

Point Contrastive Prediction with Semantic Clustering for Self-Supervised Learning on Point Cloud Videos

Xiaoxiao Sheng^{1*}, Zhiqiang Shen^{1*}, Gang Xiao^{1†}, Longguang Wang², Yulan Guo³, Hehe Fan⁴
¹Shanghai Jiao Tong University ²Aviation University of Air Force
³Sun Yat-sen University ⁴Zhejiang University
 {shengxiaoxiao, shenzhiqiang, xiaogang}@sjtu.edu.cn

Abstract

We propose a unified point cloud video self-supervised learning framework for object-centric and scene-centric data. Previous methods commonly conduct representation learning at the clip or frame level and cannot well capture fine-grained semantics. Instead of contrasting the representations of clips or frames, in this paper, we propose a unified self-supervised framework by conducting contrastive learning at the point level. Moreover, we introduce a new pre-text task by achieving semantic alignment of superpoints, which further facilitates the representations to capture semantic cues at multiple scales. In addition, due to the high redundancy in the temporal dimension of dynamic point clouds, directly conducting contrastive learning at the point level usually leads to massive undesired negatives and insufficient modeling of positive representations. To remedy this, we propose a selection strategy to retain proper negatives and make use of high-similarity samples from other instances as positive supplements. Extensive experiments show that our method outperforms supervised counterparts on a wide range of downstream tasks and demonstrates the superior transferability of the learned representations.

1. Introduction

Point cloud videos captured by 3D sensors describe the dynamics of objects and their surrounding environments, and have been applied in a wide range of fields to perceive the environment, including robotics and autonomous driving. Early point cloud understanding approaches mainly focus on the geometric modeling of static point clouds [7, 18, 46]. Recently, more attention has been paid to point cloud videos [12, 14, 40, 41]. However, since obtaining point-wise annotation for point cloud videos is labor-intensive [1, 43], conducting self-supervised learning on dy-

*These authors contributed equally.

†Corresponding author.

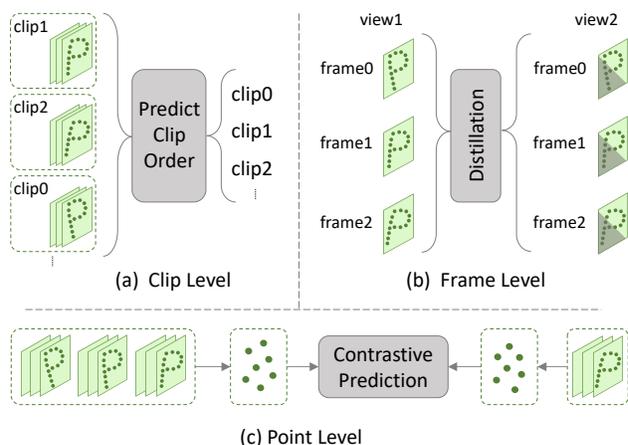


Figure 1. Existing works utilize clip-level (a) or frame-level (b) instances for point cloud video pre-training, while we focus on point-level (c) pre-training.

amic point clouds has drawn increasing interest. Despite the great success of recent self-supervised learning on images and static point clouds [4, 16, 17, 44, 45], two questions still remain for point cloud videos:

(i) *How to build a unified self-supervised framework?* Multi-granularity perception of point cloud videos is demanded in different tasks, such as classification, semantic segmentation, and part segmentation. Existing works conduct self-supervised learning by predicting the orders of randomly shuffled clips or distilling spatiotemporal knowledge based on complete-to-partial sequences [9, 35] (Fig. 1(a-b)). Representations learned by these paradigms focus more on frame-level semantics and cannot well capture fine-grained semantic cues. Therefore, building a unified self-supervised framework that can learn representation rich in multi-granularity semantics is highly demanded.

(ii) *How to achieve effective learning between local samples?* To build a unified self-supervised framework for multiple point cloud video tasks, it is necessary to learn

fine-grained semantics at the level of local samples. Traditional contrastive learning constructs two views from the same instance as positives and pushes away all other instances [2–4, 6, 16, 17, 37, 45]. Since dynamic point clouds are highly redundant in the temporal dimension, directly applying previous approaches to local samples may introduce massive undesired negatives. Therefore, how to conduct effective learning on local samples to obtain fine-grained semantics still remains under-investigated.

In this paper, we propose a unified point-based contrastive prediction framework, termed as PointCPSC, for self-supervised learning on point cloud videos. We conduct representation learning at the point level by contrasting local superpoints of predictions and targets (Fig. 1(c)). Regarding challenge (i), we propose a new pretext task to align the predicted prototypes and target prototypes, as well as soft category assignments between predictions and targets. For challenge (ii), we propose a negative sample selection strategy and employ higher similar samples from other instances as positive supplements. Compared with the frame-based self-supervised framework, our method achieves more effective representation modeling at a finer granularity, and can be applied to multiple point cloud video understanding tasks. The main contributions of our paper are summarized as follows:

- We propose a unified self-supervised contrastive learning framework for point cloud videos. Our framework facilitates the representations to capture both fine-grained dynamics and hierarchical semantics for multiple downstream tasks.
- We introduce a new pretext task by achieving the semantic alignment between predictions and targets. This facilitates our self-supervised framework to capture semantic information on multiple scales.
- We design a feature similarity based sample selection strategy to retain proper negatives and positive neighbors for effective representation learning.
- Our framework produces remarkable performance on a wide range of downstream tasks. We also perform extensive ablation studies and visualized analysis to demonstrate the effectiveness of our method.

