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

Motivation

- Many vision applications require…
  - Real-time latency
  - High throughput
  - Energy efficiency

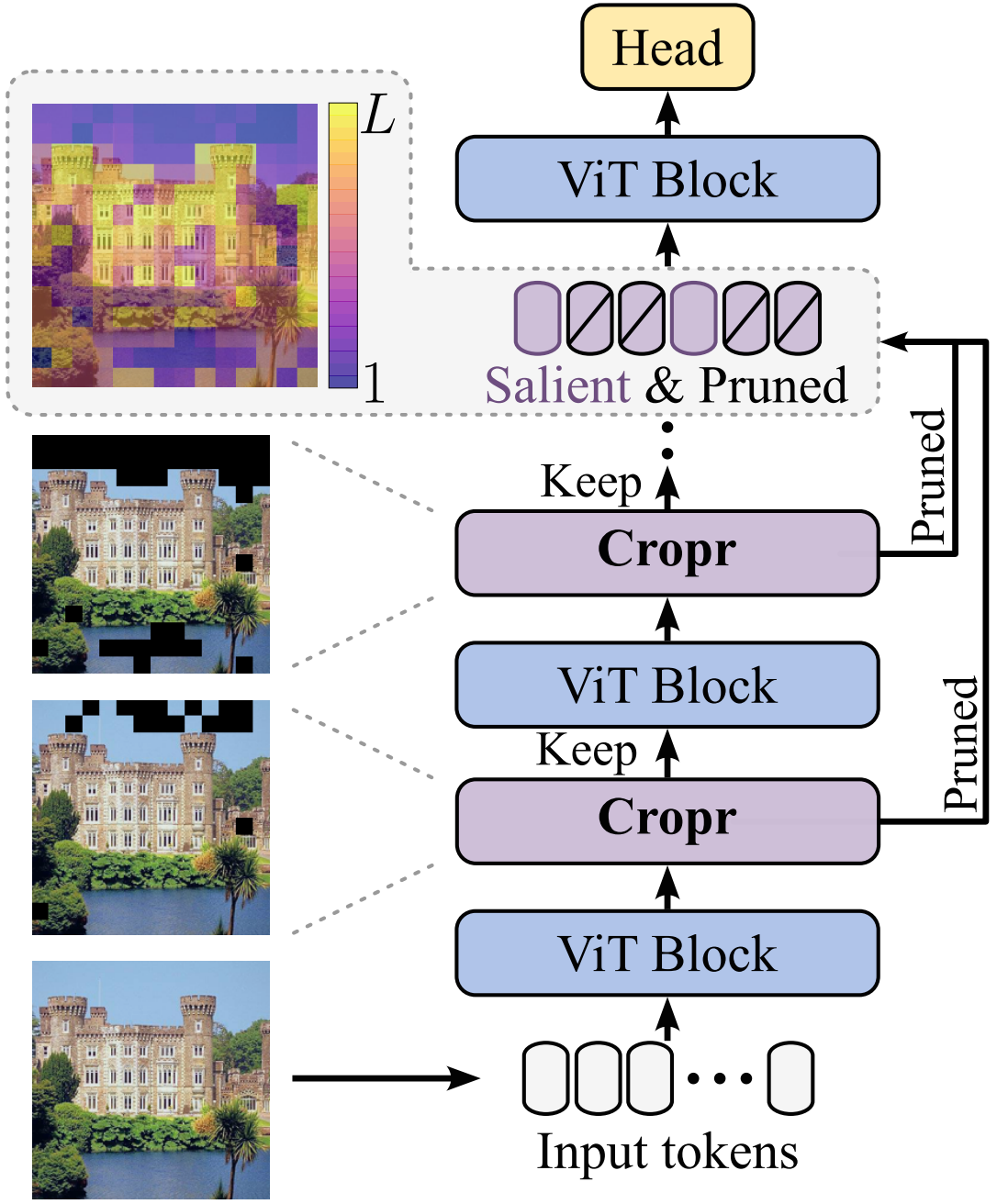
- ViTs exhibit inefficiencies
  - Square self-attention matrix
  - One token per patch, no pooling

