All Tokens Matter: Token Labeling for Training Better Vision Transformers

Zihang Jiang [1] Qibin Hou [1] Li Yuan [1] Daquan Zhou [1] Yujun Shi [1]
Xiaojie Jin [2] Anran Wang [2] Jiashi Feng [1]
National University of Singapore [1] ByteDance [2]

Outline

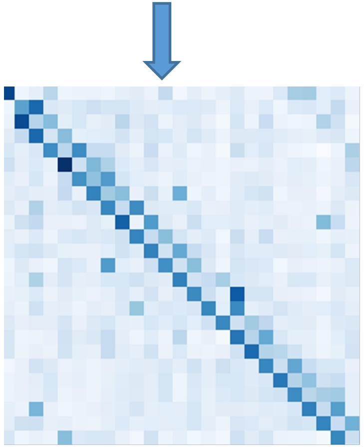
- **Background**
- **Problems and improvements**
- **Introduction**
- **Method**
- **Experiment & analysis**

Background

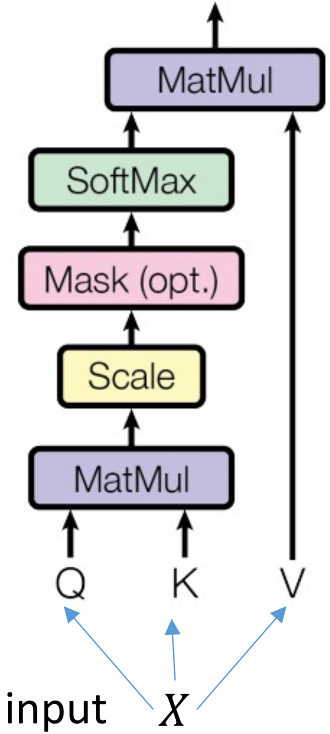
- **Attention mechanism**

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Attention weight for
Pair-wise relationship



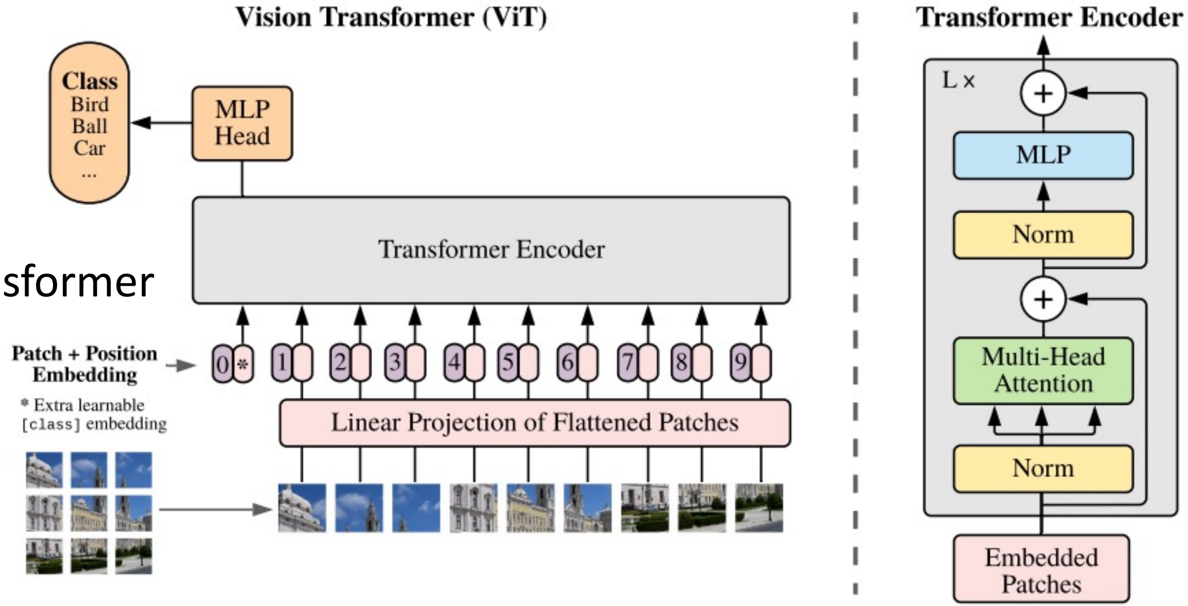
Scaled Dot-Product Attention



Background

- Vision Transformer:

