Complete-to-Partial 4D Distillation for Self-Supervised Point Cloud Sequence Representation Learning – Supplementary Materials

This document provides a list of supplemental materials to support the main paper.

- **Overview of STRL and VideoMAE** - We introduce more about STRL and VideoMAE in Section A for better comparison and understanding of our method.
- **Fine-tuning on Synthia 4D** - We show the result on Synthia 4D semantic segmentation in Section B. The performance improvement shows that our method is also effective in outdoor scenarios.
- **Data-efficient 4D Representation Learning** - We evaluate the data efficiency of our method in Section C. The result shows the strong data efficiency of our method.
- **Additional Ablation Studies**
  - In Section D, we explore the influence of different time windows in the distillation framework.
  - In Section E, we examine whether to use a single MLP or multiple MLPs to predict motion.
- **Visualization on HOI4D Action Segmentation** - We give a visualization on HOI4D action segmentation in Section F to intuitively demonstrate the outstanding performance of our method.
- **Implementation Details** - We provide additional implementation details of our method in Section G.

A. Overview of STRL and VideoMAE

In our experiments, we compare our method with existing work for self-supervised representation learning on video and point cloud: STRL [4] and VideoMAE [2]. In this section, we will introduce more about these two works.

STRL [4] learns the point cloud representation through the interactions of two networks: the online network and the target network. By using spatial augmentation and synthetic sequence generation, the network is capable of capturing spatio-temporal representation in a self-supervised manner.

VideoMAE [2] is a simple extension of Masked Autoencoders [3] for video representation learning. Through randomly masking spacetime patches in videos and pixel-level reconstruction, the method can outperform supervised pre-training by large margins.

Table 1. Evaluation for semantic segmentation on Synthia 4D dataset [5]

Method	Frames	Bldn	Road	Sdwlk	Fence	Vegittn	Pole	Car	T.Sign	Pedstrn	Bicycl	Lane	T.Light	mIoU
P4Transformer [1]	1	96.76	98.23	92.11	95.23	**98.62**	97.77	95.46	80.75	85.48	0.00	74.28	74.22	82.41
P4Transformer [1]	3	96.73	98.35	**94.03**	95.23	98.28	98.01	95.60	81.54	85.18	0.00	75.95	79.07	83.16
P4Transformer+C2P [1]	3	**97.02**	**98.54**	93.21	**95.52**	97.80	**98.12**	**95.87**	**84.81**	**88.19**	0.00	**77.62**	**82.60**	**84.11**

B. Fine-tuning on Synthia 4D

Partial-view Sequence Generation. We use perspective projection to get partial-view point cloud for indoor scenario experiments in the main paper. However, due to the different property of outdoor scenarios, we do spherical projection to get the range map to do the occlusion sampling. A detailed introduction of spherical projection can be found in [6]. We sample the camera trajectory by continuous horizontal movement around the original camera position.

Setup. Synthia 4D [5] is a synthetic dataset generated from Synthia dataset [5]. It consists of six videos of driving scenarios where both objects and cameras are moving. Following previous work [1], we use the same training/validation/test split, with 19,888/815/1,886 frames, respectively. The pre-training clip length is 12 with 4096 points in each frame. Fine-tuning is done on clip length 3 with 16384 points in each frame to keep consistent with the previous methods. The distillation network is the same as we introduced in the basic setting of the experiment part in the main paper. We use P4Transformer [1] as the backbone. The mean Intersection over Union (mIoU) is used as the evaluation metric.

Result. As reported in Table 1, our method has a considerable improvement. This indicates that our method is also general to outdoor scenarios. Specifically, we observe that our method has a large improvement on several small objects which further reflects that the network has a better understanding of geometry and motion.

C. Data-efficient 4D Representation Learning

We evaluate our method under limited training data on the HOI4D Action Segmentation task. For all data-efficient experiments, our limited data are randomly sampled from the full dataset of HOI4D Action Segmentation dataset. For pre-training and fine-tuning, we use the same setup as we describe in Section 4 in the main paper. As shown in Table 2 and Figure 1. Our pre-training method shows consistently outstanding performance in the case of lack of data, compared with VideoMAE [2]. This indicates that our method can still formulate a good 4D representation under limited data.

Table 2. Data-efficient learning on HOI4D Action Segmentation.

