When Token Compression Breaks: Structural Pruning vs. Token Reduction for Robust ViT Segmentation under High Compression

Tien-Phat Nguyen and Ngai-Man Cheung

Temasek Laboratories, Singapore University of Technology and Design, Singapore
{tienphat_nguyen, ngaiman_cheung}@sutd.edu.sg

Abstract. Vision Transformers (ViTs) are strong backbones for semantic segmentation, but their computational cost limits deployment. Recent token compression methods for efficient transformer-based segmentation reduce this cost by decreasing the number of tokens. However, existing evaluations primarily focus on low-to-moderate compression, leaving their behavior under aggressive compression and corrupted inputs unclear. Meanwhile, structural pruning provides an orthogonal route to efficiency by removing redundant components in the ViT architecture, but is rarely compared to token compression under a unified protocol. To bridge this gap, we benchmark representative token compression and structural pruning methods for ViT-based semantic segmentation under matched FLOPs on ADE20K and Cityscapes, together with their common-corruption variants ADE20K-C and Cityscapes-C. Our results reveal a consistent trend on both clean and corrupted inputs: token compression is highly effective at mild reductions but degrades sharply when compression becomes severe, consistent with substantial information loss from overly aggressive token reduction. In contrast, structural pruning exhibits a smoother degradation curve and is more stable at high compression. Motivated by these findings, we study a *prune-then-merge* pipeline that applies moderate token compression on top of a moderately pruned backbone. At comparable FLOPs, this combined strategy consistently achieves a better accuracy-robustness trade-off at high compression, offering a practical recipe for deployment-oriented ViT segmentation. Code is available at `https://github.com/phatnguyencs/vit-seg-compression`.

Keywords: Token Compression · Compact Transformers · Network Pruning · Semantic Segmentation

1 Introduction

Vision Transformer (ViT) [8] encoders are now a common choice for semantic segmentation, producing high-performance architectures [19,30,35,39]. However, their deployment remains challenging as self-attention scales quadratically with token count, and high-resolution segmentation requires processing large spatial

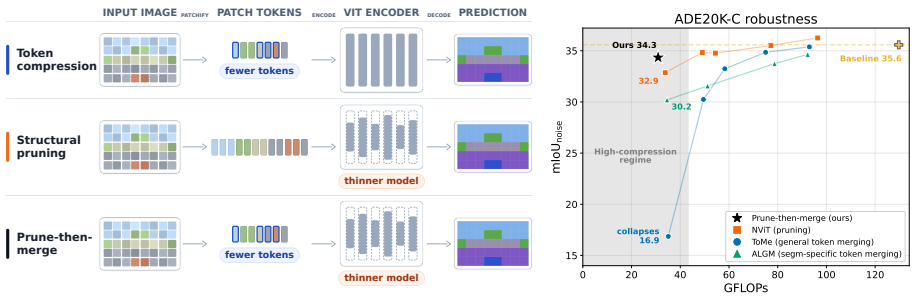

Fig. 1: Compression strategies and their robustness-compute trade-off. *Left:* token compression reduces the number of processed tokens, structural pruning reduces model width/capacity, and our *prune-then-merge* pipeline combines both techniques. *Right:* a summary view of the robustness-compute trade-off on ADE20K-C, measured by corrupted-input mIoU versus compute. Token compression can preserve performance at mild reductions but drops sharply in the high-compression regime. Structural pruning degrades more smoothly, while *prune-then-merge* provides a stronger trade-off at comparable FLOPs.

token grids [2]. This motivates compression and acceleration techniques that reduce computation while preserving segmentation quality. At the same time, real-world segmentation systems must remain reliable under distribution shifts such as noise, blur, and weather artifacts [17]. Therefore, clean accuracy and FLOPs alone are insufficient for evaluating compressed segmentation models, and corruption robustness must also be considered.

To improve the efficiency of ViT segmentation models, two major compression strategies have been widely explored. Token compression [22, 24, 26, 31] is a data-adaptive strategy that reduces the number of tokens processed by the transformer, for example by pruning redundant tokens or merging spatially similar tokens. Structural pruning [10,29,36], in contrast, is an architecture-centric strategy that removes redundant model components, such as attention heads, embedding dimensions, or FFN channels, to obtain a permanently compact backbone. Fig. 1 left summarizes these two strategies and our *prune-then-merge* combination, which applies token compression on top of a structurally pruned backbone.

Despite extensive progress in both compression directions, two questions remain underexplored for semantic segmentation. First, representative token compression and structural pruning methods are rarely compared under a unified segmentation pipeline with matched FLOPs. This makes it difficult to assess their performance and corruption robustness across a wide range of compression levels. Because these methods act on different dimensions (tokens versus network capacity), their degradation behaviors can differ substantially, especially under aggressive compression. Second, although pruning and token compression are potentially complementary, it remains unclear whether applying moderate token compression on top of a pruned backbone yields a stronger accuracy-robustness trade-off than using either technique alone at comparable FLOPs.