Goal: Improve inference efficiency

… while maintaining high task performance

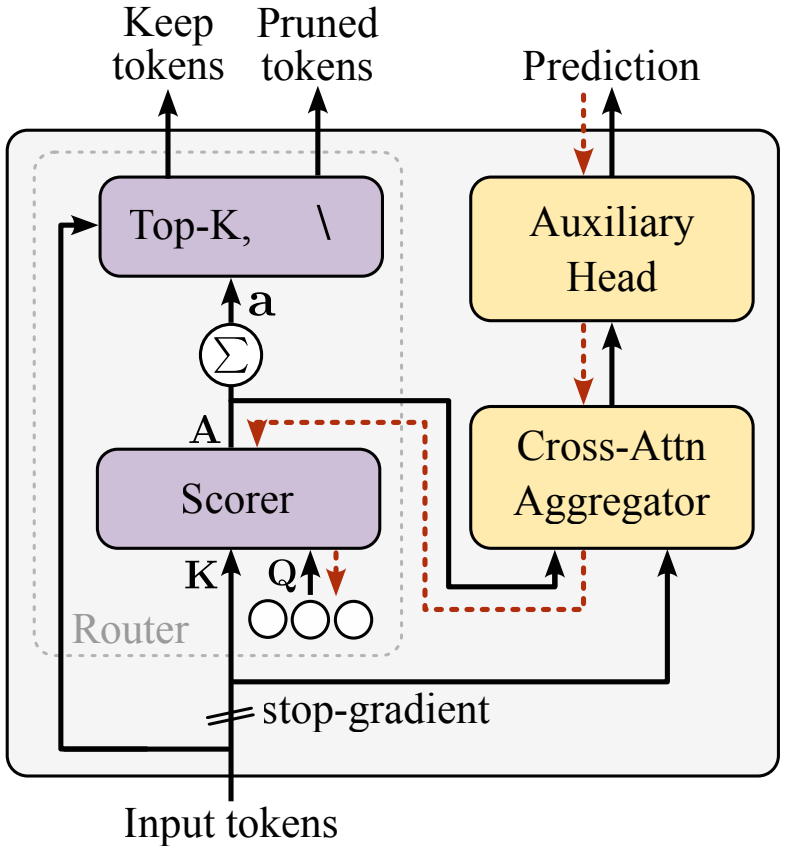
Overview

- **Cro**ss-attention **pr**uning modules prune the least relevant tokens for a given task
- Retained tokens forwarded to next layer
- Accelerates ViTs by **1.5-4x** while maintaining high task performance
- Pruned tokens are reintroduced before last block using Last Layer Fusion
- Applicable to **quite a few vision tasks**: classification, semantic segmentation, object detection, instance segmentation

