UC San Diego

Tencent AI Lab

香 港 大 學
THE UNIVERSITY OF HONG KONG

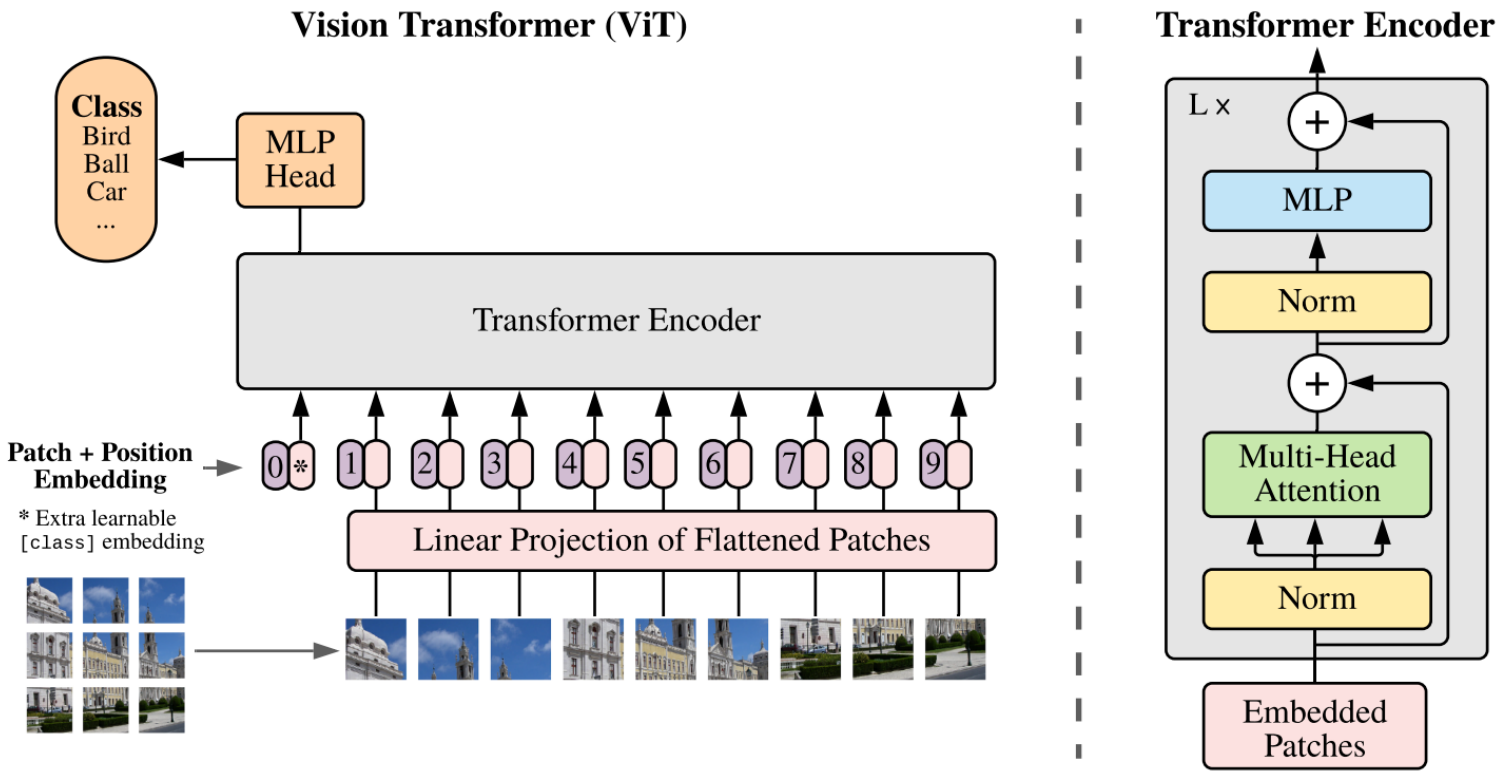
Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations

Youwei Liang[1], Chongjian Ge[2], Zhan Tong[3], Yibing Song[3], Jue Wang[3], Pengtao Xie[1]

[1] UC San Diego [2] The University of Hong Kong [3] Tencent AI Lab

Vision Transformers (ViTs)

- Divide an image into patches, with each patch as a token
- Construct attention computation among tokens
- Is it possible to use a subset of the tokens to make ViTs more efficient?