To address these gaps, we benchmark representative token compression and structural pruning methods within a controlled flat-token ViT segmentation pipeline on ADE20K [40] and Cityscapes [7], together with their corruption variants ADE20K-C and Cityscapes-C [17], under matched-FLOPs comparisons across a wide range of compression levels. Our benchmark shows that the two compression strategies exhibit distinct degradation behaviors. Token compression can preserve performance at low-to-moderate reduction, particularly for segmentation-aware methods, but degrades sharply under severe compression, with substantial drops in both clean performance and corruption robustness. In contrast, structural pruning exhibits a smoother degradation curve and is generally more stable at high compression. To better understand this behavior, we further analyze feature diversity and qualitative predictions, showing that aggressive token compression can reduce representation rank and produce less coherent predictions around fine structures and local boundaries. Motivated by these findings, we study a *prune-then-merge* pipeline: first apply moderate structural pruning to obtain a compact backbone, then apply moderate token compression on top. Under comparable FLOPs, this stacked strategy yields a stronger accuracy-robustness trade-off at high compression, providing a practical recipe for ViT segmentation. Fig. 1 right provides an overview of these findings from the robustness-compute perspective on ADE20K-C.

In summary, our contributions are as follows:

- **Unified benchmark across compression regimes.** We benchmark representative token compression and structural pruning methods for flat-token ViT segmentation under matched FLOPs on ADE20K, Cityscapes, and their corrupted variants. Our results show that token compression can preserve performance at low-to-moderate reduction but becomes fragile at high compression, while structural pruning is more stable and better preserves corruption robustness. We complement accuracy-compute curves with effective-rank analysis and qualitative examples, showing that aggressive token compression can reduce feature diversity and produce less coherent predictions around fine structures and local boundaries.
- **Practical prune-then-merge strategy.** We study a stacked pipeline that applies moderate token compression on top of a pruned backbone, achieving a stronger trade-off between accuracy and robustness than using either strategy alone under comparable compute at high compression.

2 Related Work

2.1 Vision Transformers for Semantic Segmentation

Vision Transformers were first introduced to semantic segmentation by SETR [39], which repurposed a standard ViT encoder for dense prediction by formulating segmentation as a sequence-to-sequence task. Segmenter [30] further established ViTs as effective segmentation backbones by adopting a fully transformer-based architecture with a mask transformer decoder, achieving strong results on

ADE20K and Cityscapes while explicitly highlighting the computational cost of global self-attention at high resolution. SegFormer [35] addressed part of this issue by introducing a hierarchical transformer encoder and a lightweight decoder, improving efficiency while maintaining competitive accuracy. Subsequent segmentation frameworks [5, 6, 16] have further advanced segmentation quality through improved decoder designs, training formulations, and multi-task unification, while typically relying on strong pretrained transformer backbones.

An alternative direction focuses on designing efficient transformer architectures from scratch. TopFormer [37] targets mobile settings through token pyramids, and EfficientViT [2] replaces softmax attention with linear attention to significantly reduce latency. While effective, these approaches require architectural redesign and retraining. Compression of pretrained backbones provides a complementary route for efficient segmentation, since it can reuse pretrained ViT backbones and adapt them to lower compute budgets rather than requiring a newly designed architecture [32]. Our work follows this direction by compressing pretrained ViT backbones within a controlled flat-token segmentation pipeline.

2.2 Token Compression for Efficient Vision Transformers

Token-level compression lowers the cost of transformer inference by reducing the number of tokens being processed, either through token pruning or token merging. Many approaches are developed and validated primarily on image classification, and some are explicitly designed for classification. EViT [21] uses CLS-to-token attention for selection, while ATS [11] and SPViT [20] learn recognition-oriented token selection policies. In contrast, several methods define compression using more task-independent criteria that do not rely on a classification head, such as ToMe [1] (feature-space similarity merging), Token Fusion (ToFu) [18] (bridging pruning and merging), and DiffRate [4] (learning adaptive layer-wise reduction rates). However, token compression strategies developed for classification do not transfer straightforwardly to semantic segmentation, where dense prediction requires preserving spatial coverage for background regions and boundaries [24, 26, 31]. To address this mismatch, segmentation-aware methods design criteria tailored to dense prediction: CTS [24] merges locally redundant tokens guided by semantic consistency, ALGM [26] performs local-to-global merging with adaptive token budgets across layers, and DToP [31] reduces computation via early exiting of confidently predicted tokens.

Despite these advances, evaluations remain fragmented and often focus on low-to-moderate compression ratios, where many methods preserve accuracy. Recent work [12] provides a comparative study of token reduction but focuses on classification benchmarks and does not consider segmentation. For semantic segmentation, token compression under high compression remains less explored, although severe token reduction may cause substantial information loss and degrade robustness. Moreover, prior work rarely compares token compression against structural pruning under matched FLOPs within the same segmentation pipeline, especially under common corruptions.

2.3 Structural Pruning for Vision Transformers

Structural pruning reduces inference cost by reducing architectural components (e.g., attention heads, embedding/MLP channels, or layers), producing compact models with fewer FLOPs and parameters. Recent methods adopt principled importance criteria for cross-component pruning, such as NViT [36], which uses Hessian-based saliency with latency-aware guidance, and SAViT [38], which models joint importance across heterogeneous structures. Beyond ViT-specific designs, architecture-agnostic pruning methods improve portability across model families. DepGraph [9] achieves this by automatically modeling layer/channel dependencies during pruning. Isomorphic Pruning [10] makes importance scores comparable across different substructures, allowing a unified pruning criterion that can be used across architectures such as ViTs [8] and ConvNeXts [23].