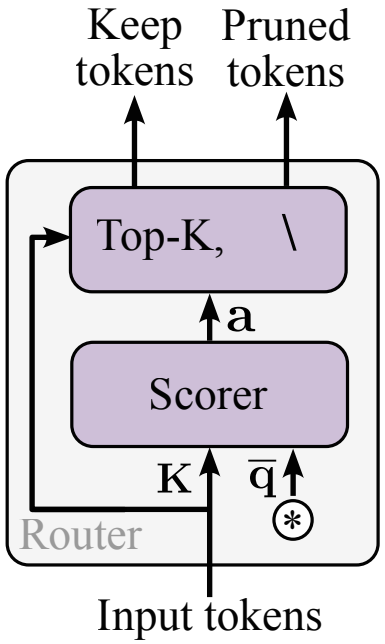


Cross-attention pruning module **Cropr**

during training



during inference



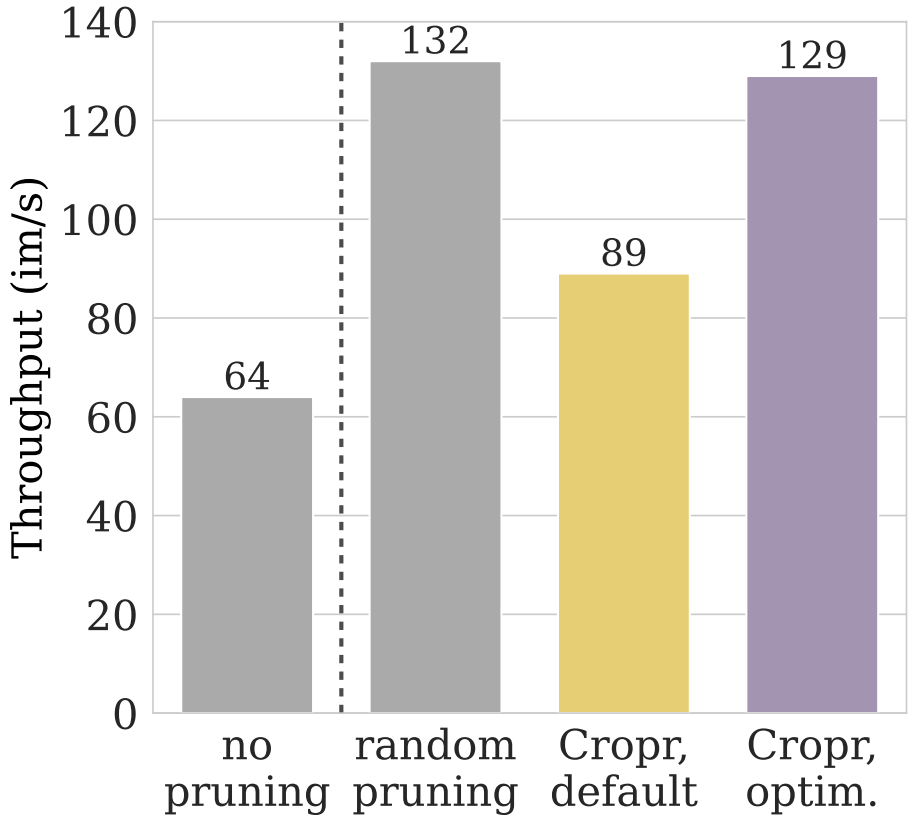


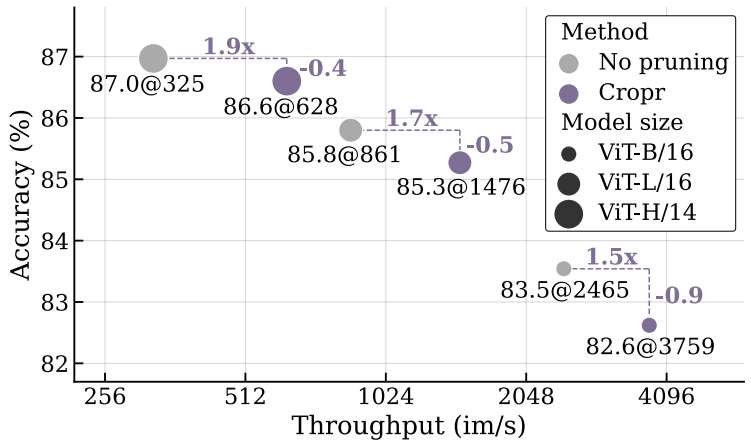
Image classification (IN-1k)

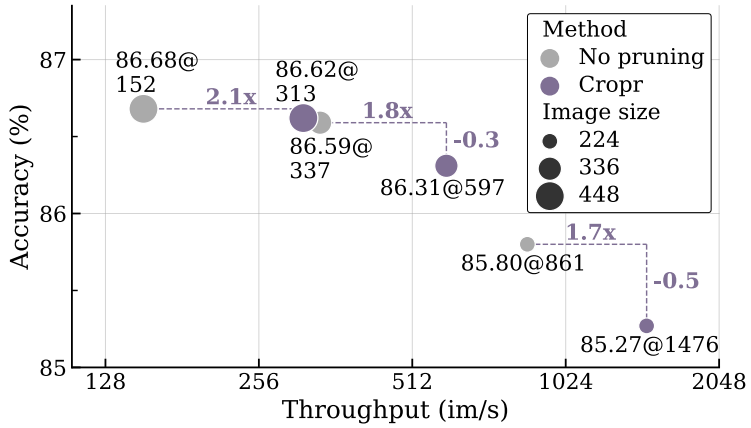
Method	Sch.	LLF	Pool	Acc.	1000 im/s
No pruning	—	—	avg	85.8	0.86 1.0×
Non-salient	↘	✓	avg	76.4	1.48 1.7×
Random	↘	✓	avg	83.8	1.50 1.7×
Variance [32]	↘	✓	avg	84.3	1.50 1.7×
Attn Top-K	↘	✓	cls	84.7	1.45 1.7×
Cropr	↘	✓	avg	**85.3**	1.48 1.7×
K-Medoids [31]	↘		avg	84.5	0.31 0.4×
ATS [14]	↘		cls	83.9	0.49 0.6×
DPC-KNN [12]	↘		avg	79.2	1.00 1.2×
EViT [26]	↘		cls	84.5	1.57 1.8×
ToMe, from [4]	↘		cls	**85.1**	1.55 1.8×
ToMe [4]	↘		avg	85.0	1.55 1.8×
Cropr	↘		avg	**85.1**	1.61 1.9×
DynamicViT [37]	↴		avg	64.4	1.32 1.5×
SiT [60]	↴		avg	83.0	1.41 1.6×
Sinkhorn [15]	↴		avg	56.5	1.40 1.6×
PatchMerger [38]	↴		avg	82.4	1.40 1.6×
Cropr	↴		avg	85.4	1.43 1.7×
Cropr	↴	✓	avg	**85.5**	1.35 1.6×