An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (https://arxiv.org/pdf/2010.11929.pdf)

Motivations

- Randomly remove tokens in a trained DeiT-S (Touvron et al., 2021a)
- In (a), removing image tokens unrelated to the visual content of the corresponding category does not deteriorate ViT predictions. In (b), removing related image tokens makes ViT predict incorrectly.

(a) Mask on backgrounds

(b) Mask on objects

Attentive Token Identification

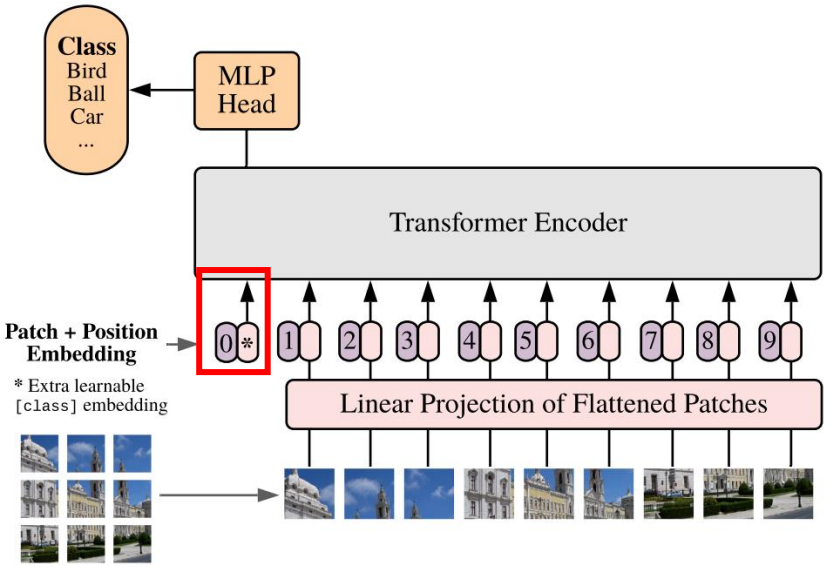
- Class token is used for classification
- Class token is a linear combination of representations of all tokens
- Class attention score determines the weight of the combination
- Important (Attentive) tokens get higher weight

$$\boldsymbol{x}_{\text{class}} = \text{Softmax}(\frac{\boldsymbol{q}_{\text{class}} \cdot \boldsymbol{K}^{\top}}{\sqrt{d}})\boldsymbol{V} = \boldsymbol{a} \cdot \boldsymbol{V}$$

