

Masked Spatio-Temporal Structure Prediction for Self-supervised Learning on Point Cloud Videos

Zhiqiang Shen^{1,5*}, Xiaoxiao Sheng^{1*}, Hehe Fan^{2†}, Longguang Wang³, Yulan Guo⁴,
Qiong Liu⁵, Hao Wen⁵, Xi Zhou^{1,5}

¹Shanghai Jiao Tong University ²Zhejiang University ³Aviation University of Air Force
⁴Sun Yat-sen University ⁵CloudWalk

Abstract

Recently, the community has made tremendous progress in developing effective methods for point cloud video understanding that learn from massive amounts of labeled data. However, annotating point cloud videos is usually notoriously expensive. Moreover, training via one or only a few traditional tasks (e.g., classification) may be insufficient to learn subtle details of the spatio-temporal structure existing in point cloud videos. In this paper, we propose a Masked Spatio-Temporal Structure Prediction (MaST-Pre) method to capture the structure of point cloud videos without human annotations. MaST-Pre is based on spatio-temporal point-tube masking and consists of two self-supervised learning tasks. First, by reconstructing masked point tubes, our method is able to capture the appearance information of point cloud videos. Second, to learn motion, we propose a temporal cardinality difference prediction task that estimates the change in the number of points within a point tube. In this way, MaST-Pre is forced to model the spatial and temporal structure in point cloud videos. Extensive experiments on MSRAAction-3D, NTU-RGBD, NvGesture, and SHREC'17 demonstrate the effectiveness of the proposed method. The code is available at <https://github.com/JohnsonSign/MaST-Pre>.

1. Introduction

In physics, motion is the phenomenon in which position changes over time. Because point clouds provide precise position information, *i.e.*, 3D coordinates, point cloud videos, which evolve over time, can accurately describe the 3D motion in the real world. Effectively understanding point cloud videos can significantly improve intelligent agents on the interaction with environments. Therefore, the community has developed a few effective methods for

*These authors contributed equally.

†Corresponding author.

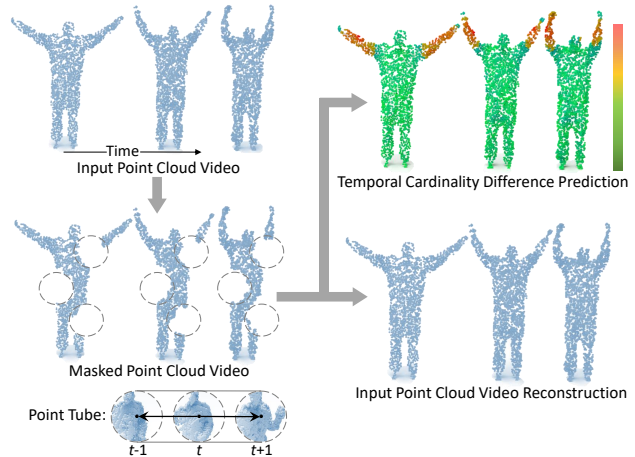


Figure 1. Our MaST-Pre is based on spatio-temporal point-tube masking. To enable a model to capture the appearance structure in point cloud videos, we ask it to reconstruct masked point tubes. To equip the model with motion modeling ability, we develop a temporal cardinality difference prediction task.

point cloud video understanding, including video classification [9–12, 40, 50] and semantic segmentation [6, 24, 42, 43]. However, most of these methods are based on supervised learning and that requires much effort to carefully annotate massive amounts of labels. Moreover, learning via only classification or segmentation may make deep neural networks take too much emphasis on the task itself but largely ignore the subtle details of the instinct spatio-temporal structure in point cloud videos. To alleviate those problems, we propose a self-supervised learning method on point cloud videos.

Self-supervised learning uses supervisory signals from the data itself and enables deep neural networks to learn from massive data without human annotations. This is important to recognize more subtle patterns in data. Networks pre-trained with self-supervised learning usually yield higher performance than when solely trained in a supervised manner [5, 15, 16, 19]. Although self-supervised

learning has been applied to images [15], videos [13, 37] and static point clouds [29, 47], it has not been promoted on 4D signals, such as point cloud videos. Visual signals in point cloud videos can be divided into appearance and motion. While appearance specifies which objects are in videos, motion describes their dynamics. Therefore, self-supervised learning on point cloud videos should carefully make the most of the appearance and motion structure.