  - Split image into patches
  - Feed patch sequence into Transformer



Problems and Improvements

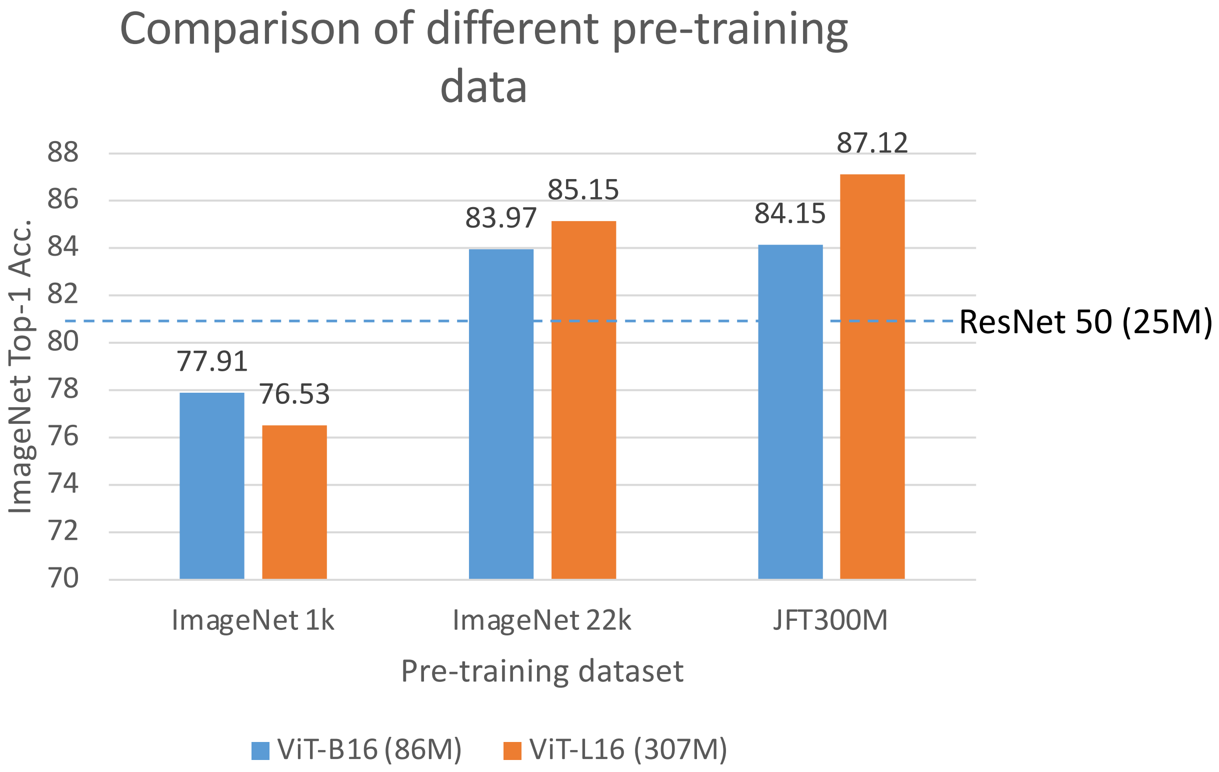
- **Problems of ViT:**
  - Need **lots of** training data to avoid **overfitting**
  - Only utilize a single class token for prediction

- **Improvements**
  - Propose **token labelling objective** to improve the training of transformer-based visual models
  - takes advantage of both the **patch tokens** and the **class tokens**

Introduction

Performance drop

- When train on “small” dataset like ImageNet 1k, ResNet50 outperforms large ViT models .