For semantic segmentation, structural pruning has shown promising results. UPop [29] reports pruning on Segmenter for ADE20K, and NViT [36] demonstrates that ImageNet-pruned ViTs retain competitive segmentation performance after fine-tuning. However, these studies mainly treat pruning as a standalone compression strategy, without systematically comparing it with token compression or evaluating its robustness under common corruptions.

The efficiency-robustness trade-off has been studied in the classification setting. Prior work [13] shows that compressed networks can retain similar clean accuracy while disproportionately degrading on atypical, noisy, and long-tail examples, and that high compression amplifies sensitivity to distribution shifts such as ImageNet-A and ImageNet-C. We extend this perspective to ViT-based semantic segmentation by evaluating two orthogonal compression axes, token reduction and structural pruning, with explicit corruption-robustness evaluation.

A few works [14, 34] explore multi-axis compression by jointly pruning architectural and token dimensions with joint objectives. In contrast, we study a practical pipeline *prune-then-merge* and evaluate this strategy under matched compute in the high-compression regime.

3 Benchmark Protocol

This section describes our unified benchmark for comparing two orthogonal efficiency strategies in ViT-based semantic segmentation: token compression and structural pruning. We first state the benchmark objectives and research questions (Sec. 3.1). We then introduce the controlled experimental setup, including the segmentation pipeline, evaluated methods, and training protocol, and efficiency/evaluation metrics (Sec. 3.2). Finally, we describe the clean datasets, corrupted variants, and how corrupted-input robustness is measured (Sec. 3.3).

3.1 Benchmark objectives

We benchmark token compression and structural pruning within ViT-based semantic segmentation, and evaluate both clean accuracy and robustness under

common corruptions. Beyond measuring accuracy under compute constraints, our goal is to characterize how robustness evolves as compression becomes more aggressive under matched FLOPs. Concretely, we ask: **(Q1)** How do token compression and structural pruning trade off clean accuracy and corruption robustness across compute budgets? **(Q2)** Which strategy degrades more gracefully in the high-compression regime?

3.2 Experiment setup and evaluation protocol

Segmentation pipeline. All methods are evaluated under the same segmentation pipeline for controlled comparison. We adopt a ViT-based segmenter [30] consisting of a ViT encoder followed by a lightweight transformer-based decoder for dense prediction. In our setting, we use DeiT-B [33] as the encoder and keep the decoder architecture unchanged across experiments. Compression is applied only to the encoder: token-compression methods reduce the encoder token sequence, while NViT reduces encoder architectural capacity. For token-compression methods, the compressed token sequence is passed to the decoder and then restored to the original spatial grid for dense prediction, following prior ViT-segmentation token-compression pipelines [24, 26]. For pruning-based models, the full spatial token grid is preserved throughout the encoder and decoder.

Evaluated methods. We choose representative methods from both families: *(i) Token compression:* ToMe as a general-purpose method and ALGM, CTS as segmentation-specific methods. *(ii) Structural pruning:* NViT, which prunes architectural redundancy. *(iii) Prune-then-merge (PtM):* a two-stage combination that applies ToMe on top of an NViT-pruned encoder. In particular, we first select a moderately pruned NViT model as the backbone, and then sweep a small set of ToMe merging rates to match the target FLOPs budget.

Training and compression protocols. For every configuration, we finetune using the same training recipe [26], which follows the Segmenter pipeline [30], and report the best validation mIoU. All models are initialized from ImageNet-pretrained DeiT-B or NViT checkpoints and finetuned on the target segmentation dataset. We use SGD with momentum 0.9, a polynomial learning-rate schedule with power 0.9, drop-path rate 0.1, without weight decay or learning-rate warmup. Following the ALGM-Segmenter configuration [26], we train ADE20K models for 64 epochs with batch size 8, learning rate 10^{-3}, and crop size 512×512. For Cityscapes, we train for 216 epochs with batch size 8, learning rate 10^{-2}, and crop size 768×768.

For token-compression methods, we follow their original compression placement: ALGM applies token merging at encoder layers 1 and 5, CTS performs token merging at the first encoder layer, and ToMe applies token merging across all encoder layers. The compression is enabled during finetuning and evaluation, rather than being applied only as a post-hoc inference adaptation. For each method, the threshold or merging rate is swept to obtain different FLOPs budgets. The ADE20K token-compression settings for the mild, moderate, and

aggressive regimes are ALGM $T = \{0.94, 0.75, 0.65\}$, CTS $R = \{0.3, 0.6, 0.7\}$, and ToMe $R = \{40, 95, 140\}$. The corresponding Cityscapes settings are ALGM $T = \{0.94, 0.88, 0.70\}$, CTS $R = \{0.4, 0.5, 0.65\}$, and ToMe $R = \{145, 190, 280\}$.

For structural pruning, we follow the NViT global-pruning protocol [36]. NViT controls pruning strength using a latency target p, defined relative to the original DeiT-B model. We use the official NViT-B/H/S configurations when they match our target compression regimes, and additionally sweep p over a small set of latency targets, including $p \in \{0.80, 0.70, 0.25\}$, to obtain FLOPs- or FPS-matched operating points for each dataset. Unless a specific official configuration is named, we refer to these pruned models as NViT.

