PointCMP: Contrastive Mask Prediction for Self-supervised Learning on Point Cloud Videos

Zhiqiang Shen[1,2*], Xiaoxiao Sheng[1*], Longguang Wang[3†], Yulan Guo[4], Qiong Liu[2], Xi Zhou[2]
[1]Shanghai Jiao Tong University [2]CloudWalk
[3]Aviation University of Air Force [4]Sun Yat-sen University
{shenzhiqiang, shengxiaoxiao}@sjtu.edu.cn, wanglongguang15@nudt.edu.cn

Abstract

Self-supervised learning can extract representations of good quality from solely unlabeled data, which is appealing for point cloud videos due to their high labelling cost. In this paper, we propose a contrastive mask prediction (PointCMP) framework for self-supervised learning on point cloud videos. Specifically, our PointCMP employs a two-branch structure to achieve simultaneous learning of both local and global spatio-temporal information. On top of this two-branch structure, a mutual similarity based augmentation module is developed to synthesize hard samples at the feature level. By masking dominant tokens and erasing principal channels, we generate hard samples to facilitate learning representations with better discrimination and generalization performance. Extensive experiments show that our PointCMP achieves the state-of-the-art performance on benchmark datasets and outperforms existing full-supervised counterparts. Transfer learning results demonstrate the superiority of the learned representations across different datasets and tasks.

1. Introduction

Recently, LiDARs have become increasingly popular in numerous real-world applications to perceive 3D environments, such as autonomous vehicles and robots. Point clouds acquired by LiDARs can provide rich geometric information and facilitate the machine to achieve 3D perception. Early works focus on parsing the real world from static point clouds [9,24,64], while recent researches pay more attention to understanding point cloud videos [14, 16, 54, 55]. Since annotating point clouds is highly time and labor consuming [1, 57], learning from point cloud videos in a self-supervised manner draws increasing interest. Although contrastive learning and mask prediction paradigms

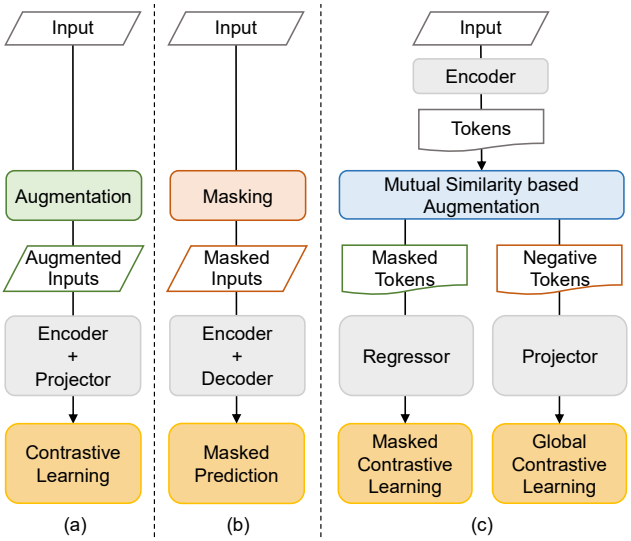


Figure 1. A comparison between (a) contrastive learning paradigm, (b) mask prediction paradigm, and (c) our method.

[6, 19, 21, 22, 58, 63] have shown the effectiveness of self-supervised learning on images or static point clouds, these methods cannot be directly extended to point cloud videos due to the following three challenges:

(i) Multiple-Granularity Information Matters. The contrastive learning paradigm [3,4,6,19,22,63] usually focuses on extracting global semantic information based on instance-level augmentations. In contrast, the mask prediction paradigm [2, 12, 21, 23, 58] pays more attention to modeling local structures while ignoring global semantics. However, since fine-grained understanding of point cloud videos requires not only local spatio-temporal features but also global dynamics [16,55], existing paradigms cannot be directly adopted.

(ii) Sample Generation. The contrastive learning paradigm is conducted by pulling positive samples while pushing negative ones [3, 6, 8, 19, 22, 51, 63], and the mask prediction paradigm learns representations by modeling the visible parts to infer the masked ones [2, 21, 39, 49, 58, 62].

*These authors contributed equally.
†Corresponding author.

Both paradigms rely heavily on the augmented samples at the input level. Further, as demonstrated in several works [18, 25, 26, 42], self-supervised learning can significantly benefit from proper hard samples. However, the spatial disorder, temporal misalignment, and uneven information density distribution impose huge challenges on hard sample generation for point cloud videos at the input level.

(iii) Leakage of Location Information. The mask prediction paradigm usually learns to reconstruct masked raw signals by modeling visible ones [2, 21, 39, 49, 58, 62]. For images, the contents are decoupled from the spatial position such that positional encoding is provided as cues to predict masked regions. However, for point clouds with only xyz-coordinates, positional encoding may be used as shortcuts to infer the masked points without capturing geometric information [30, 39].

In this paper, we propose a contrastive mask prediction framework for self-supervised learning on point cloud videos, termed as PointCMP. To address challenge (i), our PointCMP integrates the learning of both local and global spatio-temporal features into a unified two-branch structure, and simultaneously conducts self-supervised learning at different granularities (Fig. 1(c)). For challenge (ii), we introduce a mutual similarity based augmentation module to generate hard masked samples and negative samples at the feature level. To handle challenge (iii), instead of directly regressing the coordinates of masked points, token-level contrastive learning is conducted between the predicted tokens and their target embeddings to mitigate information leakage.