2. Related Work

In this section, we first present related works of contrastive learning on images and static point clouds. Then, we introduce the advanced works about dynamic point cloud modeling.

2.1. Contrastive Learning

Self-supervised learning has achieved great success in images, notably represented by instance-based discrimina-

tive methods [2–6, 16, 17, 38, 45]. This classic paradigm augments two views of an instance as a positive pair, while treating all views of other instances as negatives. Many techniques have been introduced to enhance the representation learning capability [3, 6, 16, 34, 36, 37]. He *et al.* [17] introduced dynamic queues to store massive negatives. Chen *et al.* [6] indicated that massive negatives and momentum updated encoder are not essential for contrastive learning, and a simple siamese network structure with a stop gradient can avoid mode collapse. Caron *et al.* [3] established a teacher-student self-distillation framework and aligned the two branches with a classification loss. In addition, Debidatta *et al.* [10] utilized feature similarity to mine the nearest neighbors from the support set as positive sample supplements, making positive representations robust and invariant to deformations.

Recently, contrastive learning has been extended to static point cloud understanding. PointContrast [43] generates two views of point clouds, and then utilizes the contrastive loss to pull matched point pairs and push unmatched ones. DepthContrast [47] learns global representations from two augmented depth views by setting an instance discrimination task. Although contrastive learning has achieved great success on images and static point clouds, the utilization of contrastive learning on dynamic point clouds is still under-investigated.

2.2. Dynamic Point Cloud Modeling

Currently, most point cloud video understanding methods focus on supervised learning [11–14, 20, 39–41, 48]. Liu *et al.* [20] added a temporal dimension to PointNet++ [28] to process dynamic point clouds. Wang *et al.* [39] extracted motion information from regularized voxels, and then combined these voxels with raw points for spatiotemporal modeling. Fan *et al.* [13] used stacked convolutions to extract hierarchical spatiotemporal features. P4Transformer [11] captures long relationships between tokens obtained from spatiotemporal tubes. Zhong *et al.* [48] and Wen *et al.* [41] introduced traditional techniques, such as ST-surface or primitives, into the existing network structure to effectively learn spatiotemporal representations. Niemeyer *et al.* [25] learned a temporally and spatially continuous vector field to assign a motion vector to each point, which is suitable for generative tasks such as dynamic point cloud reconstruction. Rempe *et al.* [29] learned object-centric spatiotemporal representations from normalized point clouds and proved to be effective on multiple downstream tasks.

Meanwhile, several works make attempts to conduct self-supervised learning on dynamic point clouds. Wang *et al.* [35] divided input sequences into several temporal clips and then predicted the correct order of randomly shuffled clips. Dong *et al.* [9] used complete and partial sequences as inputs to the teacher and student networks for realizing

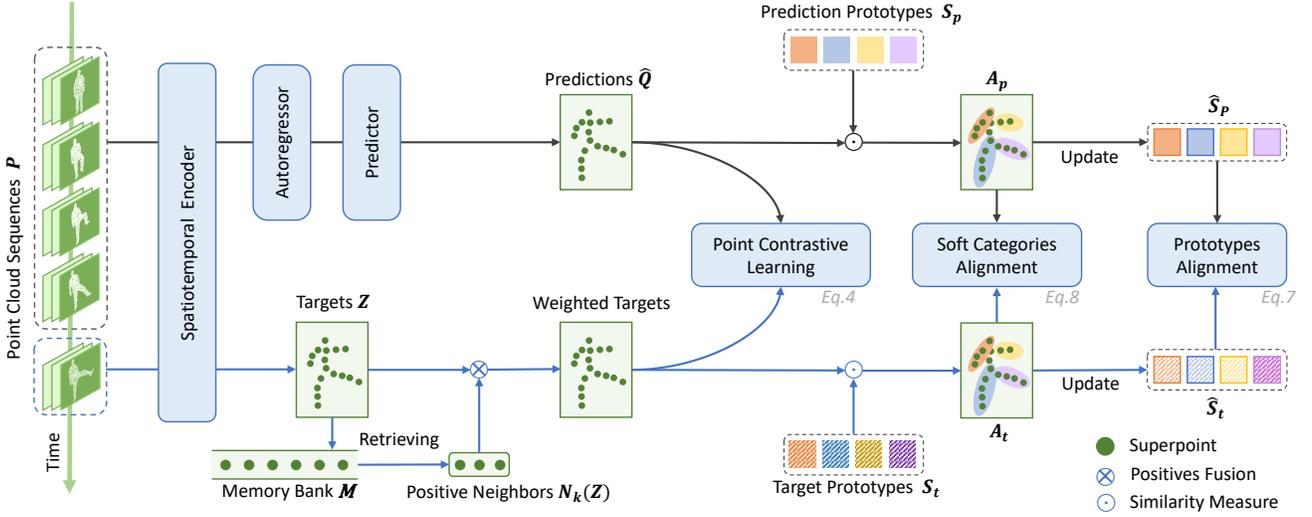


Figure 2. The framework of our PointCPSC. The samples with lower similarities in the memory bank are used as negatives. We only present the selection of positive neighbors from the memory bank for simplicity.

spatiotemporal knowledge distillation. However, the representations learned by contrasting samples at the clip or frame level cannot capture local spatiotemporal dynamics. To remedy this, Sheng *et al.* [32] proposed a self-supervised framework to learn fine-grained representations through contrastive prediction and reconstruction. Although spatiotemporal reconstruction of raw points pays more attention to fine-grained information, the learned representations are susceptible to noises and the network is difficult to be optimized. Shen *et al.* [33] combined contrastive learning and masked predictions to achieve self-supervised representation learning. However, their clip-level masking strategy is insufficient to explore fine-grained dynamics of point clouds. Different from the above methods, in this paper, we propose to conduct contrastive learning between superpoints and introduce semantic clustering as a pretext task to learn representations versatile to diverse downstream tasks.

