Masked Motion Prediction with Semantic Contrast for Point Cloud Sequence Learning

Yuehui Han[1], Can Xu[1], Rui Xu[1], Jianjun Qian[1], and Jin Xie[2,3]⋆

[1] PCA Lab, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
[2] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[3] School of Intelligence Science and Technology, Nanjing University, Suzhou, China
{hanyh, 220106011137, xu_ray, csjqian, csjxie}@njust.edu.cn

Abstract. Self-supervised representation learning on point cloud sequences is a challenging task due to the complex spatio-temporal structure. Most recent attempts aim to train the point cloud sequences representation model by reconstructing the point coordinates or designing frame-level contrastive learning. However, these methods do not effectively explore the information of temporal dimension and global semantics, which are the very important components in point cloud sequences. To this end, in this paper, we propose a novel masked motion prediction and semantic contrast (M2PSC) based self-supervised representation learning framework for point cloud sequences. Specifically, it aims to learn a representation model by integrating three pretext tasks into the same masked autoencoder framework. First, motion trajectory prediction, which can enhance the model's ability to understand dynamic information in point cloud sequences. Second, semantic contrast, which can guide the model to better explore the global semantics of point cloud sequences. Third, appearance reconstruction, which can help capture the appearance information of point cloud sequences. In this way, our method can force the model to simultaneously encode spatial and temporal structure in the point cloud sequences. Experimental results on four benchmark datasets demonstrate the effectiveness of our method. Source code is available at `https://github.com/yh-han/M2PSC.git`.

Keywords: Self-Supervised Learning · Motion Trajectory Prediction · Semantic Contrast · Point Cloud Sequences

1 Introduction

The understanding of point cloud sequences seriously affects the interaction ability between intelligent agents and environments, which is the core technology in many applications (e.g., autonomous vehicles, robots and virtual reality). In recent years, research on point cloud sequences has received intensive attention, and many point cloud sequences representation learning methods [7–10]

⋆ Corresponding authors

are emerged. Although these methods achieve outstanding performance, they are inseparable from annotated labels. As annotating point cloud sequences is time-consuming and labor-intensive, the potential of these supervised methods is limited. To alleviate this problem, research on self-supervised point cloud sequences representation learning has attracted increasing attention.

By designing pretext task, self-supervised representation learning can effectively learn discriminative sample features without annotated labels. It has outstanding performance in many fields such as images [2,15,16], videos [25,31,33], graphs [12, 14, 40] and static point clouds [13, 24, 43]. Inspired by its success in other fields, some self-supervised representation learning methods for point cloud sequences [27–30, 44] are proposed. Most methods design self-supervised point cloud sequences representation learning algorithm based on the ideas of frame temporal order prediction [35], frame-level contrastive learning [29], or reconstruction [27] to extract sample features. However, point cloud sequences have complex structures that contain information in both spatial and temporal dimensions. And these methods do not make sufficient use of the information of temporal dimension and global semantics, which may potentially affect the quality of the learned features of point cloud sequences. Therefore, in order to better learn point cloud sequences features, it is necessary to design effective pretext tasks based on more detailed spatial and temporal information exploration.

Motion trajectory is the special information unique to dynamic data (e.g., point cloud sequences, videos), which can accurately describe position or state changes over time. In self-supervised video representation learning, motion trajectory based methods have been well studied. Many methods design pretext tasks based on motion trajectory clustering [32], motion trajectory prediction [3,31] and motion trajectory tracking [34,36], and achieve excellent performance. In comparison, affected by complex data structures, motion trajectory based self-supervised point cloud sequences representation learning is understudied.

Based on the above analysis, in this paper, we propose a novel motion trajectory prediction based masked autoencoder framework for self-supervised point cloud sequences representation learning. Different from the current methods [27,29,35] that roughly use the temporal dimension information, our method can more precisely utilize it by exploring motion trajectory in point cloud sequences. Moreover, we also introduce self-supervised constraints from the perspective of global semantics and appearance. Specifically, our method consists of three self-supervised pretext tasks: motion trajectory prediction, semantic contrast and appearance reconstruction. We first encode the point cloud sequences employing a masked autoencoder framework. And based on the encoded features of the visible parts, we develop a motion trajectory decoder to predict the motion trajectories of points in the point cloud sequences. This task can make more precise use of the temporal dimension information and help the model better understand the actions in the point cloud sequences. We then construct global semantic contrast based on the visible and masked parts, which are natural contrastive samples. This task can guide the model to better learn the global semantics of point cloud sequences. Finally, we also use the point decoder to predict the coordinate of masked points, which can guide the model to learn the local structures of point cloud sequences. It should be noted that the motion trajectories as supervision signals are extracted using the pre-trained point cloud correspondence model CorrNet3D [41]. We evaluate our method on four point cloud sequences benchmark datasets, i.e., MSRAction-3D [18], NTU-RGBD [26], SHREC'17 [5] and NvGesture [23]. And experimental results show that our method achieves excellent performance, which demonstrate the effectiveness of our method.

Although there already exist method that based on motion information learning, i.e., MaST-Pre [27], it is very different from our method. MaST-Pre uses temporal cardinality difference to approximate motion changes in point cloud sequences, which only vaguely uses temporal dimension information. In contrast, our method utilizes motion trajectory prediction as the pretext task, which can provide more accurate temporal dimension information guidance for model learning. To summarize, the main contributions include:

- We propose a novel motion trajectory prediction and semantic contrast based masked autoencoder framework for self-supervised point cloud sequences representation learning.
- We develop a motion trajectory decode module, which can predict the motion trajectory of points in the point cloud sequences.
- We construct global semantic contrast based on the visible and masked parts, which can help the model better explore the semantics of point cloud sequences.
- We evaluate our method on four benchmark datasets, where M2PSC achieves excellent performance.

