Token Cropr: Faster ViTs for Quite a Few Tasks

Benjamin Bergner[1,3,*], Christoph Lippert[1,2], Aravindh Mahendran[4]
[1]Hasso Plattner Institute for Digital Engineering, University of Potsdam
[2]Hasso Plattner Institute for Digital Health at the Icahn School of Medicine at Mount Sinai
[3]Amazon [4]Google DeepMind

Abstract

The adoption of Vision Transformers (ViTs) in resource-constrained applications necessitates improvements in inference throughput. To this end several token pruning and merging approaches have been proposed that improve efficiency by successively reducing the number of tokens. However, it remains an open problem to design a token reduction method that is fast, maintains high performance, and is applicable to various vision tasks. In this work, we present a token pruner that uses auxiliary prediction heads that learn to select tokens end-to-end based on task relevance. These auxiliary heads can be removed after training, leading to throughput close to that of a random pruner. We evaluate our method on image classification, semantic segmentation, object detection, and instance segmentation, and show speedups of 1.5 – 4x with small drops in performance. As a best case, on the ADE20k semantic segmentation benchmark, we observe a 2x speedup relative to the no-pruning baseline, with a negligible performance penalty of 0.1 median mIoU across 5 seeds.

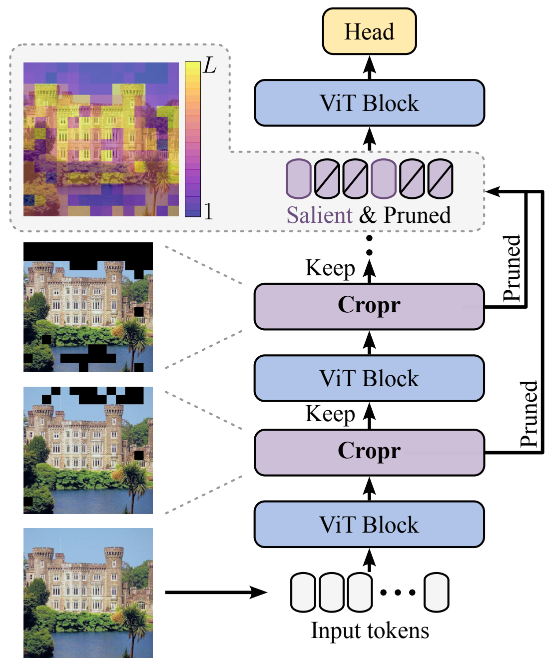

Figure 1. **Cro**ss-attention **pr**uning (**Cropr**) modules successively prune less relevant tokens, retaining only the most discriminative ones for deeper layers. Our method accelerates ViTs while maintaining high performance and is applicable to many vision tasks, from classification to segmentation and detection. The example castle images illustrate the pruning process. The heatmap visualizes which tokens were pruned at each block 1 to L in the network.

1. Introduction

The Vision Transformer [11] is a widely used architecture for computer vision tasks such as image classification, segmentation, and object detection [20]. ViTs represent images as a sequence of per-patch tokens, that they process using multi-head self-attention (MHSA) transformer blocks. The self-attention mechanism computes a pairwise dot product between all tokens, which results in a quadratic time and space complexity with respect to sequence length, $\mathcal{O}(n^2)$ [48]. For real-world applications that require low latency or a small compute budget, sequence length thus becomes a burden, especially in the light of increasing model sizes [8], image resolutions [2], and finer tokenization [3].

Images, such as the example used in Fig. 1, are spatially redundant containing non-salient background and repetitive patterns. This suggests that several patches could be processed using fewer transformer blocks, providing an opportunity to prune uninformative tokens, reduce sequence length in higher layers, and thus improve computational efficiency. However, this raises a central question: How can we accurately and efficiently assess the importance of individual tokens for a given task?

Recent token pruning methods rely on heuristics, such as self-attention scores, to identify informative tokens [14, 27]. Alternative approaches reduce token count by merging similar tokens [4, 12, 32]. However, these methods do not explicitly model the importance of a token for a given task, which can lead to a significant drop in task performance. In

*Work done prior to joining Amazon.

contrast, attribution methods such as Saliency [42], Occlusion [58] and Attention Rollout [1] estimate input contributions to a prediction, but require a full forward pass, which is not a viable option due to the associated overhead.

The question of estimating task relevance is further complicated by the diversity of task types in computer vision. Image classification, the simplest of vision tasks, has been the focus of many prior works, [14, 24, 27, 31, 34, 38, 55, 56] to name a few. Dense tasks such as semantic segmentation, however, present a new challenge for token pruning since they require predictions at the pixel level, which is inherently in conflict with the idea of pruning tokens.

We address these concerns with **Cro**ss-attention **pr**uning (**Cropr**), a simple token pruning method for ViTs that efficiently estimates per-token task relevance, while being applicable to various vision tasks. Cropr modules are applied at intermediate layers for token pruning, see Fig. 1. Each module consists of a cross-attention based aggregation mechanism coupled with an auxiliary prediction head. The latter learns to solve the task while the former ranks tokens by task relevance, forwarding only the most relevant tokens to deeper layers. The auxiliary heads can be discarded after training, which minimizes overhead and renders token pruning efficient. Lastly, pruned tokens are reintroduced later in the network, in a trick called Last Layer Fusion, to enable dense tasks. We detail our method in Sec. 3.

We evaluate our method on image classification, semantic segmentation, object detection, and instance segmentation in Sec. 4. We demonstrate strong performance even under aggressive pruning schedules. For example, when fine-tuning an EVA-02 [13] backbone, we are able to maintain 89.7% top-1-accuracy on ImageNet-1k, a drop of only 0.2 percentage points compared to the unpruned model, while achieving a $2.1\times$ speedup. We also evaluate the effect of token pruning on different encoder capacities and image resolutions, showing that our method performs particularly well at scale. An ablation study offers empirical support for our design choices and qualitative evaluations provide insights into the pruning process. We conclude in Sec. 5 with a summary of our findings and future work.