Comparison to other token reduction methods

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 ↓	448	0.3B	0.07B	88.8	259

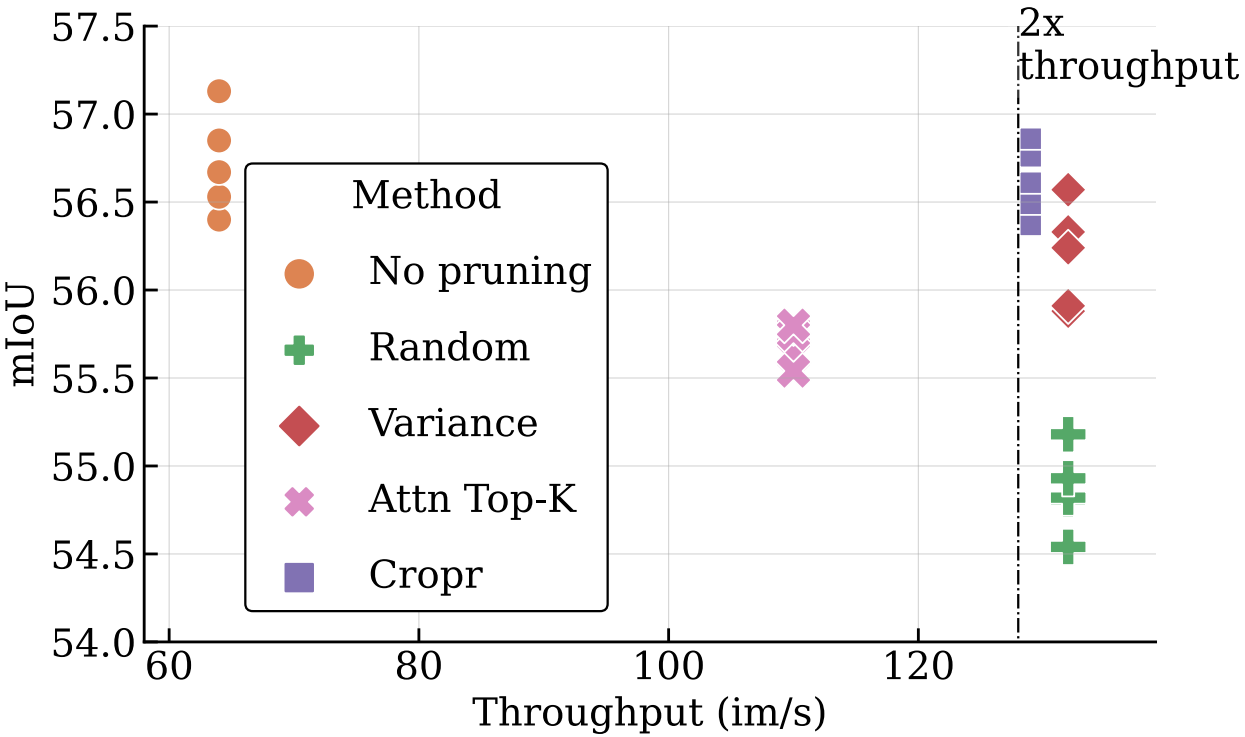
Cropr applied to a SoTA model



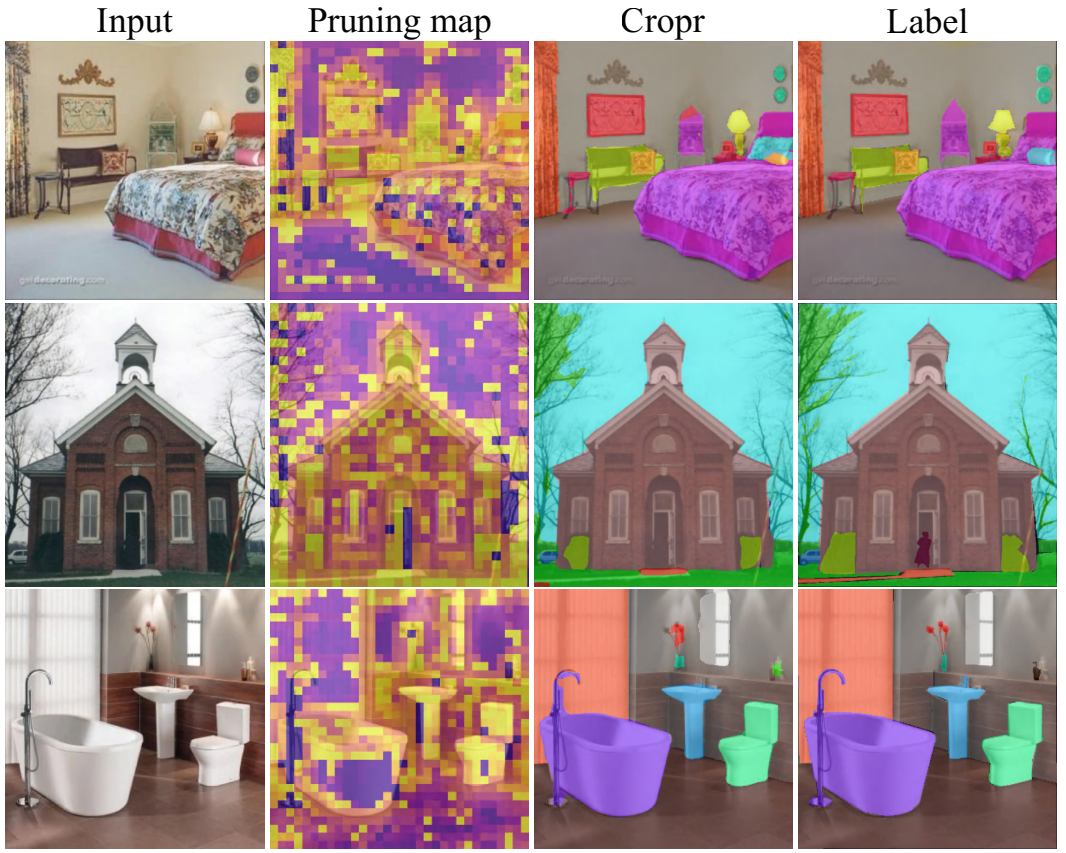


