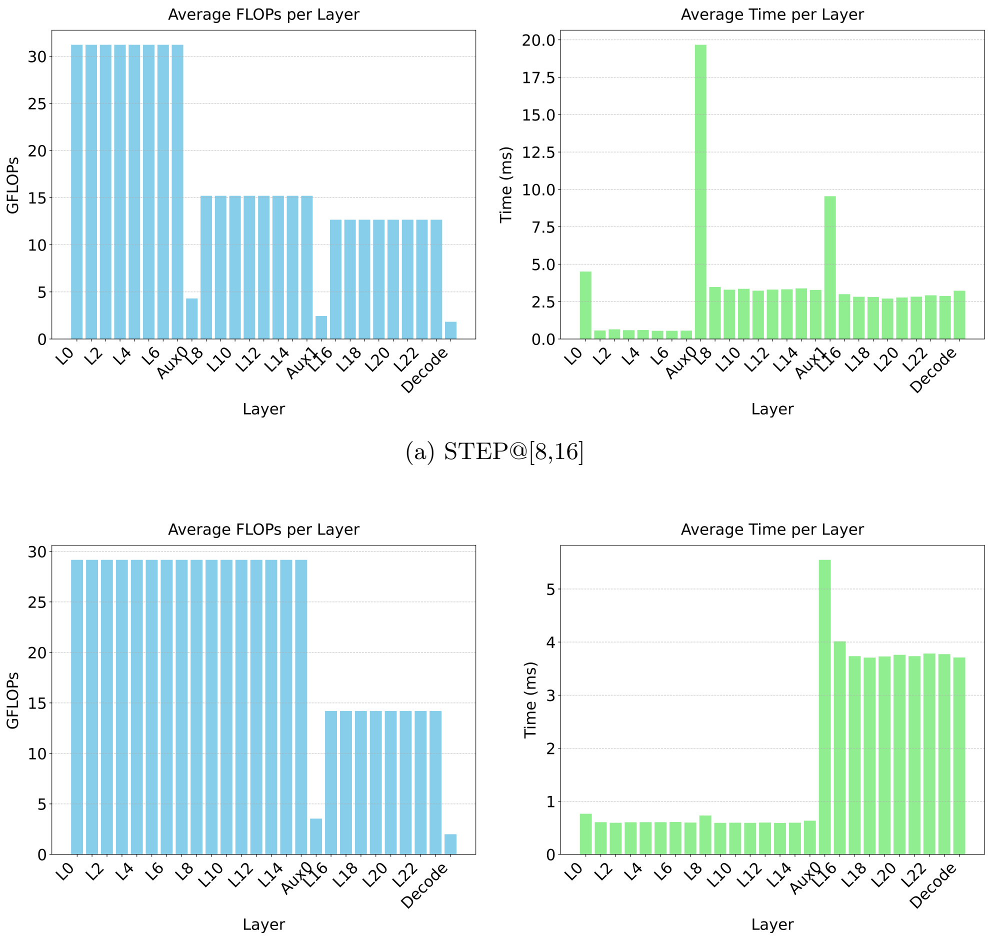


(a) STEP@[6,8]

(b) STEP@[8]

Fig. 9: Per-layer computational complexity (GFLOPs) and throughput (FPS) analysis across encoder and auxiliary pruning heads using ViT-Base as backbone on the Cityscapes dataset at 1024×1024 resolution.

network. Our experiments were conducted under varying image resolution settings, encompassing both low and high-resolution inputs. To the best of our knowledge, this is the first work to systematically assess the effect of token pruning across different resolution levels. STEP demonstrated strong scalability on both ViT-Base and ViT-Large with high-resolution images, offering substantial computational savings while preserving most of the segmentation accuracy, making it a compelling choice for efficient dense prediction in high-resolution scenarios. However, this efficiency gain does not always



(a) STEP@[8,16]

(b) STEP@[18]

Fig. 10: Per-layer computational complexity (GFLOPs) and throughput (FPS) analysis across encoder and auxiliary pruning heads using ViT-Large as backbone on the Cityscapes dataset at 1024×1024 resolution.