Efficiency metrics and evaluation. Token compression changes sequence length, whereas pruning changes architectural width/structure, so their native compression ratios are not directly comparable. We therefore use FLOPs as a common efficiency measure: for each compression technique, we measure multiple settings and compare models at similar FLOPs reductions relative to the uncompressed baseline. We report mean Intersection-over-Union (mIoU) on clean validation images ($\text{mIoU}_{\text{clean}}$) and on corrupted validation images ($\text{mIoU}_{\text{noise}}$). Throughout this work, results are presented as $\text{mIoU}_{\text{clean}}/\text{mIoU}_{\text{noise}}$. As a complementary deployment-oriented check, we report inference throughput in Sec. 4.4. FPS is measured on a single NVIDIA RTX A6000 GPU with batch size 32 and mixed precision enabled.

3.3 Dataset and corruption protocol

We evaluate on ADE20K and Cityscapes, together with their corrupted variants ADE20K-C and Cityscapes-C. For robustness evaluation, we follow the standard corruption protocol used in prior segmentation robustness work [17]: each validation image is transformed by a fixed set of corruption operators at multiple severity levels, and performance is evaluated under each corruption setting. We consider 16 corruption types grouped into four categories (Noise, Blur, Digital, Weather), each evaluated at 5 severity levels, yielding 80 corrupted variants per image. We report corrupted-input robustness by aggregating mIoU across corruption types and severity levels.

4 Benchmark Results and Insights

This section analyzes benchmark results under FLOPs-matched comparisons and highlights consistent patterns across compression levels. Rather than emphasizing isolated operating points, we focus on how performance evolves as compute decreases. Findings 1 and 2 characterize token compression across mild and aggressive regimes, while Finding 3 compares this behavior with structural pruning. Together, these results show that token redundancy can be reduced with limited accuracy loss at mild compression, but aggressive token reduction can sharply degrade both clean performance and corruption robustness. Structural pruning,

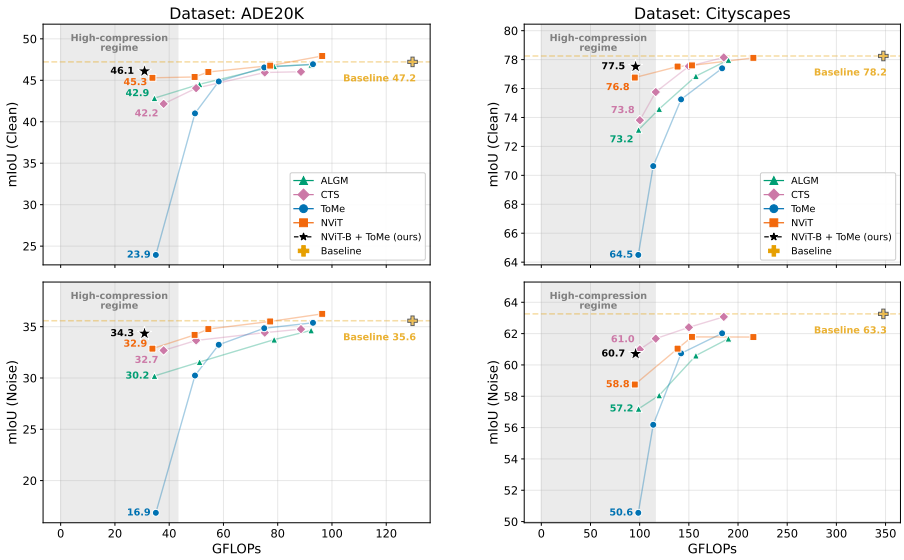

Fig. 2: Accuracy-compute trade-offs on ADE20K (left) and Cityscapes (right). We compare structural pruning (NViT), token compression methods (ToMe, ALGM, CTS), and the stack pipeline (NViT + ToMe). The top row evaluates clean accuracy $\text{mIoU}_{\text{clean}}$, while the bottom row evaluates robustness under common corruptions $\text{mIoU}_{\text{noise}}$. In both datasets, mild token compression can preserve performance, but aggressive token compression leads to sharp degradation; structural pruning shows a smoother degradation trend as compute decreases.

by contrast, reduces architectural capacity while preserving the spatial token grid, leading to a more gradual degradation trend at high compression.

4.1 Compression-regime behavior under matched FLOPs

Fig. 2 visualizes the overall accuracy-compute trade-offs of token compression, structural pruning, and their combination across different compute budgets on ADE20K and Cityscapes. Each curve shows how segmentation performance evolves as FLOPs are reduced, while Tab. 1 provides detailed numeric results for each method at different compression levels. From these results, we observe three consistent patterns across the two datasets.

Finding 1: Mild token compression can preserve near-baseline performance. Tab. 1 shows *mild* token-compression performance at low compression levels. Across both ADE20K and Cityscapes, several token-compression configurations achieve substantial compute reductions (about 1.4×–1.9×) while preserving near-baseline performance on both clean and corrupted inputs. This effect is most consistent for segmentation-aware methods (CTS, ALGM), which are