Throughput gains increase with larger models and image resolutions, while performance drop decreases.

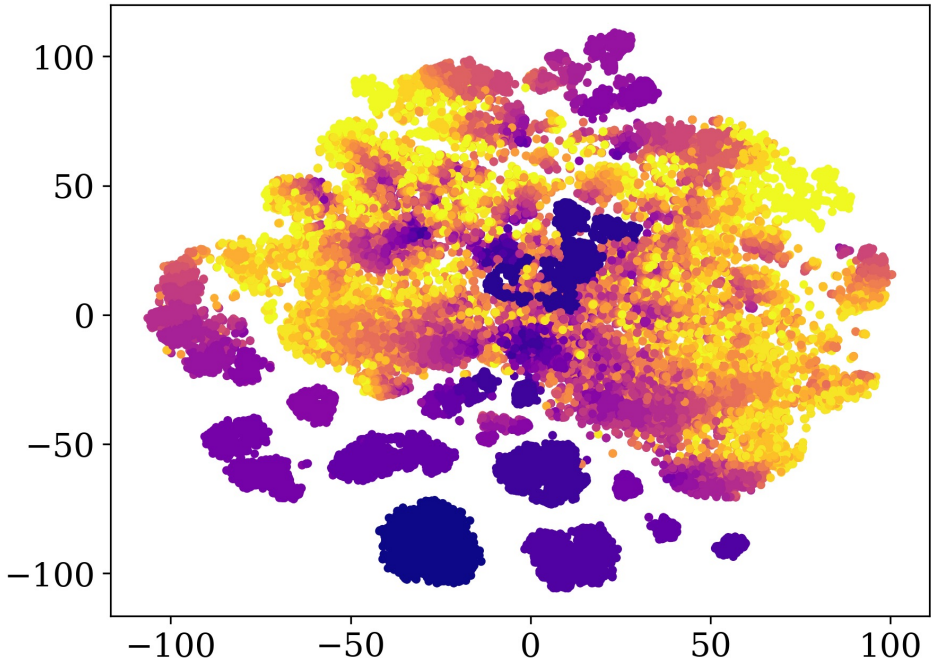
Semantic segmentation



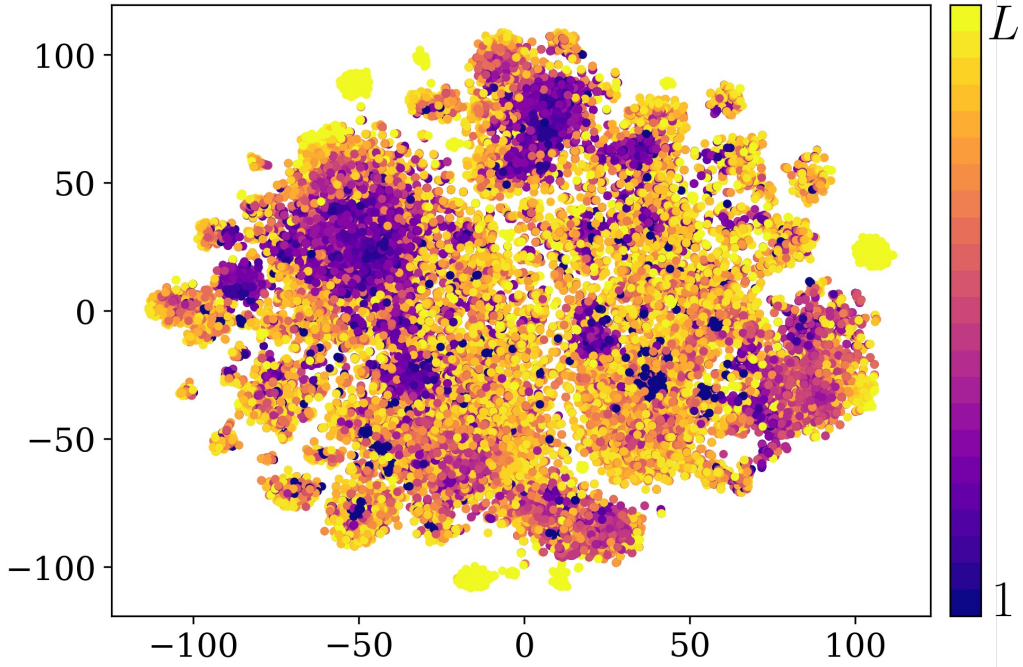
Cropr *performs comparable to* No pruning*, with 2x throughput increase.*