Class token

Class attention

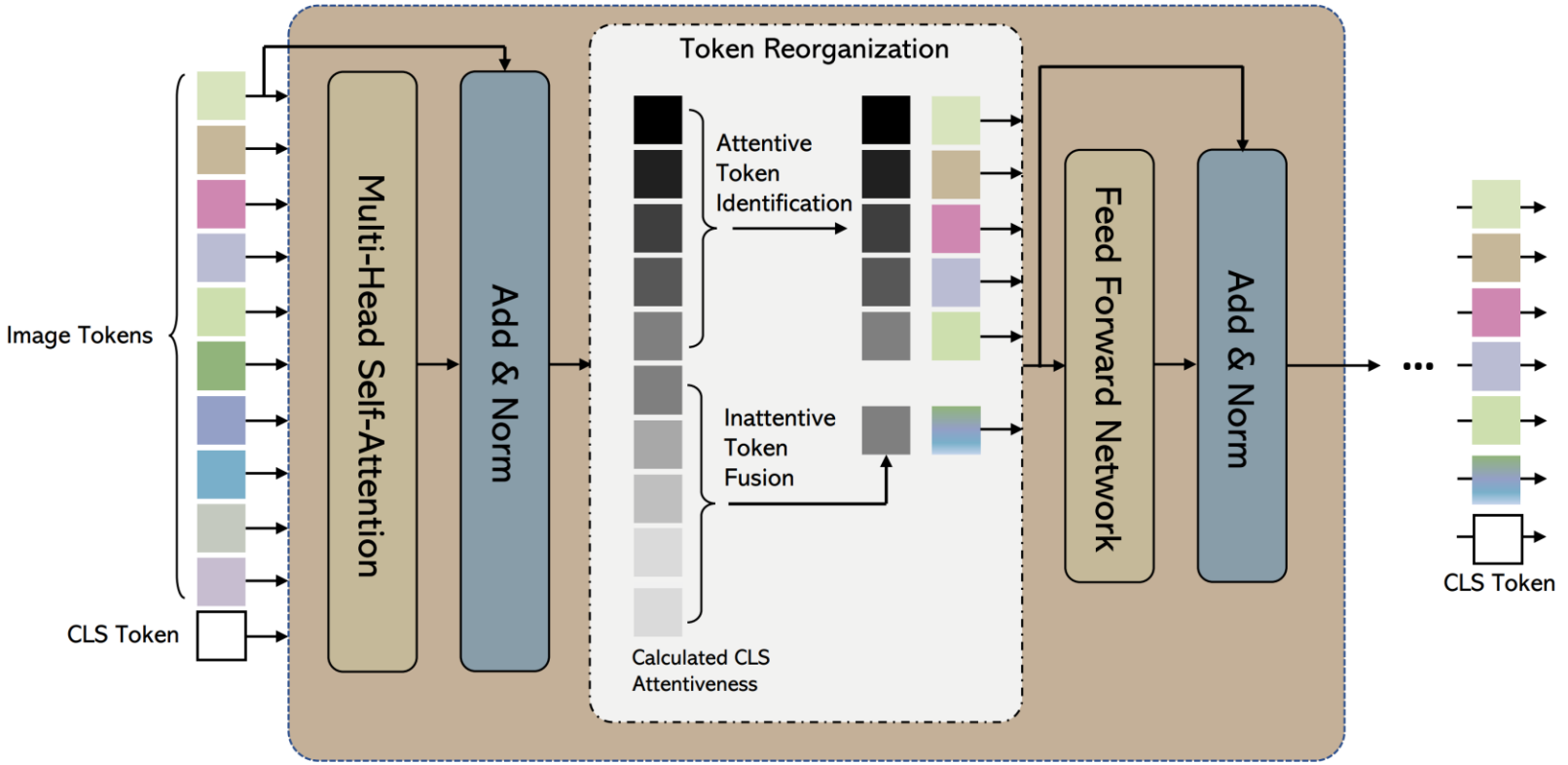
All tokens



- Use the class attention score as a selection criterion
- Keep the tokens with top class attention scores

Token reorganization

- Between the MHSA and FFN
- Class attention scores: averaged over all MHSA heads
- Keep the tokens with top class attention scores
- Hyper-parameter: token keep rate
- An extra point: fuse the inattentive tokens



Experiments

- Incorporating token reorganization to a **trained** DeiT-S
- Classification accuracy drops
- We propose to incorporate image token reorganization during the **ViT training** process

Token keep rate	1.0	0.9	0.8	0.7	0.6	0.5
Top-1 Acc (%)	79.8	79.7(-0.1)	79.2(-0.6)	78.5(-1.3)	76.8(-3.0)	73.8(-6.0)

ImageNet classification accuracy (*without training*)

Experiments

- Apply token reorganization to two ViT variants: DeiT and LV-ViT
- Train for 300 epochs on ImageNet (same setting as the vanilla models)
- Results: large speedup, small accuracy drop