Our contributions are summarized as follows:

- We develop a unified self-supervised learning framework for point cloud videos, namely PointCMP. Our PointCMP integrates the learning of multiple-granularity spatio-temporal features into a unified framework using parallel local and global branches.
- We propose a mutual similarity based augmentation module to generate hard masked samples and negative samples by masking dominant tokens and principal channels. These feature-level augmented samples facilitate better exploitation of local and global information in a point cloud video.
- Extensive experiments and ablation studies on several benchmark datasets demonstrate the efficacy of our PointCMP on point cloud video understanding.

2. Related Work

In this section, we first briefly review two mainstream self-supervised learning frameworks. Then, we present recent advances for point cloud video understanding.

2.1. Contrastive Learning

Contrastive learning has greatly promoted the development of self-supervised learning [3, 4, 6–8, 19, 22, 52, 63]. Usually, semantically consistent sample pairs are separately encoded by an asymmetric siamese network, and then contrastive loss aligns them to facilitate the encoder to learn discriminative representations [45, 50, 51]. For contrastive learning on images, data augmentation has been widely investigated to generate positive and negative samples to improve the discriminability of representations [6, 18, 32, 40].

Recently, contrastive learning has also been studied on static point clouds. Specifically, Xie *et al*. [57] used random geometric transformations to generate two views of a point cloud and associated matched point pairs in these two views using contrastive loss. Zhang *et al*. [66] constructed two augmented versions of a point cloud and used their global features to setup an instance discrimination task for pre-training.

2.2. Mask Prediction

Mask prediction has demonstrated its effectiveness in numerous computer vision tasks and draws increasing interest [2, 12, 21, 23, 49, 58]. Bao *et al*. [2] proposed a BERT-style framework [27] to predict token identities of masked patches based on visible ones. Then, Zhou *et al*. [68] developed an online tokenizer for better image BERT pre-training. Later, Feichtenhofer *et al*. [17] and Tong *et al*. [46] introduced mask prediction to videos and obtained representations rich in local details by inferring masked spatio-temporal tubes.

Recently, several efforts have been made to extend the mask prediction paradigm to point clouds. Specifically, Yu *et al*. [62] proposed PointBERT and introduced a masked point modeling task for point cloud pre-training. Pang *et al*. [39] proposed Point-MAE to reconstruct masked point coordinates using high-level latent features learned from unmasked ones. Liu *et al*. [30] proposed to use binary point classification as a pretext task for point cloud masked autoencoding.

Most existing contrastive learning and mask prediction methods rely on input-level augmentation to conduct self-supervised learning on static point clouds. Nevertheless, it is intractable to directly extend these methods to point cloud videos as more complicated augmentation operations are required to cover the additional temporal dimension. To remedy this, we propose to synthesize samples at the feature level based on mutual similarities, which enables reasonable sample generation without considering the unstructured data formats of point cloud videos.

2.3. Point Cloud Video Understanding

Spatial disorder and temporal misalignment make point cloud videos more challenging to be parsed using a neu-

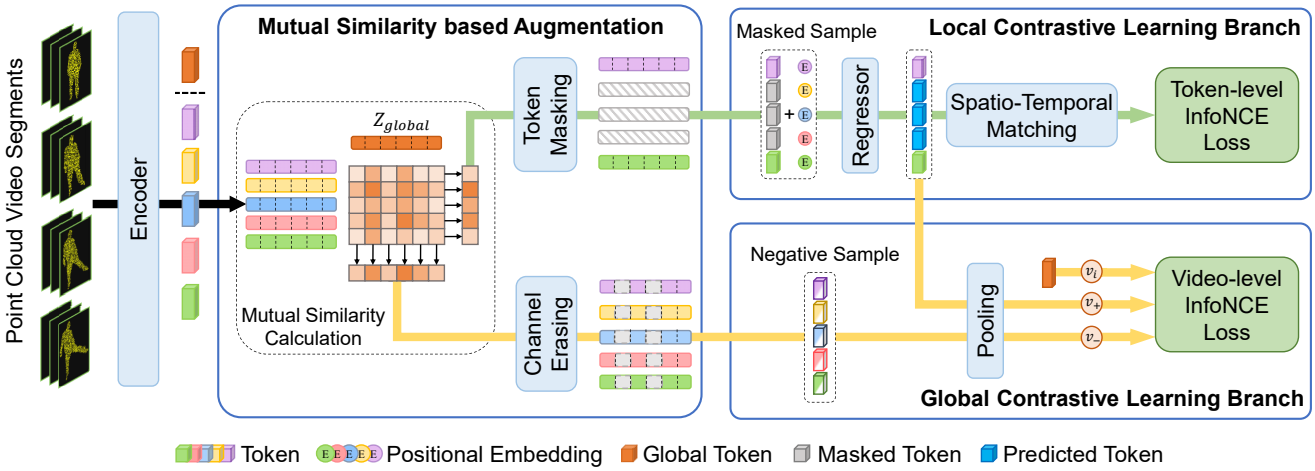


Figure 2. An overview of our PointCMP.

ral network than structured data like images. To leverage the advanced techniques developed for structured data, previous methods transform point clouds into a sequence of bird's eye views [33], voxels [10, 61], and pillar grids [60]. However, these transformations inevitably lead to the loss of geometric details. Recently, more attention has been paid to learning directly on raw points using attention-based models [13, 14, 54, 55], convolution-based models [15, 16, 31], and hand-crafted temporal descriptors [53, 67]. Specifically, Fan *et al*. [15] proposed a spatio-temporal decoupled encoder, which alternately performs spatial and temporal convolution to model raw point sequences hierarchically. Then, they further developed P4Transformer [13] that utilizes point spatio-temporal tubes to aggregate local neighborhoods into tokens.