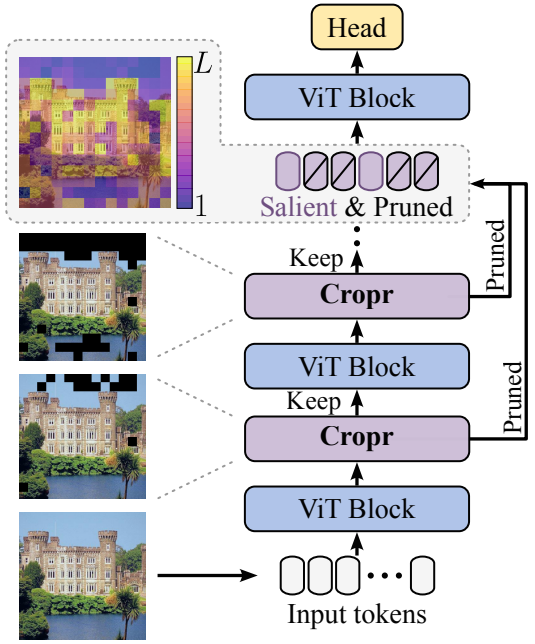
Cropr prunes tokens from stuff classes earlier, but keeps a few tokens from each class in later layers. Despite pruning, adjacent outputs of the same class appear consistent.



The token concatenation baseline merges keep and pruned tokens after the last transformer block, distinct clusters form per block.



Last Layer Fusion (LLF) fuses tokens before the last block, leading to more uniform representations and suggesting better synchronization.

