Appendix for DAPT

Xin Zhou*[1], Dingkang Liang*[1], Wei Xu[1], Xingkui Zhu[1], Yihan Xu[1], Zhikang Zou[2], Xiang Bai†[1]
[1]Huazhong University of Science and Technology, {xzhou03,dkliang,xbai}@hust.edu.cn
[2] Baidu Inc., China

A. Additional Experiments

A.1. Training Detail

We adopt downstream fine-tuning configuration following pioneer work Point-MAE [8]. More details are provided in Tab. 1. Taking fine-tuning on ScanObjectNN [9] as an example, the overall training includes 300 epochs, with a cosine learning rate [6] of 5e-4, and a 10-epoch warm-up period. AdamW optimizer [7] is used.

A.2. Scale in Dynamic Adapter

We also conduct additional experiments in Tab. 2 to further prove the effectiveness of our dynamic scale. Using the scale 0.1 suggested in AdaptFormer [1] or 4.0 suggested by He et al. [2] cannot achieve satisfying results. We also experiment with a learned scale which does not give better results. We claim that our dynamic scale offers greater adaptability for the intricate geometry of point clouds and eliminates the need to adjust scale as a hyper-parameter.

A.3. Number of Internal Prompts

In previous works [3–5], external prompts are utilized by concatenating a certain number of adjustable tokens into the transformer's input space. Therefore, this subsection investigates the impact of prompt numbers in DAPT on classification tasks.

We adopt average pooling (default), max pooling, and top-K operation to obtain internal prompts in different length. Fig. 1 displays the results on the challenging variants (i.e., PE_T50_RS) of ScanObjectNN. The results suggest that simply increasing the prompt number may even hurt the performance.

A.4. Token Selections for Head Inputs

Based on Fig. 3 in the manuscript, we conduct four other experiments on token selections for head input, as shown in Fig. 2. Interestingly, with the pooling of our prompts, we can achieve better results than the pooling of patch tokens. The only use of the pooling of Prompts exceeds only the pooling of patch tokens by 1.18%. We argue that our internal prompts better capture instance-specific features and act as global features, providing positive assistance to downstream task heads.

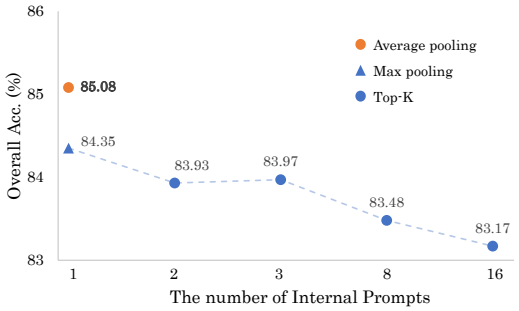


Figure 1. The effect of the number of Internal Prompts.

A.5. Inference Time

In this subsection we evaluate the inference time on classification task in Tab. 3. Our DAPT achieves 311.28 frame/s with only a negligible impact on inference speed, which is highly competitive compared to IDPT (281.18 frame/s).

B. Qualitative Analysis

B.1. t-SNE Visualizations

In Fig. 3, the t-SNE [10] feature manifold visualization displays the models following full fine-tuning, linear probing, IDPT, and our DAPT on the ScanObjectNN PB_T50_RS dataset. From Fig. 3(a), it is evident that the feature distribution extracted by the ShapeNet-pretrained model on ScanObjectNN appears less discriminative. We contend that this is mainly due to the significant domain gap between the synthetic ShapeNet and real-world ScanObjectNN datasets, demonstrating the necessity for fine-tuning on downstream tasks. With full fine-tuning in Fig. 3(b), the feature distribution becomes more discriminative as all parameters are tuned. Fig. 3(c-d) confirms that our DAPT helps the pre-trained model generate more distinguishable representations with fewer learnable parameters than IDPT.

* Equal contribution. † Corresponding author.

Table 1. Training details for downstream fine-tuning.

