Token Cropr: Faster ViTs for Quite a Few Tasks

Supplementary Material

A. Broader Impact

Our method significantly increases the throughput of ViTs, making it well suited for applications that require real-time inference, such as autonomous driving, robotics, and computer-assisted medical interventions. Our approach could also be used to accelerate high-capacity models, potentially enabling new applications that require both high performance and low latency. Edge devices such as smartphones could benefit from decreased computation to improve battery life. Since inference is performed repeatedly and often represents a greater cumulative cost than training, our method offers a broader potential contribution to sustainability by reducing carbon emissions.

That said, it is important to acknowledge that our method could also be misused to accelerate models for harmful applications, particularly due to the versatility of Cropr across various vision tasks. We neither explore such applications in this paper nor intend to pursue them in future work.

Moreover, we have not evaluated our method for equitable performance across demographic groups. Just as models can have biases against certain groups, these biases can propagate to token scoring and selection. Addressing these fairness and inclusivity concerns is critical before using token pruning methods in real-world applications. In addition, a thorough error analysis should be conducted to identify discrepancies between the pruned and unpruned models, ensuring robust and reliable performance.

B. Limitations

Limited hardware: Across experiments, we report $1.5 - 4\times$ speedups of our method over unpruned baselines, as measured on A100 NVIDIA GPUs. However, runtime gains may vary on other hardware accelerators. We use `gather` operations for token selection and concatenation, whose performance is hardware dependent.

Gap to the no-pruning baseline: While Cropr significantly reduces computation, it does not fully close the performance gap with unpruned baselines. This is particularly noticeable in smaller ViTs, schedules with high TPRs, and low-resolution images (App. D).

Pruning schedule design: The manuscript, and this supplementary in App. E, explore a variety of pruning schedules, which required manual design and task- and model-specific adaptations. In contrast, automated schedules, conditioned on user-defined constraints like target performance and throughput, would likely be more user-friendly.

Quite a few tasks but not all: We have evaluated Cropr solely on vision tasks. As discussed in the main text, Cropr could be extended to other modalities. Furthermore, as the title suggests we address quite a few tasks, but not all of them. While tasks such as fine-grained recognition are a trivial application of Cropr, other tasks such as visual question answering and image retrieval require follow-up work.

Fine-tuning requirement: Cropr requires training to adapt router and auxiliary head weights. In contrast, recent works explore post-hoc token reduction without fine-tuning. While appealing, note that these methods still achieve their best performance with fine-tuning. For example, ToMe [3], evaluated on ImageNet-1k with ViT-L MAE and $R = 8$, achieves 85.1% with fine-tuning compared to 83.9% without. ToFu [13] reaches 84.7% without fine-tuning, less than our 85.1%. We also reproduced GTP-ViT [21] with above setup and observed an accuracy of only 72.7% without fine-tuning; a lower pruning rate of $R = 4$ is required to recover performance to 84.7%. Performance gains from fine-tuning are consistent across methods, particularly at higher pruning rates, as shown in Table 2 of [21]. Furthermore, since fine-tuning from a pretrained checkpoint is still common in practice, our method in these cases requires no additional steps. It remains an open problem to supersede fine-tuning without sacrificing performance.

C. Hyperparameters

In Tabs. 1 to 4, we list hyperparameters for the datasets and models we use in our experiments. These settings are adopted from Fang et al. [8], He et al. [10], Strudel et al. [17]. Hyperparameter and design choices specific to Cropr are described in the main text.

D. Different image resolutions

We investigate the effect of image size on the performance and throughput of Cropr models. We apply Cropr with LLF to an MAE-pretrained ViT-L on ImageNet-1k at resolutions of 224, 336, and 448 pixels per side. The pruning rate R scales with image size to 8, 18, and 32 tokens per block, respectively, maintaining a TPR of 90% across all settings.

Figure 1 shows that Cropr's relative performance penalty decreases at higher resolutions, improving from -0.5 to -0.06, effectively closing the gap to the unpruned model. Furthermore, throughput gains are elevated at higher resolutions, going from a speedup of $1.7\times$ at 224^2 px to a speedup of $2.1\times$ at 448^2 px. This is perhaps due to the quadratic relationship between sequence length and compute in transformer models.