Keep rate	Top-1 Acc (%)	Top-5 Acc (%)	Throughput (images/s)	MACs	Top-1 Acc (%)	Top-5 Acc (%)	Throughput (images/s)	MACs
DeiT-S	79.8	94.9	2923	4.6	79.8	94.9	2923	4.6
	EViT without inattentive token fusion				EViT with inattentive token fusion			
0.9	79.9±0.1 (+0.1)	94.9±0.0 (-0.0)	3201 (+10%)	4.0 (-13%)	79.8±0.1 (-0.0)	95.0±0.0 (+0.1)	3197 (+9%)	4.0 (-13%)
0.8	79.7±0.0 (-0.1)	94.8±0.0 (-0.1)	3772 (+27%)	3.5 (-24%)	79.8±0.0 (-0.0)	94.9±0.0 (-0.0)	3619 (+24%)	3.5 (-24%)
0.7	79.4±0.1 (-0.4)	94.7±0.1 (-0.2)	4249 (+45%)	3.0 (-35%)	79.5±0.0 (-0.3)	94.8±0.0 (-0.1)	4385 (+50%)	3.0 (-35%)
0.6	79.1±0.2 (-0.7)	94.5±0.1 (-0.4)	4967 (+70%)	2.6 (-43%)	78.9±0.1 (-0.9)	94.5±0.0 (-0.4)	4722 (+62%)	2.6 (-43%)
0.5	78.4±0.2 (-1.4)	94.1±0.0 (-0.8)	5325 (+82%)	2.3 (-50%)	78.5±0.0 (-1.3)	94.2±0.0 (-0.7)	5408 (+85%)	2.3 (-50%)

Keep rate	Top-1 Acc (%)	Top-5 Acc (%)	Throughput (images/s)	MACs (G)
LV-ViT-S	83.3	–	2112	6.6
0.7	83.0 (-0.3)	96.3	2954 (+40%)	4.7 (-29%)
0.5	82.5 (-0.8)	96.2	3603 (+71%)	3.9 (-41%)

Using higher-resolution images

- More images tokens
- Higher accuracy, with same throughput as a vanilla model with smaller-size images

(a) DeiT-S

Model	Keep rate	Image size	Top-1 (%)	Top-5 (%)	img/s	MACs (G)
DeiT-S	1.0	224	79.8	94.9	2923	4.6
EViT	0.5	256	79.3	94.7	3788	3.1
EViT	0.5	288	80.1	95.0	3138	3.9
EViT	0.5	304	81.0	95.6	2905	4.4
EViT	0.6	256	80.0	95.0	3524	3.5
EViT	0.6	288	81.0	95.4	2927	4.5
EViT	0.7	272	80.3	95.3	2870	4.6

(b) LV-ViT-S

Model	Keep rate	Image size	Top-1 (%)	Top-5 (%)	img/s	MACs (G)
LV-ViT-S	1.0	224	83.3	–	2112	6.6
LV-ViT-S	1.0	224 ↑ 384	84.4	–	557	21.9
EViT	0.9	240	83.6	96.5	1956	6.8
EViT	0.8	256	83.6	96.6	1901	6.9
EViT	0.7	256	83.5	96.5	2102	6.2
EViT	0.7	272	83.7	96.6	1829	7.1
EViT	0.5	304	83.4	96.5	1758	7.4
EViT	0.7	272 ↑ 448	84.7	97.1	548	21.5