Configuration	Classification			Segmentation
	ScanObjectNN	ModelNet	ModelNet Few-shot	ShapeNetPart
Optimizer	AdamW	AdamW	AdamW	AdamW
Learning rate	5e-4	5e-4	5e-4	2e-4
Weight decay	5e-2	5e-2	5e-2	5e-2
Learning rate scheduler	cosine	cosine	cosine	cosine
Training epochs	300	300	150	300
Warmup epochs	10	10	10	10
Batch size	32	32	32	16
r in Dynamic Adapter	64	72	72	128
Number of points	2048	1024	1024	2048
Number of point patches	128	64	64	128
Point patch size	32	32	32	32

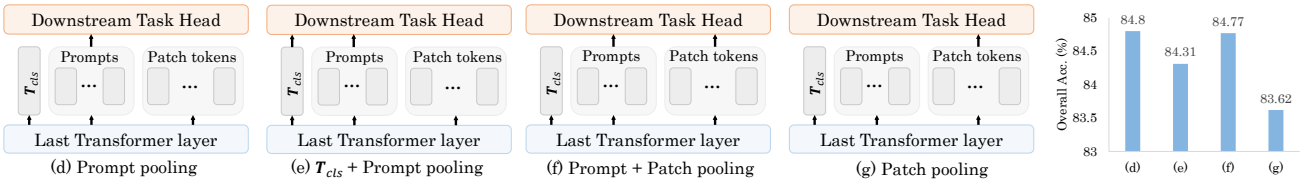
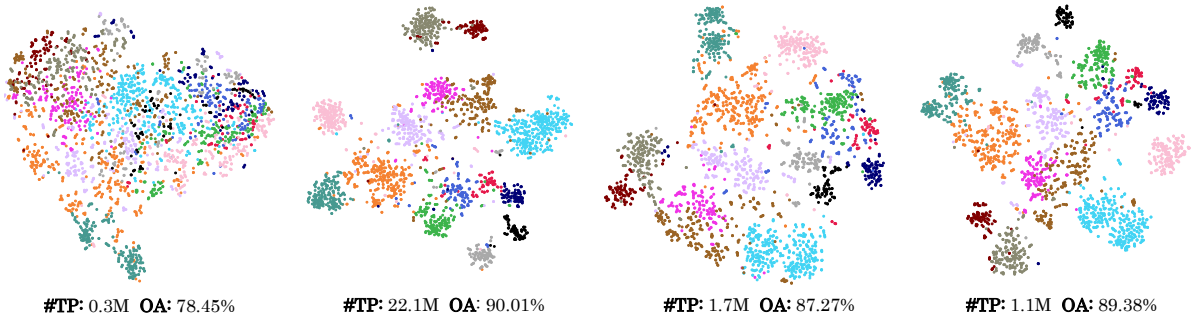


Figure 2. The effect of different inputs for downstream task head.



(a) Linear probing (b) Full fine-tuning (c) IDPT (d) DAPT (ours)

Figure 3. The t-SNE visualizations from the test sets of ScanObjectNN (PB_T50_RS) using a pre-trained ReCon with different tuning strategies. We extract the final classification features from the top linear layer for t-SNE visualizations.

Table 2. The effect of different scale settings in Dynamic Adapter.

Type	Scale	#TP (M)	PB_T50_RS
Train: fixed Inference: fixed	0.01	1.08	83.28
	0.1	1.08	83.55
	4.0	1.08	**84.35**
Train: learnable Inference: fixed	Initialized with 0.01	1.08	84.07
	Initialized with 0.1	1.08	**84.77**
	Initialized with 1.0	1.08	84.32
	Initialized with 4.0	1.08	84.56
	Initialized with 5.0	1.08	84.49
	Initialized with 10.0	1.08	84.70
Train: learnable Inference: dynamic **(Ours)**	Dynamic scale	1.09	**85.08**

Table 3. Comparison of different fine-tuning strategy on ScanObjectNN classification. Throughput is measured with a batch size of 32 on a single RTX 4090 GPU.

Method	#TP(M)	FLOPs(G)	Throughput (frame / s)	PB_T50_RS
Point-MAE [8]	22.1	4.8	323.66	85.18
IDPT [12]	1.7	7.2	281.18	84.94
DAPT (Ours)	1.1	5.0	311.28	85.08

B.2. Part Segmentation Visualizations

In Fig. 4, we visualize our DAPT part segmentation results on Point-BERT [11] baseline. We select five representative categories each, each with three viewpoints. Our DAPT requires a small number of tunable parameters while achieving satisfying segment results.

