CVPR
SEATTLE, WA JUNE 17-21, 2024

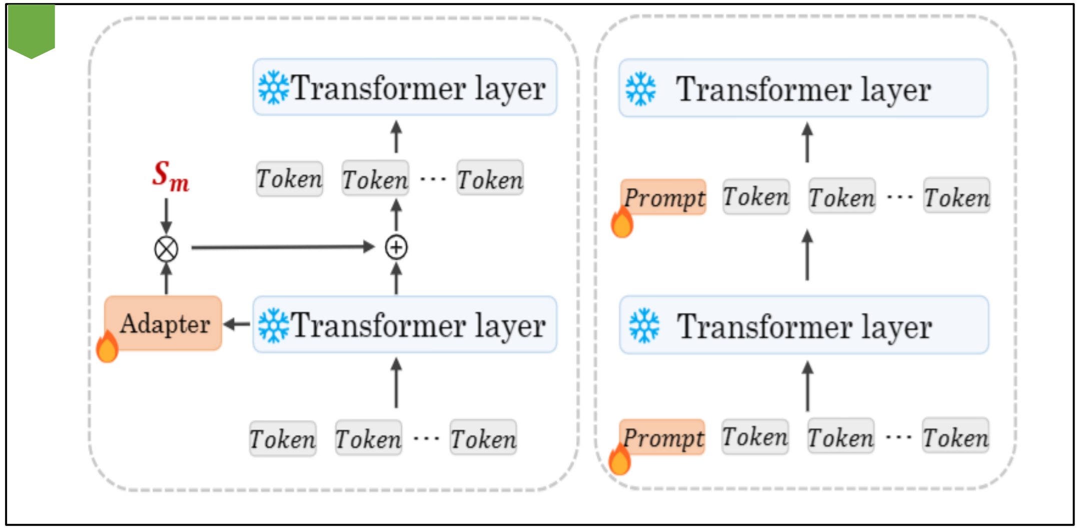
Dynamic Adapter Meets Prompt Tuning: Parameter-Efficient Transfer Learning for Point Cloud Analysis

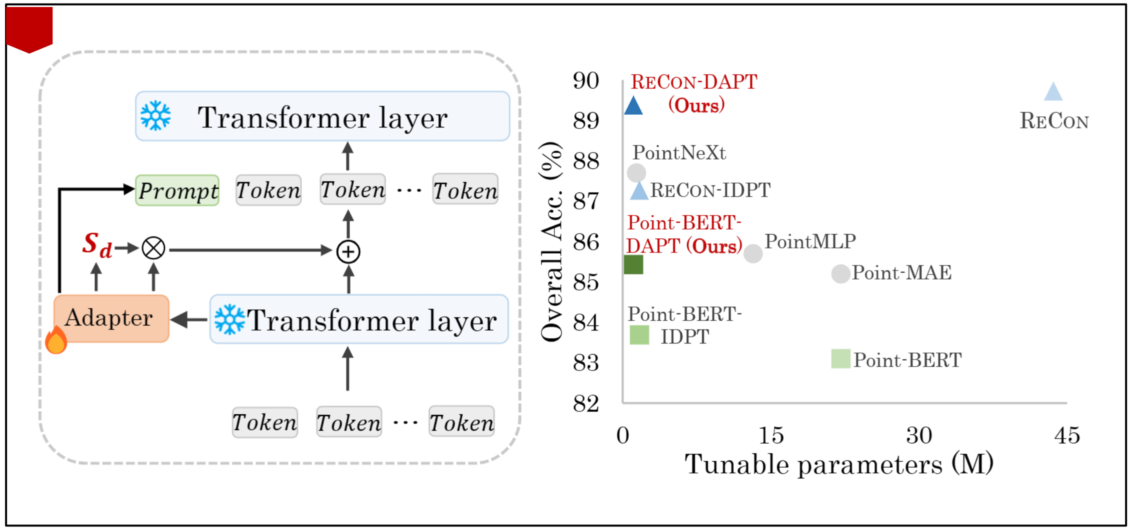
Xin Zhou [1*], Dingkang Liang [1*], Wei Xu [1], Xingkui Zhu [1], Yihan Xu [1], Zhikang Zou[2], Xiang Bai[1]

[1] Huazhong University of Science & Technology, [2]Baidu Inc.

