Supplementary Material for "Locality Guidance for Improving Vision Transformers on Tiny Datasets"

Kehan Li[1⋆], Runyi Yu[1⋆], Zhennan Wang[2], Li Yuan[1,2,✉⋆⋆], Guoli Song[2], and Jie Chen[1,2,✉⋆⋆]

[1] School of Electronic and Computer Engineering, Peking University, China
[2] Peng Cheng Laboratory, Shenzhen, China
yuanli-ece@pku.edu.cn, chenj@pcl.ac.cn

A More Ablation Study

We provide additional ablation study on the dataset scale, the CNN architecture and the fitting degree of the CNN in this section. All models with our method are trained for 100 epochs and the baseline models are trained for 300 epochs. The detailed settings are same as the ablation study in Section 4.3 of the paper.

Dataset Scale In order to investigate the performance of the proposed locality guidance on datasets of different sizes, we train DeiT-Tiny [9] on subsets of CIFAR-100 [6] training set with different proportions. The experimental results are shown in Table I and Fig. I. We find that the performance of the VT is extremely sensitive to the dataset scale, *i.e.*, the performance degenerates rapidly when the dataset getting smaller. Compared with the VT baseline, the improvement of our locality guidance can always be observed regardless of the dataset scale, and is more significant with less training samples. This means that our method mitigates the sensitivity to the dataset scale to some extent.

Table I. Ablation study on dataset scale. The performance of the VT is sensitive to dataset scale and our method mitigates the sensitivity to some extent.

Training Size		CNN Acc.	VT Acc.		
			Baseline	$+L_{guidance}$	Δ
5,000	10%	47.26	23.24	45.01	+21.77
12,500	25%	58.88	34.75	61.11	+26.36
25,000	50%	66.80	50.37	70.64	+20.27
37,500	75%	69.75	58.90	73.70	+14.80
50,000	100%	70.43	65.08	77.29	+12.21

⋆ Equal contribution.
⋆⋆ Corresponding author: Li Yuan, Jie Chen

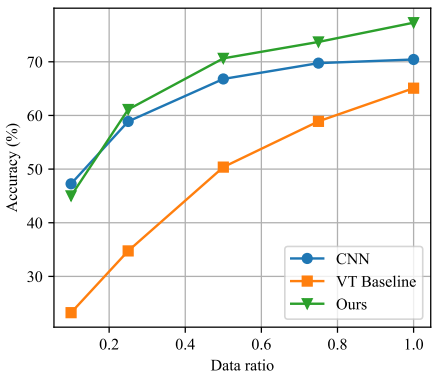

Fig. I. Accuracy as a function of data ratio for different model. The performance of the VT is more sensitive to dataset scale than the CNN. Our method can always improve the VT regardless of the dataset scale, and can mitigates the sensitivity.

CNN Architecture We adopt different commonly used CNNs as guidance models to further verify the generalizability of our method. Specifically, we employ four state-of-the-art CNN architectures, *i.e.*, VGG [8], ResNet [2], Xception [1] and DenseNet [3]. Each of them has distinct characteristics, for example, straightforward structure in VGG, residual connection in ResNet, separable convolution in Xception and dense connection in DenseNet. As for the model size, we adopt lightweight ResNet-56 [2] and DenseNet-40 [3] as the official design. For VGG and Xception, we modify the down-sampling operation and the number of channels in their original implementation to adapt 32×32 resolution of the input images. The experimental results given in Table II prove that our method is applicable to different types of CNNs, and the VT with our method can surpass the corresponding guidance models in this framework.

Table II. Ablation study on different CNNs. * indicates that we modify the original structure. Our method is applicable to different types of CNNs.