In this paper, we propose a Masked Spatio-Temporal Structure Prediction (MaST-Pre) method for self-supervised learning on point cloud videos (Fig. 1). MaST-Pre is based on a masking strategy, which has been proven effective in a range of applications. For example, because of the canonical structure, images can be easily segmented into multiple patches for masking [8, 15], which in the case of video are extended to patch tubes [13, 37]. For unstructured static point clouds, spherical support domain masking can be used for masked autoencoder [29, 47]. However, the spatial irregularity and temporal regularity make point cloud videos require a more elaborate masking strategy. Our method is based on a masked point tube mechanism, where a point tube is a local area expanding over a short time [11].

Based on point-tube masking, our MaST-Pre employs two self-supervised tasks to capture the appearance and motion structure, respectively. First, to learn the appearance structure, MaST-Pre is asked to predict the invisible parts of the input from unmasked points. Second, to capture the dynamics in point tubes, we propose the *temporal cardinality difference*, which can be calculated online from inputs without additional parameters. Cardinality can reflect basic structures (*e.g.*, line, edge, and plane) of static point clouds [21]. In this paper, we extend it to a temporal version so that it can model the dynamics of point cloud videos. Intuitively, the temporal cardinality difference characterizes the flow of points within a short time. Therefore, inferring the temporal cardinality difference of masked point tubes facilitates MaST-Pre to learn motion-informative representations. Our contributions are summarized as follows:

- We design a 4D scheme of masked prediction for self-supervised learning on point cloud videos, termed as MaST-Pre. Our MaST-Pre jointly learns the appearance and motion structure of point cloud videos.
- We propose the temporal cardinality difference, a simple and effective motion feature directly captured from raw input points. It explicitly guides MaST-Pre to learn motion-informative representations.
- Extensive experiments and ablation studies on several benchmark datasets validate that our MaST-Pre learns rich representations of point cloud videos.

2. Related Work

In this section, we first briefly review visual mask prediction for self-supervised learning. Then, we present recent advances in self-supervised learning on point clouds and dynamics modeling for point cloud videos.

2.1. Visual Mask Prediction

Mask prediction has been proven to be an excellent self-supervised task for visual representation learning [8, 15, 45]. By reconstructing target signals from the masked input, mask prediction enables the network to learn rich representations and boosts self-supervised learning [2, 4, 39, 41]. Chen *et al.* [4] extended GPT [3] to operate the pixel sequence for prediction. Bao *et al.* [2] and Wang *et al.* [39] introduced another successful framework, BERT [19], to predict the identities of masked tokens.

Then, He *et al.* [15] proposed MAE as a scalable vision learner to predict the pixels of masked patches. Feichtenhofer *et al.* [13] and Tong *et al.* [37] extended MAE to video representation learning by masking patch tubes. Wei *et al.* [41] developed MaskFeat to predict the HOG features of masked spatio-temporal tubes for self-supervised video pre-training. The regular structure of images and videos makes it easy to obtain patches or patch tubes, which facilitates the design of masking strategies. Pang *et al.* [29] extended MAE to unstructured static point clouds and designed a masking strategy based on the local spatial neighborhoods. However, point cloud videos are not only spatially irregular but also temporally misaligned across frames [9, 11]. To remedy this, we design a point-tube masking strategy.

2.2. Self-supervised Learning on Point Clouds

Contrastive learning has made significant progress on static point clouds [17, 32, 44, 49]. Xie *et al.* [44] proposed PointContrast to discriminate two geometric views of matched points using contrastive loss. Hou *et al.* [17] introduced contextual contrastive learning to PointContrast for data-efficient point pre-training. Rao *et al.* [32] mapped the local and global features to shared representation space and applied a contrastive loss on them. Zhang *et al.* [49] used the instance discrimination task on two augmented versions of a point cloud, while Huang *et al.* [18] pre-trained static point clouds using spatio-temporal augmentations. They transformed different views at point, region, object, and scene levels, and then used contrastive learning to judge their semantic consistency. However, there are limited augmentation methods that can guarantee the semantic consistency of point clouds, let alone point cloud videos.