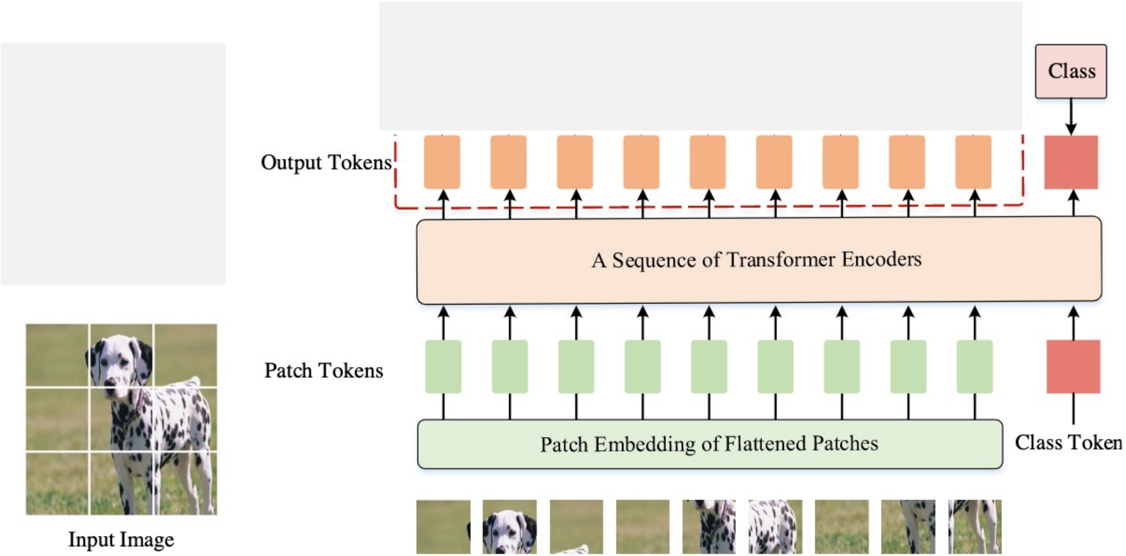


Introduction

Standard training vs Token labeling

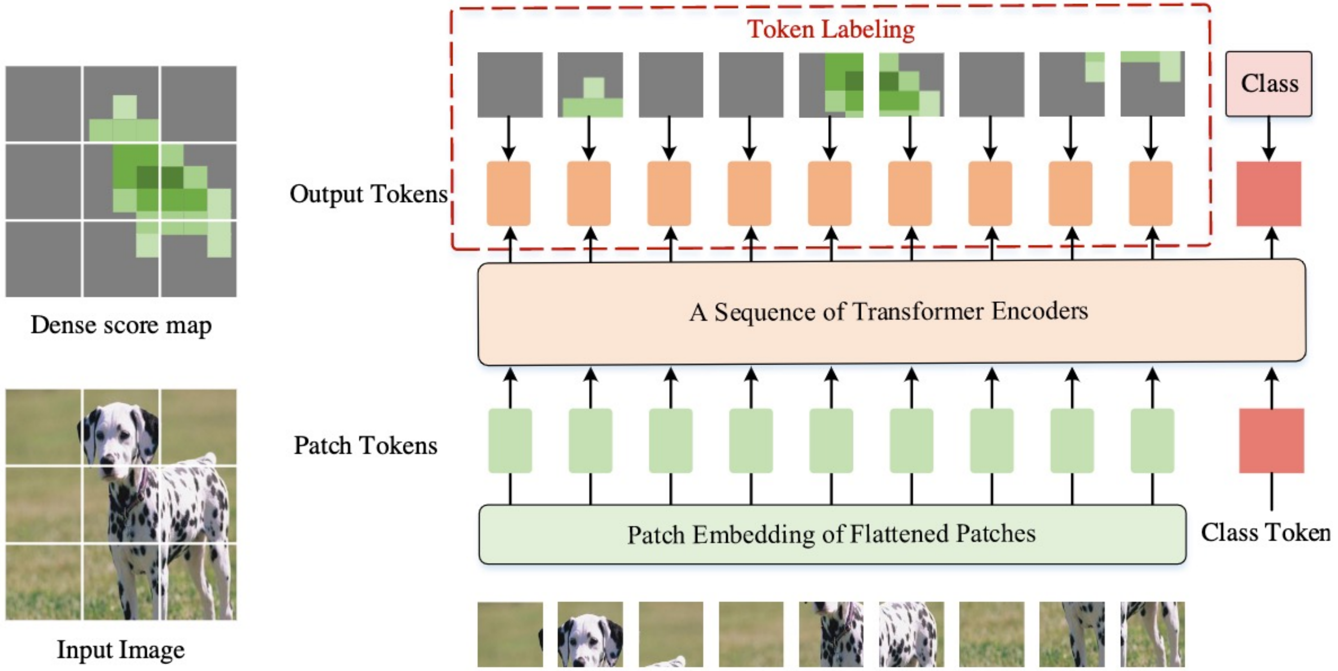
Improve training using

dense token-level supervision



Method

Detail of token labeling objective



Method

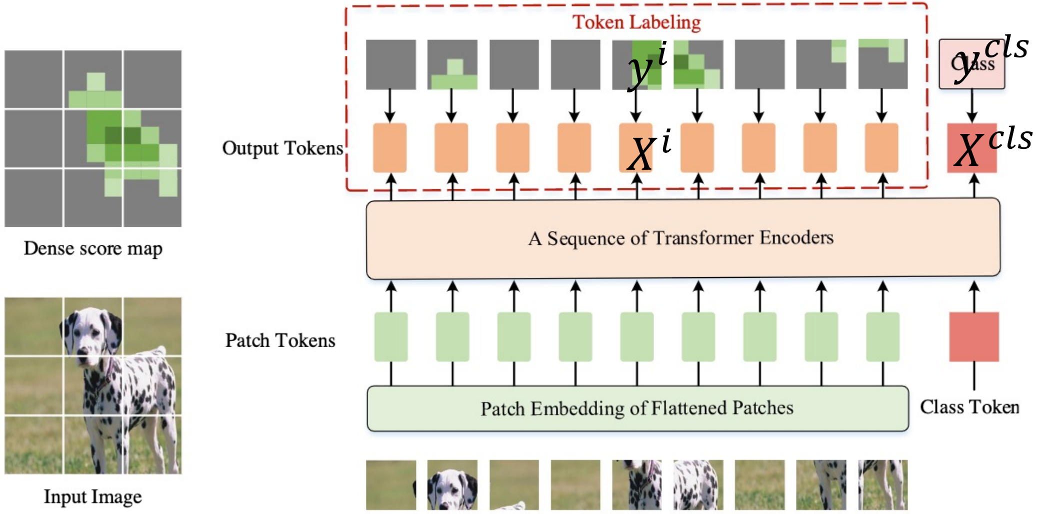
Loss function