Config	Value
checkpoint	MAE-pretrained [10]
learning rate	4e-3
layer-wise lr decay [1, 5]	0.65 (B), 0.75 (L, H)
learning rate schedule	cosine decay [14]
optimizer	AdamW [15]
optimizer hparams	β_1, β_2, ϵ = 0.9, 0.999, 1e-8
weight decay	0.05
input size per side	224, 336 or 448
patch size	16 (B, L), 14 (H)
batch size	1024
epochs	100 (B), 50 (L/H)
warm-up epochs	5
label smoothing [18]	0.1
drop path [11]	0.1 (B), 0.2 (L), 0.3 (H)
augmentation	RandAug(9, 0.5) [6]
random resized crop	(0.08, 1)
cutmix [22]	1.0
mixup [23]	0.8
CLS token	✓

Table 1. **ImageNet-1k** image classification hyperparameters for **MAE**-pretrained encoders.

Config	Value
checkpoint	IN-21K fine-tuned EVA-02-L [8]
learning rate	2e-5
layer-wise lr decay [1, 5]	0.85
learning rate schedule	cosine decay [14]
optimizer	AdamW [15]
optimizer hparams	β_1, β_2, ϵ = 0.9, 0.999, 1e-8
weight decay	0.05
input size per side	448
patch size	14
batch size	512
epochs	20
warm-up epochs	2
label smoothing [18]	0.2
drop path [11]	0.15
augmentation	RandAug(9, 0.5) [6]
random resized crop	(0.08, 1)
cutmix [22]	✗
mixup [23]	✗
CLS token	✓

Table 2. **ImageNet-1k** image classification hyperparameters for **EVA-02**-pretrained encoders.

Config	Value
checkpoint	MIM pretrained EVA-02-L [8]
learning rate	2e-5
layer-wise lr decay [1, 5]	0.9
learning rate schedule	polynomial decay [4]
optimizer	AdamW [15]
optimizer hparams	β_1, β_2, ϵ = 0.9, 0.999, 1e-8
weight decay	0.05
input size per side	512
patch size	16
batch size	8
epochs	64
warm-up epochs	0
drop path [11]	0.2
CLS token	✓

Table 3. **ADE20k** semantic segmentation hyperparameters.

Config	Value
checkpoint	Objects365 fine-tuned EVA-02 [8]
learning rate	4e-5
layer-wise lr decay [1, 5]	0.8
learning rate schedule	constant
optimizer	AdamW [15]
optimizer hparams	β_1, β_2, ϵ = 0.9, 0.999, 1e-8
weight decay	0.1
input size per side	1536
patch size	16
batch size	64
training steps	40k
drop path [11]	0.3
large-scale jittering [9]	✓
attention window size	16
global attn block ids	3, 6, 9, 12, 15, 18, 21, 24
max numbers of detection	100
softNMS [2]	IoU threshold = 0.6
maskness scoring [12, 20]	maskness threshold = 0.5
EMA decay [16]	0.999
CLS token	✗

Table 4. **COCO** object detection and instance segmentation hyperparameters.

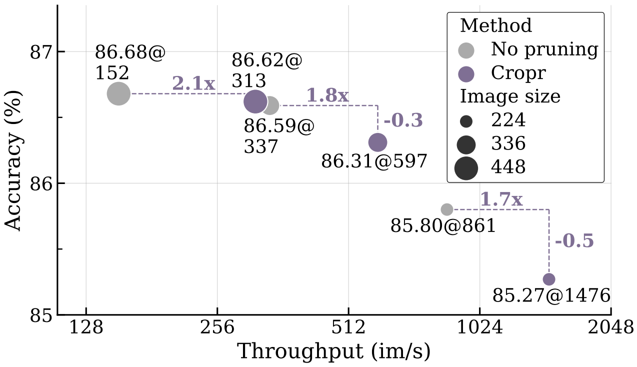


Figure 1. Performance-throughput trade-off plot for different image sizes on ImageNet-1K. Token pruning in higher-resolution images provides more speedup and less performance drop.

E. Throughput ablations

In this section, we evaluate different pruning rates R, investigate the effect of keep token sequence lengths on runtime, and compare different numerical precision modes and FlashAttention [7]. ViT-L is employed for all ablations.

Different pruning rates. We ablate the pruning rate R in our image classification setting, fine-tuning an MAE-pretrained ViT on ImageNet-1K with Cropr and LLF. We vary the pruning rate from $R = 0$ (no pruning) to $R = 8$ (value used in the manuscript). We report top-1 accuracy and throughput in Tab. 5. For light schedules, with $R \leq 2$, performance is maintained with up to 8% higher throughput. When allowing for a drop of 0.1 accuracy points, the model can be accelerated up to 35% using $R = 5$.