2. Related Work

Several methods have been proposed to reduce sequence length in vision transformers by pruning / merging tokens.

Token pruning for classification. A common strategy is to leverage attention scores from the class token (CLS) into image tokens as a bottom-up cue for pruning [14, 27, 31, 55]. Tokens with lower attention scores are regarded as less important and pruned out. Notably, Haurum et al. [16] show that a simple Top-K selector is a strong baseline. However, modern fused kernel implementations [7, 36] often restrict direct access to attention matrices, thus requiring alternative strategies. We instead take a top-down approach, leveraging signals from auxiliary heads to retain task-relevant tokens.

Other approaches use parametrized modules to predict which tokens to keep [24, 38], but introduce additional layers and losses that may interfere with the primary task. Cropr modules apply a stop-gradient to avoid gradient interference and limit additional parameters to a single query token at inference time. Another common design choice is to make token pruning adaptive, pruning more tokens for simpler inputs [10, 14, 23, 26, 34, 45, 56]. In contrast, we use a throughput-optimized static approach that prunes a constant number of tokens to enable batching across inputs.

Token pruning beyond classification. Very few works apply token pruning beyond classification: Liu et al. [30], Tang et al. [44] extend it to semantic segmentation by adding auxiliary heads to prune tokens based on confidence. We compare against Tang et al. [44] in our experiments. Liu et al. [29] use 2-layer MLPs for token pruning in object detection and instance segmentation, achieving moderate speedups of up to 34% in small networks. Instead, we omit extra layers at inference time and apply Cropr to a larger ViT, achieving a $1.9\times$ speedup and 63.0 AP^{box}. We believe that Cropr significantly advances the state of the art in token pruning by being fast, maintaining high performance, and being applicable to various vision tasks.

Token merging. The assumption behind token merging is that similar token representations contribute redundantly and can thus be combined. Hard merging methods combine similar tokens into non-overlapping groups, e.g. through clustering [12, 32, 59] or bipartite matching [4]. In contrast, soft merging methods create summary tokens by learning convex combinations of spatial tokens [15, 19, 39, 62]. For instance, Renggli et al. [39] and Jaegle et al. [19] employ cross-attention with learnable queries for this purpose. We also use cross-attention with learnable queries, but for token selection as opposed to merging, where attention scores reflect task relevance and aggregated tokens are used only in the auxiliary heads.

Pruning and merging. Recently, pruning and merging concepts have also been applied jointly [5, 21, 52]. Many pruning methods additionally aggregate pruned tokens into one or a few new tokens [24, 27, 31, 50, 55]. Cropr similarly reactivates pruned tokens but by simply concatenating them with retained tokens before the final transformer block, without resorting to any token summarization.

3. Methods

Given a sequence of per-patch tokens, our goal is to increase the inference efficiency of ViTs by successively reducing the number of tokens as they propagate through the network. To this end, we add Cropr modules on top of ViT

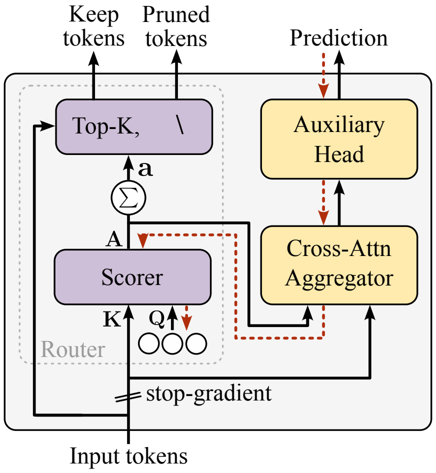

Figure 2. Cropr module during training. The router scores and separates salient keep tokens from uninformative tokens to be pruned. The scorer's attention matrix, $\mathbf{A}$, is reused in the aggregator whose output is used to make intermediate predictions. Gradient flow indicated as a dotted red line feeds back into the scorer and queries.

blocks, each of which selects the most discriminative tokens while pruning the least informative ones. In this way, computation in subsequent layers is reduced while relevant information is preserved, minimizing the impact of pruning on task performance. To select the most discriminative tokens, Cropr modules use a cross-attention based routing and aggregation mechanism that receives task-specific training signals from an auxiliary head (Sec. 3.1). By slightly customizing these components, our method can be applied to various vision tasks, such as image classification, segmentation, and object detection (Sec. 3.2). In particular, models for dense tasks such as semantic segmentation make pixel-wise predictions and thus require information from all tokens. We propose Last Layer Fusion (LLF) as a simple but effective approach to recover information from pruned tokens (Sec. 3.3). During inference, it is possible to introduce further optimizations to slim down our module and improve throughput (Sec. 3.4). We end this section with a realistic example that illustrates our pruning schedule (Sec. 3.5). An implementation of Cropr is provided at: https://github.com/benbergner/cropr.

3.1. Module description

The Cropr module is illustrated in Fig. 2. Each module takes tokens $\mathbf{X} \in \mathbb{R}^{M \times D}$ as input and outputs disjoint sets of "keep" and "pruned" tokens, $\mathbf{X}^k \in \mathbb{R}^{K \times D}$ and $\mathbf{X}^p \in \mathbb{R}^{R \times D}$ respectively, where $K = M - R$ and R is the pruning rate. Each module consists of four components: a scorer and a selector, which together form the router, as well as an aggregator and a task head.