Token labeling loss:

$$L_{tl} = \frac{1}{N}\sum_{i=1}^{N} H(X^i, y^i).$$

Total loss:

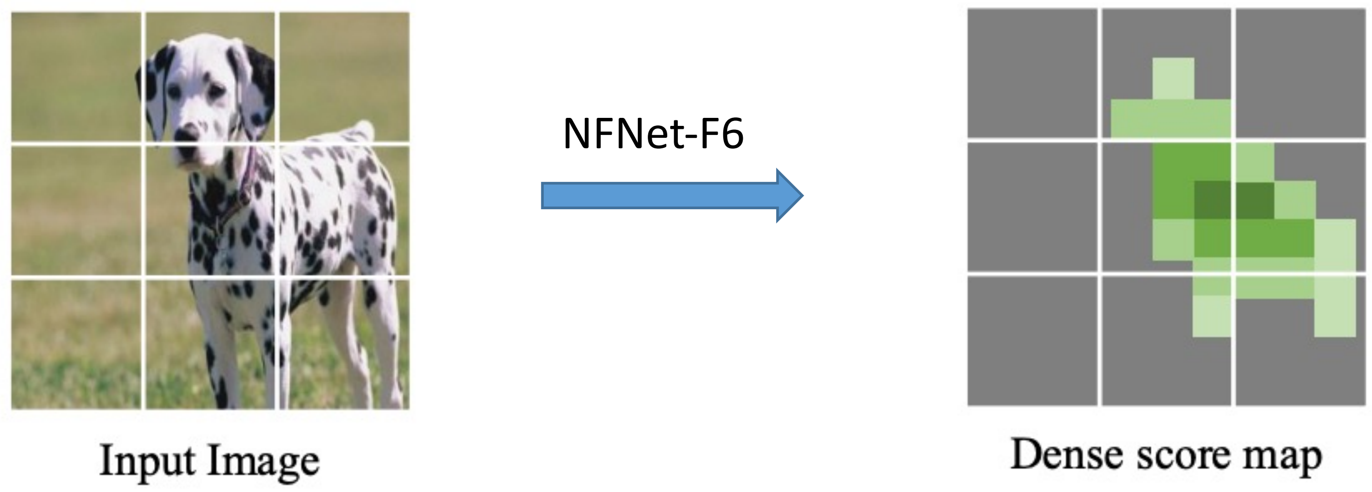
$$\begin{aligned} L_{total} &= H(X^{cls}, y^{cls}) + \beta \cdot L_{tl}, \\ &= H(X^{cls}, y^{cls}) + \beta \cdot \frac{1}{N}\sum_{i=1}^{N} H(X^i, y^i), \end{aligned}$$