R	Acc.	im/s	Comments
0	85.8	861 1.00×	No performance drop
1	85.8	883 1.03×	
2	85.8	934 1.08×	
3	85.7	996 1.16×	0.1% Accuracy drop
4	85.7	1067 1.24×	
5	85.7	1160 1.35×	
6	85.6	1244 1.44×	
7	85.5	1357 1.58×	
8	85.3	1476 1.71×	

Table 5. Accuracy and throughput for varying pruning rates on ImageNet-1k using an MAE-pretrained ViT-L.

Being divisible by 8? Small changes in the number of keep tokens has a surprisingly large impact on throughput. We evaluated this effect across image sizes 512, 1024, and 2048, with corresponding patch sequence lengths $M = 1024, 4096$, and 16384, respectively, with a patch size of 16 (ignoring the CLS token). Cropr is applied without LLF.

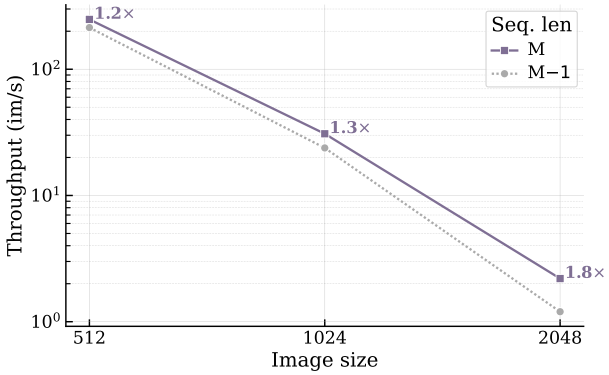


Figure 2. Effect of sequence length M on throughput for different image sizes. Annotations denote speedups. A mere reduction of 1 token, instead of giving a negligible speedup, results in significant throughput drops. Both the x and y-axis are log scaled.

We compare the throughput of two models in Fig. 2. The solid line uses pruning rates R of 40, 160, and 640 tokens per block for each image size respectively, resulting in a TPR of 90% across image sizes. The dotted line on the other hand artificially sets the sequence lengths to $M - 1$, i.e. subtracting one patch with otherwise identical settings, resulting in initial sequence lengths of 1023, 4095, and 16383.

As seen in the plot, despite the reduction of one token in the dotted line case, the throughput drops significantly. At the highest resolution, this is in fact a 1.8× slowdown. This slowdown is likely due to worse memory alignment and thread utilization in the accelerator. We hypothesize that schedules where the number of remaining tokens is divisible by 8 are likely to achieve the highest throughput and used that as a rule of thumb when designing pruning schedules for all our experiments.

Numerical precision and FlashAttention. In the main paper, all models were run using automatic mixed precision (AMP). Changes to this setting primarily affect model throughput. Here, we add to that and report throughputs for models that use (a) FP32 numerical precision, and (b) AMP in combination with FlashAttention [7]. Cropr is applied without LLF, setting R as in the previous ablation to achieve a TPR of 90% for all image sizes.

As shown in Fig. 3, Cropr improves over the unpruned baseline in terms of throughput in all three settings. Relative speedups are higher for larger images, in line with the findings in App. D. Notably, for images at a resolution of 2048×2048, Cropr achieves a speedup of up to 8.9× when using AMP.

AMP + Flash Attention is the fastest setting overall. But even in this optimized regime, Cropr delivers a significant speedup between 1.7× and 2.3×.