2 Related Work

2.1 Supervised Learning on Point Cloud Sequences

Affected by the complex spatio-temporal structure, the learning of point cloud sequences is a challenging task. Currently, supervised learning is still the mainstream in point cloud sequence learning, which focuses on designing effective model structures to capture spatial and temporal information in point cloud sequences. Deep Learning on Dynamic 3D Point Cloud Sequences (MeteorNet) [20] proposes the Meteor module to aggregate the feature representation of points in the spatio-temporal neighborhood, and gradually expand the aggregation range through module stacking. Point Spatio-temporal Convolution (PSTNet) [9] proposes to decouple the space and time in point cloud sequences and employs spatial convolution and temporal convolution to model the local structure of points and the dynamics of the spatial regions, respectively. Different from the spatio-temporal decoupling operation in PSTNet, Point 4D Transformer network (P4Transformer) [7] proposes to directly aggregate information in spatio-temporal neighborhood. It first develops a point 4D convolution to encode the spatio-temporal local structures, and then employs the transformer to capture the global appearance and motion information across the entire point cloud sequences. Also based on transformer, Point Primitive Transformer (PPTr) [39] proposes to leverage the primitive plane as mid-level representation to capture the long-term spatial-temporal context in point cloud sequences. Following the idea of introducing traditional techniques, Kinematics-inspired Neural Networ (Kinet) [45] proposes to generalize the kinematic concept of ST-surfaces to the feature space to effectively learn spatio-temporal feature representations of point cloud sequences. To better preserve the spatio-temporal structure, Point Spatio-Temporal Transformer (PST-Transformer) [8] proposes to adaptively searche related or similar points across the entire point cloud sequence by performing self-attention on point features. Although these supervised learning methods achieve outstanding performance, the over-reliance on manually annotated labels limits their potential in point cloud sequence understanding. Therefore, in order to alleviate the dependence on manually annotated labels, self-supervised point cloud sequence representation learning is receiving increasing attention.

2.2 Self-supervised Learning on Point Cloud Sequences

Self-supervised point cloud sequence representation learning aims to employ pretext tasks to guide the learning of the model, thereby getting rid of the dependence on manually annotated labels. Recurrent Order Prediction (ROP) [35] proposes to learn the 4D spatio-temporal features by predicting the temporal order of sampled and shuffled point cloud clips. Inspired by discrimination and generation tasks in self-supervised learning, Sheng et al. [29] proposes the Contrastive Prediction and Reconstruction (CPR) based self-supervised point cloud sequence representation learning method. It employs reconstruction and local contrast to enhance the prediction ability of subsequent segments, and utilizes global contrast to improve the coding ability of multi-frame point cloud sequences. Rather than utilizing point cloud frames as contrastive samples, Contrastive Mask Prediction (PointCMP) [28] proposes to construct contrastive samples in the feature space. It develops a mutual similarity based augmentation module to generate hard masked samples and negative samples by masking dominant tokens and principal channels. In order to capture fine-grained semantics, different from the clip or frame based method, Point Contrastive Prediction with Semantic Clustering (PointCPSC) [30] proposes a point level based contrastive learning framework. Inspired by visual mask prediction, Masked Spatio-Temporal Structure Prediction (MaST-Pre) [27] proposes a masked autoencoder framework for point cloud sequences based on point reconstruction and temporal cardinality difference prediction. For better learning 4D representations, Complete-to-Partial 4D Distillation (C2P) [44] proposes a teacher-student knowledge distillation framework to guide the learning of the model.

2.3 Mask Prediction for Vision

Mask prediction has achieved great success in visual self-supervised representation learning [1, 15, 24, 33, 43], and the core idea of which is to reconstruct the masked signals. In the field of self-supervised image representation learning, Masked Autoencoders (MAE) [15] takes pixel reconstruction of masked image patches as the pretext task to guide the learning of model. As for self-supervised video representation learning, Masked Autoencoders As Spatiotemporal Learners (MAE-ST) [11] and Masked Autoencoders for Video (VideoMAE) [33] introduce the ideas of MAE into video learning. They use the pixel reconstruction of masked video tubes as the pretext task to learn the spatio-temporal feature representations in videos. Different from pixel reconstruction in MAE-ST and VideoMAE, Masked Feature Prediction (MaskFeat) [38] proposes to predict the Histograms of Oriented Gradients (HOG) features of the masked video regions. Encouraged by the success of MAE in image and video, Masked Autoencoders for Point Cloud (Point-MAE) [24], Multi-scale Masked Autoencoders (Point-M2AE) [42] and Discriminative Mask Pre-training Transformer framework (MaskPoint) [19] propose point coordinate reconstruction or point discrimination based framework for self-supervised 3D point cloud representation learning.

2.4 Motion Trajectory for Vision

Motion trajectories can describe changes in position or state over temporal dimension, which is the important source of supervision signals in self-supervised video representation learning [3, 31, 32, 34, 36]. Tokmakov et al. [32] propose a dense trajectory clustering based unsupervised video representation learning framework, which take clusters formed in improved dense trajectories space as initial supervision signals for video clustering. In order to better explore temporal clues, Masked Motion Encoding (MME) [31] proposes a motion trajectory reconstruction based method for self-supervised video representation learning. It first generates the trajectories of points in multiple frames, and then forces the model to predict trajectories based on the learning of visible patches. Based on tracking the movement of video objects, Chen et al [3] propose a unified framework to ground physical objects and events from dynamic scenes and language.