Despite the huge success of self-supervised learning methods in video understanding [5, 20, 29, 38, 41, 44, 48, 56, 65], self-supervised point cloud video understanding is still under-investigated. Recently, Wang *et al*. [47] designed a pretext task, namely recurrent order prediction (ROP), to predict the temporal order of shuffled point cloud segments for self-supervised learning. However, this method can only capture clip-level temporal structures and cannot exploit finer spatio-temporal details. To parse a point cloud video, it is important for a self-supervised method to capture both spatio-temporal local structures and global semantics. To this end, we develop a unified PointCMP framework that can enable networks to simultaneously learn information with different granularities.

3. Method

The architecture of our PointCMP is illustrated in Fig. 2. Given a point cloud video, it is first uniformly divided into L segments. Then, these segments are fed to an encoder to produce tokens $\boldsymbol{Z} \in \mathbb{R}^{L\times N\times C}$ by aggregating local spatio-temporal information, where N means the token number aggregated from each segment and C is the number of channels. Meanwhile, a global token $\boldsymbol{Z}_{global} \in \mathbb{R}^{C}$ with global semantics is also obtained following [15]. Next, $\boldsymbol{Z}_{global}$ and $\boldsymbol{Z}$ are passed to a mutual similarity based augmentation module for online sample generation. Afterwards, a local contrastive learning branch and a global contrastive learning branch are employed to capture multi-granularity information.

3.1. Mutual Similarity based Augmentation

Hard samples have been demonstrated to be critical to the performance of self-supervised learning [18, 26, 42]. However, it is challenging to generate hard samples for orderless and unstructured point cloud videos at the input level. To address this issue, we introduce a mutual similarity based augmentation module to synthesize hard samples at the feature level.

Hard Masked Samples. Our intuition is that reconstruction is easier when tokens sharing higher similarities with the global token are visible. Therefore, we are motivated to mask these tokens to synthesize hard masked samples. Specifically, the similarity $\boldsymbol{s}^i$ between the i-th token $\boldsymbol{z}^i$ and the global token $\boldsymbol{Z}_{global}$ is calculated as:

$$\boldsymbol{s}^i = \frac{\boldsymbol{z}^i}{\|\boldsymbol{z}^i\|_2} \cdot \frac{\boldsymbol{Z}_{global}}{\|\boldsymbol{Z}_{global}\|_2}. \tag{1}$$

Then, the top 40% tokens with the highest similarities are selected as dominant ones. Note that, point patches corresponding to adjacent tokens usually share overlapped regions [62]. That is, token-level masking may introduce shortcuts for mask prediction. To remedy this, segment-level masking is adopted as different segments are isolated. Specifically, L_m segments with the most dominant tokens are selected with all tokens ($\mathbb{R}^{L_m\times N\times C}$) being masked. By masking these tokens that share high similarity with the global token, the difficulty of mask prediction is largely in-

creased. It is demonstrated in Sec. 4.6 that our hard masked samples can facilitate the encoder to achieve much higher accuracy.

Hard Negative Samples. Our major motivation is that different channels contain information of various importance, and the channels with higher correlation with the global token are more discriminative. Consequently, we synthesize hard negative samples by erasing these channels. Specifically, the correlation of the c-th channel in the i-th token $\boldsymbol{s}_c^i$ is calculated as:

$$\boldsymbol{s}_c^i = \frac{\boldsymbol{z}_c^i}{\|\boldsymbol{z}^i\|_2} \cdot \frac{\boldsymbol{z}_c^{global}}{\|\boldsymbol{Z}_{global}\|_2}, \tag{2}$$

where $\boldsymbol{z}_c^{global}$ is the c-th channel of the global token. Then, we rank $\boldsymbol{s}_c^i$ to obtain the order of each channel $\boldsymbol{o}_c^i$, and sum up the resultant ranks across all tokens:

$$\boldsymbol{A}_c = \sum_{i=1}^{L\times N} \boldsymbol{o}_c^i, \tag{3}$$

Next, the top 20% channels are selected as principal channels and erased to produce hard negative samples.

3.2. Local Contrastive Learning Branch

In the local branch, we first generate positional embedding for each token by feeding its spatio-temporal coordinate (x, y, z, t) to a linear layer. Then, these positional embeddings are summed with their tokens and fed to a regressor to predict masked tokens using the context and position cues. Next, the predicted tokens $\boldsymbol{Z}_{pre} \in \mathbb{R}^{L_m\times N\times C}$ are passed to a spatio-temporal matching module, as shown in Fig 3. Specifically, $\boldsymbol{Z}_{pre}$ is pooled to obtain a global representation $\mathbb{R}^{L_m\times C}$, which is then added to $\boldsymbol{Z}_{pre}$. Afterwards, the resultant token is fed into a decoder to predict their position $\boldsymbol{P}_{pre} \in \mathbb{R}^{L_m\times N\times 3}$. Here, a three-layer Transformer [13] is adopted as the regressor and FoldingNet [59] is used as the decoder.