3. Method

The overall framework of our PointCPSC is presented in Fig. 2. A point cloud video is denoted as $\mathbf{P} \in \mathbb{R}^{T \times N \times 3}$, where T is sequence length and N is the number of points in each frame. We equally divide the video into L segments. After all segments are processed by the spatiotemporal encoder, the former $L-1$ segments are fed into transformer autoregressor to predict the L -th target segment in the latent space. We follow previous works [6, 17] to implement a predictor to further transform the predictions. To make local contrasts more effective and model comprehensive positive representations, we propose to select appropriate negatives and beneficial positive neighbors. Meanwhile, we also per-

form semantic clustering to adapt the self-supervised framework for multiple downstream tasks.

3.1. Point Contrastive Prediction

Following [13], spatiotemporal tubes are defined as tubes within spatial radius s and temporal radius t centered on certain points. We first encode these spatiotemporal tubes to obtain the embeddings of superpoints. These superpoints aggregate local information and can well preserve local semantics, thereby facilitating the learning of fine-grained information.

Specifically, after the L -th segment is encoded by the spatiotemporal encoder, we obtain the target embeddings $\mathbf{Z} \in \mathbb{R}^{l \times r \times c}$, where l , r , and c are frame length, superpoint number, and feature dimension, respectively. We then take the representations of the $L-1$ segment as predictions, which are denoted as $\mathbf{Q} \in \mathbb{R}^{l \times r \times c}$. However, owing to the disorder of point clouds, the predictions are not aligned with the corresponding targets. We take target positions as anchors to search for neighbors within predictions and perform feature interpolation to obtain updated predictions $\hat{\mathbf{Q}} \in \mathbb{R}^{l \times r \times c}$.

Negative Selection. Due to the high redundancy in a point cloud video, numerous superpoints contain similar semantics. For effective contrastive learning, samples with high similarities are discarded. Specifically, we use dynamically updated memory bank, denoted as \mathbf{M} , to store history target embeddings. During per-training, we calculate the similarities between the current target and those in \mathbf{M} as follows:

$$sim = \cos(\mathbf{m}, \mathbf{z}), \mathbf{m} \in \mathbf{M}, \quad (1)$$

where sim represents the similarity between history embedding \mathbf{m} and current superpoint \mathbf{z} . For embeddings in \mathbf{M} , we sort their similarities with \mathbf{z} and retain 70% negatives with the lowest similarity for contrastive learning.

Positive Neighbors. Instead of directly employing the embeddings of the same spatiotemporal position as positives, we propose to explore favorable positive neighbors by utilizing feature similarity:

$$N_k(\mathbf{z}) = \operatorname{argmax}_{\mathbf{m} \in \mathbf{M}} (\cos(\mathbf{m}, \mathbf{z}), \operatorname{top}_n = K), \quad (2)$$

where $N_k(\mathbf{z})$ represents the retrieved K neighbors related to the current target superpoint \mathbf{z} . Following [15], we adaptively introduce these positive neighbors into local Info Noise Contrastive Estimation (InfoNCE) loss [26] to perform contrastive learning between predictions and targets, which is represented as follows:

$$\mathbf{w} = \operatorname{Softmax}(N_k(\mathbf{z}) \cdot \mathbf{z}) \in \mathbb{R}^K, \quad (3)$$

$$\mathcal{L}_l = -\log \frac{\sum_{j=0}^K w_j \exp(\mathbf{z}^T \mathbf{q} / \tau)}{\sum_{j=0}^K w_j \exp(\mathbf{z}^T \mathbf{q} / \tau) + \sum_{\mathbf{q}' \in \Psi} \exp(\mathbf{z}_i^T \mathbf{q}' / \tau)}, \quad (4)$$

where $\mathbf{q} \in \{\hat{\mathbf{q}}_+ \cup N_k(\mathbf{z})\}$ is the positive set that contains the positive sample $\hat{\mathbf{q}}_+$ and neighbors $N_k(\mathbf{z})$, Ψ is the negative set, w_0 is set as 1 and means the weight between \mathbf{z} and $\hat{\mathbf{q}}_+$, $w_{j=1, \dots, K}$ is calculated using Eq.3, and τ is temperature hyper-parameter.

Overall, we make point contrastive prediction more effective by selecting proper negatives, and robust representations are learned by supplying positive neighbors from other instances. The feature similarity is utilized as adaptive weights to combine neighbors and positive samples. We further investigate how to utilize retrieved positive neighbors in ablation studies.

3.2. Semantic Clustering

Local spatiotemporal representations are learned based on the above contrastive learning framework. As multiple-granularity semantics are critical to diverse downstream tasks, we introduce a semantic clustering task on dynamic point clouds.