3 Method

In this section, we present our masked motion prediction and semantic contrast based self-supervised point cloud sequence representation learning method. As shown in Fig. 1, given a point cloud sequence, we first extract the motion trajectories of the points between multiple point cloud frames. We then employ the masked autoencoder based framework to encode the feature representations of point cloud sequence. At the same time, we add a motion trajectory decoder to predict the motion trajectories of the points, and use the existing visible and masked parts to construct contrastive samples. Finally, we utilize three pretext tasks to guide the learning of the model, i.e., motion trajectory prediction, global semantic contrast and appearance reconstruction.

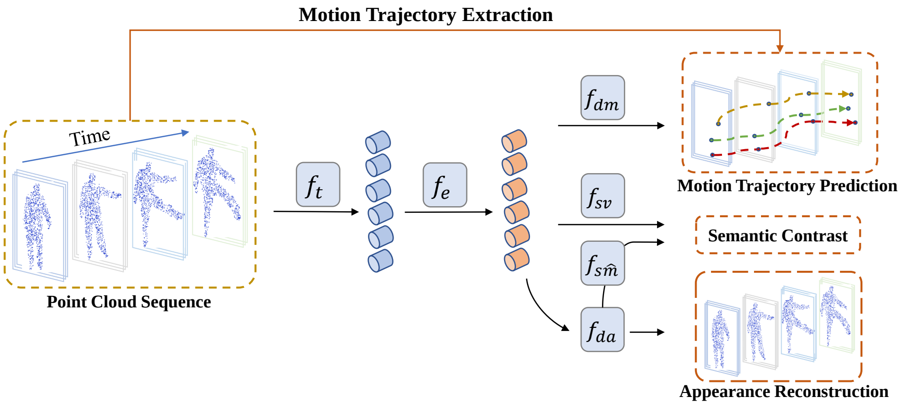

Fig. 1: The architecture of our method. We integrate three pretext tasks into the same masked autoencoder framework, i.e., motion trajectory prediction, semantic contrast, and appearance reconstruction. Note that we employ CorrNet3D to extract the motion trajectories of the point cloud sequences. f_t represents the obtaining, masking and embedding operations of the point tubes.

3.1 Motion Trajectory for M2PSC

Before introducing the proposed method, we first provide some preliminary concepts. Let $\boldsymbol{P} \in \mathbb{R}^{L\times N\times 3}$ represent a point cloud sequence, where L denotes the sequence length and N represents the point number in each frame. The goal of self-supervised point cloud sequence representation learning is to pre-train a point cloud sequence encoder based on pretext tasks without using manually annotated labels. And then the pre-trained encoder is transferred to downstream tasks to improve the performance of the model.

Motion Trajectory Extraction: Motion trajectory is a kind of data that accurately describes position or state changes over temporal dimension, which is often treated as supervision signals in self-supervised video representation learning methods [3, 31, 32, 34, 36]. Inspired by this, we introduce motion trajectories prediction for self-supervised point cloud sequence representation learning to alleviate the problem of insufficient utilization of temporal dimension information in current methods. However, due to the huge difference between the video and the point cloud sequence, we cannot directly use the method in the video to extract the motion trajectory of the point cloud sequence. Therefore, we need a motion trajectory extraction method adapted to point cloud sequences. To this end, we consider resorting to CorrNet3D [41], which is used for non-rigid shape correspondence of 3D human point clouds. For more information about CorrNet3D, please refer to the supplementary materials.

Since CorrNet3D is used for shape correspondence between two point clouds, it cannot be directly used to extract the motion trajectories of the point cloud sequence. Therefore, we divide the motion trajectory extraction of the point cloud sequence into two steps: extraction of the corresponding points between

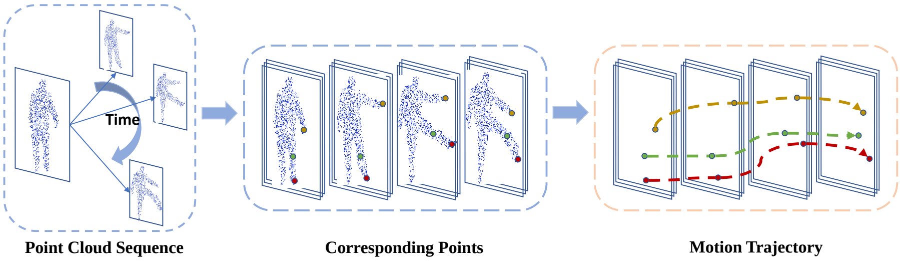

Fig. 2: Example of motion trajectory extraction of the point cloud sequence.

different point cloud frames and extraction of motion trajectories in the point cloud sequence. As shown in Fig. 2, specifically, we first calculate the point correspondence between the first point cloud frame and subsequent frames based on CorrNet3D, respectively. From this, we obtain the location information of the corresponding points in different point cloud frames. Take point i in the first point cloud frame as an example, the location information of the relevant points in point cloud sequence can be expressed as:

$$\hat{\boldsymbol{T}}_i = \{\boldsymbol{p}_i^1, \boldsymbol{p}_i^2, ..., \boldsymbol{p}_i^L\} \tag{1}$$

where $\boldsymbol{p}_i^L = (x_L, y_L, z_L)$ represent the coordinates of the corresponding point in point cloud frame L.