Method

Label generation

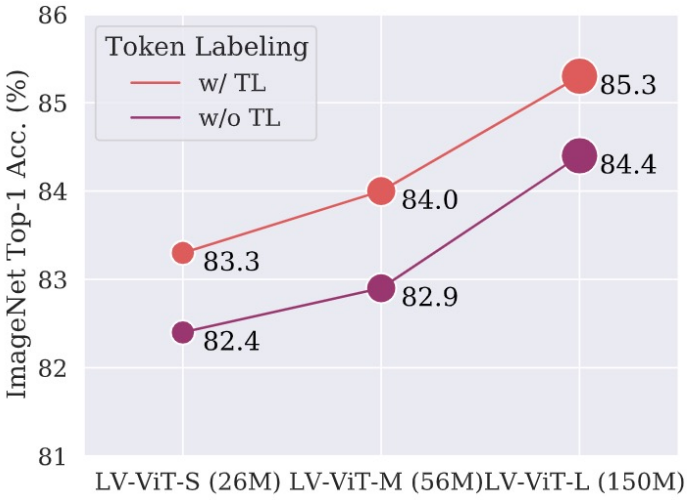
We use state-of-the-art model NFNet-F6 as machine annotator



Experiment

Model scaling with token labeling

- Consistent improvement with respect to different model size.



Name	Depth	Embed dim.	MLP Ratio	#Heads	#Params	Throughput (im/s)	Test size	Top-1 Acc. (%)
LV-ViT-T	12	240	3.0	4	8.5M	2032.6	224	79.1
LV-ViT-S	16	384	3.0	6	26M	1018.2	224	83.3
LV-ViT-M	20	512	3.0	8	56M	668.9	224	84.1
LV-ViT-L	24	768	3.0	12	150M	204.8	288	**85.3**

Experiment

Comparison with SOTA On ImageNet

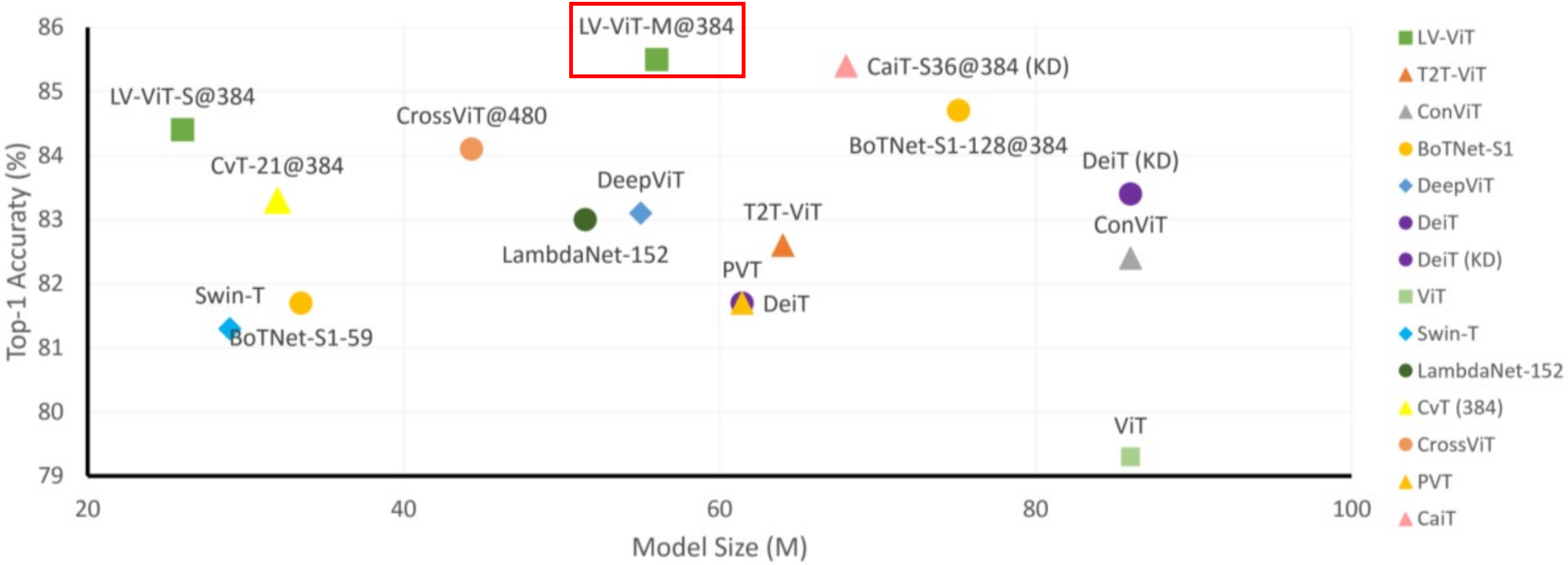
State-of-the-art Performance
with less params and computation