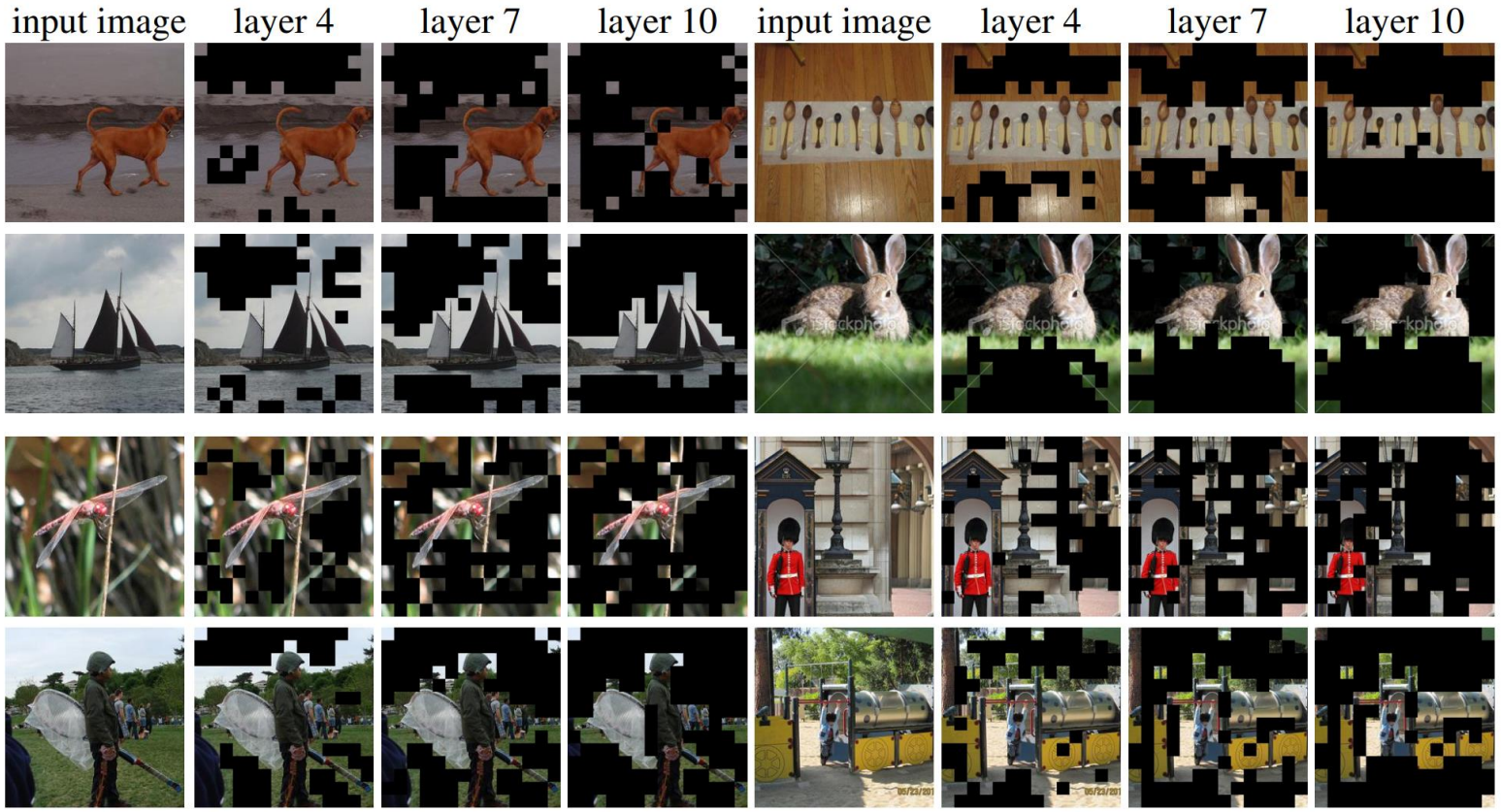
Comparison with DynamicViT

- EViT: requires lower training budget, obtains higher accuracy

Model	Pre-trained	Epochs	Keep rate	Top-1 (%)	#Params (M)	Throughput (img/s)	MACs (G)
DynamicViT-DeiT-S	✓	30	0.5	77.5	22.8	5579	2.2
EViT-DeiT-S	✓	30	0.5	78.5 (+1.0)	22.1 (-0.7)	5549	2.3
EViT-DeiT-S	✓	100	0.5	79.1 (+1.6)	22.1 (-0.7)	5549	2.3
DynamicViT-DeiT-S	✓	30	0.7	79.3	22.8	4439	3.0
EViT-DeiT-S	✓	30	0.7	79.5 (+0.2)	22.1 (-0.7)	4478	3.0
EViT-DeiT-S	✓	100	0.7	79.8 (+0.5)	22.1 (-0.7)	4478	3.0
DynamicViT-DeiT-S	×	300	0.5	73.1	22.8	5579	2.2
EViT-DeiT-S	×	300	0.5	78.5 (+5.4)	22.1 (-0.7)	5549	2.3
DynamicViT-DeiT-S	×	300	0.7	77.6	22.8	4439	3.0
EViT-DeiT-S	×	300	0.7	79.5 (+1.9)	22.1 (-0.7)	4478	3.0