Based on the location information of the corresponding points in different frames, we then calculate the related location movement of points between two adjacent frames. Take frame t and $t+1$ as an example, we can define it as:

$$\triangle \boldsymbol{p}_i^t = \boldsymbol{p}_i^{t+1} - \boldsymbol{p}_i^t \tag{2}$$

In this way, we can obtain the motion trajectories of point i in the point cloud sequence.

$$\boldsymbol{T}_i = (\triangle \boldsymbol{p}_i^1, \triangle \boldsymbol{p}_i^2, ..., \triangle \boldsymbol{p}_i^{L-1}) \tag{3}$$

where $(\cdot,\cdot)$ denotes the concatenation operation, $\boldsymbol{T}_i \in \mathbb{R}^C$, C represents the dimension of the motion trajectories and $C = 3 \times (L-1)$. Finally, we can obtain the motion trajectories of the points in the point cloud sequence.

$$\boldsymbol{T} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, ..., \boldsymbol{T}_N\} \tag{4}$$

where $\boldsymbol{T} \in \mathbb{R}^{N \times C}$. It should be noted that motion trajectory extraction only occurs in the pre-training stage. Compared with MaST-Pre [27] that uses temporal cardinality difference to estimate the number changes of points in the point tubes to simulate motion information, our method can more accurately characterize the motion state in point cloud sequences.

Motion Trajectory Prediction: After obtaining the motion trajectory of the point cloud sequence, we integrate it into the masked autoencoder framework to improve the model's ability to explore the temporal information of the point cloud sequence. Following MaST-Pre [27], we first divide the input point cloud sequence into basic units, i.e., point tubes. Specifically, we sample K keypoints $\hat{\boldsymbol{p}}$ from point cloud sequence $\boldsymbol{P}$ using farthest point sampling (FPS), where K represents the number of keypoints. For keypoint $\hat{\boldsymbol{p}}_i$, the corresponding point tube is defined as $\boldsymbol{Tube}_{\hat{\boldsymbol{p}}_i} = \{\boldsymbol{p}|\boldsymbol{p} \in \boldsymbol{P}, \mathcal{D}_s(\boldsymbol{p}, \hat{\boldsymbol{p}}_i) < r, \mathcal{D}_t(\boldsymbol{p}, \hat{\boldsymbol{p}}_i) < \frac{l}{2}\}$, where $\boldsymbol{p}$ represents the point in the input point cloud sequence, $\mathcal{D}_s$ represents the Euclidean distance, $\mathcal{D}_t$ is the difference in frame timestamps of two points, r represents the spatial neighborhood radius, l denotes the number of frames in a point tube. Next, we randomly sample n points in each spatial neighborhood. Then, we divide a point cloud sequence into K point tubes, and each point tube includes $l \times n$ points.

After obtaining the point tubes, we use random masking to divide the point tubes into visible and masked parts, and follow the baseline [7, 27] to obtain the embedded representations of the visible point tubes. Then, we utilize the encoder $f_e(\cdot)$ of the masked autoencoder to obtain the feature representations of visible point tubes, i.e., $\boldsymbol{Z}_v$. It should be noted that during pre-training only the visible point tubes with spatio-temporal positional embeddings pass through the encoder, while in downstream tasks it is the entire point cloud sequence. (For more detailed information, please refer to the supplementary materials).

In order to construct the task of motion trajectory prediction, we propose a motion trajectory decoding module, which is similar to the encoder but a lightweight vanilla Transformer. Then, we feed the feature representations of visible point tubes as well as learnable tokens corresponding to the points into the motion trajectory decoder to predict the motion trajectories of the points. This can be defined as:

$$\boldsymbol{T}^{pre} = f_{dm}(\boldsymbol{Z}_v, \boldsymbol{t}_p) \tag{5}$$

where $\boldsymbol{T}^{pre} \in \mathbb{R}^{N \times C}$ represents the predicted motion trajectories of the points, $f_{dm}(\cdot)$ denotes motion trajectory decoder, $\boldsymbol{t}_p \in \mathbb{R}^{N \times F}$ represents the learnable tokens corresponding to the points in the first point cloud frame, F represents the feature dimension of the tokens. Note that both $\boldsymbol{Z}_v$ and $\boldsymbol{t}_p$ are added with the corresponding spatio-temporal positional embeddings.

Finally, we can define the motion trajectory prediction loss as:

$$\mathcal{L}_m = \frac{1}{N} \sum_{i=1}^{N} \left\{ \sum_{j=1}^{C} (\boldsymbol{T}_{ij}^{pre} - \boldsymbol{T}_{ij})^2 \right\} \tag{6}$$

By using accurate motion trajectories, our pretext task can guide the model to more efficiently explore the temporal dimension information of point cloud sequences, thereby improving the performance of downstream tasks.

3.2 Semantic Contrast and Appearance Reconstruction

As a high-level feature representation, semantics plays an important role in point cloud sequence understanding. Therefore, in order to enhance the model's ability to explore the semantics of point cloud sequences, we integrate contrastive learning into the masked autoencoder framework and propose the global semantic contrast based on sample masking. Different from the current methods that usually employ continuous frames or complete feature representations of point cloud sequence to construct contrastive samples, the proposed global semantic contrast treats existing visible and masked point tubes as contrastive samples. It increases the difficulty of contrastive learning by constructing more differentiated contrastive samples, thereby forcing the model to explore more effective information from point cloud sequences.