As discussed in Sec. 1, the positional embeddings may lead to leakage of location information when inferring the coordinates for masked points. To remedy this, we adopt a contrastive loss to associate the representations of predicted tokens $\boldsymbol{Z}_{pre}$ and corresponding groundtruth tokens $\boldsymbol{Z}_{gt}$ learned by the encoder. Specifically, the tokens located at $\boldsymbol{P}_{gt}$ are obtained through trilinear interpolation by querying $\boldsymbol{Z}_{pre}$ located at $\boldsymbol{P}_{pre}$, resulting in $\hat{\boldsymbol{Z}}_{pre}$. For the i-th token $\boldsymbol{z}_i \in \hat{\boldsymbol{Z}}_{pre}$, the corresponding token in $\boldsymbol{Z}_{gt}$ is adopted as the positive sample $\boldsymbol{z}_+$. Meanwhile, other tokens are regarded as negative samples. This avoids directly using the token position correspondence to construct sample pairs. The InfoNCE loss [37] is used for training:

$$\mathcal{L}_{\boldsymbol{z}_i} = -\log \frac{\exp\left(\boldsymbol{z}_i^T \boldsymbol{z}_+/\tau\right)}{\exp\left(\boldsymbol{z}_i^T \boldsymbol{z}_+/\tau\right) + \sum_{\boldsymbol{z}_j\in\Phi} \exp\left(\boldsymbol{z}_i^T \boldsymbol{z}_j/\tau\right)}, \tag{4}$$

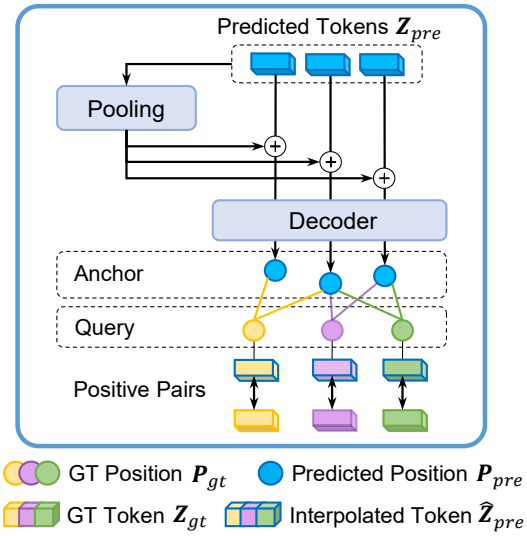


Figure 3. Network Architecture of our spatio-temporal matching module.

where τ is a temperature parameter and Φ is a negative sample set. Through token-level contrastive learning, the encoder can alleviate the shortcuts of positional encoding to capture fine-grained local information.

3.3. Global Contrastive Learning Branch

In the global branch, we focus on learning discriminative representations at the video level. We take the global token $\boldsymbol{Z}_{global}$ as the query $\boldsymbol{v}_i$, and the resultant tokens produced by the regressor are pooled to obtain the positive sample $\boldsymbol{v}_+$, as shown in Fig. 2. Meanwhile, the hard negative sample $\boldsymbol{v}_-$ in addition with samples from other videos in batch $\mathcal{B}$ are passed to a max-pooling layer, resulting in negative samples. Then, all samples are projected into the latent space and the InfoNCE loss [37] is adopted for training:

$$\mathcal{L}_{\boldsymbol{v}_i} = -\log \frac{\exp\left(\boldsymbol{v}_i^T \boldsymbol{v}_+/\tau\right)}{\exp\left(\boldsymbol{v}_i^T \boldsymbol{v}_-/\tau\right) + \sum_{\boldsymbol{v}_j\in\{\mathcal{B}\cup\boldsymbol{v}_+\}}^{i\neq j} \exp\left(\boldsymbol{v}_i^T \boldsymbol{v}_j/\tau\right)}. \tag{5}$$

Overall, the total loss of our PointCMP is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{NCE}^{local} + \mathcal{L}_{NCE}^{global}, \tag{6}$$

where $\mathcal{L}_{NCE}^{local}$ refers to the InfoNCE loss in the local branch (Eq. 4), and $\mathcal{L}_{NCE}^{global}$ refers to the InfoNCE loss in the global branch (Eq. 5). With both loss terms, our PointCMP can simultaneously learn both local and global information.

4. Experiments

In this section, we first present the datasets and implementation details used in the experiments. Then, we compare our PointCMP to previous methods under four

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

Methods		#Frames				
		4	8	12	16	24
Supervised Learning	MeteorNet [31]	78.11	81.14	86.53	88.21	88.50
	Kinet [67]	79.80	83.84	88.53	91.92	93.27
	PST^2 [54]	81.14	86.53	88.55	89.22	-
	PPTr [55]	80.97	84.02	89.89	90.31	92.33
	P4Transformer [13]	80.13	83.17	87.54	89.56	90.94
	PST-Transformer [14]	81.14	83.97	88.15	91.98	93.73
	PSTNet [15]	81.14	83.50	87.88	89.90	91.20
	PSTNet++ [16]	81.53	83.50	88.15	90.24	92.68
End-to-end Fine-tuning	**PSTNet + PointCMP**	**84.02**	**89.56**	**91.58**	**92.26**	**93.27**
Linear Probing	**PSTNet + PointCMP**	78.11	88.55	90.24	91.92	92.93

widely used protocols, including end-to-end fine-tuning, linear probing, semi-supervised learning, and transfer learning. Finally, we conduct ablation studies to demonstrate the effectiveness of our method.

4.1. Datasets and Implementation Details

Datasets. We conduct experiments on 3D action recognition and 3D gesture recognition tasks. Four benchmark datasets are employed, including NTU-RGBD [43], MSRAction-3D [28], NvGesture [36], and SHREC'17 [11].

