Appendix for "A General and Efficient Training for Transformer via Token Expansion"

Wenxuan Huang[1*] Yunhang Shen[2*] Jiao Xie[3] Baochang Zhang[4] Gaoqi He[1]
Ke Li[2] Xing Sun[2] Shaohui Lin[1,5✉]

[1]School of Computer Science and Technology, East China Normal University, Shanghai, China
[2]Tencent Youtu Lab, China [3]Xiamen University, China [4]Beihang University, China
[5]Key Laboratory of Advanced Theory and Application in Statistics and Data Science - MOE, China

osilly0616@gmail.com, shenyunhang01@gmail.com, jiaoxie1990@126.com, bczhang@buaa.edu.cn
gqhe@cs.ecnu.edu.cn, tristanli.sh@gmail.com, winfred.sun@gmail.com, shaohuilin007@gmail.com

1. Implementation Details

1.1. Details of Applying ToE to DeiT and LV-ViT

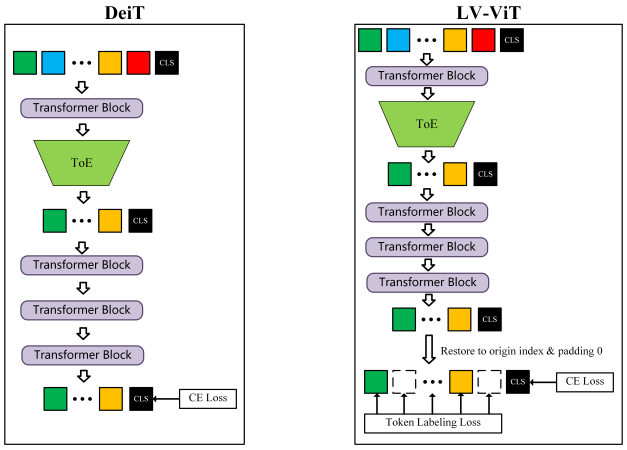

Figure 1. Details of applying ToE to DeiT and LV-ViT during training. Dotted cubes denote the tokens are all-zero vectors.

For DeiT [1] and LV-ViT [2], we apply ToE to the output tokens of the first block. For the training of DeiT, we simply reduce the tokens by ToE. But for LV-ViT requiring the token index, we employ *zero-padding* on the reduced output tokens of last Transformer block and restore the tokens to their original index. The details are presents in Fig. 1. We also use the same way (by adding ToE at the first block output tokens) to combine our ToE with EfficientTrain to achieve the better performance, which is summarized in Tab. 4 of the main paper.

1.2. Details of Breaking the Restriction of Hyper-parameter Consistency

Firstly, for the training of DeiT, we follow the hyper-parameters of original paper [1]. We set the batch size to be $1,024$, learning rate to be $1e-3$ using a cosine scheduler with warmup, and the decay to the minimal learning rate of $1e-5$. We employ the AdamW optimizer, whose weight decay is set to $5e-2$.

In Tab. 2 of the main paper, we relax the restriction of hyper-parameter consistency to achieve better results. We will decribe the following training details for the ToE $^{\text{Hyper}}_{0.4}$ and ToE $^{\text{Hyper}}_{0.5}$. In fact, we only change the minimal learning rate and use the more elaborate training schedule. Specifically, we set minimal learning rate to $2e-4$ and change default training schedule of ToE from [0→100, 101→200, 201→300] for three stages with default average splitting epochs to [0→130, 131→260, 261→300].

1.3. Training Details of Fine-tuning

Following the fine-tuning process in [1], we fine-tune DeiT for $1,000$ epochs with an initial learning rate of $3e-5$, and the batch size of 768 per GPU for four GPUs on CIFAR-10/100 [3] [1] [2]. The input image size of 32×32 are resized to 224×224. Other hyper-parameters and strategies are the same as the pre-training process on ImageNet-1K [4].

1.4. Details of Training time

The detailed training time per training epoch for applying ToE to the models in different training stages are presented in Tab. 1. The training time is averagely measured by 3 times running.

*Equal contribution. ✉Corresponding author.