Specifically, we first parameterize two group prototypes for prediction and target embeddings. During pre-training, these two group prototypes are gradually learned. The distances from predictions to their corresponding prototypes are calculated to obtain soft category distributions for each predicted superpoint. The same operation is also performed to obtain soft category distributions for each target superpoint. Intuitively, the embeddings of predictions and targets should follow the same category probability distribution. Meanwhile, the two group prototypes should also follow approximate distributions. We denote the initial target prototypes as $\mathbf{S}_t = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_k] \in \mathbb{R}^{k \times c}$ and the semantic

clustering is achieved as follows:

$$\mathbf{A}_t = \operatorname{Softmax}_k(\mathbf{Z} \cdot \mathbf{S}_t^T) \in \mathbb{R}^{l \times r \times k}, \quad (5)$$

$$\hat{\mathbf{S}}_t = \frac{1}{\sum_{i,j} \mathbf{A}_t[i,j]} \sum_{i,j} \mathbf{A}_t[i,j] \odot \mathbf{Z}[i,j] \in \mathbb{R}^{k \times c}, \quad (6)$$

where \mathbf{A}_t represents soft category assignments of \mathbf{Z} , $\hat{\mathbf{S}}_t$ are updated prototypes of \mathbf{Z} , and \odot is a Hadamard product. Similarly, the soft category distributions \mathbf{A}_p and updated prototypes of predictions $\hat{\mathbf{S}}_p$ can be obtained.

Following [42], an extra predictor is utilized to further transform $\hat{\mathbf{S}}_p$. Finally, an InfoNCE loss [26] is employed to align $\hat{\mathbf{S}}_p$ and $\hat{\mathbf{S}}_t$ as follows:

$$\mathcal{L}_c = -\log \frac{\exp(\hat{\mathbf{s}}_t^T \hat{\mathbf{s}}_p / \tau)}{\exp(\hat{\mathbf{s}}_t^T \hat{\mathbf{s}}_p / \tau) + \sum_{\hat{\mathbf{s}}'_p \in \phi} \exp(\hat{\mathbf{s}}_t^T \hat{\mathbf{s}}'_p / \tau)}, \quad (7)$$

where $(\hat{\mathbf{s}}_t, \hat{\mathbf{s}}_p)$ is a positive pair, and ϕ is the negative set that contains unmatched prototypes.

Moreover, we utilize Kullback-Leibler Divergence (KL) loss to achieve the alignment of soft category distributions between predictions and targets:

$$\mathcal{L}_k = \sum_{i=1}^k \mathbf{a}_p^i (\log \mathbf{a}_p^i - \log \mathbf{a}_t^i), \quad (8)$$

where \mathbf{a}_p^i and \mathbf{a}_t^i are i -th category probabilities of predictions and targets, respectively.

Overall, the total loss of our self-supervised framework consists of three parts:

$$\mathcal{L}_{total} = \mathcal{L}_l + \lambda_1 \mathcal{L}_c + \lambda_2 \mathcal{L}_k, \quad (9)$$

where λ_1 and λ_2 are hyper-parameters for balance. By performing the alignment of superpoint categories and prototypes, our method can well capture multiple-granularity semantics.

Note that, prototypes and soft category alignments are performed when pre-training on dynamic point cloud semantic segmentation. This is because soft category alignment is capable of extracting fine-grained point-level information which is beneficial for segmenting objects. While pre-training on action recognition, we only conduct prototypes alignment to provide high-level semantics.

4. Experiments

Firstly, the dataset benchmarks and implementation details are introduced, and then we compare the performance of PointCPSC with previous methods under multiple settings. Extensive ablation studies are also conducted to demonstrate the effectiveness of each sub-module in our framework. Finally, we present qualitative analysis and visualizations to verify our motivation.

Table 1. Action recognition accuracy (%) on MSRAction-3D.

Methods		#Frames			
		8	12	16	24
Supervised Learning	MeteorNet [20]	81.14	86.53	88.21	88.50
	Kinet [48]	83.84	88.53	91.92	93.27
	PST ² [40]	86.53	88.55	89.22	-
	PPTr [41]	84.02	89.89	90.31	92.33
	P4Transformer [11]	83.17	87.54	89.56	90.94
	PST-Transformer [12]	83.97	88.15	91.98	93.73
	PSTNet [13]	83.50	87.88	89.90	91.20
	PSTNet++ [14]	83.50	88.15	90.24	92.68
End-to-end Fine-tuning	PointCPSC	88.89	90.24	92.26	92.68
Linear Probing	PointCPSC	86.87	89.56	88.89	90.24

4.1. Datasets and Pre-training Details

We perform point cloud action recognition on MSRAction-3D [19] and NTU-RGBD [31], 4D semantic segmentation on Synthia 4D [8], and gesture recognition on NvGesture [24].

The MSRAction-3D [19] dataset records 567 human action sequences with Kinect, including 20 action categories performed by 10 subjects. We follow [20] to obtain 270 training videos and 297 test videos.

The NTU-RGBD [31] dataset collects 56880 videos recorded by three cameras from different angles, with a total of 40 subjects and 60 categories. There are 40,320 training videos and 16,560 test videos under a cross-subject setting.

The Synthia 4D dataset [8] contains 6 videos of different driving scenarios, generated from the Synthia dataset [30]. Following [8, 20], this dataset is split into 19,888 training frames, 815 validation frames, and 1,886 test frames.

The NvGesture [24] dataset collects 1532 dynamic sequences with 25 categories. We follow [23] to obtain 1050 training videos and 482 test videos.

Pre-training on action recognition. PSTNet [13] is utilized as our encoder to conduct experiments. During training, we use 88 clips as one batch, where each clip contains 24 1024-point frames. The frame interval of sampling for MSRAction-3D and NTU-RGBD are set to 1 and 2, respectively. The number of neighbors for the ball query is set to 9. The spatial search radius is set to 0.5 and 0.1 for MSRAction-3D and NTU-RGBD, respectively. Following [13], random scaling is adopted for data augmentation. The AdamW optimizer [21] with a cosine decay scheduler is employed for optimization. We pre-train the model for 200 epochs with an initial learning rate of 0.0008. The temperature hyper-parameter is set to 0.01.