Hightlights

- Existing Adapter tuning utilizes additional residual blocks with manual scale. Prompt tuning usually introduces extra random initialized prompts into the input space.
- Our DAPT leverages a simple Dynamic Adapter that generates a dynamic scale for each token and seamlessly integrates it with Prompt tuning.





: Tunable parameters : Frozen parameters S_m : Manual scale S_d : Dynamic scale

Background

- Pre-training on 3D datasets is gaining significant interest. Several works utilize self-supervised methods and achieve excellent performance.
- Two mainstream methods:
  - Mask modeling
  - Contrastive learning

Background

- Finetuning all parameters for pre-trained model may lead to
  - Catastrophic forgetting and break the rich prior
  - Fine-tuning for each point cloud analysis dataset requires a separate weights copy, and the storage space overhead may becomes a burden as the number of datasets increases
  - The computational cost requirements escalate dramatically, especially for larger batch sizes, leading to a substantial increase in GPU memory usage, limiting its accessibility for researchers with weak hardware.

Background

- Parameter-Efficient Transfer Learning fixes most of the parameters and adjusting only a selected few.
- Two mainstream methods:
  - **Adapter tuning**: often require manual scale setting as a crucial hyper-parameter, while the value remains constant during inference.
  - **Prompt tuning**: usually adds external random initialized prompts as extra inputs.