resulted in a proportional throughput improvement. The dCTS alone showed particularly strong robustness to increasing input resolutions. Across all configurations, a small but consistent drop in mIoU was observed compared to the baseline, suggesting that some relevant tokens may have been prematurely discarded or merged. This highlights the importance of carefully tuning the fusion thresholds within STEP especially when operating under high-resolution regimes. The early pruning mechanism using ATM-based auxiliary heads allows up to 48% of tokens to be halted. While this further reduces computational complexity, it significantly slows down inference, regardless of image resolution. This is likely due to hardware inefficiencies introduced by the masking mechanism. Further work could focus on analyzing current solution and improving algorithm design to better leverage GPU parallelism and avoid GPU-to-CPU data transfers.

References

[1] Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 568–578 (2021)

[2] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., *et al.*: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6881–6890 (2021)

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

[4] Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7262–7272 (2021)

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

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

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

[8] Kerssies, T., Cavagnero, N., Hermans, A., Norouzi, N., Averta, G., Leibe, B., Dubbelman, G., Geus, D.: Your vit is secretly an image segmentation model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)

[9] Yoo, J., Ko, D., Kim, G.: Ccaseg: Decoding multi-scale context with convolutional cross-attention for semantic segmentation. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV), pp. 9461–9470 (2025)

[10] Yeom, S., Klitzing, J.: U-mixformer: Unet-like transformer with mix-attention

for efficient semantic segmentation. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV), pp. 7710–7719 (2025)

[11] Hu, X., Jiang, L., Schiele, B.: Training vision transformers for semi-supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4007–4017 (2024)

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

[13] Yuan, Z., Xue, C., Chen, Y., Wu, Q., Sun, G.: 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, pp. 191–207. Springer, Berlin, Heidelberg (2022). https://doi.org/10.1007/978-3-031-19775-8_12 . https://doi.org/10.1007/978-3-031-19775-8_12

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

[15] Huang, X., Shen, Z., Dong, P., Cheng, K.-T.: Quantization variation: A new perspective on training transformers with low-bit precision. Transactions on Machine Learning Research (2024)

[16] Shang, Y., Liu, G., Kompella, R., Yan, Y.: Quantized-vit efficient training via fisher matrix regularization. In: MultiMedia Modeling: 31st International Conference on Multimedia Modeling, MMM 2025, Nara, Japan, January 8–10, 2025, Proceedings, Part III, pp. 270–284. Springer, Berlin, Heidelberg (2025). https://doi.org/10.1007/978-981-96-2064-7_20 . https://doi.org/10.1007/978-981-96-2064-7_20

[17] Wu, K., Zhang, J., Peng, H., Liu, M., Xiao, B., Fu, J., Yuan, L.: Tinyvit: Fast pretraining distillation for small vision transformers. In: European Conference on Computer Vision (ECCV) (2022)

[18] Yang, Z., Li, Z., Zeng, A., Li, Z., Yuan, C., Li, Y.: ViTKD: Feature-based Knowledge Distillation for Vision Transformers . In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1379–1388. IEEE Computer Society, Los Alamitos, CA, USA (2024). https://doi.org/10.1109/CVPRW63382.2024.00145 . https://doi.ieeecomputersociety.org/10.1109/CVPRW63382.2024.00145

[19] Proust, M., Poreba, M., Szczepanski, M., Haroun, K.: Step: Supertoken and early-pruning for efficient semantic segmentation. In: VISIGRAPP 2025-20th

International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pp. 56–61 (2025). https://doi.org/10.5220/0013132800003912 . https://www.scitepress.org/Papers/2025/131328/131328.pdf

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

[21] Havtorn, J.D., Royer, A., Blankevoort, T., Bejnordi, B.E.: MSViT: Dynamic mixed-scale tokenization for vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 838–848 (2023)

[22] Chen, M., Lin, M., Li, K., Shen, Y., Wu, Y., Chao, F., Ji, R.: Cf-vit: A general coarse-to-fine method for vision transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37