Semantic Contrast: In masked autoencoder framework for self-supervised learning of point cloud sequences, visual and masked point tubes are natural contrastive samples. Therefore, we directly treat visible and masked point tubes from the same point cloud sequence as positive samples, and the others as negative samples. Specifically, based on the feature representations of the visible point tubes, we first develop a semantic prediction module to predict the global semantics of point cloud sequences. This can be defined as:

$$\boldsymbol{Z}^{gv} = f_{sv}(\boldsymbol{Z}_v) \tag{7}$$

where $f_{sv}(\cdot)$ represents semantic prediction module, which consists of a two-layer multi-layer perceptron and global pooling operation. As for the masked point tubes, following MaST-Pre [27], we first use the decoder to predict its feature representation. This can be defined as:

$$\boldsymbol{Z}_{\hat{m}} = f_{da}(\boldsymbol{Z}_v, \boldsymbol{t}_{\hat{m}}) \tag{8}$$

where $\boldsymbol{Z}_{\hat{m}}$ denotes the feature representations of the masked point tubes, $\boldsymbol{t}_{\hat{m}}$ represents the learnable tokens corresponding to the masked point tubes. Note that $\boldsymbol{Z}_v$ and $\boldsymbol{t}_{\hat{m}}$ are added with the corresponding spatio-temporal positional embeddings. Then, we employ the developed semantic prediction module to predict the global semantics of point cloud sequence.

$$\boldsymbol{Z}^{g\hat{m}} = f_{s\hat{m}}(\boldsymbol{Z}_{\hat{m}}) \tag{9}$$

where $f_{s\hat{m}}(\cdot)$ represents semantic prediction module, which has the same structure as $f_{sv}(\cdot)$ but different parameters. Take a pair of positive samples $\boldsymbol{Z}_i^{gv}$ and $\boldsymbol{Z}_i^{g\hat{m}}$ as an example, we can define the contrastive loss as:

$$l(\boldsymbol{Z}_i^{gv}, \boldsymbol{z}_i^{g\hat{m}}) = -log \frac{exp(s(\boldsymbol{Z}_i^{gv}, \boldsymbol{Z}_i^{g\hat{m}})/\tau)}{\sum\limits_{j=1, j\neq i}^{B} exp(s(\boldsymbol{Z}_i^{gv}, \boldsymbol{Z}_j^{gv})/\tau) + \sum\limits_{j=1}^{B} exp(s(\boldsymbol{Z}_i^{gv}, \boldsymbol{Z}_j^{g\hat{m}})/\tau)} \tag{10}$$

where B is the minibatch size, $s(\cdot)$ denotes the cosine similarity function, τ is a temperature parameter, we set it to 0.01. Finally, we can get the global semantic contrast loss between $\boldsymbol{Z}^{gv}$ and $\boldsymbol{Z}^{g\hat{m}}$.

$$\mathcal{L}_s = \frac{1}{2B} \sum_{i=1}^{B} [l(\boldsymbol{Z}_i^{gv}, \boldsymbol{z}_i^{g\hat{m}}) + l(\boldsymbol{Z}_i^{g\hat{m}}, \boldsymbol{z}_i^{gv})] \tag{11}$$

By using visible and masked point tubes to construct global semantic contrast, it can enhance the guidance of the pretext task and enhance the model's ability to explore the global semantics of point cloud sequences.

Appearance Reconstruction: In addition to motion trajectory prediction and global semantic contrast, we also introduce the appearance reconstruction task. Following MaST-Pre [27], based on the predicted feature representations of the masked point tubes, we employ prediction heads to obtain the predicted point coordinates $\boldsymbol{P}^{pre}$ of masked point tubes. Next, the l_2 Chamfer Distance [6] is introduced to calculate the reconstruction loss between a prediction $\boldsymbol{P}^{pre}$ and the ground truth $\boldsymbol{P}^{gt}$. This can be defined as:

$$l_a = \frac{1}{l}\sum_{i=1}^{l}\left\{\frac{1}{|\boldsymbol{P}_i^{pre}|}\sum_{a\in\boldsymbol{P}_i^{pre}}\underset{b\in\boldsymbol{P}_i^{gt}}{min}\|a-b\|_2^2+\frac{1}{|\boldsymbol{P}_i^{gt}|}\sum_{b\in\boldsymbol{P}_i^{gt}}\underset{a\in\boldsymbol{P}_i^{pre}}{min}\|b-a\|_2^2\right\} \tag{12}$$

where $\boldsymbol{P}^{pre}\in\mathbb{R}^{l\times n\times 3}$, $\boldsymbol{P}^{gt}\in\mathbb{R}^{l\times n\times 3}$, $\boldsymbol{P}^{gt}\in\boldsymbol{P}$. Then, the final reconstruction reconstruction loss can be defined as:

$$\mathcal{L}_a = \frac{1}{K_{\hat{m}}}\sum_{i=1}^{K_{\hat{m}}} l_a^i \tag{13}$$

where $K_{\hat{m}}$ is the number of reconstructed point tubes. Overall, the total loss of our method is defined as:

$$\mathcal{L} = \mathcal{L}_m + \mathcal{L}_s + \mathcal{L}_a \tag{14}$$

Based on the above three learning objectives, our method can guide the model to simultaneously encode temporal and spatial cues to better understand point cloud sequences.

4 Experiments

In this section, extensive experiments are conducted on four benchmark point cloud sequence datasets, i.e., MSRAction-3D [18], NTU-RGBD [26], NvGesture [23] and SHREC'17 [5]. Following MaST-Pre [27], we employ end-to-end fine-tuning, semi-supervised learning and transfer learning to evaluate the performance of our method on action recognition and gesture recognition downstream tasks, respectively. Besides, we conduct ablation studies to verify the effectiveness of our proposed method.

4.1 Implementation Details of Pre-training

All experiments in our method are implemented using PyTorch [4], and during pre-training, we sample 24 frames from each point cloud sequence and sample 1024 points from each frame. Following [27], we set the frame sampling stride of MSRAction-3D and NTU-RGBD datasets to 1 and 2, respectively. As for the spatial downsampling rate and the temporal downsampling rate, we set it to 32 and 2, respectively. Meanwhile, we set the temporal kernel size l of each point tube to 3. The spatial neighborhood radius r is set to 0.3 and 0.1 on MSRAction-3D and NTU-RGBD datasets respectively, and the number of points in each spatial neighborhood is set to 32. Besides, the masking ratio to divide the visible and masked point tubes is set to 0.75.