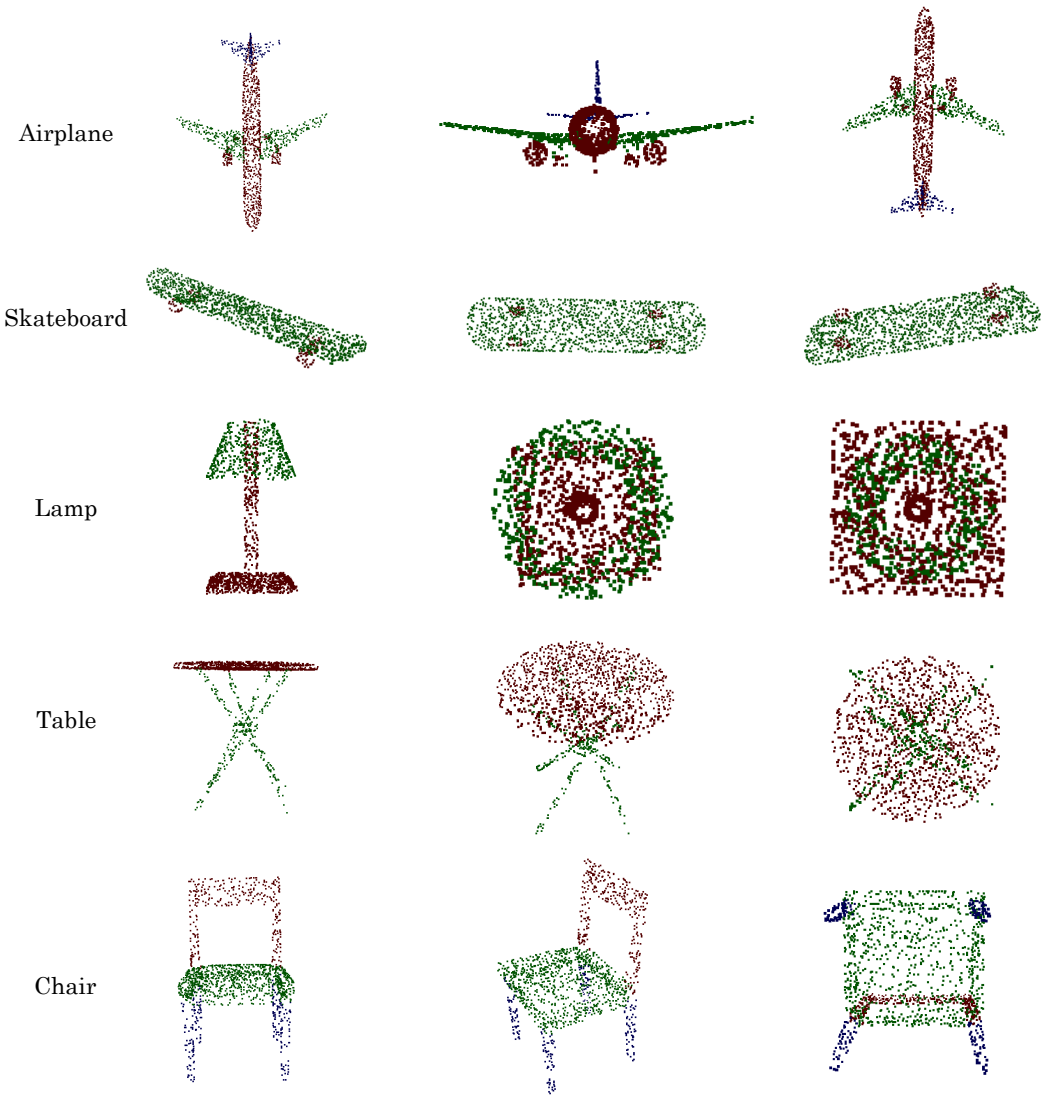


Figure 4. Qualitative results for part segmentation. We show our prediction projection images from three different viewpoints.

References

[1] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *Proc. of Advances in Neural Information Processing Systems*, 2022. 1

[2] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *Proc. of Intl. Conf. on Learning Representations*, 2021. 1

[3] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proc. of European Conference on Computer Vision*, 2022. 1

[4] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proc. of Conf. on Empirical Methods in Natural Language Processing*, 2021.

[5] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Annual Meeting of the Association for Computational Linguistics*, 2021. 1

[6] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *Proc. of Intl. Conf. on Learning Representations*, 2017. 1

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. of Intl. Conf. on Learning Representations*, 2019. 1

[8] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proc. of European Conference on Computer Vision*, 2022. 1, 2

[9] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification

model on real-world data. In *Porc. of IEEE Intl. Conf. on Computer Vision*, 2019. 1

[10] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008. 1

[11] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2022. 2

[12] Yaohua Zha, Jinpeng Wang, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Instance-aware dynamic prompt tuning for pre-trained point cloud models. In *Porc. of IEEE Intl. Conf. on Computer Vision*, 2023. 2