CNN Arc.	CNN Acc.	VT Acc.
None	-	65.08
VGG*[8]	75.65	76.44
ResNet-56[2]	70.43	**77.29**
Xception*[1]	74.05	75.35
DenseNet-40[3]	72.80	77.16

Fitting Degree of the CNN. We explore the influence of the fitting degree of the CNN by training the CNN with different training schedule. From the ex-

perimental results in Table III, it is obvious that the proposed method improves the VT a lot even if the CNN only gets a little knowledge about dataset, reflecting that the CNN can provide guidance for the VT robustly. However, it is reasonable that this guidance will produce ambiguity to a certain extent when the CNN can not fully understand the input.

Table III. Ablation study on fitting degree of the CNN. Our method can improve the VT even if the CNN can not understand images well.

CNN Epochs	CNN Acc.	VT Acc.
None	-	65.08
20	48.17	69.85
50	61.20	72.46
100	66.91	74.71
300	72.80	**77.29**

The Type of Guidance Model We conduct experiments for utilizing pre-trained VT (PVT [10]) and lightweight VT trained from scratch (Mobile ViT [5]) as guidance model and show the accuracy as well as the speed for training the target VT (DeiT [9]) in Table IV. Among them, we find that locality guidance through tiny CNN (ResNet-56) is the most efficient solution.

Table IV. Ablation study on the type of guidance model. The guidance through lightweight CNN is more efficient.

Guidance Model	Time/iter.	VT Acc.
None	0.17s	65.08
Pre-train(PVT)	0.37s	74.74
Lite VT(Mobile ViT)	0.29s	75.88
Lite CNN(ResNet-56)	0.19s	**77.29**

Loss function Following recent attempts on feature-based knowledge distillation, as well as the stability and easy implementation, we simply choose L_2 loss in our work. Here we also conduct experiments on different loss variants, as given in Table V, which reflects the robust of the proposed locality guidance framework.

Table V. Ablation study on loss function. Results reflect the robust of the proposed method.

Loss Function	VT Acc.
None	65.08
KL	77.41
L_1	77.27
L_2	77.29

B Implementation Details

The factor β described in Equation (6) plays a role on balancing imitation and self-learning, which can be affected by model structure and dataset. However, through extensive experimentation, we find that it is stable in most cases. Therefore, we set β to 2.5 in most cases, except that we set it to 0.5 for PVTv2 on Flowers and Chaoyang dataset, and for ConViT on Chaoyang dataset. In addition, for better comparing the complexity of models used in experiments, we list the number of parameters as well as the FLOPs of each model in Tabel VI.

Table VI. Summary of model complexity. We list the lightweight guidance CNNs, VTs and the CNN baseline in sequence.

Model	#Params	FLOPs	Model	#Params	FLOPs
CNN Guidance Models			*Transformers*		
DenseNet-40	1.06M	0.28G	DeiT-Tiny	5.7M	1.3G
VGG*	2.82M	0.37G	T2T-ViT-7	4.2M	1.1G
Xception*	0.97M	0.06G	PiT-Tiny	4.9M	0.7G
ResNet-20	0.28M	0.04G	PVT-Tiny	13.2M	1.9G
ResNet-56	0.86M	0.13G	PVTv2-B0	3.4M	0.6G
ResNet-110	1.74M	0.26G	ConViT-Tiny	6.0M	1.0G
CNN Baseline			ResNet-18	11.7M	1.8G
ResNet-50	25.6M	4.1G	ResNet-101	44.7M	7.9G

C Comparing with Dual Network Structure

In this section, we show a comparison with the dual network structure (*i.e.*, Conformer [7]) on tiny dataset, which also combines the VT with a full CNN. The differences of the two methods are shown in Fig. II. The main differences lies in two aspects. a) During training, the bidirectional data flow between the two branches is constructed in Conformer, while no data flow between the two branches exists in our method. b) During inference, the transformer branch and the CNN branch produce the result together in Conformer, while the VT outputs the prediction individually in our method.