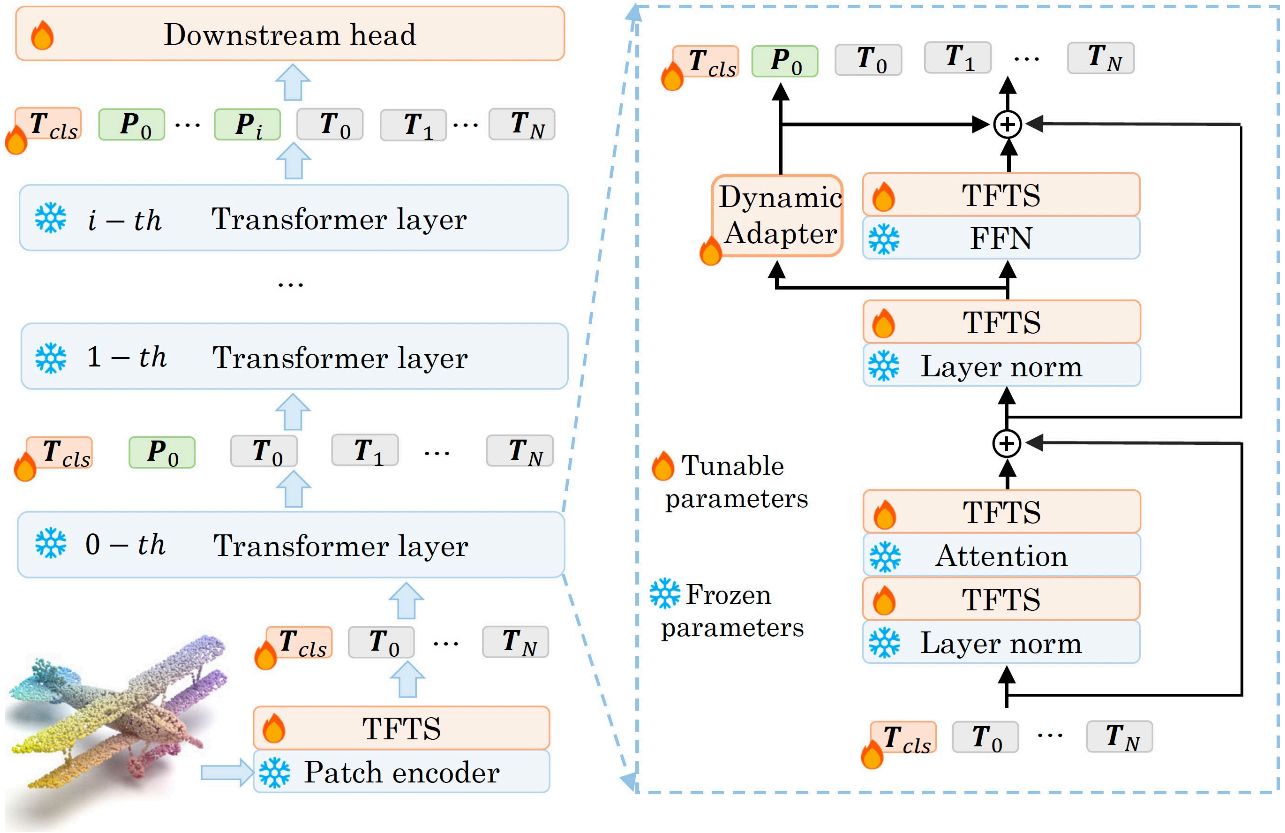
Background

- Existing tuning strategies achieve promising results in NLP and 2D vision, they lack targeted design.
- Can not achieve satisfying results on hard point cloud datasets.

Tuning Strategy	#TP(M)	OBJ_BG	OBJ_ONLY	PB_T50_RS
Point-MAE	22.1	90.02	88.29	85.18
Linear probing	0.3	87.26(-2.76)	84.85(-3.44)	75.99(-9.19)
+ Adapter	0.9	89.50(-0.52)	88.64(+0.35)	83.93(-1.25)
+ VPT	0.4	87.26(-2.76)	87.09(-1.20)	81.09(-4.09)

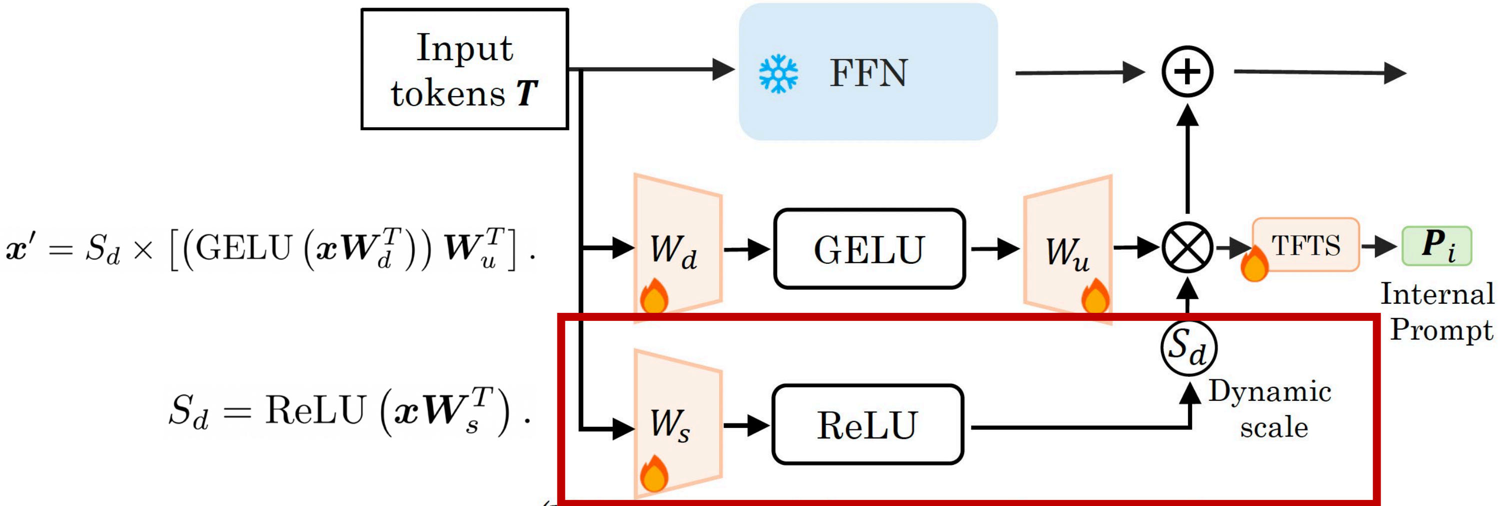
Overall

- During the fine-tuning, we fix the entire backbone, only fine-tuning the newly added parameters.