[1]https://github.com/facebookresearch/dino/issues/144
[2]https://github.com/facebookresearch/deit/issues/45

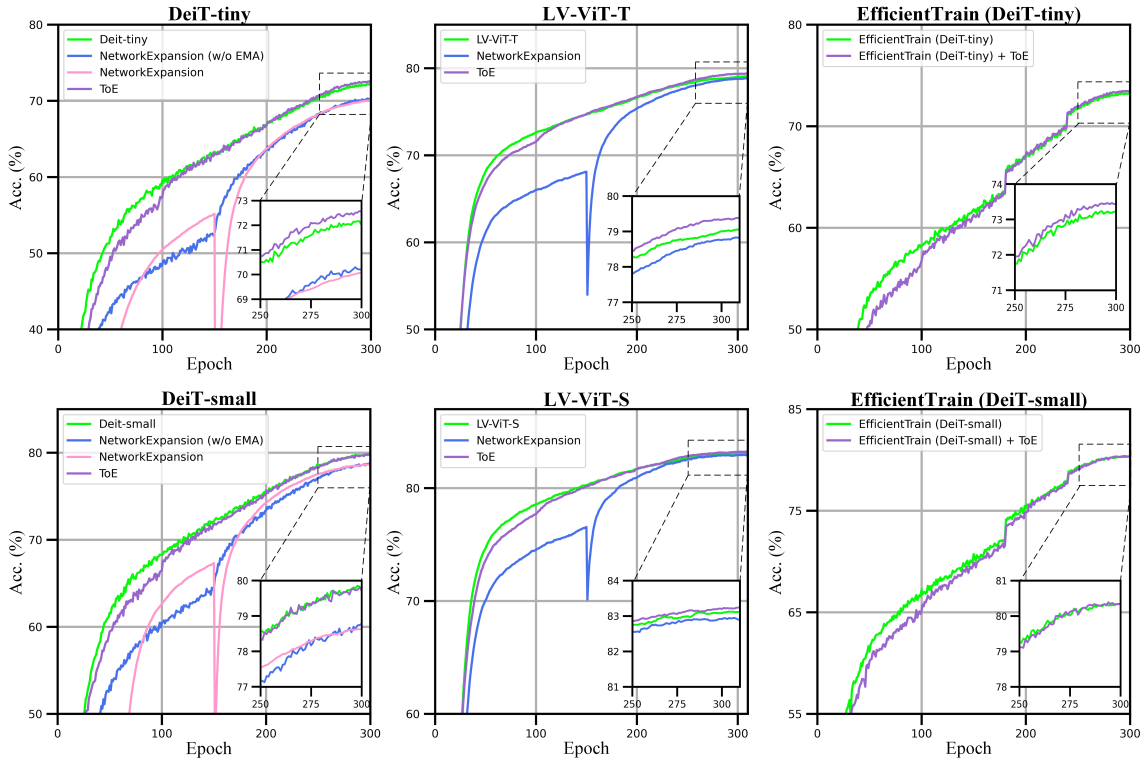

Figure 2. Validation Top-1 accuracy of DeiT-tiny&small and LV-ViT-T&S on ImageNet-1k during training with different methods. DeiT does *not* use the EMA strategy by default, while LV-ViT uses the EMA strategy by default.