Method	ADE20K			Cityscapes		
	GFLOPs↓	$\text{mIoU}_{\text{clean}}$ ↑	$\text{mIoU}_{\text{noise}}$ ↑	GFLOPs↓	$\text{mIoU}_{\text{clean}}$ ↑	$\text{mIoU}_{\text{noise}}$ ↑
Baseline	$129.70_{\times 1.0}$	47.22	35.58	$347.77_{\times 1.0}$	78.25	63.26
Mild compression						
ALGM	$92.31_{\times 1.4}$	46.88	34.64	$190.27_{\times 1.8}$	77.98	61.68
CTS	$88.57_{\times 1.5}$	46.03	34.76	$185.52_{\times 1.9}$	**78.15**	**63.07**
ToMe	$93.00_{\times 1.4}$	<u>46.94</u>	<u>35.38</u>	$183.83_{\times 1.9}$	77.40	<u>62.02</u>
NViT	$96.40_{\times 1.4}$	**47.93**	**36.25**	$215.61_{\times 1.6}$	<u>78.12</u>	61.78
Moderate compression						
ALGM	$51.21_{\times 2.5}$	44.52	31.56	$157.19_{\times 2.2}$	76.86	60.60
CTS	$49.93_{\times 2.6}$	44.05	33.66	$150.03_{\times 2.3}$	<u>77.54</u>	**62.40**
ToMe	$49.51_{\times 2.6}$	41.01	30.24	$142.11_{\times 2.5}$	75.25	60.74
NViT-H	$49.37_{\times 2.6}$	<u>45.40</u>	<u>34.20</u>	$138.55_{\times 2.5}$	77.52	<u>61.04</u>
NViT+ToMe *(PtM)*	$49.02_{\times 2.7}$	**46.79**	**34.83**	$141.46_{\times 2.5}$	**77.58**	60.69
Aggressive compression						
ALGM	$34.58_{\times 3.8}$	42.85	30.20	$98.94_{\times 3.5}$	73.15	57.20
CTS	$37.95_{\times 3.4}$	42.16	32.69	$100.18_{\times 3.5}$	73.81	**60.99**
ToMe	$35.08_{\times 3.7}$	23.94	16.86	$98.60_{\times 3.5}$	64.50	50.56
NViT-S	$33.82_{\times 3.8}$	<u>45.30</u>	<u>32.86</u>	$95.05_{\times 3.7}$	<u>76.76</u>	58.75
NViT-B+ToMe *(PtM)*	$34.84_{\times 3.7}$	**45.71**	**34.33**	$95.87_{\times 3.6}$	**77.52**	<u>60.71</u>

Table 1: Full benchmark under matched FLOPs on ADE20K and Cityscapes at three compression levels. We compare segmentation-aware token compression (ALGM, CTS), general token compression (ToMe), structural pruning (NViT), and the proposed prune-then-merge (PtM). GFLOPs subscripts give the compute-reduction factor. Best $\text{mIoU}_{\text{clean}}$/$\text{mIoU}_{\text{noise}}$ per dataset and level in **bold**, second best <u>underlined</u>. For NViT rows without an official B/H/S suffix, the latency target p is swept per dataset to match the corresponding compute regime.

designed to maintain spatial information during token reduction. For instance, on ADE20K, CTS reduces compute from 129.70 to 88.57 GFLOPs (1.5×), with only a small drop in $\text{mIoU}_{\text{clean}}$/$\text{mIoU}_{\text{noise}}$ from 47.22/35.58 to 46.03/34.76. On Cityscapes, CTS achieves a 1.9× compute reduction (347.77 to 185.52 GFLOPs) with relatively similar performance (78.15/63.07 vs. 78.25/63.26). Overall, these results indicate that ViT-based segmenters contain substantial token redundancy that can be reduced with limited accuracy loss at mild compression when spatial structure is preserved.

Finding 2: At high compression, token compression degrades sharply. At high-compression settings (3.4×-3.8× FLOPs reduction), token compression drops substantially in both clean performance and corruption robustness. As shown by the steep drops in Fig. 2, this degradation becomes much sharper than in the mild-compression regime. According to Tab. 1, the degradation is particularly severe for general token merging. ToMe collapses on ADE20K from the baseline 47.22/35.58 to 23.94/16.86 at 35.08 GFLOPs (3.7× reduction), suggesting substantial loss of spatial information when an excessive number of tokens are merged across encoder layers. On Cityscapes, ToMe also shows a large perfor-

mance drop to 64.50/50.56. CTS and ALGM decrease more gracefully but still exhibit noticeable performance loss under extreme compression. For instance, CTS maintains comparatively stronger performance (73.81/60.99 on Cityscapes at 100.18 GFLOPs), yet still falls below the baseline by a clear gap.

Overall, these results suggest that aggressive token reduction can discard or distort critical spatial information required for dense prediction, leading to significant degradation in both accuracy and robustness. This behavior contrasts with structural pruning (Finding 3), which reduces model capacity more gradually and therefore exhibits a smoother degradation pattern. We provide further analysis in Sec. 4.3.

Finding 3: Model pruning is more stable at high compression. Fig. 2 shows that structural pruning exhibits a much smoother degradation pattern as compute decreases. Unlike token compression, which can drop sharply when many tokens are merged, NViT reduces model capacity while preserving the full spatial token grid. Empirically, performance decreases steadily rather than collapsing. On ADE20K, the baseline achieves 47.22/35.58 at 129.70 GFLOPs, NViT achieves 47.93/36.25, 45.40/34.20, and 45.30/32.86 at 96.40 (1.4×), 49.37 (2.6×), and 33.82 (3.8×) GFLOPs respectively. On Cityscapes, the aggressive NViT-S still achieves 76.76/58.75 at 95.05 GFLOPs (3.7×), compared with the baseline 78.25/63.26. Overall, these results suggest that structural pruning provides a more stable high-compression path than aggressive token reduction.