[23] Ronen, T., Levy, O., Golbert, A.: Vision transformers with mixed-resolution tokenization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4613–4622

[24] Mahmud, T., Yaman, B., Liu, C.-H., Marculescu, D.: PaPr: Training-Free One-Step Patch Pruning with Lightweight ConvNets for Faster Inference (2024). https://arxiv.org/abs/2403.16020

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

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

[27] Kim, S., Shen, S., Thorsley, D., Gholami, A., Kwon, W., Hassoun, J., Keutzer, K.: Learned token pruning for transformers. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. KDD '22, pp. 784–794. Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3534678.3539260 . https://doi.org/10.1145/3534678.3539260

[28] Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., *et al.*: 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, pp. 620–640 (2022). Springer

[29] Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not all patches are what

you need: Expediting vision transformers via token reorganizations. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=BjyvwnXXVn_

[30] Meng, L., Li, H., Chen, B.-C., Lan, S., Wu, Z., Jiang, Y.-G., Lim, S.-N.: AdaViT: Adaptive Vision Transformers for Efficient Image Recognition . In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12299–12308. IEEE Computer Society, Los Alamitos, CA, USA (2022). https://doi.org/10.1109/CVPR52688.2022.01199 . https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01199

[31] Song, Z., Xu, Y., He, Z., Jiang, L., Jing, N., Liang, X.: CP-ViT: Cascade Vision Transformer Pruning Via Progressive Sparsity Prediction. https://doi.org/10.48550/arXiv.2203.04570

[32] Marin, D., Chang, J.-H.R., Ranjan, A., Prabhu, A., Rastegari, M., Tuzel, O.: Token pooling in vision transformers for image classification. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 12–21 (2023). https://doi.org/10.1109/WACV56688.2023.00010

[33] Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your ViT but faster. In: International Conference on Learning Representations (2023)

[34] Tang, Q., Zhang, B., Liu, J., Liu, F., Liu, Y.: Dynamic token pruning in plain vision transformers for semantic segmentation. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 777–786. IEEE Computer Society, Los Alamitos, CA, USA (2023). https://doi.org/10.1109/ICCV51070.2023.00078 . https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00078

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

[36] Liu, X., Wu, T., Guo, G.: Adaptive sparse vit: Towards learnable adaptive token pruning by fully exploiting self-attention, pp. 1222–1230 (2023). https://doi.org/10.24963/ijcai.2023/136

[37] Marchetti, M., Traini, D., Ursino, D., Virgili, L.: Efficient token pruning in vision transformers using an attention-based multilayer network. Expert Systems with Applications **279**, 127449 (2025) https://doi.org/10.1016/j.eswa.2025.127449

[38] Wang, H., Dedhia, B., Jha, N.K.: Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers, pp. 16070–16079 (2024). https://doi.org/10.1109/CVPR52733.2024.01521

[39] Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., Zhang, L., Xu, C., Sun, X.: Evo-vit: Slow-fast token evolution for dynamic vision transformer.

In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 2964–2972 (2022)

[40] Courdier, E., Sivaprasad, P.T., Fleuret, F.: PAUMER: Patch Pausing Transformer for Semantic Segmentation (2023). https://arxiv.org/abs/2311.00586

[41] Liu, Y., Zhou, Q., Wang, J., Wang, Z., Wang, F., Wang, J., Zhang, W.: Dynamic token-pass transformers for semantic segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1816–1825 (2024). https://doi.org/10.1109/WACV57701.2024.00184

[42] Liu, Y., Gehrig, M., Messikommer, N., Cannici, M., Scaramuzza, D.: Revisiting token pruning for object detection and instance segmentation. In: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 2646–2656. IEEE Computer Society, Los Alamitos, CA, USA (2024). https://doi.org/10.1109/WACV57701.2024.00264 . https://doi.ieeecomputersociety.org/10.1109/WACV57701.2024.00264