Pre-training on semantic segmentation. The encoder in P4Transformer [11] is adopted to conduct experiments. The 04 sequence of the Synthia 4D dataset is employed for pre-training. We sample 4-frame clips with each frame con-

taining 4096 points for training. The frame interval is set to 1, and the spatial search radius and the number of neighbors for the ball query are set to 0.9 and 32, respectively. The data augmentation strategy in [11] is adopted in the experiments. We employ the same optimization strategies as those for pre-training on action recognition.

4.2. End-to-end Fine-tuning

We first perform pre-training on MSRAction-3D, and then add a new classifier after the encoder for fine-tuning. Two linear layers with a batch normalization layer are adopted as the classifier. Following the previous works [11, 13, 20], we test the performance with various lengths of frames. 2048 points are sampled for each frame. The spatial search radius and the number of neighbors for the ball query are set to 0.3 and 9, respectively. We finetune the pre-trained model for 35 epochs and employ a warmup strategy. We compare the performance of our PointCPSC with previous supervised methods in Table 1. As we can see, PointCPSC consistently outperforms the baseline method PSTNet under different frames. This demonstrates the effectiveness of our method, which helps the model to learn semantic information that is beneficial to the point cloud action recognition task.

After pre-training on Synthia 4D, we follow [11, 27] to add a decoder and a classifier for fine-tuning. During fine-tuning, 3-frame clips with each frame containing 16384 points are sampled. The spatial search radius and the number of neighbors for the ball query are set to 0.9 and 32, respectively. We finetune the pre-trained model for 150 epochs and adopt the warmup strategy. We compare our PointCPSC with other supervised methods and the results are presented in Table 2. The PointCPSC with 3 frames achieves 84.47 mIOU, which is 2% higher than that of P4Transformer with 1 frame. This indicates that temporal context information benefits semantic segmentation. Compared with the baseline P4Transformer, PointCPSC achieves significant improvements, especially in small object segmentation, including traffic signs, pedestrians, lanes, and traffic lights. This validates that our self-supervised framework can well fit fine-grained downstream tasks.

4.3. Linear Probing

After pre-training on MSRAction-3D, we evaluate the pre-trained encoder under the setting of linear probing. The same experimental setups as fine-tuning are adopted. As shown in Table 1, our results are competitive compared to previous methods, where the performance of PointCPSC with 8 frames outperforms all supervised methods. Furthermore, our method with 12 frames achieves the accuracy of 89.56%, surpassing the baseline PSTNet [13] with notable margins. These results demonstrate that our pre-training can learn beneficial high-level semantics.

Table 2. Semantic segmentation accuracy (%) on the Synthia 4D dataset.

Methods	Input	Frame	Bldn	Road	Sdwk	Fence	Vegittn	Pole	Car	T. Sign	Pedstrn	Bicycl	Lane	T. Light	mIOU
Minkowski [8]	voxel	3	90.13	98.26	73.47	87.19	99.10	97.50	94.01	79.04	92.62	0.00	50.01	68.14	77.46
PointNet++ [28]	point	1	96.88	97.72	86.20	92.75	97.12	97.09	90.85	66.87	78.64	0.00	72.93	75.17	79.35
MeteorNet [20]	point	3	98.10	97.72	88.65	94.00	97.98	97.65	93.83	84.07	80.90	0.00	71.14	77.60	81.80
PSTNet [13]	point	1	96.32	98.07	85.40	94.66	97.16	97.51	94.83	76.65	76.99	0.00	75.39	76.45	80.79
PSTNet [13]	point	3	96.91	98.33	90.83	95.00	96.96	97.61	95.15	77.45	85.68	0.00	75.71	77.28	82.24
P4Transformer [11]	point	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 [11]	point	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
PointCPSC	point	3	95.88	98.31	94.13	96.32	97.12	98.55	95.74	85.35	87.11	0.00	78.85	86.28	84.47

Table 3. Action recognition accuracy (%) on NTU-RGBD under cross-subject setting.

Methods	Accuracy (%)
3DV-Motion [39] (voxel)	84.5
3DV-PointNet++ [39] (voxel+point)	88.8
Kinet [48]	92.3
P4Transformer [11]	90.2
PST-Transformer [12]	91.0
PSTNet [13]	90.5
PSTNet++ [14]	91.4
PointCPSC (50% Semi-supervised)	88.0

4.4. Semi-supervised Learning

We first pre-train the models on NTU-RGBD, and then conduct semi-supervised fine-tuning with 50% training data under the cross-subject setting. The spatial radius is set as 0.5. We finetune 20 epochs and adopt a warmup strategy. The other experimental setups and optimization strategies are the same as those used for fine-tuning on MSRAction-3D. As shown in Table 3, we compare the performance of PointCPSC with previous supervised methods. Our method with only 50% annotated data achieves the accuracy of 88.0%. This clearly demonstrates the effectiveness of our self-supervised pre-training, which learns advantageous information to assist in semi-supervision.

4.5. Transfer Learning

We conduct pre-training on NTU-RGBD and then transfer the pre-trained encoder to gesture recognition to demonstrate the generalization of the pre-trained representations. We finetune the pre-trained model on the NvGesture dataset for 50 epochs. During fine-tuning, 32 1024-point frames are sampled. The batch size and initial learning rate are set to 32 and 0.02, respectively. The SGD optimizer with cosine decay strategy is adopted for optimization. We compare our PointCPSC with other supervised methods and the results are shown in Table 4. It can be seen that our method facilitates the baseline PSTNet to achieve higher accuracy. This validates that our method has superior generalization capability and the learned representations are beneficial for

Table 4. Gesture recognition accuracy (%) on NvGesture.