Prediction-based methods on point clouds also attract a lot of attention. Yang *et al.* [46] proposed a folding-based autoencoder that deforms a 2D grid to reconstruct the target 3D point cloud. Recently, mask prediction has been extended to static point clouds. Liu *et al.* [22] designed a point

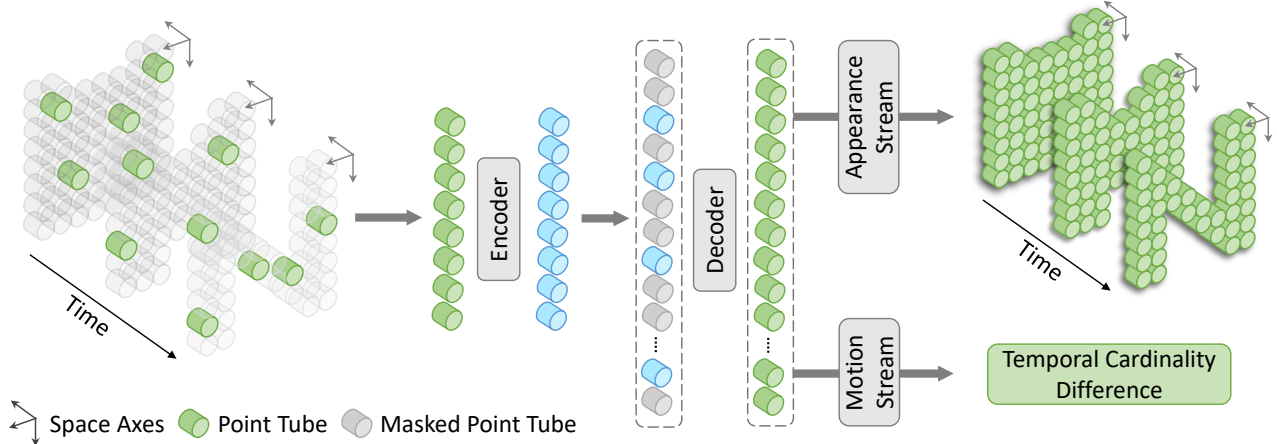


Figure 2. Illustration of the proposed MaST-Pre method. First, given a point cloud video, MaST-Pre divides it into several point tubes and masks part of them. Then, based on an encoder-decoder architecture, MaST-Pre attempts to recover masked point tubes and predict their temporal cardinality difference.

discrimination task for masked patches. Yu *et al.* [47] proposed PointBERT with an offline point tokenizer. Pang *et al.* [29] used a simpler task, reconstructing masked point coordinates, for point pre-training in an end-to-end manner. However, these methods solely focus on the geometric representation learning of static point clouds. In this paper, on top of learning appearance information, an explicit motion information learning method is designed for point cloud videos.

2.3. Dynamics Modeling for Point Cloud Videos

Supervised learning dominates point cloud video research. [11, 12] and [9, 10, 42, 43] use convolution-based methods and attention-based methods to implicitly learn the long-term features of point cloud videos, respectively. In addition, [24, 40, 50] apply empirical-based dynamics methods modeling point cloud videos. Wang *et al.* [40] introduced temporal rank pooling [14] to capture the frame-level dynamics. Zhong *et al.* [50] proposed a two-stream framework and used feature-level ST-surfaces [30] in the dynamics learning branch. Liu *et al.* [24] used a scene flow estimator [23] or an alternative grouping method to do point tracking for dynamics modeling. Although these methods are effective, they require complex calculations or additional modules.

Self-supervised learning on point cloud videos is understudied [34, 35]. Wang *et al.* [38] pre-trained the encoder by predicting the temporal order of shuffled segments to learn the dynamics of point cloud videos. Zhang *et al.* [48] developed complete-to-partial 4D distillation to predict the representations of point cloud frames within a short time window. However, these methods learn motion information using clip-level or frame-level pretext tasks. In this paper, we propose the temporal cardinality difference prediction

for fine-grained dynamics learning on point cloud videos.