[43] Yin, H., Vahdat, A., Alvarez, J.M., Mallya, A., Kautz, J., Molchanov, P.: A-ViT: Adaptive tokens for efficient vision transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10809–10818 (2022)

[44] Zeng, W., Jin, S., Liu, W., Qian, C., Luo, P., Ouyang, W., Wang, X.: 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, pp. 11101–11111 (2022)

[45] Zeng, W., Jin, S., Xu, L., Liu, W., Qian, C., Ouyang, W., Luo, P., Wang, X.: Tcformer: Visual recognition via token clustering transformer. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)

[46] Li, J., Wang, Y., ZHANG, X., Shi, B., Jiang, D., Li, C., Dai, W., Xiong, H., Tian, Q.: Ailurus: A scalable vit framework for dense prediction. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems, vol. 36, pp. 30979–30996. Curran Associates, Inc., ??? (2023). https://proceedings.neurips.cc/paper_files/paper/2023/file/62c9aa4d48329a85d1e36d5b6d0a6a32-Paper-Conference.pdf

[47] Marin, D., Chang, J.-H.R., Ranjan, A., Prabhu, A., Rastegari, M., Tuzel, O.: Token pooling in vision transformers for image classification. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 12–21 (2023)

[48] Norouzi, N., Orlova, S., De Geus, D., Dubbelman, G.: Algm: Adaptive local-then-global token merging for efficient semantic segmentation with plain vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision

and Pattern Recognition, pp. 15773–15782 (2024)

[49] Haroun, K., Martinet, J., Chehida, K.B., Allenet, T.: Leveraging local similarity for token merging in vision transformers. In: ICONIP 2024-31th International Conference on Neural Information Processing (2024)

[50] Haroun, K., Allenet, T., Chehida, K.B., Martinet, J.: Dynamic hierarchical token merging for vision transformers. In: VISAPP-2025-20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (2025)

[51] Lee, D.H., Hong, S.: Learning to merge tokens via decoupled embedding for efficient vision transformers. In: Conference on Neural Information Processing Systems (2024)

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

[53] Kim, M., Gao, S., Hsu, Y.-C., Shen, Y., Jin, H.: Token fusion: Bridging the gap between token pruning and token merging. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1383–1392 (2024)

[54] Wu, X., Zeng, F., Wang, X., Chen, X.: PPT: Token Pruning and Pooling for Efficient Vision Transformers (2024). https://arxiv.org/abs/2310.01812

[55] Chen, M., Shao, W., Xu, P., Lin, M., Zhang, K., Chao, F., Ji, R., Qiao, Y., Luo, P.: Diffrate: Differentiable compression rate for efficient vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17164–17174 (2023)

[56] Chen, D., Lin, K., Deng, Q.: Ucc: A unified cascade compression framework for vision transformer models. Neurocomputing **612**, 128747 (2025) https://doi.org/10.1016/j.neucom.2024.128747

[57] Mao, J., Shen, Y., Guo, J., Yao, Y., Hua, X., Shen, H.: Prune and merge: Efficient token compression for vision transformer with spatial information preserved. IEEE Transactions on Multimedia **PP**, 1–14 (2025) https://doi.org/10.1109/TMM.2025.3535405

[58] Huang, H., Zhou, X., Cao, J., He, R., Tan, T.: Vision transformer with super token sampling. arXiv:2211.11167 (2022)

[59] Zeng, W., Jin, S., Liu, W., Qian, C., Luo, P., Ouyang, W., Wang, X.: 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, pp. 11101–11111 (2022)

[60] Rao, Y., Liu, Z., Zhao, W., Zhou, J., Lu, J.: Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(9), 10883–10897 (2023) https://doi.org/10.1109/TPAMI.2023.3263826

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

[62] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015) https://doi.org/10.1007/s11263-015-0816-y

[63] MMSegmentation Contributors: MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. https://github.com/open-mmlab/mmsegmentation (2020)

[64] Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1209–1218. IEEE Computer Society, Los Alamitos, CA, USA (2018). https://doi.org/10.1109/CVPR.2018.00132 . https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00132

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

[66] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)