Method

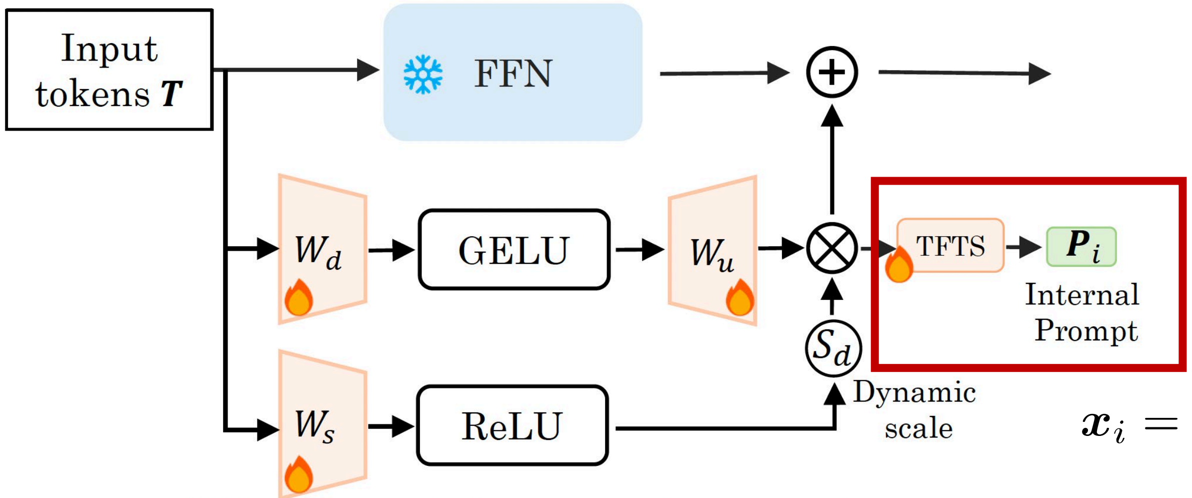
- The Dynamic Adapter adopt a parallel MLP to generate dynamic scale S_d based on variant point cloud features.
- The ReLU to select the positive scale and set the rest as zero.



(b) Dynamic Adapter with internal prompt

Method

- We leverage the Dynamic Adapter to generate the prompt derived from the original model’s internal output using the proposed Dynamic Adapter, called Internal Prompt tuning.