4.2 Prune-then-merge as a practical high-compression strategy

The observations from Findings 1-3 suggest that token compression and structural pruning degrade differently under high compression. Token compression reduces token redundancy and becomes fragile when compression is aggressive, whereas pruning reduces model capacity more gradually while preserving the spatial token grid. This motivates a strategy: apply moderate structural pruning to obtain a compact backbone, then apply moderate token merging to further reduce computation. We therefore evaluate a *prune-then-merge (PtM)* pipeline, where token merging (ToMe) is applied on top of a pruned backbone (NViT). As shown in Tab. 1, this combined configuration achieves a strong accuracy-robustness trade-off at similar FLOPs in the high-compression regime. At high compression, PtM achieves 45.71/34.33 mIoU at 34.84 GFLOPs on ADE20K and 77.52/60.71 at 95.87 GFLOPs on Cityscapes, corresponding to 3.7× and 3.6× FLOPs reduction, respectively. It outperforms both token compression and structural pruning on ADE20K, and gives the best clean mIoU with competitive robustness on Cityscapes. These results suggest that pruning and token compression are complementary under high compression. Pruning first reduces model capacity while retaining the spatial layout required for dense prediction, after which moderate token merging can further reduce computation without the sharp information loss observed under aggressive token compression alone. When applied sequentially, this pipeline provides a practical recipe for achieving strong accuracy-robustness trade-offs under high compression.

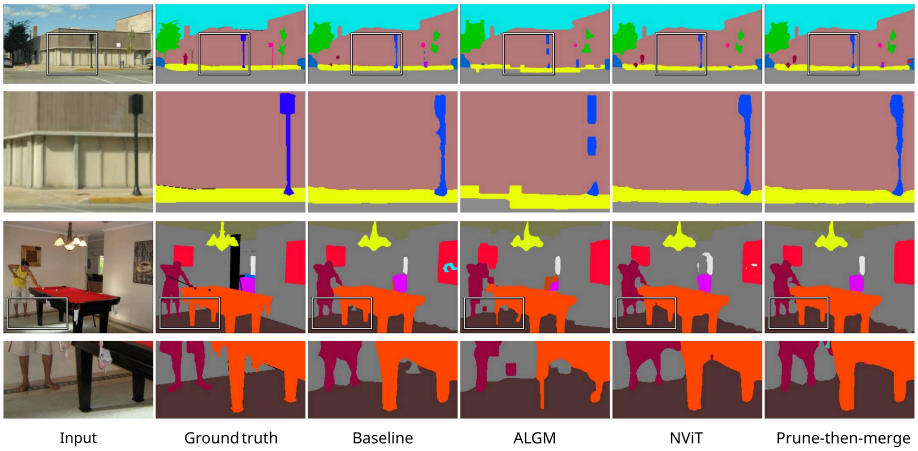

Fig. 3: Qualitative comparison under aggressive compression on ADE20K. We show two representative examples with zoomed regions highlighting fine structures and local boundaries. Compared with NViT and prune-then-merge, ALGM produces less spatially coherent predictions in these regions.

4.3 Why does aggressive token compression break?

Effective-rank analysis. We further analyze feature diversity using entropy-based effective rank [27] as a spectral diagnostic of representation dimensionality and collapse in dense prediction [3]. For each image, we reconstruct the encoder features on the original token grid and denote the resulting feature matrix as $X \in \mathbb{R}^{N \times C}$, where N is the number of tokens and C is the feature dimension. Following [27], we compute $\mathrm{eRank}(X)$ and report $\mathrm{eRank}(X)/\min(N, C)$, which normalizes by the maximum attainable rank, and measures the fraction of available rank used by the features. Lower values indicate that the features are concentrated along fewer dominant directions, suggesting stronger dimensional collapse [15, 41].

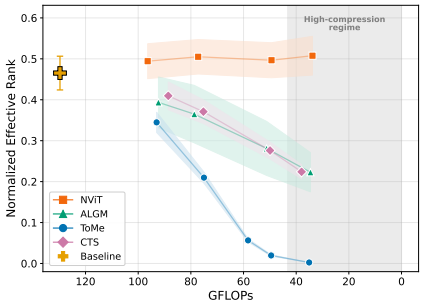

Fig. 4: Normalized effective rank on ADE20K-val (markers and bands show mean ± std over images).

Fig. 4 shows that token compression methods exhibit a clear decline in normalized effective rank as compression increases. The drop is most severe for ToMe, indicating that aggressive merging concentrates the reconstructed encoder representation into fewer dominant feature directions. Segmentation-aware methods such as ALGM and CTS show a more gradual decline, but still lose feature diversity at higher compression. In contrast, NViT maintains a stable normalized effective rank across compression levels, suggesting that structural