	Network	Params	FLOPs	Train size	Test size	Top-1(%)	Real Top-1 (%)
CNNs	EfficientNet-B5 [34]	30M	9.9B	456	456	83.6	88.3
	EfficientNet-B7 [34]	66M	37.0B	600	600	84.3	–
	Fix-EfficientNet-B8 [34, 38]	87M	89.5B	672	800	85.7	90.0
	NFNet-F3 [3]	255M	114.8B	320	416	85.7	89.4
	NFNet-F4 [3]	316M	215.3B	384	512	85.9	89.4
	NFNet-F5 [3]	377M	289.8B	416	544	86.0	89.2
Transformers	ViT-B/16 [15]	86M	55.4B	224	384	77.9	83.6
	ViT-L/16 [15]	307M	190.7B	224	384	76.5	82.2
	T2T-ViT-14 [46]	22M	5.2B	224	224	81.5	–
	T2T-ViT-14↑384 [46]	22M	17.1B	224	384	83.3	–
	CrossViT [6]	45M	56.6B	224	480	84.1	–
	Swin-B [25]	88M	47.0B	224	384	84.2	–
	TNT-B [16]	66M	14.1B	224	224	82.8	–
	DeepViT-S [59]	27M	6.2B	224	224	82.3	–
	DeepViT-L [59]	55M	12.5B	224	224	83.1	–
	DeiT-S [36]	22M	4.6B	224	224	79.9	85.7
	Distilled DeiT-S [36]	22M	4.6B	224	224	81.2	86.8
	DeiT-B [36]	86M	17.5B	224	224	81.8	86.7
	DeiT-B↑384 [36]	86M	55.4B	224	384	83.1	87.7
	Distilled DeiT-B [36]	87M	17.5B	224	224	83.4	88.3
	BoTNet-S1-128 [31]	79.1M	19.3B	256	256	84.2	-
	BoTNet-S1-128↑384 [31]	79.1M	45.8B	256	384	84.7	-
	CaiT-S36↑384 [37]	68M	48.0B	224	384	85.4	89.8
	CaiT-M36 [37]	271M	53.7B	224	224	85.1	89.3
	CaiT-M36↑448 [37]	271M	247.8B	224	448	86.3	90.2
Ours LV-ViT	LV-ViT-S	26M	6.6B	224	224	83.3	88.1
	LV-ViT-S↑384	26M	22.2B	224	384	84.4	88.9
	LV-ViT-M	56M	16.0B	224	224	84.1	88.4
	LV-ViT-M↑384	56M	42.2B	224	384	85.4	89.5
	LV-ViT-L	150M	59.0B	288	288	85.3	89.3
	LV-ViT-L↑448	150M	157.2B	288	448	85.9	89.7
	LV-ViT-L↑448	150M	157.2B	448	448	86.2	89.9
	LV-ViT-L↑512	151M	214.8B	448	512	86.4	90.1