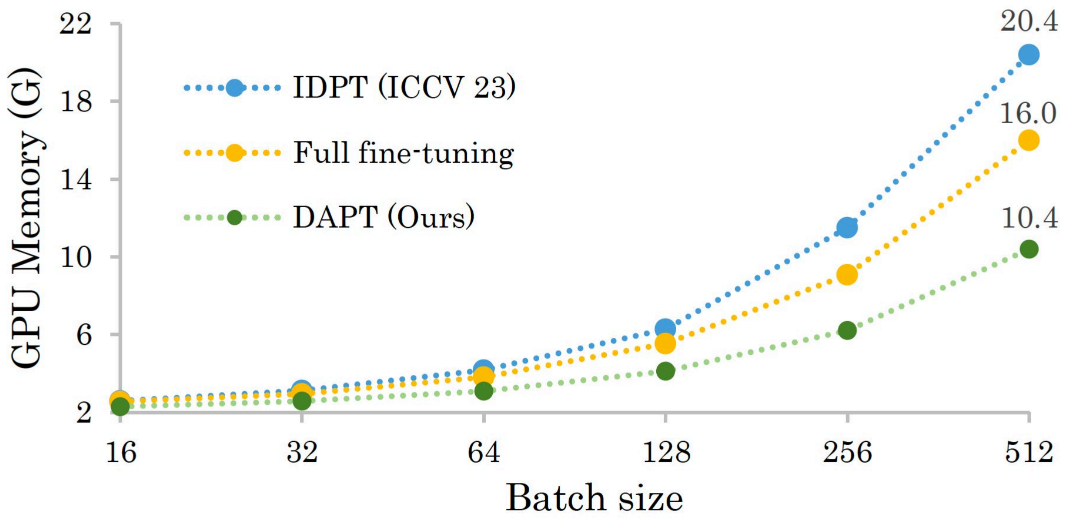


$$x_i = L_i\left([T_{cls}; P_0, \ldots, P_{i-1}; T]\right).$$

(b) Dynamic Adapter with internal prompt

Method

- DAPT can consistently reduce training memory as the batch size increases. With 512 batch size, our DAPT significantly reduces GPU memory usage by **35% and 49%** compared to full fine-tuning and IDPT.



(c) The comparison of training GPU memory

Experiments

- For the ScanObjectNN and ModelNet40, we achieve the best performance on most sub-sets.

Method	Reference	Tunable params. (M)	FLOPs (G)	ScanObjectNN			ModelNet40	
				OBJ_BG	OBJ_ONLY	PB_T50_RS	Points Num.	OA (%)
Self-Supervised Representation Learning (Full fine-tuning)								
OcCo [47]	ICCV 21	22.1	4.8	84.85	85.54	78.79	1k	- / 92.1
Point-BERT [55]	CVPR 22	22.1	4.8	87.43	88.12	83.07	1k	- / 93.2
MaskPoint [28]	ECCV 22	22.1	-	89.70	89.30	84.60	1k	- / 93.8
Point-MAE [37]	ECCV 22	22.1	4.8	90.02	88.29	85.18	1k	- / 93.8
Point-M2AE [60]	NeurIPS 22	15.3	3.6	91.22	88.81	86.43	1k	- / 94.0
ACT [8]	ICLR 23	22.1	4.8	93.29	91.91	88.21	1k	- / 93.7
ReCon [41]	ICML 23	43.6	5.3	94.15	93.12	89.73	1k	- / 93.9
Self-Supervised Representation Learning (Efficient fine-tuning)								
Point-BERT [55] (baseline)	CVPR 22	22.1 (100%)	4.8	87.43	88.12	83.07	1k	92.7 / 93.2
+ IDPT [57]	ICCV 23	1.7 (7.69%)	7.2	88.12(+0.69)	88.30(+0.18)	83.69(+0.62)	1k	92.6(-0.1) / 93.4(+0.2)
+ DAPT (ours)	-	**1.1** (**4.97%**)	5.0	91.05(+3.62)	89.67(+1.55)	85.43(+2.36)	1k	93.1(+0.4) / 93.6(+0.4)
Point-MAE [37] (baseline)	ECCV 22	22.1 (100%)	4.8	90.02	88.29	85.18	1k	93.2 / 93.8
+ IDPT [57]	ICCV 23	1.7 (7.69%)	7.2	91.22(+1.20)	90.02(+1.73)	84.94(-0.24)	1k	93.3(+0.1) / 94.4(+0.6)
+ DAPT (ours)	-	**1.1** (**4.97%**)	5.0	90.88(+0.86)	90.19(+1.90)	85.08(-0.10)	1k	93.5(+0.3) / 94.0(+0.2)
ReCon [41] (baseline[2])	ICML 23	22.1 (100%)	4.8	94.32	92.77	90.01	1k	92.5 / 93.0
+ IDPT* [57]	ICCV 23	1.7 (7.69%)	7.2	93.29(-1.03)	91.57(-1.20)	87.27(-2.74)	1k	93.4(+0.9) / 93.5(+0.5)
+ DAPT (ours)	-	**1.1** (**4.97%**)	5.0	94.32(0.00)	92.43(-0.34)	89.38(-0.63)	1k	93.5(+1.0) / 94.1(+1.1)

Experiments

- We evaluate few-shot and part segmentation performance of DAPT on the ModelNet40 and ShapeNetPart datasets, respectively.