Method	r_{FLOPs} ↑	FPS↑	Speedup↑	$\text{mIoU}_{\text{clean}}$ ↑	$\text{mIoU}_{\text{noise}}$ ↑
Baseline	1.00×	23.69	1.00×	47.22	35.58
Mild speedup (∼1.4×)					
ALGM	1.41×	33.45	1.41×	46.88	34.64
CTS	1.46×	33.69	1.42×	46.03	34.76
ToMe	1.39×	32.87	1.39×	**46.94**	35.38
NViT	1.68×	32.22	1.36×	46.77	**35.51**
Moderate speedup (∼2.5×)					
ALGM	2.53×	60.01	2.53×	44.52	31.56
CTS	2.60×	58.44	2.47×	44.05	**33.66**
ToMe	2.62×	61.50	2.60×	41.01	30.24
NViT-S	3.84×	57.54	2.43×	**45.30**	32.86
Aggressive speedup (∼3.5×)					
ALGM	3.75×	78.90	3.33×	42.85	30.20
CTS	3.42×	72.21	3.05×	42.16	32.69
ToMe	3.70×	84.96	3.59×	23.94	16.86
NViT	7.61×	87.34	3.69×	41.01	26.85
NViT-B+ToMe *(PtM)*	4.75×	88.10	3.72×	**45.17**	**33.92**

Table 2: Wall-clock comparison on ADE20K grouped by measured FPS speedup among segmentation-aware token compression (ALGM, CTS), general token compression (ToMe), structural pruning (NViT), and prune-then-merge (PtM). The FLOPs reduction ratio r_{FLOPs} and FPS speedup are reported relative to the uncompressed baseline. **Bold** indicates the best clean and corruption mIoU among compression methods within each speed group.

pruning better preserves feature diversity in this setting by retaining the spatial token grid.

Qualitative results. Fig. 3 provides qualitative examples that illustrate the high-compression behavior observed in our quantitative results. Under aggressive token compression, ALGM merges local tokens more aggressively, which results in less coherent predictions around thin structures and local boundaries. In the first example, ALGM fragments the pole-like object and introduces class confusion across the sidewalk-road boundary, with sidewalk and road labels bleeding into each other. In the second example, similar artifacts appear around the table legs and the boundaries between the person, table, and floor. NViT and prune-then-merge maintain more coherent spatial predictions in these regions.

4.4 FPS-matched validation

We further evaluate inference speed (FPS). Tab. 2 compares representative ADE20K configurations at similar wall-clock speedups. The results show a similar regime-level trend. At mild-to-moderate speedups, ALGM, CTS, and NViT

remain effective, while general token merging (ToMe) already degrades more noticeably at moderate speedups. At the aggressive speedup level, this gap becomes larger: ToMe drops sharply to 23.94/16.86 mIoU at around 3.6× FPS speedup, whereas ALGM and CTS token compression methods remain stronger but still lose accuracy compared to their moderate-speedup configurations. In contrast, prune-then-merge achieves the strongest clean/corruption trade-off among the tested high-speed configurations. At around 3.7× FPS speedup, PtM (NViT-B+ToMe) maintains 45.17/33.92 mIoU, close to the uncompressed baseline of 47.22/35.58 and ahead of other single-axis compression methods.

We note that, under this segmentation pipeline, pruning-only models require much larger FLOPs reductions to achieve similar FPS speedups. NViT-S reduces FLOPs by 3.84× but achieves only a 2.43× FPS speedup. To compare pruning in the aggressive wall-clock regime, we therefore include a more aggressively pruned NViT variant, obtained with a smaller latency target. This variant reaches a 3.69× FPS speedup, but its clean/corrupted mIoU drops to 41.01/26.85. We hypothesize that this mismatch happens because token compression and structural pruning affect runtime computation differently. Following prior ViT-segmentation token-compression pipelines [24, 26], token compression is applied in the encoder, and the reduced token set is passed to the unchanged decoder before full spatial predictions are reconstructed. Pruning-only models, in contrast, reduce architectural capacity while preserving the full spatial token grid throughout. More generally, FLOPs reduction does not necessarily yield proportional latency speedup, as wall-clock performance also depends on memory access, tensor shapes, and kernel efficiency [25, 28].

5 Conclusion

Token compression has emerged as a popular approach for improving the efficiency of Vision Transformers (ViTs) while maintaining strong performance, including for semantic segmentation. However, existing evaluations often emphasize low-to-moderate compression, leaving behavior under high compression, where larger compute reduction may be required for deployment, less systematically analyzed. In this work, we presented a unified benchmark of representative token compression and structural pruning methods for ViT-based semantic segmentation across a wide range of FLOPs budgets on both clean inputs and common-corruption variants. Our benchmark reveals a clear regime-dependent trend: token compression can preserve performance under mild reduction, but degrades sharply under aggressive compression. In contrast, structural pruning exhibits a smoother degradation curve and is typically more stable at high compression. Motivated by these findings, we studied a practical *prune-then-merge* pipeline that applies moderate token merging on top of a moderately pruned backbone. Under comparable compute budgets at high compression, this strategy achieves a stronger accuracy-robustness trade-off than applying either token compression or structural pruning alone.

Future work includes extending the benchmark to additional recognition tasks, such as classification and detection, and to broader backbone families beyond flat-token ViTs, such as hierarchical transformers commonly used for high-resolution vision. In addition, we aim to study more principled ways to co-design pruning and token compression for improved robustness at aggressive compression levels.

