Adapting Pre-trained 3D Models for Point Cloud Video Understanding via Cross-frame Spatio-temporal Perception

Supplementary Material

A. Implementation Details

A.1. Pre-trained Models

The 3D point cloud pre-training model [1–3, 6–10] is our basic model. We apply our CSA strategy to three static point cloud pre-trained models: Point-BERT [6], Point-MAE [3], and PointGPT-S [1]. These models share the same backbone structure: the Transformer depth is 12, each block contains a 6-head Self-Attention [5] layer and an MLP, and the feature dimension is 384. The FeedForward Network (FFN) following each self-attention layer is implemented as an MLP, with dimensions of (384, 1536, 384).

A.2. Cross-frame Spatio-temporal Adaptation

Our Cross-frame Spatio-temporal Adaptation (CSA) strategy tunes only the parameters of the Point Tube Adapter, the Position Encoder, the Geometric Constraint Temporal Adapter (GCTA), and the Head during downstream task training. As mentioned in Sec 4, each anchor frame is selected every two frames, and the number of anchor points within it is 64. We set the temporal length of the point tube to $l = 3$, the neighborhood space radius to $r_s = 0.3$, and randomly sample 32 points from the neighborhood space. The detailed architecture of each module within our CSA is shown in Table 1.

Module	Block	Input Size	Output Size
Point Tube Adapter	Grouping	(24, 2048, 3)	(12, 64, 32, 3)
	Conv1d	(12, 64, 32, 3)	(12, 64, 32, 384)
	Maxpooling	(12, 64, 32, 384)	(12, 64, 384)
Position Encoder	Linear	(12, 64, 3)	(12, 64, 384)
GCTA	Grouping	(12, 64, 384)	(12, 64, k, 384)
	Concatenation	(12, 64, k, 384)	(12, 64, k, 768)
	Conv2d	(12, 64, k, 384)	(12, 64, k, 48)
	Maxpooling	(12, 64, k, 48)	(12, 64, 48)
	Conv1d	(12, 64, 48)	(12, 64, 384)
Head	Maxpooling	(12, 64, 384)	384
	Linear	384	N_{cls}

Table 1. Detailed architecture of our models. k is the number of neighbors for Cross-frame KNN. N_{cls} is the number of classes.

B. Influence of Geometric Constraint Temporal Adapter Implementation

We construct the cross-frame KNN graph based on the spatial distance between all point tubes and aggregate features according to it in GCTA. The Table 2 shows the performance comparison with the dynamic graph construction method, which constructs a unique KNN graph in the feature space in each block. The results show that the neighbors in Euclidean space are more helpful for the model to capture long-term dynamics than those in semantic space.

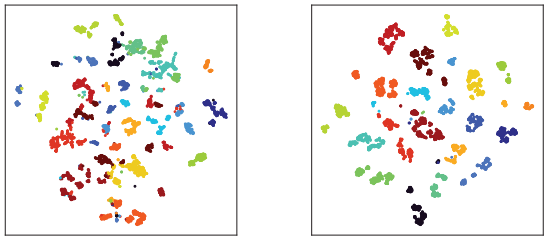
Graph	Accuracy(%)
Dynamic Feature KNN Graph	94.08
Cross-frame KNN Graph	95.12

Table 2. Comparison of graph construction methods on the MSR-Action3D.

C. Effect of Geometric Constraint Temporal Adapter

C.1. Feature Visualization

We qualitatively evaluate the effect of GCTA on the model's ability to encode point cloud sequences by visualizing the learned features. We compare the model without GCTA to the model with GCTA. We use t-SNE [4] to project the learned features into a two-dimensional space. As shown in Figure 1, the features are more compact and discriminative with GCTA than without it.



(a) Model without GCTA. (b) Model with GCTA.

Figure 1. Feature visualization using t-SNE. Features belonging to the same category are represented by the same color in the visualization.

C.2. Architecture Design

We further evaluate the impact of different adapter architecture. Specifically, we take the model with GCTA removed as the baseline, and evaluate the performance of three adapter types: the MLP adapter, the self-attention adapter, and our proposed GCTA. As shown in Table 3,

while both the MLP and the self-attention adapters improve upon the baseline, GCTA achieves the highest accuracy by effectively combining spatial and temporal information. The results indicate that GCTA significantly enhances model accuracy.

Adapter	Accuracy(%)
-	87.11
MLP	89.55
Self-Attention	89.90
GCTA	95.12

Table 3. Comparison of different architecture of adapter in Transformer blocks.

C.3. Effect of GCTA in Different Blocks

We add GCTA to only part of the Transformer blocks to evaluate the performance of the model. We conduct the following experiments: (1) Only in odd-numbered blocks, (2) Only in even-numbered blocks, (3) Only in the first 6 blocks, (4) Only in the last 6 blocks. For each strategy, we conduct 5 experiments and take the average as the result. The results are shown in Table 4.

Blocks	Accuracy(%)
Odd	94.01
Even	93.87
First 6	92.89
Last 6	94.56
All	94.77

Table 4. Performance of GCTA in different Transformer blocks.

Adding GCTA only in odd-numbered or even-numbered blocks leads to performance degradation. Applying GCTA only to the first six blocks results in the lowest accuracy of 92.89%, while to the last six blocks yield an accuracy of 94.56%. This finding indicates that early blocks may not be as effective at modeling complex spatiotemporal relationships as later blocks. The highest accuracy of 94.77% is achieved when GCTA is applied to all blocks. This indicates that utilizing GCTA throughout the entire model maximizes its effectiveness, allowing for comprehensive integration of spatial and temporal features.

D. Visualization

We visualize a few attention distribution examples in Figure 2. The results show that more attention is distributed to the moving parts in some frames, which shows that our strategy can accurately identify motion information and locate frames with more intense motion.

Figure 2. Visualization of self-attention distribution for 12 anchor frames in each video. Warmer color indicates more attention.

References

[1] Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-

training from point clouds. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[2] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? *arXiv preprint arXiv:2212.08320*, 2022.

[3] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022. 1

[4] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008. 1

[5] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 1

[6] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322, 2022. 1

[7] Yaohua Zha, Rongsheng Li, Tao Dai, Jianyu Xiong, Xin Wang, and Shu-Tao Xia. Sfr: Semantic-aware feature rendering of point cloud. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[8] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14161–14170, 2023.

[9] Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6962–6970, 2024.

[10] Yaohua Zha, Naiqi Li, Yanzi Wang, Tao Dai, Hang Guo, Bin Chen, Zhi Wang, Zhihao Ouyang, and Shu-Tao Xia. Lcm: locally constrained compact point cloud model for masked point modeling. *arXiv preprint arXiv:2405.17149*, 2024. 1