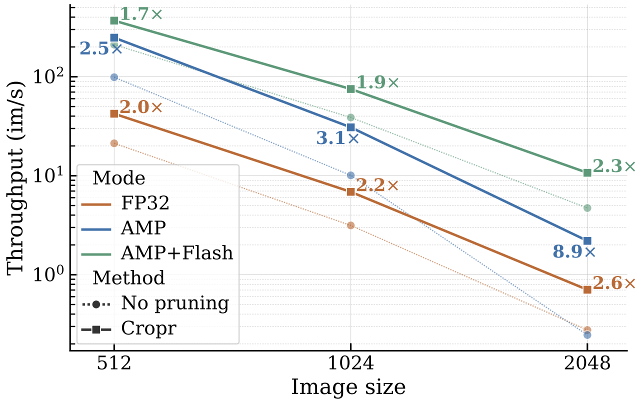
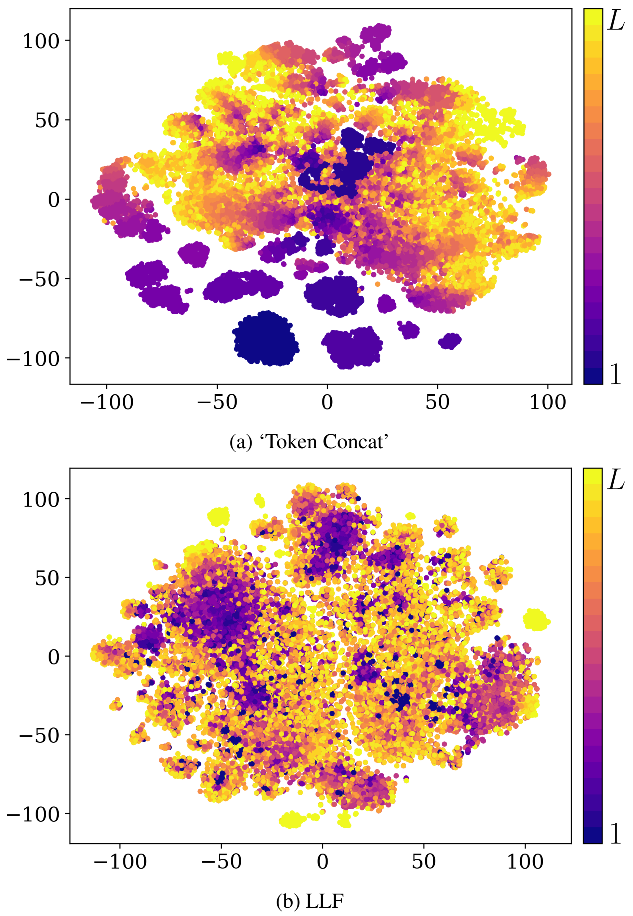


Figure 3. Throughput ablations for FP32, AMP, and AMP with FlashAttention across image sizes. Annotations denote speedups of Cropr over the unpruned baselines.

F. t-SNE visualizations of LLF's effect

In Sec. 4.4 Tab. 5 of the manuscript, we compared LLF and the 'Token Concat' baseline. Whereas 'Token Concat' performs token concatenation after the last transformer block, LLF does it after the penultimate block, enabling the pruned tokens and kept tokens to attend into each other and to loosely speaking synchronize. We visualize this effect in Fig. 4 using t-SNE [19] down-projected tokens.

We apply t-SNE to the ADE20k validation set, and for visual clarity we plot only the top-1 scoring token within the respective pruned token sets per block. Points are then colored according to the block number of the block after which they were pruned. As seen in the 'Token Concat' case, Fig. 4a, tokens pruned after different blocks occupy different regions in the embedding space, which might be challenging for the linear prediction head trying to map them into class labels. In the LLF case, Fig. 4b, the embedding space is more uniformly occupied by tokens pruned at different stages, supporting our hypothesis that LLF helps synchronize these tokens. We argue that this may be easier for the linear prediction head to then learn a projection into class logits.



(a) 'Token Concat'

(b) LLF

Figure 4. t-SNE projections of tokens extracted right before the prediction head. Tokens are colored according to the block after which they were pruned. We compare two fusion methods: (a) 'Token Concat', (b) LLF. The latter has a more uniform distribution suggesting that LLF helped synchronize these tokens.

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2021. 2

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, pages 5561–5569, 2017. 2

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

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2018. 2

[5] K Clark. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020. 2

[6] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, pages 3008–3017, 2020. 2

[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. 3

[8] 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. 1, 2

[9] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple

copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, pages 2918–2928, 2021. 2

[10] 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. 1, 2

[11] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661. Springer, 2016. 2

[12] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, pages 6409–6418, 2019. 2

[13] 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

[14] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 2

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[16] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 2

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

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 2

[19] Laurens Van Der Maaten. Learning a parametric embedding by preserving local structure. In *Artificial intelligence and statistics*, pages 384–391. PMLR, 2009. 4

[20] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Solo: A simple framework for instance segmentation. *IEEE TPAMI*, 44(11):8587–8601, 2021. 2

[21] 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

[22] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 2

[23] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2