- **NTU-RGBD**. The NTU-RGBD dataset consists of 56,880 videos with 60 action categories performed by 40 subjects. Following the cross-subject setting of [43], this dataset is split into 40,320 training videos and 16,560 test videos.
- **MSRAction-3D**. The MSRAction-3D dataset contains 567 videos with 23k frames. It consists of 20 fine-grained action categories performed by 10 subjects. Following [15], this dataset is split into 270 training videos and 297 test videos.
- **NvGesture**. The NvGesture dataset is comprised of 1532 videos with 25 gesture classes. Following [35], this dataset is split into 1050 training videos and 482 test videos.
- **SHREC'17**. The SHREC'17 dataset consists of 2800 videos in 28 gestures. Following [11], 1960 videos are used as the training set and 840 videos are adopted as the test data.

Pre-training Details. During pre-training, 16 frames were sampled as a clip from each point cloud video, with 1024 points being selected for each frame. The frame sampling stride was set to 2 and 1 on NTU-RGBD and MSRAction-3D, respectively. Then, we divided each clip into 4 segments and random scaling was utilized for data augmentation. Our model was pre-trained for 200 epochs with a batch size of 80, and linear warmup was utilized for the first 5 epochs. The initial learning rate was set to 0.0003 with a cosine decay strategy. The spatial search radius was initially set to 0.5/0.1 on NTU-RGBD/MSRAction-3D and the number of neighbors for the ball query was set to 9. The temperature parameter τ was set to 0.01/0.1 in the local/global InfoNCE loss term.

4.2. End-to-end Fine-tuning

We first evaluate our representations by fine-tuning the pre-trained encoder with a linear classifier in a supervised manner. The MSRAction-3D dataset was used for both pre-training and fine-tuning. During fine-tuning, 2048 points were selected for each frame and the pre-trained model was trained for 35 epochs with a batch size of 24. The initial learning rate was set to 0.015 with a cosine decay strategy. Following [15], the initial spatial search radius was set to 0.5 and the number of neighbors for the ball query was set to 9. Quantitative results are presented in Table 1.

As we can see, our PointCMP introduces significant accuracy improvements over the baseline trained in a fully supervised manner. Especially, the accuracy achieved using 8/12 frames is improved from 83.50%/87.88% to 89.56%/91.58%. This shows that our PointCMP can learn beneficial knowledge from point cloud videos in a self-supervised manner, which contributes to higher accuracy after fine-tuning.

4.3. Linear Probing

We then conduct experiments to validate the effectiveness of our PointCMP via linear probing. The MSRAction-3D dataset was used for both pre-training and linear probing. Specifically, the pre-trained encoder is frozen and an additional linear classifier is added for supervised training. The experimental settings are the same as Sec. 4.2.

From Table 1, we can see that the pre-trained encoder using PointCMP outperforms the fully supervised baseline

Table 2. Action recognition accuracy on NTU-RGBD under cross-subject setting.

Methods	Accuracy (%)
Kinet [67]	92.3
P4Transformer [13]	90.2
PST-Transformer [14]	91.0
PSTNet [15]	90.5
PSTNet++ [16]	91.4
PSTNet+PointCMP (50% Semi-supervised)	88.5

Table 3. Action recognition accuracy (%) of transfer learning on MSRAction-3D. Accuracy improvements against the supervised baseline are shown in subscript.

Methods	Input	#Frames	
		8	16
4D MinkNet [10]+ROP [47]	Point+RGB	86.31	-
MeteorNet [31]+ROP [47]	Point+RGB	85.40$_{+4.26}$	-
PSTNet + PointCMP	Point	88.53$_{+5.03}$	91.58$_{+1.68}$

even under the linear probing setting. Our method surpasses the baseline under most frame settings with notable margins (e.g., 88.55%/90.24% vs. 83.50%/87.88% under 8/12 frames). This clearly demonstrates the high quality of the representations learned by PointCMP.

4.4. Semi-supervised Learning

We also conduct experiments to evaluate our PointCMP under the setting of semi-supervised learning. The cross-subject training set of NTU-RGBD was used for pre-training. Specifically, we used only 50% training set of NTU-RGBD to fine-tune the pre-trained encoder in a supervised manner. Following [15], the initial spatial search radius was set to 0.1, the number of neighbors for the ball query was set to 9, and 2048 points were samples for each frame. The model was fine-tuned for 50 epochs with a batch size of 24. The initial learning rate was set to 0.015 with a cosine decay strategy.

Table 2 compares the quantitative results produced by our PointCMP and previous fully supervised approaches. Averaged accuracy over 3 experiments is reported for our method. It can be observed that our PointCMP achieves comparable performance to the fully supervised baseline even with only 50% data (88.5% vs. 90.5%). This further demonstrates the superiority of the representations learned by our PointCMP.

4.5. Transfer Learning

To evaluate the generalization performance of our PointCMP, we conduct experiments by transferring pre-trained encoder to other datasets or tasks. Specifically, the encoder was first pre-trained on the cross-subject training set of NTU-RGBD, and then fine-tuned with an additional MLP head on MSRAction-3D, NvGesture, and SHREC'17.

Table 4. Gesture recognition accuracy (%) of transfer learning on NvGesture (NvG) and SHREC'17 (SHR).