The **scorer** assigns scores $\mathbf{a} \in \mathbb{R}^M$ to the set of tokens. These scores are then passed to the **selector**, which retains the K highest scoring tokens and prunes the remaining R tokens:

$$\mathbf{X}^k = \text{Top-K}\left(\mathbf{X} \mid \mathbf{a}\right), \tag{1}$$

$$\mathbf{X}^p = \mathbf{X} \setminus \mathbf{X}^k, \tag{2}$$

where $\setminus$ is set subtraction. The scorer itself is modeled after a cross-attention module, with learnable queries $\mathbf{Q} \in \mathbb{R}^{N \times D}$ and a key matrix $\mathbf{K} \in \mathbb{R}^{M \times D}$, which is conditioned on the input tokens $\mathbf{X}$:

$$\mathbf{A} = \mathbf{Q} \times \mathbf{K}(\mathbf{X})^\top . \tag{3}$$

Cross-attn modules typically use linear query, key and value projections, multiple attention heads and a LayerNorm (LN) [11, 48]. We found that neither of these components is necessary for achieving high task performance in our setting. This allows us to streamline our module while increasing throughput (Tab. 4a). We map $\mathbf{A}$ to $\mathbf{a}$ by summing the attention matrix over the query axis. For $N > 1$,

$$\mathbf{a} = \sum_{n=1}^{N} \mathbf{A}_n, \qquad \mathbf{A}_n \in \mathbb{R}^M. \tag{4}$$

This concludes the **router** design.

To learn scores that reflect a token's contribution to a prediction, the **aggregator** uses the attention matrix $\mathbf{A}$ to compute weighted averages of the input tokens, which are then passed to an auxiliary head. Thus, over the course of training, the scorer will assign more weight to tokens that are discriminative, and these tokens will then be retained for processing by the following transformer blocks. We found it beneficial to increase capacity in the aggregator by incorporating the transformer block's feed-forward module, adding a LN and an MLP with a residual connection (Tab. 4c). Thus, for $\mathbf{X}' = \text{softmax}\left(\frac{\mathbf{A}}{\sqrt{D}}\right)\mathbf{X}$,

$$\text{aggregator}\left(\mathbf{X}|\mathbf{A}\right) = \text{MLP}(\text{LN}(\mathbf{X}')) + \mathbf{X}'. \tag{5}$$

Aggregated outputs are processed by task-specific heads to make intermediate predictions, which in turn provide gradients for training the aggregator, scorer, and queries.

Finally, note that a stop-gradient is applied before the scoring and aggregation blocks. Conceptually, this has the advantage of isolating the auxiliary heads from the backbone. Thus, the encoder is not affected by conflicting gradients from auxiliary losses. This is also computationally efficient during training, since gradients from Cropr components do not backprop through the encoder.

3.2. Task-specific designs

Our scorer and aggregator employ a flexible query mechanism, similar to that of Perceiver IO [19], enabling arbitrary output shapes and easy adaptation to various tasks. In this section, by adjusting the number of learnable queries, designing auxiliary heads and loss functions, we instantiate Cropr for each vision task, as follows.

Image classification. The scorer uses a single learnable query, $N = 1$. The aggregator then outputs a single token, which is processed using a LN and linear projection exactly as in the final classification head. The latter outputs logits for all classes. A softmax cross-entropy loss is used.

Semantic segmentation. Both main and auxiliary heads adopt the linear head of Segmenter [43]. The scorer uses one learnable query per patch token, $N = h \times w$, to obtain grid-structured representations from $\mathbf{X}$. The aggregator output is processed using a LN and linear projection, like in image classification, but independently per-patch location, followed by a per-patch softmax cross-entropy loss. To reduce computational complexity in the auxiliary heads, instead of upsampling the logits to the input resolution as in Segmenter, the labels are downsampled to the feature map resolution. The downsampled labels can then be reused across Cropr modules.

Joint detection and instance segmentation. We apply Cropr to Cascade Mask R-CNN [6, 17] for this task. But because this multi-stage detector is computationally expensive, it is less practical for use in auxiliary heads. We propose a proxy auxiliary head and loss that provides a strong signal for both tasks: multi-label classification. The intuition here is that object detection and instance segmentation both require identifying all object categories present in an image. In more detail, ground-truth labels are encoded as binary vectors, where each dimension corresponds to a class's presence. The scorer then uses a single learnable query like in image classification, $N = 1$. A LN and linear projection then map the aggregated token into as many logits as there are classes in the dataset. A sigmoid activation function and a binary cross-entropy loss are used in this multi-label setting.

3.3. Last Layer Fusion (LLF)

In dense tasks, such as semantic segmentation, predictions are made at the pixel level. However, this is hard when a significant portion of the input is dropped. In addition, many task heads require a spatial feature map for upsampling, which is not maintained during pruning.

We address these challenges with LLF, an efficient and effective approach that reactivates pruned tokens and preserves information from all image patches. Specifically, pruned tokens from all Cropr modules are inserted alongside retained tokens output by the penultimate ViT block (Fig. 1) at their respective spatial locations. In other words the pruned tokens are skipped to the final ViT block and not entirely discarded. The final ViT block processes this combined sequence, allowing previously pruned tokens to attend to deep features of retained tokens. We present t-SNE [47] plots to visualize its effect in App. F. We disable DropPath [25] in the final ViT block to ensure token fusion.

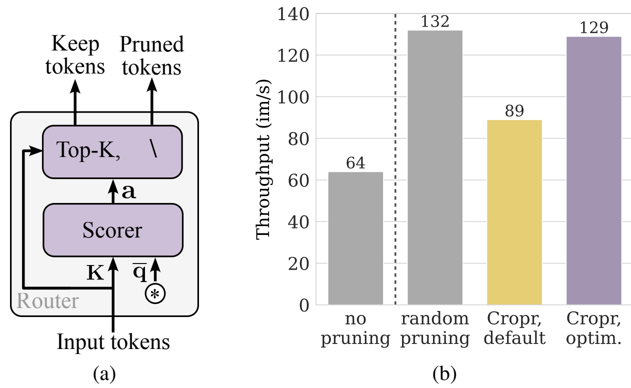

Figure 3. Cropr module during inference. (a) The aggregation function and the auxiliary head are removed. All queries are aggregated into a single query. (b) These optimizations speed up Cropr, with throughput comparable to that of a random selector. Results are shown for semantic segmentation.

LLF introduces no additional parameters while outperforming other fusion methods (Tab. 5). Note that LLF is not specific to Cropr; in fact, we equip several baselines with it in our experiments.