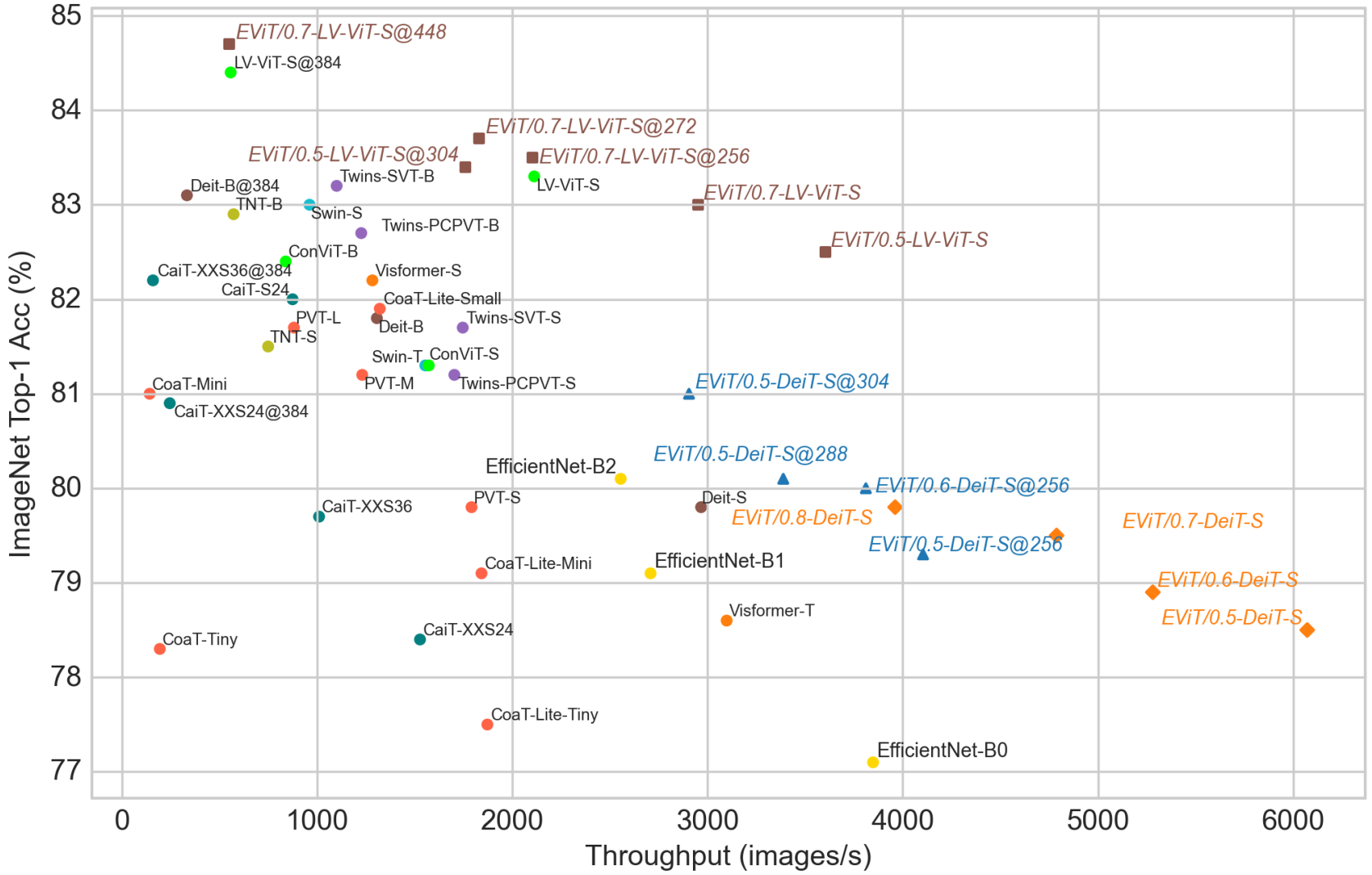
Visualization of the removed tokens

- Removed tokens are mostly on backgrounds or information-sparse area



Comparison with other ViTs

- EViT achieves a competitive trade-off between accuracy and throughput



Thank you!

- Paper: https://arxiv.org/abs/2202.07800
- Code: https://github.com/youweiliang/evit

Paper

Code