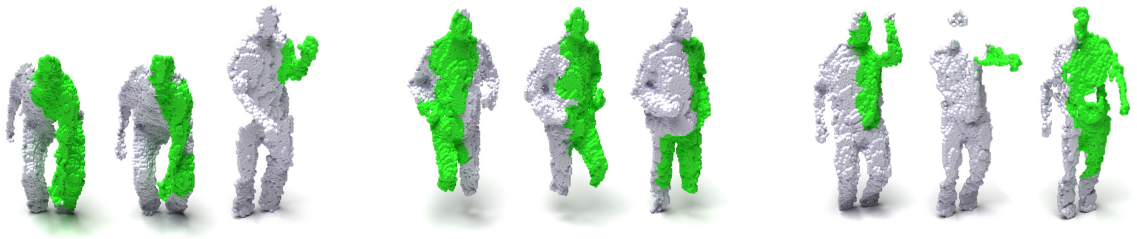
Methods	NvG	SHR
FlickerNet [34]	86.3	-
PLSTM-base [35]	85.9	87.6
PLSTM-early [35]	87.9	93.5
PLSTM-PSS [35]	87.3	93.1
PLSTM-middle [35]	86.9	94.7
PLSTM-late [35]	87.5	93.5
Kinet [67]	89.1	95.2
PSTNet (35 Epochs) [15]	78.9	87.0
PSTNet (100 Epochs) [15]	88.4	92.1
PSTNet + PointCMP (35 Epochs)	84.0	90.8
PSTNet + PointCMP (100 Epochs)	89.2	93.3

Transfer to MSRAction-3D. We first fine-tuned the pre-trained encoder on MSRAction-3D following the experimental settings in Sec. 4.2. We compare our PointCMP with ROP [47] in Table 3. Note that, since the official code for ROP is unavailable, we report its performance on 4D MinkNet [10] and MeteorNet [31] for comparison. Although PSTNet uses only points as input, our PointCMP facilitates this baseline to surpass ROP by over 2% accuracy. In addition, our PointCMP introduces more significant accuracy improvements as compared to ROP (5.03% vs. 4.26%).

Transfer to NvGesture and SHREC'17. The encoder was further transferred from action recognition to gesture recognition through fine-tuning on NvGesture and SHREC'17. Specifically, the pre-trained model was fine-tuned for 100 epochs with a batch size of 16. The initial learning rate was set to 0.01 with a cosine decay strategy. During fine-tuning, 32 frames were utilized with 512/256 points sampled for each frame on NvGesture/SHREC'17. We compare our fine-tuned models to previous supervised state-of-the-art methods in Table 4. As we can see, after fine-tuning for 100 epochs, our PointCMP facilitates PSTNet to produce very competitive accuracy. In addition, our PointCMP also allows for faster convergence such that more significant improvements are achieved after fine-tuning for only 35 epochs (*e.g.*, 78.9% vs. 84.0% on NvGesture). This also shows the superior generalization capability cross different tasks of the representations learned by our PointCMP.

4.6. Ablation Studies

Architecture Design. Our PointCMP employs a two-branch structure to simultaneously extract both local and global information, and adopts the mutual similarity based augmentation module to generate hard samples. To demonstrate the effectiveness of these architecture designs, we developed models A1 and A2 with only local and global



(a) Pick up & Throw (b) Jogging (c) Tennis Serve

Figure 4. Visualization of hard masked samples. Points corresponding to dominant tokens are marked in green.

Table 5. Ablation studies on architecture designs.

	Local Branch	Global Branch	Similarity-based Augmentation	Accuracy (%)
A1	✓			89.22
A2		✓		49.49
A3	✓	✓		89.76
A4 (Ours)	✓	✓	✓	**91.92**

Table 6. Ablation studies on hard masked samples.

	Granularity	Mask	Masking Ratio			
			25%	50%	75%	90%
B1	Token	Random	71.72	71.72	76.77	78.11
B2	Token	Similarity-based	70.03	81.82	84.18	88.55
B3	Segment	Random	90.81	88.15	79.80	-
B4 (Ours)	Segment	Similarity-based	**91.92**	**90.24**	**86.53**	-

Table 7. Ablation studies on hard negative samples.

	Hard Sample	Strategy	Accuracy (%)
C1	×	-	90.52
C2	✓	Random	91.29
C3 (Ours)	✓	Similarity-based	**91.92**

Table 8. Ablation studies on the spatio-temporal matching module.

	Architecture	Matching Module	Accuracy (%)
D1	Local	×	86.20
D2	Local	✓	89.22
D3	Local & Global	×	90.24
D4 (Ours)	Local & Global	✓	**91.92**

branch, respectively. Then, model A3 is introduced by removing the mutual similarity based augmentation module. Quantitative results are presented in Table 5.

As we can see, with only local or global branch, the performance of model A1 and A2 are limited (89.22% and 49.49%). This is because, both local and global information contribute to the recognition of point cloud videos. When these two branches are combined, complementary information can be exploited such that better accuracy is achieved by model A3 (89.76%). However, without the mutual similarity based augmentation module, model A3 still suffers an accuracy drop of 2.16% as compared to A4. This further validates the effectiveness of our mutual similarity based augmentation module.

Hard Masked Samples. The masking strategy contributes to the quality of hard masked samples and plays a critical role in the local branch of our PointCMP. Consequently, we conduct experiments to study different masking strategies and compare their results in Table 6.

As we can see, segment-wise masking strategy significantly outperforms token-wise masking strategy under different masking ratios. As compared to token-wise strategy, segment-wise strategy can better avoid the leakage of information caused by overlapped point patches, which facilitates the network to better exploit local structures in a point cloud video. Moreover, similarity-based masks introduce notable performance gains on segment-wise strategy, with accuracy being improved from 90.81%/88.15% to 91.92%/90.24%. This demonstrates the effectiveness of our hard masked samples.

We further visualize the points corresponding to dominant tokens with high similarity to the global token in Fig. 4. As we can see, tokens corresponding to moving body parts (e.g., arms in Fig. 4(c)) are highlighted, which is consistent with our intuition. This demonstrates the feasibility of our mutual similarity based augmentation to synthesize reasonable hard samples. With these discriminative regions being masked, the encoder is encouraged to leverage more context for mask prediction, with representations of higher quality being learned.