3.4. Efficient inference

The Cropr cross-attention transformer blocks and auxiliary heads constitute a significant computational overhead. Note, however, that these components shown in yellow in Fig. 2 are only required to train the scorer. At inference time, they can be safely discarded leaving just the router for token selection as illustrated in Fig. 3a.

The scorer still scales as $\mathcal{O}(N \times M)$, which is costly when the number of queries N is large, as is the case in semantic segmentation where N scales with image resolution. But since the aggregator has now been discarded, the cross-attention matrix need not be materialized. Only the summed up scores, $\mathbf{a}$, are needed. Applying the distributive property of matrix multiplication, it is then easy to show that Eq. (4) can be reduced to a vector-matrix multiplication, $\mathcal{O}(M)$:

$$\mathbf{a} = \sum_{n=1}^{N} \left(\mathbf{Q}\mathbf{K}^\top\right)_n = \sum_{n=1}^{N} \mathbf{Q}_n \mathbf{K}^\top \quad (6)$$

$$= \left(\sum_{n=1}^{N} \mathbf{Q}_n\right) \mathbf{K}^\top = \overline{\mathbf{q}}\mathbf{K}^\top, \quad (7)$$

where $\overline{\mathbf{q}} \in \mathbb{R}^D$ is an aggregated query that can be precomputed. Each Cropr module can thus be simplified to a router consisting of an efficient scoring function and a Top-K selector. With these improvements, the throughput of Cropr is comparable to that of a random selector (Fig. 3b).

3.5. Pruning schedule

We explain the pruning schedule using a working example. Consider a ViT-L with 24 blocks, a 224×224 input image, and a patch size of 16, resulting in 196 patch tokens. Unless stated otherwise, we insert Cropr modules after every block, a per-block schedule that prunes R tokens at a time. We aim to have most tokens removed by the end of the network.

Without LLF, pruning is applied after every block except the last. In our example, setting $R = 8$, we prune 23×8 tokens, leaving 12 tokens, for a total pruning ratio (TPR) of 94%. With LLF, pruning is performed after every block except the last two, resulting in 20 output tokens and a TPR of 90%. In this case, pruning is not performed after the penultimate block because the pruned tokens would be immediately reinserted.

We observed that for high-resolution images, maintaining the number of keep tokens as a multiple of 8 improves throughput (App. E). Since patch sequence lengths are commonly divisible by 8, we set R as a multiple of 8 whenever possible. Additionally, because ViTs typically employ a classification (CLS) token, we increase R by 1 in the first module. Following common practice [4, 14, 27], the CLS token is never pruned.

4. Experiments

We evaluate Cropr on four vision tasks across different ViT architectures, network capacities, and image resolutions. To show that Cropr selects task-relevant tokens, we compare it to challenging baselines: (1) no pruning (upper bound baseline), (2) random pruning, (3) variance pruning [33], ranking tokens based on per-patch pixel variance averaged over RGB channels, and (4) Attn Top-K, which selects tokens based on self-attention scores and has been shown to be among the best performing methods [16]. For a fair comparison, we use LLF with (2), (3), and (4). In addition to task-specific metrics, we report FLOPs / throughput (optimal across batch sizes) for a single forward pass at inference time, using automatic mixed precision (AMP) and an NVIDIA A100 GPU. Hyperparameters are listed in App. C.

4.1. Image classification

Comparison to baselines & prior art. We fine-tune ViT-L on ImageNet-1k [40] using a pretrained masked autoencoder (MAE) following the setup of He et al. [18], and apply the pruning schedule from our working example (Sec. 3.5). Table 1 shows a comprehensive evaluation of our method in three different scenarios. First, we compare our method against the baselines (2) - (4). Cropr outperforms all pruning baselines with comparable throughput. We also include results for a non-salient selector, which inverts Cropr by pruning the most relevant tokens. As expected, this approach performs worse than random pruning.

Method	Sch.	LLF	Pool	Acc.	1000 im/s
No pruning	—	—	avg	85.8	0.86 1.0×
Non-salient	$\searrow$	✓	avg	76.4	1.48 1.7×
Random	$\searrow$	✓	avg	83.8	1.50 1.7×
Variance [33]	$\searrow$	✓	avg	84.3	1.50 1.7×
Attn Top-K	$\searrow$	✓	cls	84.7	1.45 1.7×
Cropr	$\searrow$	✓	avg	**85.3**	1.48 1.7×
K-Medoids [32]	$\searrow$		avg	84.5	0.31 0.4×
ATS [14]	$\searrow$		cls	83.9	0.49 0.6×
DPC-KNN [12]	$\searrow$		avg	79.2	1.00 1.2×
EViT [27]	$\searrow$		cls	84.5	1.57 1.8×
ToMe, from [4]	$\searrow$		cls	**85.1**	1.55 1.8×
ToMe [4]	$\searrow$		avg	85.0	1.55 1.8×
Cropr	$\searrow$		avg	**85.1**	1.61 1.9×
DynamicViT [38]	↴		avg	64.4	1.32 1.5×
SiT [62]	↴		avg	83.0	1.41 1.6×
Sinkhorn [15]	↴		avg	56.5	1.40 1.6×
PatchMerger [39]	↴		avg	82.4	1.40 1.6×
Cropr	↴		avg	85.4	1.43 1.7×
Cropr	↴	✓	avg	**85.5**	1.35 1.6×

Table 1. ImageNet-1k results. Following He et al. [18], we use average pooling, only reverting to CLS pooling if a method requires the CLS token. Cropr is competitive or outperforms other pruning and merging methods while being runtime-efficient. $\searrow$: $R = 8$. ↴ : $R = 50$, prune after $\{6, 12, 18\}$-th block.

Next, in the middle of Tab. 1, Cropr w/o LLF is compared to prior works. Our method is competitive in performance and throughput. The latter especially varies significantly across methods, with K-Medoids [32] and ATS [14] being slower than the unpruned baseline.