Following [27], we employ P4Transformer [7] as the encoder, and the number of layers of vanilla Transformers corresponding to the MSRAction-3D and NTU-RGBD datasets are set to 5 and 10, respectively. Both the motion trajectory decoder and the decoder for obtaining the feature representations of the masked point tubes are the 4-layer transformer. Besides, same as MaST-Pre, we utilize PST-Transformer [8] as the encoder on MSRAction-3D dataset. During pre-training, the number of iterations is set to 200 and linear warmup is used for the first 10 epochs. We use the AdamW optimizer [17] and initial learning rate of 0.001 to optimize model parameters. And cosine decay strategy is also utilized.

4.2 Action Recognition

We evaluate the performance of the pre-trained model based on the action recognition task on the MSRAction-3D and NTU-RGBD datasets. Specifically, we first discard the part of the pre-trained network after the encoder and replace it with the action recognition classifier. Then, we train the network using two different supervised approaches, i.e., end-to-end fine-tuning and semi-supervised learning.

End-to-end Fine-tuning: We conduct experiments in an end-to-end fine-tuning setting on the MSRAction-3D and NTU-RGBD datasets, respectively. And in experiment, the same dataset is used for pre-training and fine-tuning.

MSRAction-3D dataset. During the fine-tuning process, we sample 24 frames from each point cloud sequence and sample 2048 points from each frame. Following [27], we set the spatial search radius r to 0.7. Besides, we use the AdamW optimizer and the initial learning rate of 0.0005 to optimize the model parameters. And cosine decay strategy is also utilized. Table. 1 lists the performance of action recognition with different methods on MSRAction-3D dataset. It should be noted that considering the differences in the backbone network of different methods, for a fairer comparison we mainly compare with methods based on the same backbone network (i.e., P4Transformer [7] and PST-Transformer [8]). As shown in Table. 1, our method achieves the best action recognition results. Compared with the supervised methods P4Transformer and PST-Transformer, the results of both MaST-Pre and our method are improved. This demonstrates the effectiveness of the self-supervised point cloud sequence representation learning method based on the masked autoencoder framework. Besides, although both are based on the masked autoencoder framework, our method has obvious performance advantages compared to MaST-Pre. In the case of using P4Transformer, the action recognition results of our method even have a performance gain of 1.74% compared to MaST-Pre. This demonstrates that the pretext tasks we proposed based on motion trajectory prediction and global semantic contrast can help the model more effectively capture the spatio-temporal and semantic

Table 1: Performance comparison of action recognition with different methods on MSRAction-3D dataset.

	Algorithm	Accuracy (%)
Supervised Learning	MeteorNet [20]	88.50
	PSTNet [9]	91.20
	PSTNet++ [10]	92.68
	Kinet [45]	93.27
	PPTr [39]	92.33
	P4Transformer [7]	90.94
	PST-Transformer [8]	93.73
End-to-end Fine-tuning	PSTNet + PointCPSC [30]	92.68
	PSTNet + CPR [29]	93.03
	PSTNet + PointCMP [28]	93.27
	P4Transformer + MaST-Pre [27]	91.29
	PST-Transformer + MaST-Pre [27]	94.08
	P4Transformer + M2PSC (ours)	**93.03**
	PST-Transformer + M2PSC (ours)	**94.84**

information in point cloud sequences, thereby improving the performance of the pre-trained model in downstream tasks.

NTU-RGBD dataset. During the fine-tuning process, the relevant settings are the same as pre-training, except that the number of iterations is 20 and the initial learning rate is set to 0.0005. Table. 2 lists the performance of action recognition with different methods on NTU-RGBD dataset, which includes seven supervised methods and four self-supervised methods. Also for a fairer comparison, we mainly focus on methods based on the same backbone network (i.e., P4Transformer [7]). As shown in Table. 2, under the end-to-end fine-tuning setting, our method achieves better action recognition results compared with MaST-Pre. This further demonstrates that the pretext task based on motion trajectory prediction can more efficiently explore motion information in point cloud sequences, and semantic contrast can provide effective guidance for the model to learn global semantic information.

Semi-supervised Learning: We also conduct experiments on the NTU-RGBD dataset to verify the performance of the pre-trained model in the semi-supervised learning setting. It should be noted that the cross-subject training set of NTU-RGBD dataset is used in the pre-training process, while only half of the training set is used in semi-supervised learning setting. And other settings for semi-supervised learning experiment are the same as end-to-end fine-tuning on the NTU-RGBD dataset. As shown in Table. 2, even using only limited annotated data, our method still achieves competitive performance. This demonstrates that our method has excellent point cloud sequence understanding capabilities.

4.3 Gesture Recognition

In addition to the action recognition task, we also conduct experiments on the NvGesture and SHREC'17 datasets to verify the performance of the pre-trained model in the gesture recognition task based on the transfer learning setting.

Table 2: Performance comparison of action recognition with different methods on NTU-RGBD dataset.

Algorithm	Accuracy (%)
3DV-Motion [37]	84.5
3DV-PointNet++ [37]	88.8
PSTNet [9]	90.5
PSTNet++ [10]	91.4
Kinet [45]	92.3
P4Transformer [7]	90.2
PST-Transformer [8]	91.0
PSTNet + PointCPSC [30](50% Semi-supervised)	88.0
PSTNet + PointCMP [28] (50% Semi-supervised)	88.5
PSTNet + CPR [29](End-to-end Fine-tuning)	91.0
P4Transformer + MaST-Pre [27](50% Semi-supervised)	87.8
P4Transformer + MaST-Pre [27](End-to-end Fine-tuning)	90.8
P4Transformer + M2PSC (ours)(50% Semi-supervised)	**88.7**
P4Transformer + M2PSC (ours)(End-to-end Fine-tuning)	**91.3**

Table 3: Performance comparison of gesture recognition with different methods on NvGesture (NvG) and SHREC'17 (SHR) datasets.