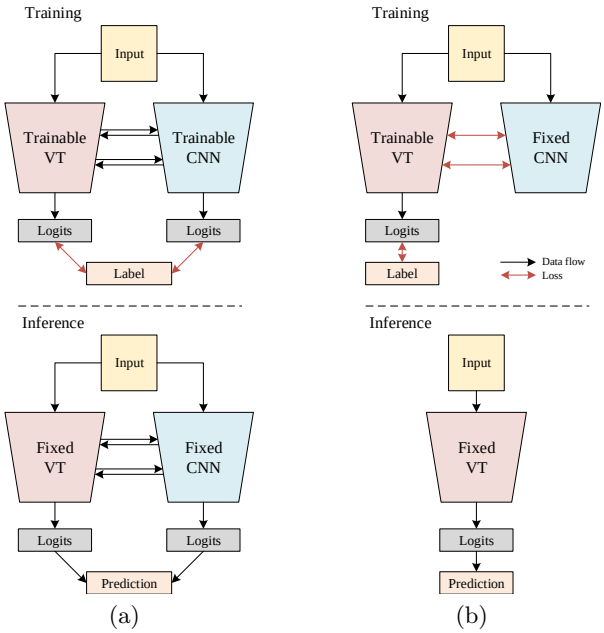

Fig. II. Main differences between the dual network structure [7] and our framework. a) The dual network structure. The CNN branch is needed during both training and inference. A bidirectional data flow between the two branches is constructed. b) Our framework. The CNN branch is only needed for calculating guidance loss and the transformer decides by itself during both training and inference.

We compare Conformer with two VTs with our method, all of them have comparable model size. Trough experimental results in Table VII, we show that although it performs well on medium-size datasets, the dual network structure is not very applicable for tiny datasets. It may be caused by the interaction between the two branches, through which the transformer branch may takes harmful information for the CNN branch.

Table VII. Comparison with dual network structure. The dual network structure can not perform well on tiny datasets.

Model	Epoch	Top-1 Acc.	
		CIFAR-100	Flowers
CvT-13 [4]	100	73.50	54.29
T2T-ViT-14 [11]	100	65.16	31.73
Conformer-Tiny [7]	100	64.62	55.64
Conformer-Tiny [7]	300	71.39	66.45
CvT-13 + $L_{guidance}$	100	76.55	65.13
T2T-ViT-14 + $L_{guidance}$	100	**77.84**	**67.71**

D Offline Processing

Offline processing can further reduce the computational cost in our method. Specifically, the feature maps produced by the CNN are stored so that the forward process of the CNN can be executed only once. To keep the alignment of spatial sizes, we do the same data augmentation (*e.g.*, resize and crop) as the training images on their corresponding feature maps when training. However, there are still misalignments caused by color transformation in data augmentation, making it impossible to obtain same feature maps as online processing. The experimental results are shown in Table VIII. We keep all settings the same with the main results, except that the features of the CNN are stored instead of forwarding the CNN once in each training iteration. It can be concluded that the misalignment caused by color transformation from data augmentation leads to a decrease on performance. However, the improvement is still apparent with the offline processing. The offline processing provides a trade-off between speeding up the training process and higher accuracy, which can be used when the computational resources are extremely limited.

Table VIII. Results on offline processing. The improvement is still apparent with the offline processing.

Model	$L_{guidance}$	Offline	Top-1 Acc.	Model	$L_{guidance}$	Offline	Top-1 Acc.
			65.08				73.58
DeiT-Tiny	✓		78.15	PiT-Tiny	✓		78.48
	✓	✓	76.09		✓	✓	77.48

References

1. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
3. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
4. Liu, Y., Sangineto, E., Bi, W., Sebe, N., Lepri, B., Nadai, M.: Efficient training of visual transformers with small datasets. Advances in Neural Information Processing Systems **34** (2021)
5. Mehta, S., Rastegari, M.: Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In: International Conference on Learning Representations (2021)
6. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing. pp. 722–729. IEEE (2008)
7. Peng, Z., Huang, W., Gu, S., Xie, L., Wang, Y., Jiao, J., Ye, Q.: Conformer: Local features coupling global representations for visual recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 367–376 (2021)
8. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
9. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021)
10. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 568–578 (2021)
11. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.H., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 558–567 (2021)