%Data	Scratch	VideoMAE [2]	C2P(ours)
10%	44.0	45.7$_{(+1.7)}$	53.4$_{(+9.4)}$
20%	53.9	54.8$_{(+0.9)}$	69.9$_{(+16.0)}$
40%	69.9	69.8$_{(-0.1)}$	75.4$_{(+5.5)}$
80%	76.7	77.3$_{(+0.6)}$	79.0$_{(+2.3)}$

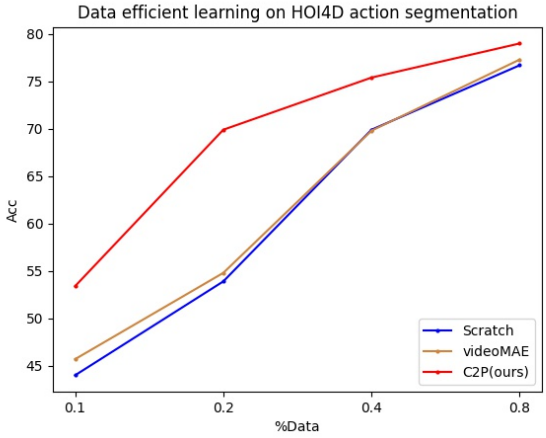

Figure 1. Data-efficient learning on HOI4D Action Segmentation. Our C2P method turns out to be more competitive as the number of data decreases, while training from scratch and the VideoMAE method shows significant performance degradation.

D. Different Time Windows

We believe a sequence-to-sequence distillation framework encourages the network not only to perceive motion information through geometric consistency but also to obtain complete geometric understanding on the basis of temporal cues. So the time window needs to be carefully designed because it determines the source of knowledge for sequence-to-sequence distillation. We set the time windows to 1 (3D distillation), 3 (our setting), and 5 for experiments on HOI4D action segmentation.

Results are shown in Table 3. When the size of the time window is set to 1, 3D distillation is clearly not enough to learn a good 4D representation due to the sacrifice of the benefits from temporal information. When the time window size is set to 3, the network is able to learn temporal information between multiple frames, resulting in a better ability to leverage temporal information and better performance. As we continue to increase the time window to 5, we observe some performance degradation which may be resulted by the complexity and difficulty of optimization. Specifically, per-frame prediction needs more mlp-heads, which makes the network not easy to integrate spatial-temporal information among different frames. To verify this, we conduct another experiment with time window size set to 5. We use two predictors to predict the i-2 frame and i+2 frame respectively. As shown in Table 3, we get a result similar to the best performance. The above experiments show that our method likewise gains improvement with a larger time window size, but the problem of optimization should be taken into account in deciding which frames are selected to do the distillation.

Table 3. Comparison of different time window size.

Window Size	Accuracy
1	79.84
3	**81.10**
5	79.76
5 ($\pm$2 frames)	80.75

E. A Single MLP or Multiple MLPs

In our 4D-to-4D distillation framework, the network learns to aggregate temporal information by predicting frame-wise features. Considering the difference between frames, we use frame-wise predictors to gain better prediction results. We believe frame-wise predictor plays a positive role in attaining better spatial-temporal information. Experiments have been conducted to verify the impact of different predictor choices. For a time window size set to 3, we use a single mlp-head predictor to replace the frame-wise predictor and get a degradation of 0.92 than the original result, indicating frame-wise predictors are more capable of capturing spatial-temporal information. With the time window size set to 5, we use two predictors to predict the previous two frames and the latter two frames respectively. We get a degradation of 0.66 compared with the results shown in Table 3. Experiments show that single predictor for multiple frames makes it difficult to aggregate cross-time information and our frame-wise predictor is quite important for a better integration of spatial-temporal information.

Table 4. Comparison of mlp-head choice.

Window Size	Accuracy
1 mlp for 3 frames	80.18
2 mlps for 5 frames	80.09

F. Visualization on HOI4D Action Segmentation

We give a visualization of HOI4D Action Segmentation in this section to intuitively demonstrate the outstanding performance of our method. As shown in Figure 2, each row from the top to the bottom: ground truth, train from scratch, pretrain with VideoMAE, pretrain with STRL and pretrain with C2P. Results show that our method outperforms the others in the continuity and accuracy of the segmentation, which results in a better performance on all the metrics.

G. Implementation Details

In this section, we introduce the details of the implementation of HOI4D Action Segmentation experiments.