Methods	Input	NvGesture
FlickerNet [22]	point	86.3
PLSTM-base [23]	point	87.6
PLSTM-early [23]	point	93.5
PLSTM-PSS [23]	point	93.1
PLSTM-middle [23]	point	94.7
PLSTM-late [23]	point	93.5
Kinet [48]	point	89.1
PSTNet [13] (50epochs)	point	86.1
PointCPSC (50epochs)	point	87.3

gesture recognition on point cloud videos.

4.6. Ablation Studies

We conduct ablation studies on MSRAction-3D and Synthia 4D. On MSRAction-3D, 16-frame clips are sampled and the other hyper-parameters are the same as end-to-end fine-tuning. On Synthia 4D, 4096-point frames are sampled and the models are finetuned for 75 epochs. All hyper-parameters except for the ablated ones are kept the same for fair comparison.

The Negatives with Appropriate Ratios. For current batch targets, we calculate their feature similarities with history embeddings stored in the memory bank. We rank the similarities in descending order and use different ratios of embeddings as negatives. The results are shown in Table 5. It can be observed that our method achieves the highest accuracy with 70% negatives and more negatives introduce moderate performance drops. This indicates that there exist negatives with high similarity in the memory bank, namely undesired negatives, and they should be abandoned in pre-training.

The Utilization of Positive Neighbors. Although the positive neighbors are retrieved based on feature similarity, how to utilize these neighbors still needs further exploration. Three different schemes are compared and the results are presented in Table 6. Compared with integrating positive neighbors with feature similarity as softmax weight (B1), the accuracy of directly adding K positive pairs (B2)

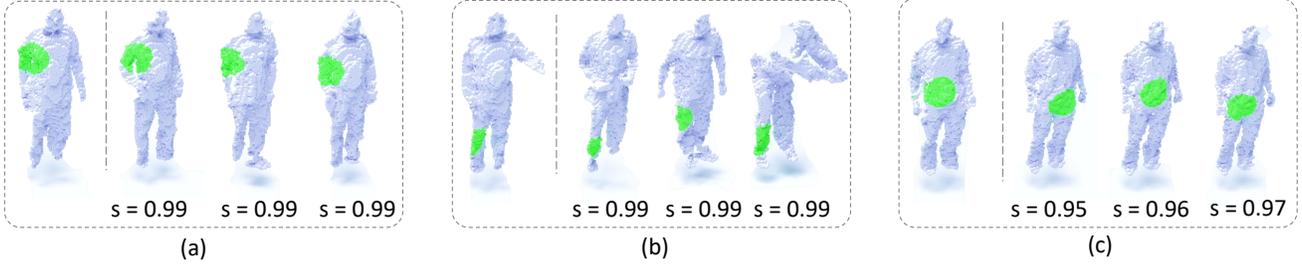


Figure 3. The visualization of positive neighbors. The neighbors with high similarities come from different actions or subjects. The number denotes the similarity.

Table 5. The negatives with appropriate ratios.

	A1	A2 (Ours)	A3	A4
Negatives Ratio (%)	60	70	80	90
Accuracy (%)	91.38	92.26	90.91	90.23

Table 6. Ablation study on positive neighbors.

	Weighting Scheme	Accuracy (%)
B1	Softmax Weighting	89.90
B2	Add K positive pairs	90.91
B3 (Ours)	Feature Weighted Fusion	92.26

Table 7. Results achieved using different numbers of positive neighbors.

	C1	C2 (Ours)	C3	C4
Numbers	1	3	5	10
Accuracy (%)	91.58	92.26	91.25	90.57

has increased by 1%. When combining the target and its neighbors with their similarity, the performance is optimal. By utilizing weight fusion, the comprehensive representations of positive samples are constructed and they are more generalized for performing contrastive learning.

The Number of Positive Neighbors. The highly similar neighbors are mined from other instances as positive supplements. We evaluate the number of positive neighbors and the results are shown in Table 7. As we can see, the performance is improved as the number of positive neighbors is increased from 1 to 3. However, further increase of positive neighbors cannot introduce accuracy gains but lead to lower performance. Consequently, 3 positive neighbors are used as the default setting in our experiments.

The Size and Cost of Memory Bank. We investigate the performance of memory banks with different sizes in terms of running time and memory consumption. Specifically, we evaluate models with the memory bank size of 256, 512, and 1024. The results are shown in Table 8. It can be observed that our method achieves higher accuracy with a larger memory bank, producing an accuracy of 92.26% with a memory bank of size 1024. Meanwhile, running

Table 8. Time (mins/epoch), memory (MiB), and accuracy (%) achieved using memory banks with different sizes.

	Size	Time	Memory	Accuracy (%)
D1	256	1.2	8647	90.91
D2	512	1.7	9435	91.57
D3 (Ours)	1024	2.1	10276	92.26

Table 9. Accuracy (%) achieved using different numbers of prototypes.

	Prototypes	MSRAction-3D	Synthia 4D
E1 (Ours)	10	92.26	71.13
E2	20	90.91	70.53
E3	30	90.91	70.28

time and memory consumption have an acceptable increase. Consequently, 1024 is used as the default size of the memory bank in our experiments.

The Size of Prototypes on Different Benchmarks. We also study the number of prototypes on different benchmarks. The results are shown in Table 9. Our method achieves the highest accuracy on MSRAction-3D with the prototype number of 10. Intuitively, the prototypes aggregated from superpoint representations can be viewed as human body parts with specific semantics. From this point of view, introducing too many prototypes may suffer semantic-less fragments and decrease the performance. On Synthia 4D, our method achieves the highest accuracy of 71.13%. It maybe because this prototype number is close to the object categories in this dataset. This indicates that a suitable number of prototypes that fit the dataset well is beneficial to the final performance.