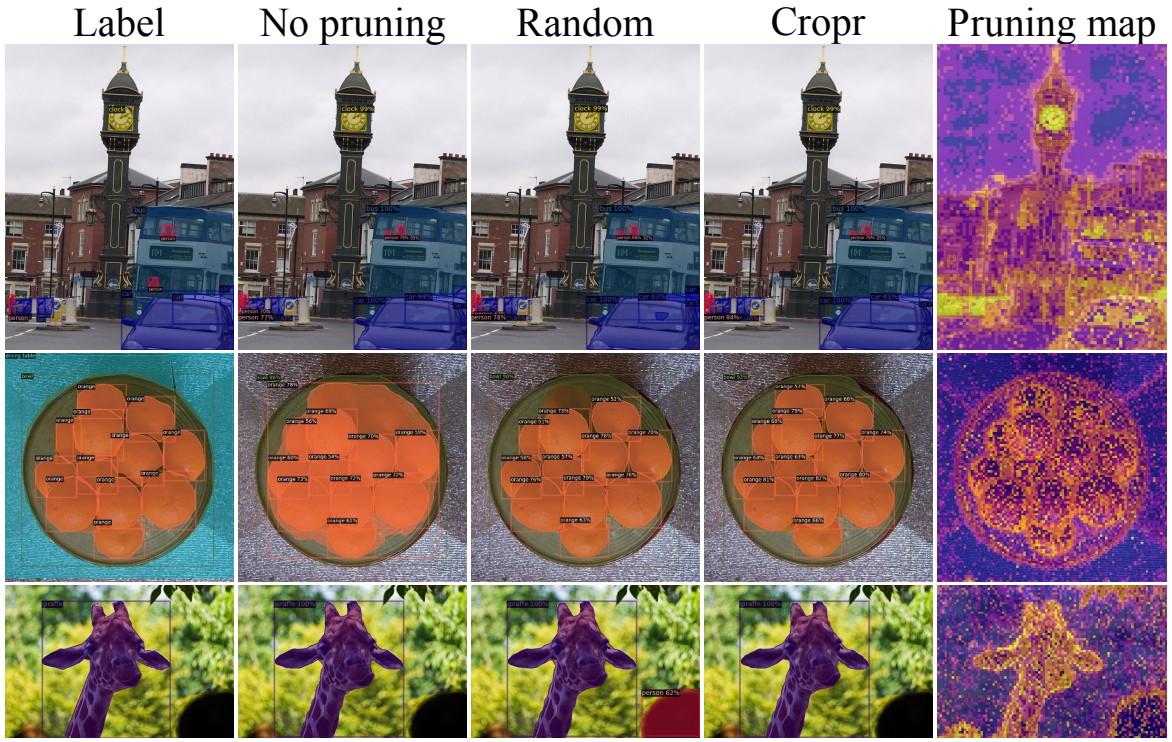


Method	#Params	GFlops	mIoU
No pruning	304M	311	56.7
Cross-Attn	319M	184	49.3
Token Concat	304M	172	51.8
Cross-Attn + Concat	319M	186	51.1
MHSA + Concat	318M	186	55.2
DToP	308M	174	50.1
LLF	304M	183	**56.6**

Token fusion ablation on ADE20k. Median mIoU across 5 seeds.

LLF performs best, without additional parameters.

Object detection and instance segmentation

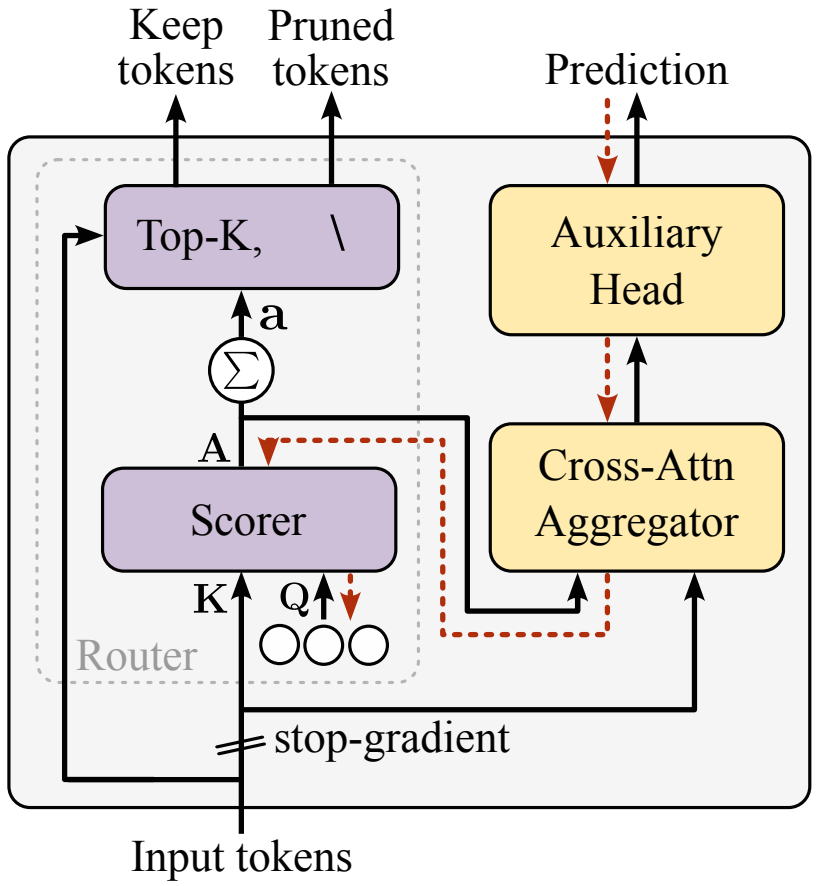


Cropr pruning maps highlight task-relevant objects.

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×

On COCO, we prune 97% of all tokens with a 2.4x speedup at a moderate performance loss.

Architecture ablations



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.