Experiment

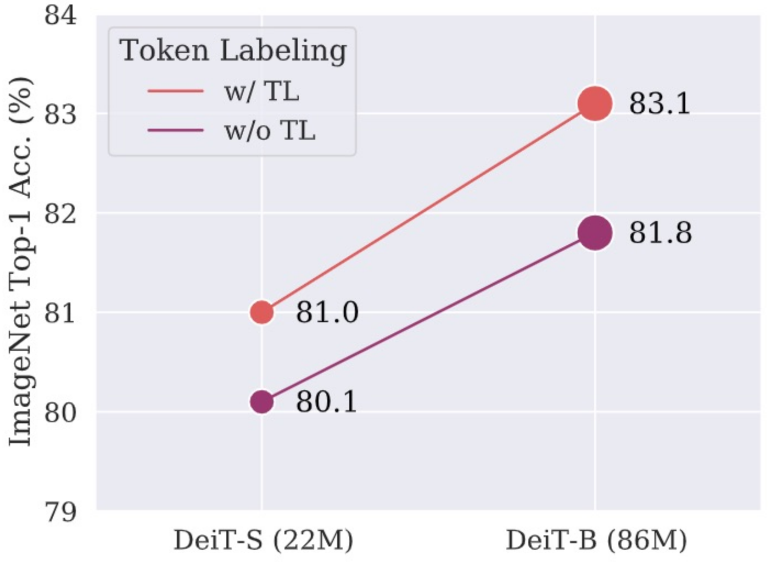
Comparison with SOTA on ImageNet

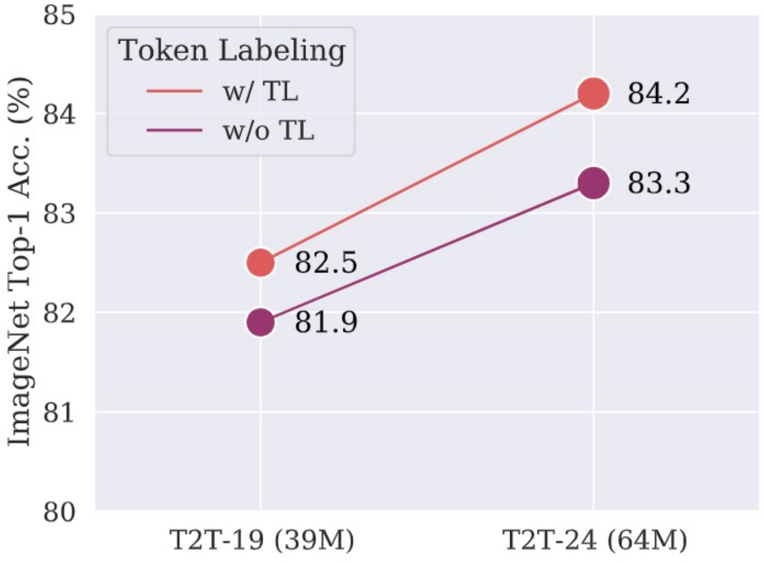
State-of-the-art Performance with less params and computation