Lastly, we observe that some methods do not converge with our block-wise pruning schedule. Hence, at the bottom of Tab. 1, we present results for a lighter 3-stage schedule, where $R = 50$ tokens are pruned after blocks 6, 12, and 18, resulting in 46 final tokens (TPR of 77%). In this scenario, Cropr performs best and shows competitive throughput.

Compared to the unpruned baseline, Cropr exhibits a minor performance drop of 0.3–0.7 accuracy points, while achieving a 1.6–1.9$\times$ speedup. In App. E, we show that using a lighter schedule does not affect performance at all.

Cropr at scale. What effect does network capacity have on performance and throughput in Cropr models compared to the unpruned baseline? This question is especially relevant given the trend toward larger models [8]. To study this we apply Cropr with LLF to ViT-B/16, L/16, and H/14, consisting of 12, 24, and 32 blocks, and set R for these model sizes to 16, 8, and 8 tokens per block, resulting in TPRs of 82, 90, and 94%, respectively. Figure 4 shows that the relative performance penalty of Cropr decreases as the model size increases, going from -0.9 in ViT-B to -0.4 in

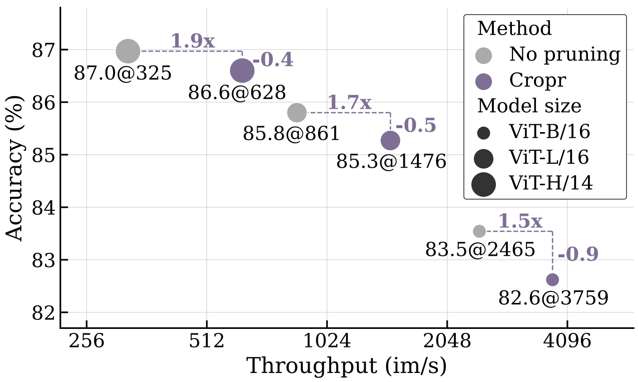

Figure 4. Performance-throughput tradeoff plot for different model sizes on ImageNet-1k. Token pruning in larger models provides more speedup and less performance drop.

ViT-H, despite higher TPRs. This observation is likely due to the fact that in deeper models, pruning is distributed over more layers resulting in fewer tokens being dropped early on. Furthermore, Cropr's speedup improves at scale since more layers benefit from reduced token counts, going from $1.5\times$ in ViT-B to $1.9\times$ in ViT-H. We observe similar effects when scaling image resolution (App. D).

Application to a SoTA model. We experiment with the EVA-02-L, a state-of-the-art open-source ViT [13]. We start training from an IN-21K fine-tuned checkpoint, resizing images to 448×448 and setting the patch size to 14. Table 2 presents the results. We first train EVA-02 without pruning, resulting in an accuracy of 89.9%, which is comparable to the 90.0% reported in Fang et al. [13]. We then train a Cropr pruned version where we set $R = 40$ and enable LLF, resulting in a TPR of 86%. We observe an accuracy of 89.7%, a drop of only 0.2 percentage points, while being $2.1\times$ faster with $\sim 41\%$ fewer FLOPs.

In addition, we report results for a more aggressive pruning schedule without LLF (marked $\downarrow$), where a single Cropr module, applied after the 3rd block, prunes 825 tokens (80% of the total). Compared to the unpruned baseline, this results in a moderate drop of 1.1 percentage points, but provides a FLOP reduction of $\sim 76\%$ and a speedup of $4.1\times$.

The rest of Tab. 2 lists results for a selection of other state-of-the-art models. After pruning, EVA-02 remains in 3rd place for accuracy, while being twice as fast. With the more aggressive schedule, our model is the fastest by a large margin, while still outperforming some of the other models.

4.2. Semantic segmentation

We experiment on the ADE20k dataset [61] and fine-tune Segmenter [43] with a linear decoding head (see Sec. 3.2). The encoder is replaced with EVA-02-L, following the settings of Fang et al. [13]. Images are processed at a resolution of 512×512 with a patch size of 16, resulting in 1024 patches. Models are trained for 64 epochs. For evaluation, we resize the max edge to 512 px and pad the smaller edge while maintaining the aspect ratio. This 1-shot evaluation approach is optimized for throughout and is more challenging than the common single-scale evaluation setting, which averages predictions from a sliding window.

Method	Res	#Par	FLOPs	Acc.	im/s
EVA-02	448	0.3B	0.31B	89.9	64
EVA-02 + Cropr	448	0.3B	0.18B	89.7	132
EVA-02 + Cropr $\downarrow$	448	0.3B	0.07B	88.8	259
CAFormer-B36 [57]	384	0.1B	0.07B	88.1	187
RegNetY 128GF [37]	384	0.6B	0.38B	88.2	148
EfficientNet-L2 [53]	800	0.5B	0.48B	88.4	33
ConvNeXt V2-H [51]	512	0.7B	0.60B	88.9	60
BEiT v2 ViT-L [35]	384	0.3B	0.18B	89.0	193
MaxViT-XL [46]	512	0.5B	0.54B	89.5	44
ViT-L, distilled [8]	384	0.3B	0.18B	89.6	193
BEiT-3 ViT-g/14 [49]	336	1.0B	0.58B	89.6	85
DaViT-Giant [9]	512	1.4B	1.04B	90.4	52
ViT-G/14 [60]	518	1.8B	2.52B	90.5	20

Table 2. Comparison of ImageNet-1k classification models. Our EVA-02 + Cropr variants remain competitive with SoTA models and achieve speedups of $2 - 4\times$ with small performance drops compared to the upper-bound baseline, EVA-02. $\downarrow$: prune 80% of all tokens after the 3rd block, w/o LLF.

In this setting the unpruned model achieves 56.7% median mIoU across 5 seeds, outperforming Seg-L-Mask/16 (51.8% mIoU [43]), despite Seg-L-Mask/16 operating at a higher resolution of 640×640, using a more complex mask transformer decoder, and employing the simpler single-scale evaluation setting. We attribute this to our use of the EVA-02 pretrained backbone.