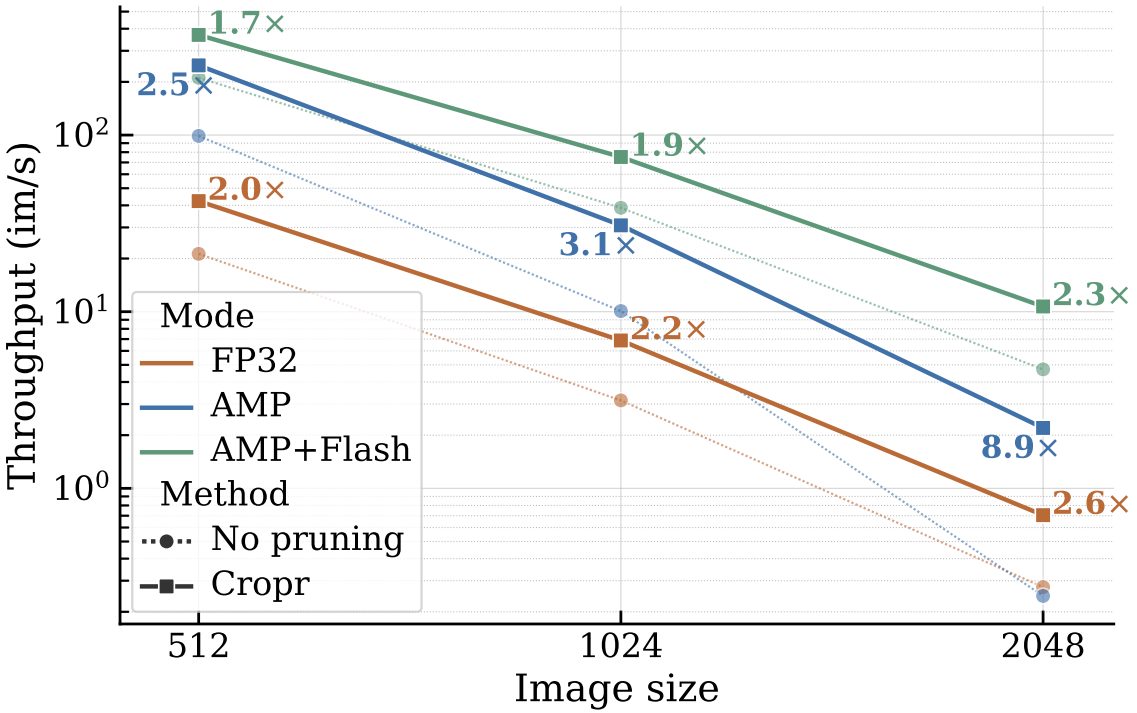
MLP	Acc.	GFlops	im/s
✗	85.0	34.2	1476
✓	85.3	34.2	1476

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

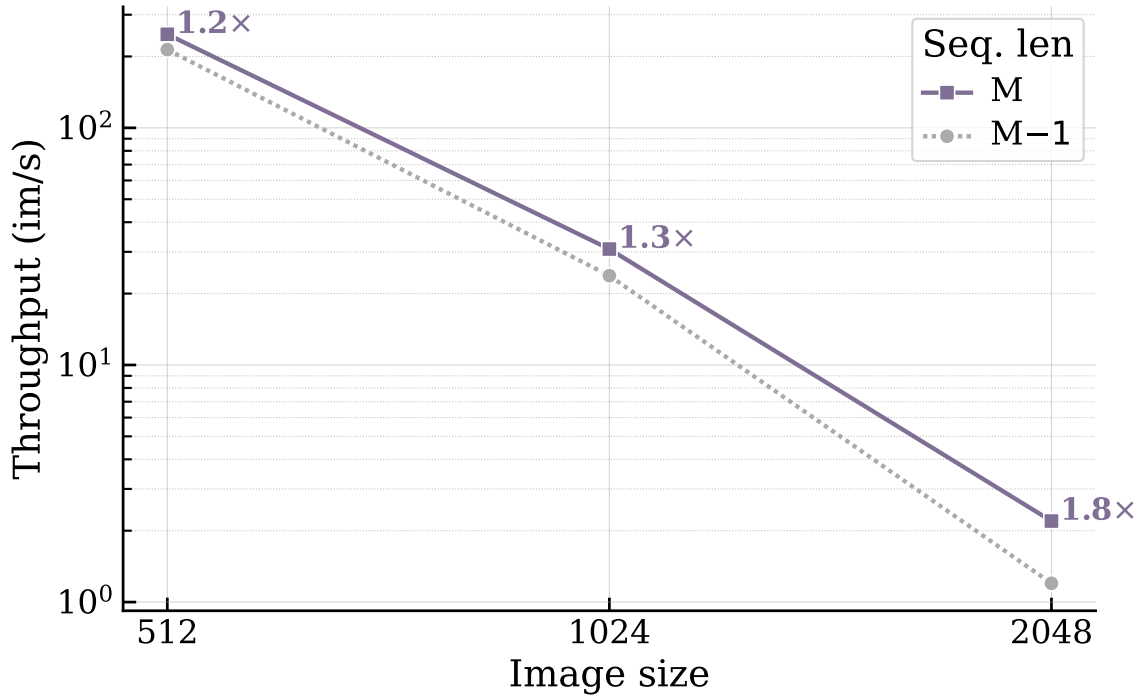
Stop grad.	Acc.
✗	85.0
✓	85.3

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

Throughput ablations



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



Effect of sequence length M on throughput for different image sizes. A mere reduction of 1 token, instead of giving a negligible speedup, results in significant throughput drops.

Thanks

Benjamin Bergner
bergner.benjamin@gmail.com

Christoph Lippert
christoph.lippert@hpi.de

Aravindh Mahendran
aravindhm@google.com

Paper

Code