Experiment

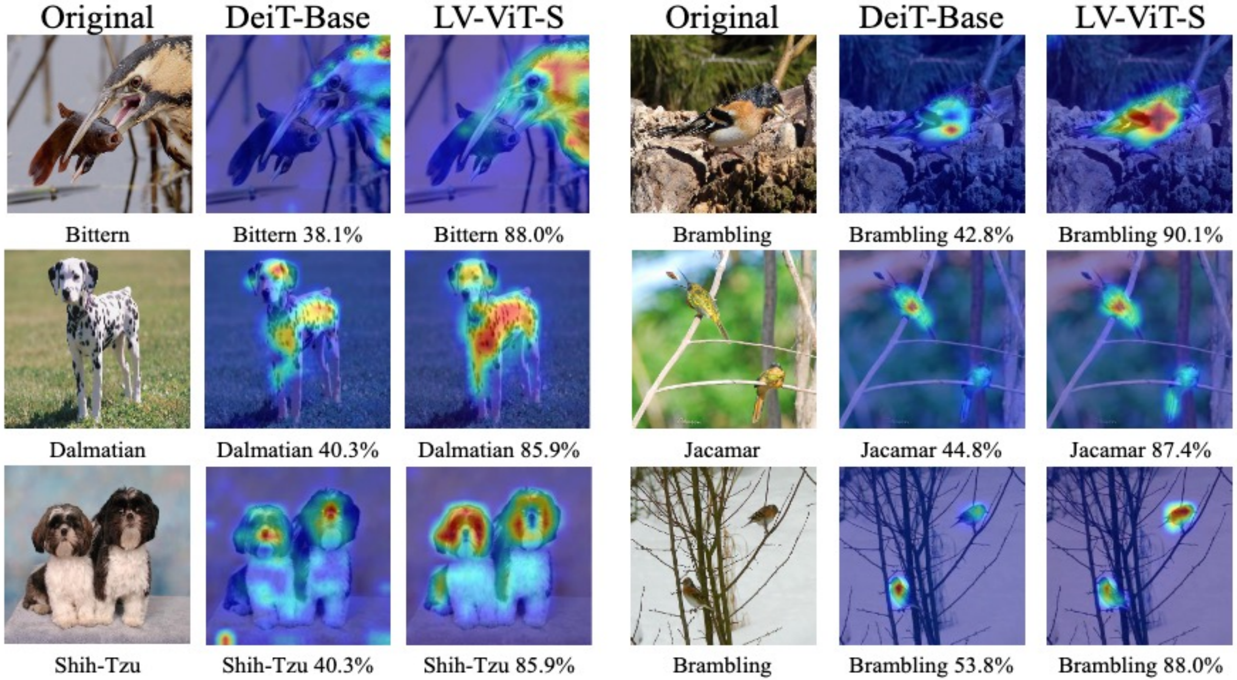
Robustness to different models





Experiment

Visualization



Experiment

Performance on downstream tasks (ADE20k Segmentation)

Table 5: Transfer performance of the proposed LV-ViT in semantic segmentation. We take two classic methods, FCN and UperNet, as segmentation architectures and show both single-scale (SS) and multi-scale (MS) results on the validation set.

Method	Token Labeling	Model Size	mIoU (SS)	P. Acc. (SS)	mIoU (MS)	P. Acc. (MS)
LV-ViT-S + FCN	✗	30M	46.1	81.9	47.3	82.6
LV-ViT-S + FCN	✓	30M	47.2	82.4	48.4	83.0
LV-ViT-S + UperNet	✗	44M	46.5	82.1	47.6	82.7
LV-ViT-S + UperNet	✓	44M	47.9	82.6	48.6	83.1

Experiment

Performance on downstream tasks (ADE20k Segmentation)

Table 6: Comparison with previous work on ADE20K validation set. As far as we know, our LV-ViT-L + UperNet achieves the best result on ADE20K with only ImageNet-1K as training data in pretraining. †Pretrained on ImageNet-22K.

	Backbone	Segmentation Architecture	Model Size	mIoU (MS)	Pixel Acc. (MS)
CNNs	ResNet-269	PSPNet [54]	-	44.9	81.7
	ResNet-101	UperNet [44]	86M	44.9	-
	ResNet-101	Strip Pooling [23]	-	45.6	82.1
	ResNeSt200	DeepLabV3+ [9]	88M	48.4	-
Transformers	DeiT-S	UperNet	52M	44.0	-
	ViT-Large†	SETR [56]	308M	50.3	83.5
	Swin-T [26]	UperNet	60M	46.1	-
	Swin-S [26]	UperNet	81M	49.3	-
	Swin-B [26]	UperNet	121M	49.7	-
	Swin-B† [26]	UperNet	121M	51.6	-
LV-ViT	LV-ViT-S	FCN	30M	48.4	83.0
	LV-ViT-S	UperNet	44M	48.6	83.1
	LV-ViT-M	UperNet	77M	50.6	83.5
	LV-ViT-L	UperNet	209M	**51.8**	**84.1**

Conclusion

- **Propose a novel token labeling objective for training better vision transformer models efficiently.**

- **Token labeling method is robust with respect to different model architecture, different model size and different machine annotator.**

- **Token labeling is also beneficial for downstream tasks like semantic segmentation.**

Thanks