When applying Cropr, we activate LLF and prune $R = 40$ tokens after each of the first 22 blocks, resulting in a TPR of 86%. To facilitate learning, a curriculum over R is used for the first 32 epochs, increasing R linearly from 1 to 40.

Comparison to baselines. We compare Cropr to baselines, (1) – (4). The pruning baselines use LLF to be applicable in a segmentation setting. Further, Attn Top-K now uses the averaged self-attention matrix to score patches, as the CLS token is not used in the head. Each model is run five times with different random seeds, and the results are summarized in Fig. 5. Cropr scores a median mIoU of 56.6%, which is only 0.1% points worse than the no-pruning baseline, while being $2.0\times$ faster. Furthermore, our model exhibits a higher median performance compared to all pruning baselines at a similar throughput. Interestingly, we found that all baselines, even a random pruner, achieve decent performance by leveraging LLF.

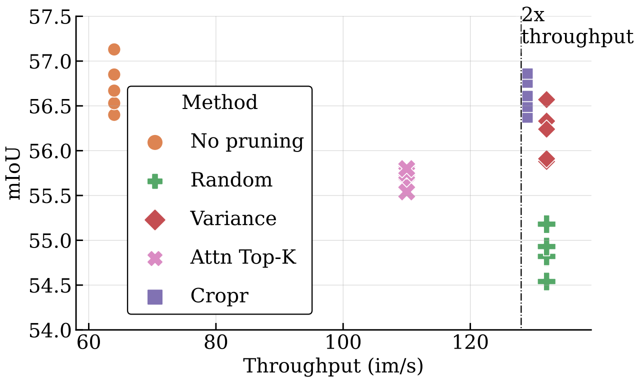

Figure 5. Semantic segmentation results on ADE20k. Cropr performs comparable to a no-pruning baseline, while achieving a 2× speedup, marked using the dashed vertical line. 5 seeds / method.

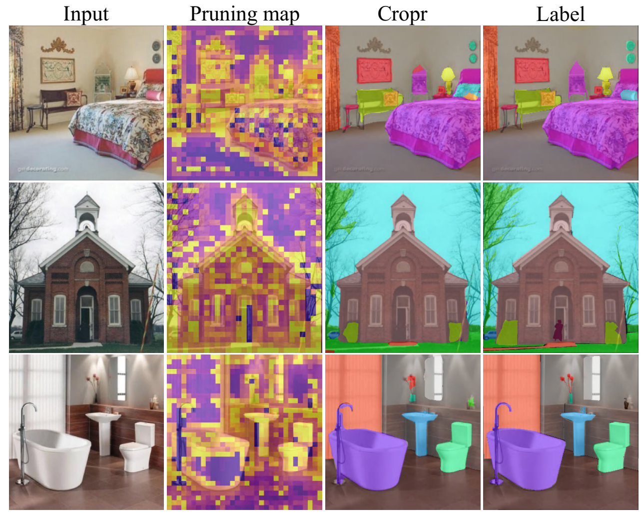

Figure 6. Visualizations for semantic segmentation. Cropr prunes tokens from stuff classes (e.g., sky, floor, wall) earlier, but keeps a few tokens from each class in later layers. Despite pruning, adjacent outputs of the same class appear consistent.

Comparison to prior works. We reimplement DToP's logit fusion approach [44], but using the same settings as our method for a fair comparison. DToP uses auxiliary heads to select tokens based on prediction confidence, and then concatenates the logits of both pruned and retained tokens to obtain the final prediction. As in our method, we use a LN and a linear output projection as auxiliary heads. Unlike DToP, LLF fuses features rather than logits, and the gradients from Cropr components do not backprop through the encoder. Table 5 shows that Cropr with LLF clearly outperforms DToP's logit fusion approach.

Liu et al. [30] also use auxiliary heads, but concatenate pruned and retained token features prior to the task head. We compare to this fusion approach in Tab. 5, showing that LLF outperforms 'Token Concat'. Moreover, they apply token pruning to a ViT-S-based Segmenter on ADE20k and report a 0.35 drop in mIoU with a 18% FLOP reduction relative to the no-pruning baseline. We achieve a median drop of only 0.1 mIoU while reducing FLOPs by 41%.

Qualitative evaluation. Figure 6 shows pruning heatmaps for indoor and outdoor scenes. In this task outputs for each pixel contribute to the evaluation metric, making it challenging to determine which information to prune. We found that attention is primarily directed to salient objects; however, a few background patches are also retained in later layers, likely due to their overall relevance to the task. Furthermore, despite pruning, we observe consistent predictions even for smaller, difficult to segment objects. This is likely facilitated by LLF, which enables early-pruned tokens to attend to deeper representations of neighboring tokens.

4.3. Object detection and instance segmentation

We benchmark Cropr on COCO [28] using the EVA-02-L backbone, initialized from an Objects365 [41] fine-tuned checkpoint. Following Fang et al. [13], we use Cascade Mask R-CNN as the task head to support both detection and segmentation. Images are resized to 1536×1536 with patch size 16, yielding $96^2 = 9216$ patches. As in Fang et al. [13], global attention is intermixed with window attention. With a window size of 16, this yields an initial grid of 6×6 windows. To support pruning while maintaining the window size, a 5-stage pruning schedule is applied. At each stage i, the number of tokens is reduced to $(96 - i * 16)^2$, resulting in a TPR of 97%. Pruning occurs after blocks 5, 8, 11, 14, and 20, just before the global attention layers. LLF is applied, allowing the task head to be used without modifications.

Table 3 shows that Cropr outperforms baselines (2) – (4) in both detection and segmentation. The performance gap between Cropr and the unpruned model is moderate, which is expected given the high TPR. Despite the optimized window-attention-based architecture, Cropr achieves a 54% reduction in FLOPs (Unpruned baseline: 2790 GFlops vs. Cropr: 1273 GFlops), along with a $2.4\times$ speedup in the encoder and a $1.9\times$ speedup in the overall model.

Figure 7 demonstrates that Cropr modules focus on task-relevant image regions corresponding to target objects. Interestingly, even the random pruner can serve as an effective detector with LLF, albeit with more errors.