Hard Negative samples. To demonstrate the effectiveness of hard negative samples in the global branch of our PointCMP, model C1 is introduced by excluding hard samples during training. That is, only samples in other videos are employed as negatives. Furthermore, we conduct experiments to study different channel erasing strategies. Quantitative results are presented in Table 7.

It can be observed that model C1 suffers an accuracy drop of 1.40% as compared to C3 when hard negative samples are excluded. Using random channel erasing to generate hard negative samples, model C2 improves C1 with accuracy being increased from 90.52% to 91.29%. With our mutual similarity based augmentation module, hard negative samples of higher quality can be synthesized such that better performance can be achieved. This validates the

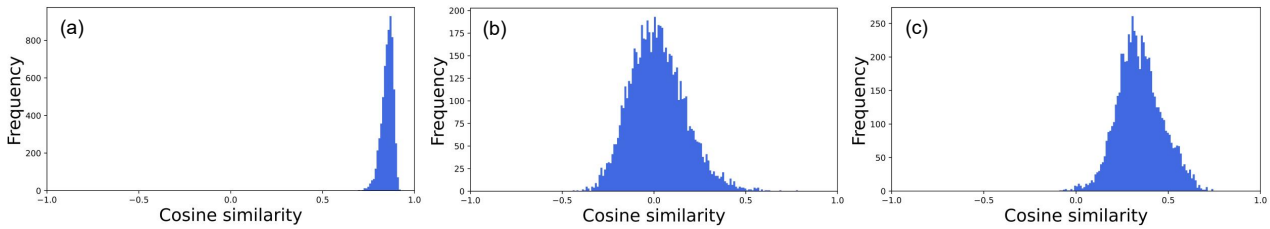


Figure 5. Visualization of cosine similarity histograms between the representations of query samples and their paired (a) positive samples, (b) negative samples, and (c) hard negative samples generated by channel erasing.

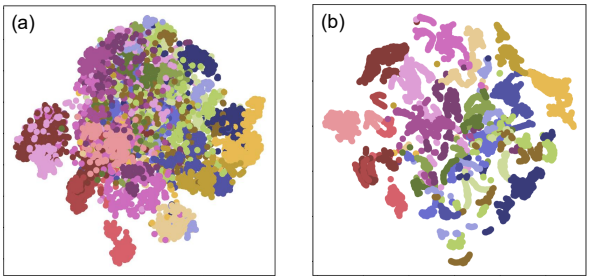


Figure 6. The t-SNE visualization of representation distributions on MSRAction-3D (a) after pre-training only with global contrastive learning and (b) after pre-training using our PointCMP.

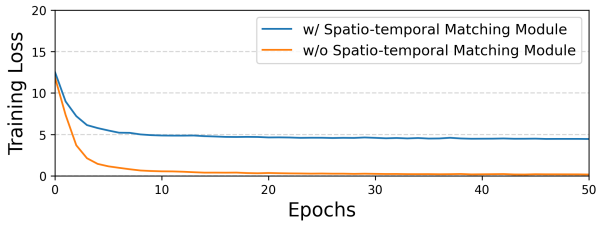


Figure 7. Evolution of local contrastive learning loss during pre-training on MSRAction-3D.

effectiveness of the hard negatives generated by principal channel erasing.

Following [18], we visualize cosine similarity of representations learned from different sample pairs in Fig. 5 to study the importance of our hard negative samples. A model pre-trained on MSRAction-3D without using hard negatives is utilized for analyses. As shown in Fig. 5(a), the similarities of positive pairs are close to 1 with an average value of 0.875. On the contrary, negative pairs are gathered around 0 with an average value of 0.013. For our hard negatives, their average similarity score is increased to 0.315, which means these samples remain difficult for the pre-trained encoder if they are not included for training. This further shows the necessity of our hard negatives.

Spatio-temporal Matching Module. In the local branch of our PointCMP, a spatio-temporal matching module is adopted to conduct local contrastive learning. To study its effectiveness, we first developed model D2 with only the local branch. Then, we introduced model D1 and D3 by removing this matching module from D2 and D4, respectively. Quantitative results are presented in Table 8. As we can see, the spatio-temporal matching module facilitates D4 to produce an accuracy improvement of 1.68% and introduces a more significant improvement of 3.02% to D2. We further visualize the evolution of the local contrastive loss (i.e., $\mathcal{L}_{NCE}^{local}$ in Eq. 6) in Fig. 7. Without the spatio-temporal matching module, the loss decreases rapidly to near 0 and the networks cannot be further optimized. This is because the leakage of location information is leveraged by the network as shortcuts without capturing geometric information. In contrast, our matching module alleviates positional information leakage and increases the hardness of learning to help the network ultimately achieve higher accuracy (Table 8).

Representation Visualization. We further visualize the feature distributions using t-SNE to demonstrate the effectiveness of our PointCMP. With only global branch as many previous methods do, the learned representations have blurred boundaries between different categories with limited discriminative capability, as shown in Fig. 6(a). In contrast, the representations extracted using our PointCMP can better exploit both global and local information with clearer boundaries between different categories, as shown in Fig. 6(b). This clearly demonstrates the high discrimination of the representations learned by our method.

5. Conclusion

In this paper, we develop a self-supervised learning framework termed PointCMP for point cloud videos. Our PointCMP unifies the complementary advantages of contrastive learning and mask prediction paradigms to simultaneously learn both global and local spatio-temporal features at different granularities. To promote the training of PointCMP, we propose a mutual similarity based augmentation module to generate hard masked and negative samples at the feature level. Experiments on benchmark datasets show that our PointCMP achieves state-of-the-art performance on both action and gesture recognition tasks.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of China (No. U20A20185, 61972435), the Guangdong Basic and Applied Basic Research Foundation (2022B1515020103), and Shanghai Science and Technology Innovation Action Plan (21DZ203700).