The Effectiveness of Self-supervised Tasks. We evaluate the effectiveness of local contrastive prediction, positive neighbors, and the pretext task of semantic clustering on different datasets. The results are shown in Table 10. Note that, we only perform prototype alignment on MSRAction-3D. On MSRAction-3D, the supplements of positive neighbors with the local contrastive prediction branch improve the accuracy to 91.98%. When prototype alignment is in-

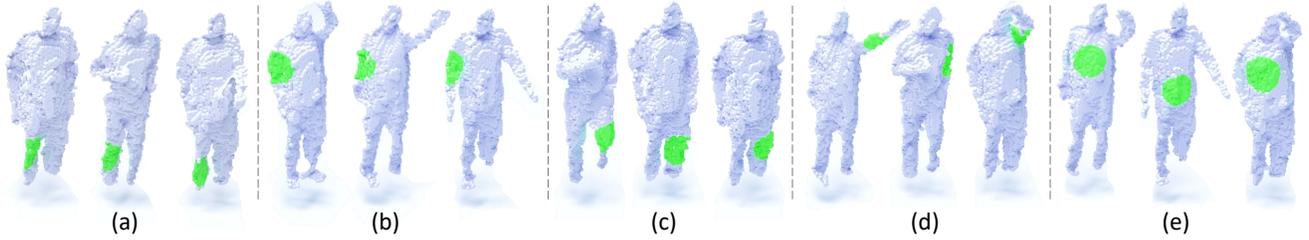


Figure 4. The visualization of the prototypes learned in pre-training. Different prototypes correspond to specific human body regions.

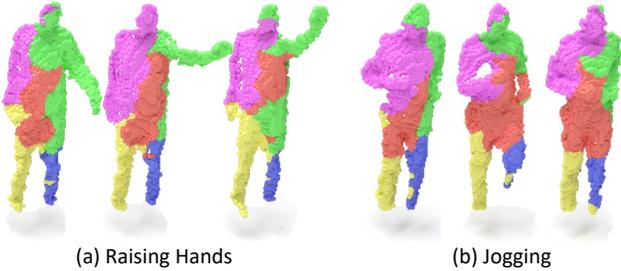


Figure 5. The visualization of human motion segmentation.

roduced, our method achieves an accuracy of 92.26%. On Synthia 4D, the retrieved positive neighbors also contribute to performance improvement. More importantly, joint prototype alignment and soft category alignment further improve the segmentation accuracy with a large margin. This demonstrates that our method can well capture fine-grained cues that benefit the semantic segmentation.

4.7. Qualitative Analysis

Positive Neighbors. During pre-training, several highly similar superpoints stored in the memory bank are selected based on feature similarity. The corresponding raw points areas of superpoints are visualized in Fig. 3. These superpoints come from diverse videos and categories, but present highly similar human body regions. Since our self-supervised pre-training aims to model local dynamics, these highly similar superpoints from other instances should be treated as positive neighbors. This motivates us to design the strategy of sample selection, to achieve effective contrast and learn robust representations.

Prototypes Visualization. We explore what the prototypes have learned by classifying the superpoints aggregated from raw point cloud sequences with pre-trained prototypes. We randomly select several prototypes and evaluate four videos. The visualization results are shown in Fig. 4. Each prototype corresponds to a specific region of human bodies. This demonstrates that our self-supervised framework effectively models local structures and learns high-level semantics beneficial for downstream tasks.

Potential Applications. We visualize the learned proto-

Table 10. Ablation results on different benchmarks.

	Tasks	MSR (%)	Syn (%)
F1	Local Contrastive Prediction	91.38	70.01
F2	F1 + Sample Selection Strategy	91.98	70.45
F3	F2 + Prototype Alignment	92.26	70.73
F4	F2 + Soft Category Alignment	-	70.67
F5	F2 + Prototype Alignment + Soft Category Alignment	-	71.13

types on two point cloud sequences in Fig. 5, where each color represents a prototype. When visualizing, the prototypes with the same semantics are incorporated. It can be seen that these pre-trained prototypes embed specific human body parts. This demonstrates that the pretext task of semantic clustering models human parts from superpoint representations. Intuitively, the prior information learned in pre-training is beneficial for non-rigid motion segmentation. Besides, the soft category assignments of points may benefit interactive annotation tasks.

5. Conclusions

We propose a unified self-supervised framework for pre-training on point cloud videos. To adapt the self-supervised framework for diverse object-centric and scene-centric downstream tasks, we design the pretext task of semantic clustering, which achieves hierarchical semantic alignment between predictions and targets. In addition, we retain proper negatives for effective contrast and select highly similar negatives as positive neighbors for robust representations. Extensive experiments and ablation studies are performed to demonstrate the effectiveness of our self-supervised framework.

Acknowledgments. This work was partially supported by the Fundamental Research Funds for the Central Universities (No.226-2023-00048), the National Natural Science Foundation of China (No.61673270, 61973212, 61972435, 61602499), Artificial Intelligence Key Laboratory of Sichuan Province (2022RZY02), and Guangdong Basic and Applied Basic Research Foundation (2019A1515011271).