Pre-training setup We use SGD optimizer to train the network. The learning rate is set to be 0.01 and we use a learning rate warmup for 10 epochs, where the learning rate increases linearly for the first 10 epochs. The dimension of the features used to calculate the contrastive loss is set to 2048. The temperature τ used when calculating contrastive loss is set to 0.07. The time window size is set to 3. As for P4DConv, we set the spatial stride to 32, the radius of the ball query is set to 0.9 and the number of samples is set to 32 by default. With batch-size set to 8, our pre-training method can be implemented on two NVIDIA GeForce 3090 GPUs.

Fine-tuning setup We use SGD optimizer to fine-tune the network. The learning rate is set to be 0.05 and we use a learning rate warmup for 5 epochs likewise. To achieve better performance, we apply learning rate decay on 20 and 35 epochs with a decay ratio set to 0.5. For model parameters, the hyper-parameters of the model are exactly the same as pre-training. With batch-size set to 8, our fine-tuning method can be implemented on two NVIDIA GeForce 3090 GPUs.

References

[1] Hehe Fan, Yi Yang, and Mohan S. Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 14204–14213, 2021. 1

[2] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners, 2022. 1, 2

[3] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1

[4] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. *arXiv preprint arXiv:2109.00179*, 2021. 1

[5] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 1

[6] Xin Zheng and Jianke Zhu. Efficient lidar odometry for autonomous driving. *IEEE Robotics and Automation Letters*, 6(4):8458–8465, 2021. 1

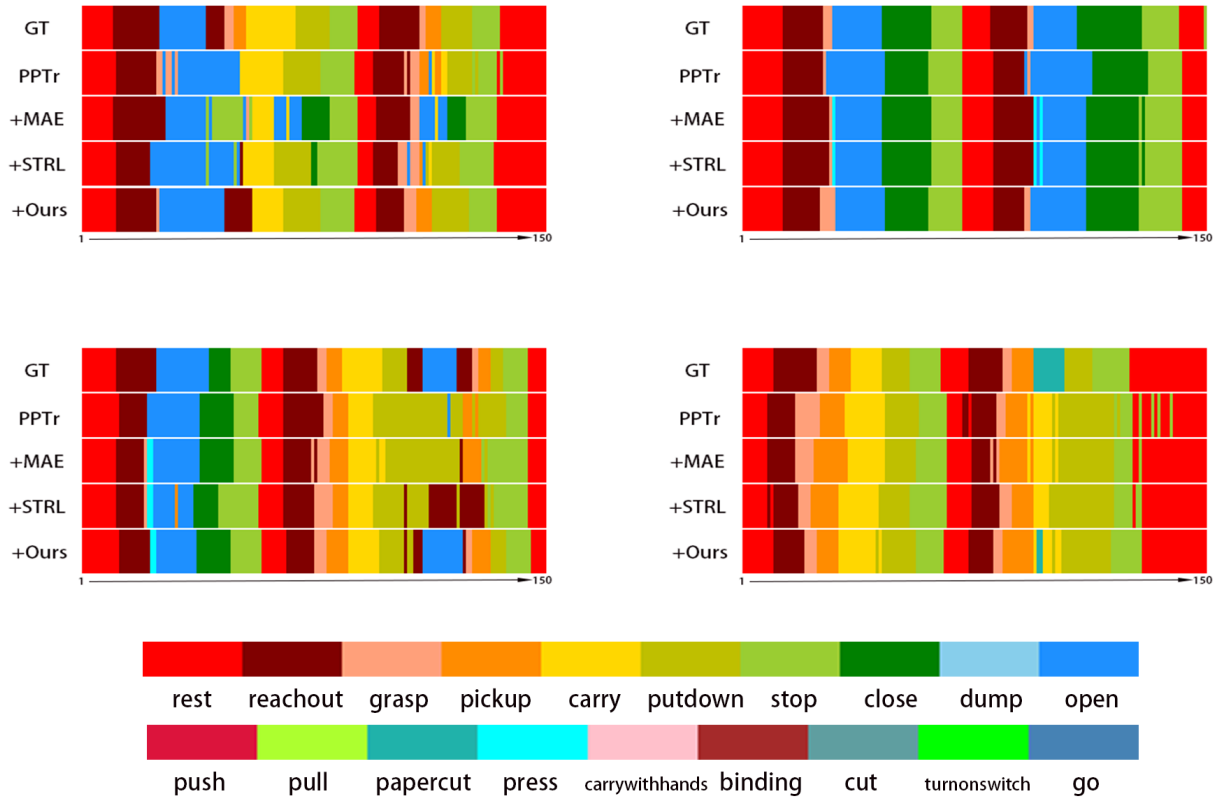

Figure 2. Visualization of results of action segmentation on HOI4D.