4.4. Ablation study

Cropr module design. In Tab. 4a, we compare our simplified cross-attention design (single head, w/o QKV and head projections, w/o LN) to a more complex MHA design (16 heads, w/ QKV and head projections, w/ LN). The simpler approach outperforms MHA in both efficiency and performance metrics. In Tab. 4b, we evaluate an alternative

Method	AP^{box}	AP^{mask}	im/s (enc.)	im/s
No pruning	64.2	55.4	5.8 1.0×	4.5 1.0×
Random	60.6	51.9	14.0 2.4×	8.5 1.9×
Variance	62.0	53.0	13.9 2.4×	8.5 1.9×
Attn Top-K	62.6	53.6	10.8 1.9×	7.3 1.6×
Cropr	**63.0**	**54.0**	13.9 2.4×	8.5 1.9×

Table 3. Object detection and instance segmentation results on COCO val, showing throughput of the encoder and overall model.

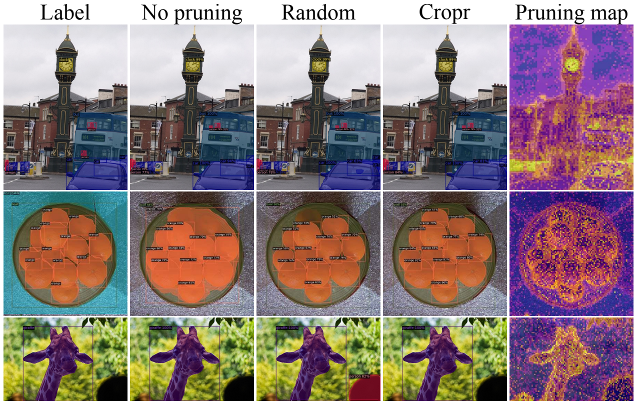

Figure 7. Bounding box and instance segmentation predictions for Cropr, as well as the unpruned and random baselines. Cropr pruning maps highlight relevant objects. First row: All methods accurately detect most objects. Second row: Only Cropr detects all oranges. Third row: Random pruner incorrectly detects a person.

Method	Acc.	GFlops	im/s
MHA	85.2	36.8	1352
Simple	85.3	34.2	1476

(a) **Cross-attn.** A simple 1-head cross-attention design w/o projection layers performs slightly better and is more efficient.

Method	Acc.
Sampling	85.1
Top-K	85.3

(b) **Selection methods.** Top-K vs. sampling from the attention distribution.

MLP	Acc.	GFlops	im/s
✗	85.0	34.2	1476
✓	85.3	34.2	1476

(c) **MLP.** Adding MLPs to the aggregator improves performance w/o overhead at inference time.

Stop grad.	Acc.
✗	85.0
✓	85.3

(d) **Gradient mode.** Stopping gradient flow works best.

Table 4. Cropr ablations on ImageNet-1k, with LLF enabled.

Method	#Params	GFlops	mIoU
No pruning	304M	311	56.7*
Cross-Attn	319M	184	47.7
Token Concat	304M	172	51.8
Cross-Attn + Concat	319M	186	51.5
MHSA + Concat	318M	186	<u>55.2</u>
DToP	308M	174	50.1*
LLF	304M	183	**56.6***

Table 5. Token fusion ablation on ADE20k. LLF performs best, without additional parameters. *: Median mIoU across 5 seeds.

selection method, which samples without replacement from the cross-attention distribution. Sampling is less effective than Top-K. Table 4c indicates that incorporating an MLP (w/ LN and residual connection) into the aggregation module improves token selection. Crucially, this modification does not impact efficiency metrics at inference time, as the aggregator is removed after training. Finally, Tab. 4d shows that stopping the gradient flow in Cropr leads to improved results, likely because gradient interference is prevented.

Token fusion. We compare LLF to several alternatives in Tab. 5: 'Cross-Attn' applies a cross-attention block with grid-shaped learned queries cross-attending into the keep tokens output by the last layer, that is, pruned tokens are not reactivated. Further note that this cross-attention block is trained from scratch. 'Token Concat' reactivates pruned tokens by concatenating them after the last layer. 'Cross-Attn + Concat' combines the two, cross-attending into concatenated tokens after the last layer. 'MHSA + Concat' is similar but uses a full self-attention transformer block trained from scratch instead. Lastly, 'DToP' is the logit fusion approach discussed in Sec. 4.2. All methods are evaluated for semantic segmentation on the ADE20k dataset.

Not reactivating pruned tokens, as in 'Cross-Attn' performs very poorly. 'Token Concat', 'Cross-Attn + Concat' and 'DToP' reactivate pruned tokens but do not support self-attention between pruned and retained tokens and thus underperform. In contrast 'MHSA + Concat' and LLF allow attention between tokens, resulting in higher mIoU. Notably, LLF outperforms MHSA without introducing any additional parameters compared to the unpruned baseline.

5. Conclusion

The experiments show that ViTs can be accelerated with small performance penalties by pruning the least informative tokens for a given task. We showcase the versatility of our approach by applying it beyond classification to semantic and instance segmentation, as well as object detection. That said, it is not without limitations. We discuss these in App. B within the supplementary material.

Future work could extend Cropr to additional vision tasks by adapting the auxiliary heads. Furthermore, the token-based nature of our method suggests broader applicability to other modalities, such as language and audio.