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] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022. 1, 2

[3] 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. 1, 2

[4] 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. 1, 2

[5] Peihao Chen, Deng Huang, Dongliang He, Xiang Long, Runhao Zeng, Shilei Wen, Mingkui Tan, and Chuang Gan. RSPNet: Relative speed perception for unsupervised video representation learning. In *AAAI*, 2021. 3

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 2

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 1, 2

[9] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3D object detection. In *CVPR*, 2022. 1

[10] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 3, 6

[11] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, Joris Guerry, Bertrand Le Saux, and David Filliat. SHREC'17 Track: 3D hand gesture recognition using a depth and skeletal dataset. In *3DOR*, 2017. 5

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2

[13] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4D transformer networks for spatio-temporal modeling in point cloud videos. In *CVPR*, 2021. 3, 4, 5, 6

[14] 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*, 2022. 1, 3, 5, 6

[15] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. PSTNet: Point spatio-temporal convolution on point cloud sequences. In *ICLR*, 2021. 3, 5, 6

[16] 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*, 2021. 1, 3, 5, 6

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

[18] Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. In *NeurIPS*, 2021. 2, 3, 8

[19] 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

[20] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *NeurIPS*, 2020. 3

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 2

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2

[23] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020. 1, 2

[24] 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

[25] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In *ECCV*, 2022. 2

[26] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *NeurIPS*, 2020. 2, 3

[27] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2

[28] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3D points. In *CVPRW*, 2010. 5

[29] Hanwen Liang, Niamul Quader, Zhixiang Chi, Lizhe Chen, Peng Dai, Juwei Lu, and Yang Wang. Self-supervised spatiotemporal representation learning by exploiting video continuity. In *AAAI*, 2022. 3

[30] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *ECCV*, 2022. 2

[31] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. MeteorNet: Deep learning on dynamic 3D point cloud sequences. In *ICCV*, 2019. 3, 5, 6

[32] Zicheng Liu, Siyuan Li, Di Wu, Zihan Liu, Zhiyuan Chen, Lirong Wu, and Stan Z. Li. AutoMix: Unveiling the power of mixup for stronger classifiers. In *ECCV*, 2022. 2

[33] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and Furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In *CVPR*, 2018. 3

[34] Yuecong Min, Xiujuan Chai, Lei Zhao, and Xilin Chen. FlickerNet: Adaptive 3D gesture recognition from sparse point clouds. In *BMVC*, 2019. 6

[35] Yuecong Min, Yanxiao Zhang, Xiujuan Chai, and Xilin Chen. An efficient PointLSTM for point clouds based gesture recognition. In *CVPR*, 2020. 5, 6

[36] 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

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 4

[38] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. VideoMoCo: Contrastive video representation learning with temporally adversarial examples. In *CVPR*, 2021. 3

[39] Yatian Pang, Wenxiao Wang, Francis E.H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *ECCV*, 2022. 1, 2

[40] Xiangyu Peng, Kai Wang, Zheng Zhu, Mang Wang, and Yang You. Crafting better contrastive views for siamese representation learning. In *CVPR*, 2022. 2

[41] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, 2021. 3

[42] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *ICLR*, 2021. 2, 3

[43] 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

[44] Chen Sun, Arsha Nagrani, Yonglong Tian, and Cordelia Schmid. Composable augmentation encoding for video representation learning. In *ICCV*, 2021. 3

[45] 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

[46] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. 2

[47] 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. 3, 6

[48] Jinpeng Wang, Yuting Gao, Ke Li, Jianguo Hu, Xinyang Jiang, Xiaowei Guo, Rongrong Ji, and Xing Sun. Enhancing unsupervised video representation learning by decoupling the scene and the motion. In *AAAI*, 2021. 3

[49] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. BEVT: BERT pretraining of video transformers. In *CVPR*, 2022. 1, 2

[50] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020. 2

[51] Xiao Wang, Haoqi Fan, Yuandong Tian, Daisuke Kihara, and Xinlei Chen. On the importance of asymmetry for siamese representation learning. In *CVPR*, 2022. 1, 2

[52] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021. 2

[53] 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. 3

[54] Yimin Wei, Hao Liu, Tingting Xie, Qiuhong Ke, and Yulan Guo. Spatial-temporal transformer for 3D point cloud sequences. In *WACV*, 2022. 1, 3, 5

[55] 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, 3, 5

[56] Bohan Wu, Suraj Nair, Roberto Martin-Martin, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *CVPR*, 2021. 3

[57] 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

[58] 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, 2

[59] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, 2018. 4

[60] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. LiDAR-based online 3D video object detection with graph-based message passing and spatiotemporal transformer attention. In *CVPR*, 2020. 3

[61] Quanzeng You and Hao Jiang. Action4D: Online action recognition in the crowd and clutter. In *CVPR*, 2019. 3

[62] 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 *CVPR*, 2022. 1, 2, 3

[63] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021. 1, 2

[64] 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

[65] Zehua Zhang and David Crandall. Hierarchically decoupled spatial-temporal contrast for self-supervised video representation learning. In *WACV*, 2022. 3

[66] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3D features on any point-cloud. In *ICCV*, 2021. 2

[67] 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. 3, 5, 6

[68] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image BERT pre-training with online tokenizer. In *ICLR*, 2021. 2