Algorithm	NvG	SHR
FlickerNet [21]	86.3	-
PLSTM [22]	85.9	87.6
PLSTM-PSS [22]	87.3	93.1
Kinet [45]	89.1	95.2
P4Transformer (30 Epochs) [7]	84.8	87.5
P4Transformer (50 Epochs) [7]	87.7	91.2
P4Transformer + MaST-Pre (30 Epochs) [27]	87.6	90.2
P4Transformer + MaST-Pre (50 Epochs) [27]	89.3	92.4
P4Transformer + M2PSC (ours)(30 Epochs)	**88.0**	**90.9**
P4Transformer + M2PSC (ours)(50 Epochs)	**89.6**	**92.8**

Transfer Learning: By applying the pre-trained model to the task of other datasets, transfer learning can be used to evaluate the generalization ability of our method. Specifically, following the settings in Section. 4.1, we first pre-train the model on the NTU-RGBD dataset based on the proposed method, then we discard the part after the encoder and replace it with the gesture recognition classifier. Finally, we train the network under the end-to-end fine-tuning setting on the NvGesture and SHREC'17 datasets, respectively.

During the fine-tuning process, we use the AdamW optimizer to optimize the model parameters, and the initial learning rate is set to 0.001 and 0.0005 for the NvGesture and SHREC'17 datasets, respectively. And cosine decay strategy is also utilized. Table. 3 lists the performance of gesture recognition with different methods on NvGesture and SHREC'17 datasets, which includes five supervised methods and one self-supervised methods. As shown in Table. 3, even in the transfer learning setting, our method achieves significant performance improvements. This demonstrates that our method has excellent generalization ability.

4.4 Ablation Studies

Effectiveness of Different Pretext Tasks: To further demonstrate the effectiveness of different pretext tasks, we construct ablation studies of different pretext tasks on the MSRAction-3D dataset. Table. 4 lists the results when the number of pretext tasks is different. We can see that there are differences in the effectiveness of different pretext tasks, and the performance based on motion trajectory prediction is significantly better than other tasks. This is because the pretext task based on motion trajectory prediction can guide the model to effectively explore temporal dimension information, thereby helping the model better understand the point cloud sequence. In addition, adding different tasks to the motion trajectory prediction task can further improve the performance of the model. This is because the semantic contrast task can provide guidance for global semantic learning for the model, and the appearance reconstruction task can enhance the understanding of the local structure of the point cloud sequence. Therefore, in order to better guide the learning of the model, we use three pretext tasks at the same time.

Table 4: Ablation studies on different pretext tasks.

	Motion Prediction	Semantic Contrast	Appearance Reconstruction	Acc. (%)
M1			✓	79.65
M2		✓		80.27
M3	✓			89.73
M4		✓	✓	88.92
M5	✓		✓	92.67
M6	✓	✓		92.83
M7(Ours)	✓	✓	✓	**93.03**

5 Conclusion

In this paper, we proposed a novel masked motion trajectory prediction and global semantic contrast based self-supervised representation learning framework for point cloud sequences. It integrated motion trajectory prediction and global semantic contrast tasks into the masked autoencoder framework to improve the model's ability to explore the temporal dimension cues and global semantic information of point cloud sequences. In addition, we also employed the appearance reconstruction task to enhance the model's learning ability of local structure information of the point cloud sequences. By conducting extensive experiments on multiple benchmark datasets, we demonstrate that our method can better improve the model's ability to understand the point cloud sequences and achieve better performance in downstream tasks.

Acknowledgments

The authors would like to thank reviewers for their detailed comments and instructive suggestions. This work was supported by the National Science Fund of China (Grant Nos. 62276144, 62176124).

References

1. Chen, A., Zhang, K., Zhang, R., Wang, Z., Lu, Y., Guo, Y., Zhang, S.: Pimae: Point cloud and image interactive masked autoencoders for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5291–5301 (2023)
2. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
3. Chen, Z., Mao, J., Wu, J., Wong, K.Y.K., Tenenbaum, J.B., Gan, C.: Grounding physical concepts of objects and events through dynamic visual reasoning. arXiv preprint arXiv:2103.16564 (2021)
4. Chollet, F.: Deep learning with Python. Simon and Schuster (2021)
5. De Smedt, Q., Wannous, H., Vandeborre, J.P., Guerry, J., Le Saux, B., Filliat, D.: Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset. In: 3DOR-10th Eurographics Workshop on 3D Object Retrieval. pp. 1–6 (2017)
6. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 605–613 (2017)
7. Fan, H., Yang, Y., Kankanhalli, M.: Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14204–14213 (2021)
8. Fan, H., Yang, Y., Kankanhalli, M.: Point spatio-temporal transformer networks for point cloud video modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(2), 2181–2192 (2022)
9. Fan, H., Yu, X., Ding, Y., Yang, Y., Kankanhalli, M.: Pstnet: Point spatio-temporal convolution on point cloud sequences. arXiv preprint arXiv:2205.13713 (2022)
10. Fan, H., Yu, X., Yang, Y., Kankanhalli, M.: Deep hierarchical representation of point cloud videos via spatio-temporal decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(12), 9918–9930 (2021)
11. Feichtenhofer, C., Li, Y., He, K., et al.: Masked autoencoders as spatiotemporal learners. Advances in neural information processing systems **35**, 35946–35958 (2022)
12. Han, Y.: Generation-based multi-view contrast for self-supervised graph representation learning. ACM Transactions on Knowledge Discovery from Data **18**(5), 1–17 (2024)
13. Han, Y., Chen, J., Qian, J., Xie, J.: Graph spectral perturbation for 3d point cloud contrastive learning. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 5389–5398 (2023)
14. Han, Y., Hui, L., Jiang, H., Qian, J., Xie, J.: Generative subgraph contrast for self-supervised graph representation learning. In: European Conference on Computer Vision. pp. 91–107. Springer (2022)

15. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
16. Huang, G., Li, W., Teng, J., Wang, K., Chen, Z., Shao, J., Loy, C.C., Sheng, L.: Siamese detr. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15722–15731 (2023)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: 2010 IEEE computer society conference on computer vision and pattern recognition-workshops. pp. 9–14. IEEE (2010)
19. Liu, H., Cai, M., Lee, Y.J.: Masked discrimination for self-supervised learning on point clouds. In: European Conference on Computer Vision. pp. 657–675. Springer (2022)
20. Liu, X., Yan, M., Bohg, J.: Meteornet: Deep learning on dynamic 3d point cloud sequences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9246–9255 (2019)
21. Min, Y., Chai, X., Zhao, L., Chen, X.: Flickernet: Adaptive 3d gesture recognition from sparse point clouds. In: BMVC. vol. 2, p. 5 (2019)
22. Min, Y., Zhang, Y., Chai, X., Chen, X.: An efficient pointlstm for point clouds based gesture recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5761–5770 (2020)
23. Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., Kautz, J.: Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4207–4215 (2016)
24. Pang, Y., Wang, W., Tay, F.E., Liu, W., Tian, Y., Yuan, L.: Masked autoencoders for point cloud self-supervised learning. In: European conference on computer vision. pp. 604–621. Springer (2022)
25. Ranasinghe, K., Ryoo, M.: Language-based action concept spaces improve video self-supervised learning. arXiv preprint arXiv:2307.10922 (2023)
26. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1010–1019 (2016)
27. Shen, Z., Sheng, X., Fan, H., Wang, L., Guo, Y., Liu, Q., Wen, H., Zhou, X.: Masked spatio-temporal structure prediction for self-supervised learning on point cloud videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16580–16589 (2023)
28. Shen, Z., Sheng, X., Wang, L., Guo, Y., Liu, Q., Zhou, X.: Pointcmp: Contrastive mask prediction for self-supervised learning on point cloud videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1212–1222 (2023)
29. Sheng, X., Shen, Z., Xiao, G.: Contrastive predictive autoencoders for dynamic point cloud self-supervised learning. arXiv preprint arXiv:2305.12959 (2023)
30. Sheng, X., Shen, Z., Xiao, G., Wang, L., Guo, Y., Fan, H.: Point contrastive prediction with semantic clustering for self-supervised learning on point cloud videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16515–16524 (2023)
31. Sun, X., Chen, P., Chen, L., Li, C., Li, T.H., Tan, M., Gan, C.: Masked motion encoding for self-supervised video representation learning. In: Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2235–2245 (2023)

32. Tokmakov, P., Hebert, M., Schmid, C.: Unsupervised learning of video representations via dense trajectory clustering. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 404–421. Springer (2020)
33. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Advances in neural information processing systems **35**, 10078–10093 (2022)
34. Wang, G., Zhou, Y., Luo, C., Xie, W., Zeng, W., Xiong, Z.: Unsupervised visual representation learning by tracking patches in video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2563–2572 (2021)
35. Wang, H., Yang, L., Rong, X., Feng, J., Tian, Y.: Self-supervised 4d spatio-temporal feature learning via order prediction of sequential point cloud clips. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3762–3771 (2021)
36. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2015)
37. Wang, Y., Xiao, Y., Xiong, F., Jiang, W., Cao, Z., Zhou, J.T., Yuan, J.: 3dv: 3d dynamic voxel for action recognition in depth video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 511–520 (2020)
38. Wei, C., Fan, H., Xie, S., Wu, C.Y., Yuille, A., Feichtenhofer, C.: Masked feature prediction for self-supervised visual pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14668–14678 (2022)
39. Wen, H., Liu, Y., Huang, J., Duan, B., Yi, L.: Point primitive transformer for long-term 4d point cloud video understanding. In: European Conference on Computer Vision. pp. 19–35. Springer (2022)
40. Yu, Y., Wang, X., Zhang, M., Liu, N., Shi, C.: Provable training for graph contrastive learning. arXiv preprint arXiv:2309.13944 (2023)
41. Zeng, Y., Qian, Y., Zhu, Z., Hou, J., Yuan, H., He, Y.: Corrnet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6052–6061 (2021)
42. Zhang, R., Guo, Z., Gao, P., Fang, R., Zhao, B., Wang, D., Qiao, Y., Li, H.: Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. Advances in neural information processing systems **35**, 27061–27074 (2022)
43. Zhang, R., Wang, L., Qiao, Y., Gao, P., Li, H.: Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21769–21780 (2023)
44. Zhang, Z., Dong, Y., Liu, Y., Yi, L.: Complete-to-partial 4d distillation for self-supervised point cloud sequence representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17661–17670 (2023)
45. Zhong, J.X., Zhou, K., Hu, Q., Wang, B., Trigoni, N., Markham, A.: No pain, big gain: classify dynamic point cloud sequences with static models by fitting feature-level space-time surfaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8510–8520 (2022)