References

- [1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. CrossPoint: Self-supervised cross-modal contrastive learning for 3D point cloud understanding. In *CVPR*, 2022. 1
- [2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2
- [3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2
- [5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 2
- [6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 2, 3
- [7] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3D object detection. In *CVPR*, 2022. 1
- [8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski convolutional neural networks. In *CVPR*, 2019. 5, 6
- [9] Yuhao Dong, Zhuoyang Zhang, Yunze Liu, and Li Yi. Complete-to-partial 4D distillation for self-supervised point cloud sequence representation learning. In *CVPR*, 2023. 1, 2
- [10] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a Little Help From My Friends: Nearest-neighbor contrastive learning of visual representations. In *ICCV*, 2021. 2
- [11] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4D transformer networks for spatio-temporal modeling in point cloud videos. In *CVPR*, 2021. 2, 5, 6
- [12] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point spatio-temporal transformer networks for point cloud video modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2181–2192, 2022. 1, 2, 5, 6
- [13] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. PSTNet: Point spatio-temporal convolution on point cloud sequences. In *ICLR*, 2021. 2, 3, 5, 6
- [14] Hehe Fan, Xin Yu, Yi Yang, and Mohan Kankanhalli. Deep hierarchical representation of point cloud videos via spatio-temporal decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9918–9930, 2021. 1, 2, 5, 6
- [15] Chongjian Ge, Jiangliu Wang, Zhan Tong, Shoufa Chen, Yibing Song, and Ping Luo. Soft neighbors are positive supporters in contrastive visual representation learning. In *ICLR*, 2023. 4
- [16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap Your Own Latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 1, 2
- [17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2, 3
- [18] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8338–8354, 2021. 1
- [19] Wanqing Li, Zhengyu Zhang, and Zicheng Liu. Action recognition based on a bag of 3D points. In *CVPRW*, 2010. 5
- [20] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. MeteorNet: Deep learning on dynamic 3D point cloud sequences. In *ICCV*, 2019. 2, 5, 6
- [21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5
- [22] Yuecong Min, Xiujuan Chai, Lei Zhao, and Xilin Chen. FlickerNet: Adaptive 3D gesture recognition from sparse point clouds. In *BMVC*, 2019. 6
- [23] Yuecong Min, Yanxiao Zhang, Xiujuan Chai, and Xilin Chen. An efficient PointLSTM for point clouds based gesture recognition. In *CVPR*, 2020. 5, 6
- [24] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network. In *CVPR*, 2016. 5
- [25] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy Flow: 4D reconstruction by learning particle dynamics. In *ICCV*, 2019. 2
- [26] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 4
- [27] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *ECCV*, 2022. 5
- [28] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017. 2, 6
- [29] Davis Rempe, Tolga Birdal, Yongheng Zhao, Zan Gojcic, Srinath Sridhar, and Leonidas J Guibas. CaSPR: Learning canonical spatiotemporal point cloud representations. *NeurIPS*, 2020. 2
- [30] 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 *CVPR*, 2016. 5
- [31] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *CVPR*, 2016. 5

- [32] Zhiqiang Shen, Xiaoxiao Sheng, Longguang Wang, Yulan Guo, Qiong Liu, and Xi Zhou. PointCMP: Contrastive mask prediction for self-supervised learning on point cloud videos. In *CVPR*, 2023. 3
- [33] Xiaoxiao Sheng, Zhiqiang Shen, and Gang Xiao. Contrastive predictive autoencoders for dynamic point cloud self-supervised learning. In *AAAI*, 2023. 3
- [34] Chenxin Tao, Honghui Wang, Xizhou Zhu, Jiahua Dong, Shiji Song, Gao Huang, and Jifeng Dai. Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. In *CVPR*, 2022. 2
- [35] Haiyan Wang, Liang Yang, Xuejian Rong, Jinglun Feng, and Yingli Tian. Self-supervised 4D spatio-temporal feature learning via order prediction of sequential point cloud clips. In *WACV*, 2021. 1, 2
- [36] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020. 2
- [37] Xiao Wang, Haoqi Fan, Yuandong Tian, Daisuke Kihara, and Xinlei Chen. On the importance of asymmetry for siamese representation learning. In *CVPR*, 2022. 2
- [38] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021. 2
- [39] Yancheng Wang, Yang Xiao, Fu Xiong, Wenxiang Jiang, Zhiguo Cao, Joey Tianyi Zhou, and Junsong Yuan. 3DV: 3D dynamic voxel for action recognition in depth video. In *CVPR*, 2020. 2, 6
- [40] Yimin Wei, Hao Liu, Tingting Xie, QiuHong Ke, and Yulan Guo. Spatial-temporal transformer for 3D point cloud sequences. In *WACV*, 2022. 1, 2, 5
- [41] Hao Wen, Yunze Liu, Jingwei Huang, Bo Duan, and Li Yi. Point primitive transformer for long-term 4D point cloud video understanding. In *ECCV*, 2022. 1, 2, 5
- [42] Xin Wen, Bingchen Zhao, Anlin Zheng, Xiangyu Zhang, and Xiaojuan Qi. Self-supervised visual representation learning with semantic grouping. In *NeurIPS*, 2022. 4
- [43] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3D point cloud understanding. In *ECCV*, 2020. 1, 2
- [44] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A simple framework for masked image modeling. In *CVPR*, 2022. 1
- [45] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021. 1, 2
- [46] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not All Points Are Equal: Learning highly efficient point-based detectors for 3D LiDAR point clouds. In *CVPR*, 2022. 1
- [47] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3D features on any point-cloud. In *ICCV*, 2021. 2
- [48] Jia-Xing Zhong, Kaichen Zhou, Qingyong Hu, Bing Wang, Niki Trigoni, and Andrew Markham. No Pain, Big Gain: Classify dynamic point cloud sequences with static models by fitting feature-level space-time surfaces. In *CVPR*, 2022. 2, 5, 6