Overall, this work makes token pruning practical through a simple yet flexible method design. Beyond pruning, we hope to inspire further exploration of efficient attention mechanisms that target task-relevant information.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online, 2020. Association for Computational Linguistics. 2

[2] Benjamin Bergner, Christoph Lippert, and Aravindh Mahendran. Iterative patch selection for high-resolution image recognition. In *ICLR*, 2022. 1

[3] Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes. In *CVPR*, pages 14496–14506, 2023. 1

[4] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *ICLR*, 2023. 1, 2, 5

[5] Maxim Bonnaerens and Joni Dambre. Learned thresholds token merging and pruning for vision transformers. In *Workshop on Efficient Systems for Foundation Models @ ICML*, 2023. 2

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE TPAMI*, 43(5):1483–1498, 2019. 4

[7] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, 35:16344–16359, 2022. 2, 3

[8] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *ICML*, pages 7480–7512. PMLR, 2023. 1, 5, 6

[9] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *ECCV*, pages 74–92. Springer, 2022. 6

[10] Peiyan Dong, Mengshu Sun, Alec Lu, Yanyue Xie, Kenneth Liu, Zhenglun Kong, Xin Meng, Zhengang Li, Xue Lin, Zhenman Fang, et al. Heatvit: Hardware-efficient adaptive token pruning for vision transformers. In *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 442–455. IEEE, 2023. 2

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 3

[12] Mingjing Du, Shifei Ding, and Hongjie Jia. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems*, 99: 135–145, 2016. 1, 2, 5

[13] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024. 2, 6, 7, 1

[14] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *ECCV*, pages 396–414. Springer, 2022. 1, 2, 5

[15] Joakim Bruslund Haurum, Meysam Madadi, Sergio Escalera, and Thomas B Moeslund. Multi-scale hybrid vision transformer and sinkhorn tokenizer for sewer defect classification. *Automation in Construction*, 144:104614, 2022. 2, 5

[16] Joakim Bruslund Haurum, Sergio Escalera, Graham W. Taylor, and Thomas B. Moeslund. Which tokens to use? investigating token reduction in vision transformers. In *ICCV*, pages 773–783, 2023. 2, 5

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 4

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 5, 1, 2

[19] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *ICLR*, 2022. 2, 3

[20] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Comput. Surv.*, 54(10s), 2022. 1

[21] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *WACV*, pages 1383–1392, 2024. 2

[22] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token Fusion: Bridging the Gap between Token Pruning and Token Merging . In *WACV*, 2024. 1

[23] Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 784–794, 2022. 2

[24] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *ECCV*, pages 620–640. Springer, 2022. 2

[25] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *ICLR*, 2017. 4

[26] Ling Li, David Thorsley, and Joseph Hassoun. Sait: Sparse vision transformers through adaptive token pruning. *arXiv preprint arXiv:2210.05832*, 2022. 2

[27] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *ICLR*, 2022. 1, 2, 5

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 7

[29] Yifei Liu, Mathias Gehrig, Nico Messikommer, Marco Cannici, and Davide Scaramuzza. Revisiting token pruning for object detection and instance segmentation. In *WACV*, pages 2658–2668, 2024. 2

[30] Yuang Liu, Qiang Zhou, Jing Wang, Zhibin Wang, Fan Wang, Jun Wang, and Wei Zhang. Dynamic token-pass transformers for semantic segmentation. In *WACV*, pages 1827–1836, 2024. 2, 7

[31] Sifan Long, Zhen Zhao, Jimin Pi, Shengsheng Wang, and Jingdong Wang. Beyond attentive tokens: Incorporating token importance and diversity for efficient vision transformers. In *CVPR*, pages 10334–10343, 2023. 2

[32] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *WACV*, pages 12–21, 2023. 1, 2, 5

[33] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *NeurIPS*, 36, 2024. 5

[34] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. Ia-redˆ2: Interpretability-aware redundancy reduction for vision transformers. In *NeurIPS*, pages 24898–24911. Curran Associates, Inc., 2021. 2

[35] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 6

[36] Markus N. Rabe and Charles Staats. Self-attention does not need $o(n^2)$ memory. *arXiv preprint arXiv:2112.05682*, 2021. 2

[37] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollar. Designing network design spaces. In *CVPR*, 2020. 6

[38] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *NeurIPS*, 34:13937–13949, 2021. 2, 5

[39] Cedric Renggli, André Susano Pinto, Neil Houlsby, Basil Mustafa, Joan Puigcerver, and Carlos Riquelme. Learning to merge tokens in vision transformers. *arXiv preprint arXiv:2202.12015*, 2022. 2, 5

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 5

[41] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, pages 8430–8439, 2019. 7

[42] Karen Simonyan. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[43] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7262–7272, 2021. 4, 6, 1

[44] Quan Tang, Bowen Zhang, Jiajun Liu, Fagui Liu, and Yifan Liu. Dynamic token pruning in plain vision transformers for semantic segmentation. In *ICCV*, pages 777–786, 2023. 2, 7

[45] Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. In *CVPR*, pages 12165–12174, 2022. 2

[46] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, pages 459–479. Springer, 2022. 6

[47] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 (86):2579–2605, 2008. 4

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, et al. Attention is all you need. *NeurIPS*, 30(1):261–272, 2017. 1, 3

[49] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *CVPR*, pages 19175–19186, 2023. 6

[50] Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *CVPR*, pages 2092–2101, 2023. 2

[51] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *CVPR*, pages 16133–16142, 2023. 6

[52] Xinjian Wu, Fanhu Zeng, Xiudong Wang, and Xinghao Chen. Ppt: Token pruning and pooling for efficient vision transformers. *arXiv preprint arXiv:2310.01812*, 2023. 2

[53] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 6

[54] Xuwei Xu, Sen Wang, Yudong Chen, Yanping Zheng, Zhewei Wei, and Jiajun Liu. GTP-ViT: Efficient Vision Transformers via Graph-based Token Propagation . In *WACV*, 2024. 1

[55] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *AAAI*, pages 2964–2972, 2022. 2

[56] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *CVPR*, pages 10809–10818, 2022. 2

[57] Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. Metaformer baselines for vision. *IEEE TPAMI*, 2023. 6

[58] MD Zeiler. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2

[59] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *CVPR*, pages 11101–11111, 2022. 2

[60] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, pages 12104–12113, 2022. 6

[61] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 6

[62] Zhuofan Zong, Kunchang Li, Guanglu Song, Yali Wang, Yu Qiao, Biao Leng, and Yu Liu. Self-slimmed vision transformer. In *ECCV*, pages 432–448. Springer, 2022. 2, 5