Acknowledgements

This research is supported by Temasek Laboratories, Singapore University of Technology and Design; the National Research Foundation, Singapore under its AI Singapore Programmes (AISG Award No.: AISG2-TC-2022-007); the Agency for Science, Technology and Research (A*STAR) under its MTC Programmatic Funds (Grant No. M23L7b0021); the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

References

1. 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)
2. Cai, H., Li, J., Hu, M., Gan, C., Han, S.: Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17302–17313 (2023)
3. Chen, L., Gu, L., Fu, Y.: Frequency-dynamic attention modulation for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025)
4. 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. arXiv preprint arXiv:2305.17997 (2023)
5. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR (2022)
6. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: NeurIPS (2021)
7. 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 (2016)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale (2021)
9. Fang, G., Ma, X., Song, M., Mi, M.B., Wang, X.: Depgraph: Towards any structural pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16091–16101 (2023)

10. Fang, G., Ma, X.T., Mi, M.B., Wang, X.: Isomorphic pruning for vision models. arXiv preprint arXiv:2407.04616 (2024)
11. 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)
12. Haurum, J.B., Escalera, S., Taylor, G.W., Moeslund, T.B.: Which tokens to use? investigating token reduction in vision transformers. In: 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 773–783 (2023). https://doi.org/10.1109/ICCVW60793.2023.00085
13. Hooker, S., Courville, A., Clark, G., Dauphin, Y., Frome, A.: What do compressed deep neural networks forget? arXiv preprint arXiv:1911.05248 (2019)
14. Hou, Z., Kung, S.Y.: Multi-dimensional vision transformer compression via dependency guided gaussian process search. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 3668–3677 (2022). https://doi.org/10.1109/CVPRW56347.2022.00411
15. Huang, H., Campello, R.J.G.B., Erfani, S.M., Ma, X., Houle, M.E., Bailey, J.: Ldreg: Local dimensionality regularized self-supervised learning. In: ICLR (2024)
16. Jain, J., Li, J., Chiu, M., Hassani, A., Orlov, N., Shi, H.: OneFormer: One Transformer to Rule Universal Image Segmentation. In: CVPR (2023)
17. Kamann, C., Rother, C.: Benchmarking the robustness of semantic segmentation models. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). p. 8825–8835. IEEE (Jun 2020). https://doi.org/10.1109/cvpr42600.2020.00885
18. Kim, M., Gao, S., Hsu, Y.C., Shen, Y., Jin, H.: Token fusion: Bridging the gap between token pruning and token merging. In: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1372–1381 (2024). https://doi.org/10.1109/WACV57701.2024.00141
19. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything (2023)
20. 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. Springer (2022)
21. 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)
22. Liu, Y., Zhou, Q., Wang, J., Wang, Z., Wang, F., Wang, J., Zhang, W.: Dynamic token-pass transformers for semantic segmentation. In: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1816–1825 (2024). https://doi.org/10.1109/WACV57701.2024.00184
23. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
24. 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)
25. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). pp. 116–131 (2018)

26. Norouzi, N., Sorlova, S., de Geus, D., Dubbelman, G.: ALGM: Adaptive Local-then-Global Token Merging for Efficient Semantic Segmentation with Plain Vision Transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
27. Roy, O., Vetterli, M.: The effective rank: A measure of effective dimensionality. In: 2007 15th European signal processing conference. pp. 606–610. IEEE (2007)
28. Shen, M., Yin, H., Molchanov, P., Mao, L., Liu, J., Alvarez, J.: Structural pruning via latency-saliency knapsack. In: Advances in Neural Information Processing Systems (2022)
29. Shi, D., Tao, C., Jin, Y., Yang, Z., Yuan, C., Wang, J.: UPop: Unified and progressive pruning for compressing vision-language transformers. In: Proceedings of the 40th International Conference on Machine Learning. vol. 202, pp. 31292–31311. PMLR (2023)
30. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation (2021)
31. Tang, Q., Zhang, B., Liu, J., Liu, F., Liu, Y.: Dynamic token pruning in plain vision transformers for semantic segmentation. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 777–786 (2023)
32. Tang, Y., Wang, Y., Guo, J., Tu, Z., Han, K., Hu, H., Tao, D.: A survey on transformer compression (2024)
33. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H.: Training data-efficient image transformers & distillation through attention. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 10347–10357. PMLR (18–24 Jul 2021)
34. Wang, Z., Luo, H., WANG, P., Ding, F., Wang, F., Li, H.: VTC-LFC: Vision transformer compression with low-frequency components. In: Thirty-Sixth Conference on Neural Information Processing Systems (2022)
35. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: Neural Information Processing Systems (NeurIPS) (2021)
36. Yang, H., Yin, H., Shen, M., Molchanov, P., Li, H., Kautz, J.: Global vision transformer pruning with hessian-aware saliency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18547–18557 (June 2023)
37. Zhang, W., Huang, Z., Luo, G., Chen, T., Wang, X., Liu, W., Yu, G., Shen, C.: Topformer: Token pyramid transformer for mobile semantic segmentation. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (2022)
38. Zheng, C., Zhang, K., Yang, Z., Tan, W., Xiao, J., Ren, Y., Pu, S., et al.: Savit: Structure-aware vision transformer pruning via collaborative optimization. Advances in Neural Information Processing Systems **35**, 9010–9023 (2022)
39. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., Zhang, L.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: CVPR (2021)
40. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision **127**(3), 302–321 (2019)
41. Zhuo, Z., Wang, Y., Ma, J., Wang, Y.: Towards a unified theoretical understanding of non-contrastive learning via rank differential mechanism. In: International Conference on Learning Representations (2023)