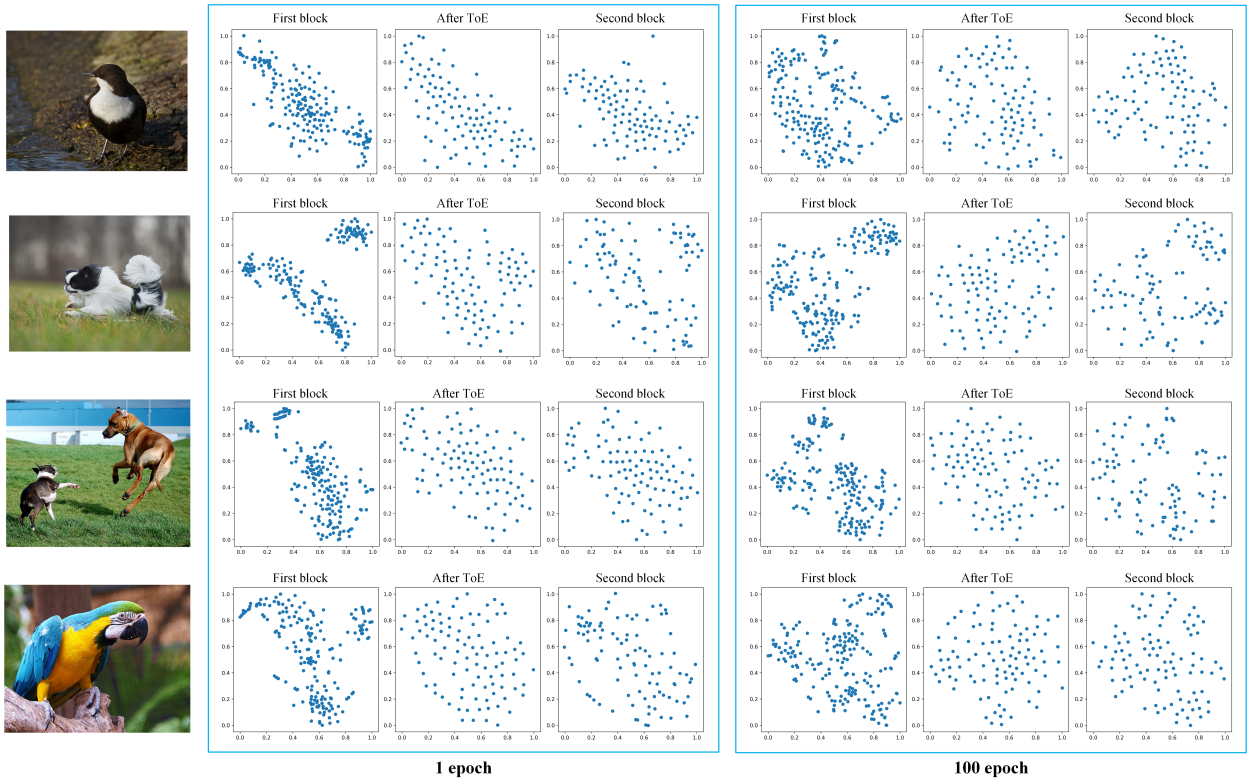

Figure 3. More visualization for the feature distribution of token set. Continuation of Fig. 2 of the main paper.

Table 1. Each epoch training time in the different training stages.

Model	Training time (GPU hours per training epoch)		
	Stage-1	Stage-2	Stage-3
DeiT-tiny + ToE $_{r_1=0.5}$	395s	542s	655s
DeiT-small + ToE $_{r_1=0.5}$	948s	1,244s	1,488s
DeiT-base + ToE $_{r_1=0.5}$	2,028s	2,784s	3,512s
DeiT-base + ToE $_{r_1=0.4}$	1,852s	2,740s	3,512s
LV-ViT-T + ToE $_{r_1=0.4}$	1,180s	1,356s	1,566s
LV-ViT-S + ToE $_{r_1=0.4}$	1,828s	2,356s	2,848s
LV-ViT-M + ToE $_{r_1=0.4}$	2,596s	3,512s	4,424s

Table 2. Results of ToE on YOLOS for COCO object detection. We use eight NVIDIA RTX A6000 GPUs with 150 epochs for YOLOS-S.

Model	Method	AP	Total GPU hours
YOLOS-S	Baseline [5]	36.1	1,193h
	ToE$_{r_1=0.5}$ (Ours)	36.0 (-0.1)	964h (1.24×)

2. Additional Results

2.1. Additional Results for Object Detection

In Tab. 2, ToE applied into YOLOS [5]. ToE reduces 229 hours with 1.24× training speedup for training YOLOS-S on COCO [6] with the only 0.1 AP drop.

2.2. More Validation Curves of Training Process

We present the validation curves of training process for integrating into ToE to DeiT, LV-ViT and EfficientTrain framework [7] in Fig. 2. For the different ViTs and efficient training frameworks, ToE can general accelerate the training process in a lossless manner.

2.3. Visualization of ToE

More visualizations of ToE as a continuation of Fig. 2 of the main paper are presented in Fig. 3. ToE preserves the distribution integrity of intermediate features in the original token set.

References

[1] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICLR*, pages 10347–10357. PMLR, 2021. 1

[2] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *NeurIPS*, 34:18590–18602, 2021. 1

[3] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 1

[5] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *NeurIPS*, 34:26183–26197, 2021. 3

[6] 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. 3

[7] Yulin Wang, Yang Yue, Rui Lu, Tianjiao Liu, Zhao Zhong, Shiji Song, and Gao Huang. Efficienttrain: Exploring generalized curriculum learning for training visual backbones. In *ICCV*, pages 5852–5864, 2023. 3