Methods	Reference	5-way		10-way	
		10-shot	20-shot	10-shot	20-shot
with Self-Supervised Representation Learning (Full fine-tuning)					
OcCo [47]	ICCV 21	94.0±3.6	95.9±2.3	89.4±5.1	92.4±4.6
Point-BERT [55]	CVPR 22	94.6±3.1	96.3±2.7	91.0±5.4	92.7±5.1
MaskPoint [28]	ECCV 22	95.0±3.7	97.2±1.7	91.4±4.0	93.4±3.5
Point-MAE [37]	ECCV 22	96.3±2.5	97.8±1.8	92.6±4.1	95.0±3.0
Point-M2AE [60]	NeurIPS 22	96.8±1.8	98.3±1.4	92.3±4.5	95.0±3.0
ACT [8]	ICLR 23	96.8±2.3	98.0±1.4	93.3±4.0	95.6±2.8
ReCon [41]	ICML 23	97.3±1.9	98.9±3.9	93.3±3.9	95.8±3.0
with Self-Supervised Representation Learning (Efficient fine-tuning)					
Point-BERT [55] (baseline)	CVPR 22	94.6±3.1	96.3±2.7	91.0±5.4	92.7±5.1
+ IDPT [57]	ICCV 23	**96.0±1.7**	97.2±2.6	91.9±4.4	93.6±3.5
+ DAPT **(ours)**	-	95.8±2.1	**97.3±1.3**	**92.2±4.3**	**94.2±3.4**
Point-MAE [37] (baseline)	ECCV 22	96.3±2.5	97.8±1.8	92.6±4.1	95.0±3.0
+ IDPT [57]	ICCV 23	**97.3**±2.1	97.9±1.1	92.8±4.1	95.4±**2.9**
+ DAPT **(ours)**	-	96.8±**1.8**	**98.0±1.0**	**93.0±3.5**	**95.5**±3.2

Few-shot learning on ModelNet40

Methods	Reference	#TP (M)	Cls. mIoU (%)	Inst. mIoU (%)
Self-Supervised Representation Learning (Full fine-tuning)				
OcCo [47]	ICCV 21	27.09	83.42	85.1
MaskPoint [28]	ECCV 22	-	84.60	86.0
Point-BERT [55]	CVPR 22	27.09	84.11	85.6
Point-MAE [37]	ECCV 22	27.06	84.19	86.1
ACT [8]	ICLR 23	27.06	84.66	86.1
Self-Supervised Representation Learning (Efficient fine-tuning)				
Point-BERT [55] (baseline)	CVPR 22	27.06	84.11	85.6
+ IDPT* [57]	ICCV 23	5.69	83.50	85.3
+ DAPT **(ours)**	-	**5.65**	83.83	85.5
Point-MAE [37] (baseline)	ECCV 22	27.06	84.19	86.1
+ IDPT [57]	ICCV 23	5.69	83.79	85.7
+ DAPT **(ours)**	-	**5.65**	84.01	85.7
ReCon [41] (baseline[2])	ICML 23	27.06	84.52	86.1
+ IDPT* [57]	ICCV 23	5.69	83.66	85.7
+ DAPT **(ours)**	-	**5.65**	83.87	85.7

Part segmentation on ShapeNetPart

Experiments

- Comparisons of parameter efficient transfer learning methods from NLP and 2D Vision on the hardest variant of ScanObjectNN.

Method	Reference	#TP (M)	PB_T50_RS
Point-MAE [28]	ECCV 22	22.1	**85.18**
Linear probing	-	0.3	75.99
+ Adapter [16]	ICML 19	0.9	83.93
+ Perfix tuning [25]	ACL 21	0.7	77.72
+ BitFit [56]	ACL 21	0.3	82.62
+ LoRA [17]	ICLR 22	0.9	81.74
+ VPT-Deep [18]	ECCV 22	0.4	81.09
+ AdaptFormer [4]	NeurIPS 22	0.9	83.45
+ SSF [26]	NeurIPS 22	0.4	82.58
+ IDPT [57]	ICCV 23	1.7	84.94
+ DAPT (**ours**)	-	1.1	**85.08**

Comparisons on ScanObjectNN PB-T50-RS

CVPR
SEATTLE, WA JUNE 17-21, 2024

THANK YOU !

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 & Technology, [2]Baidu Inc.