3. Method

The architecture of our MaST-Pre is illustrated in Fig. 2. Given a point cloud video, it is first divided into multiple point tubes. Next, a masking operation is performed, separating these point tubes into visible and invisible parts. The visible ones are fed to an encoder, and then their updated embeddings are fed to the decoder along with the masked point tubes. Afterward, two-stream prediction tasks are implemented to recover the point coordinates within masked point tubes and to infer their temporal cardinality differences, respectively. Intuitively, enabling the decoder to perform well on two-stream prediction tasks demands the encoder to learn representations rich in appearance and motion information jointly.

3.1. Masking Strategy

The masking strategy includes three steps: input division, embedding, and masking operation.

Division. Point tubes are introduced as division units of the input point cloud video. Specifically, given a point cloud video \mathbf{P} , Farthest Point Sampling is used to select N key points $\hat{\mathbf{p}}$ from the input. Next, we construct one point tube for each key point. The point tube centered at i -th key point \hat{p}_i is denoted as $\mathbf{Tube}_{\hat{p}_i} = \{p \mid p \in \mathbf{P}, \mathcal{D}_s(p, \hat{p}_i) < r, \mathcal{D}_t(p, \hat{p}_i) < \frac{l}{2}\}$, where p is one of the input points, \mathcal{D}_s is the Euclidean distance, \mathcal{D}_t is the difference in frame timestamps of two points, r is the radius of a spatial neighborhood and l is the number of frames in a point tube. Then, we use random sampling to select n points in each spatial neighborhood.

In this way, a point cloud video is divided into N point tubes, and each point tube contains $l \times n$ points. To en-

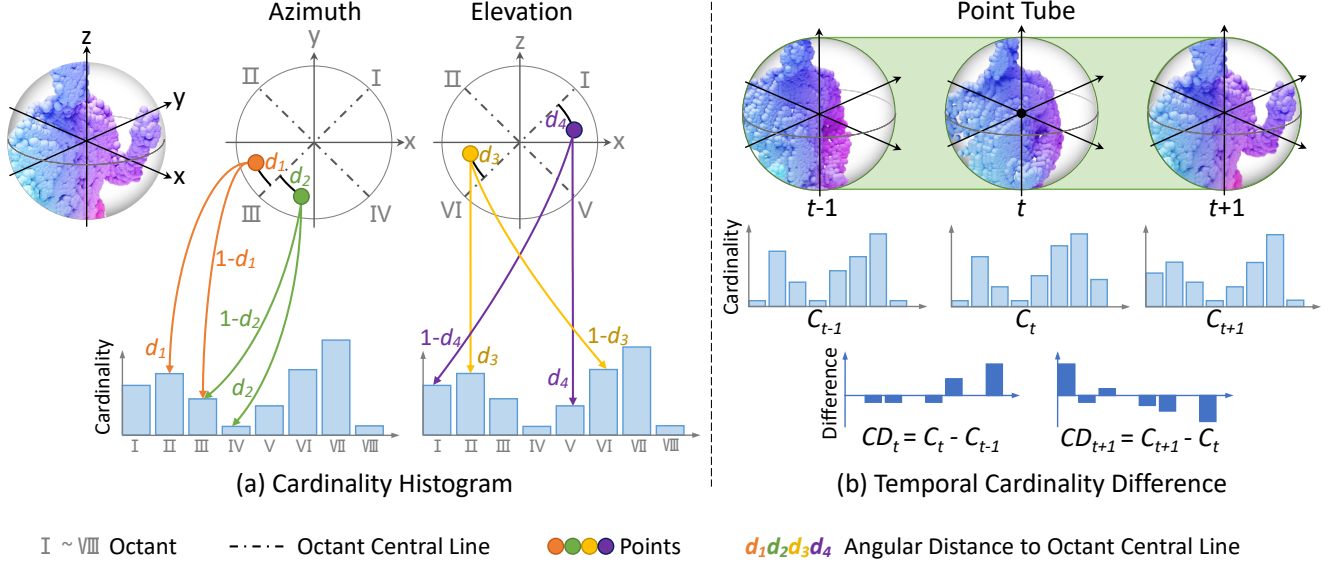


Figure 3. Illustration of Cardinality Histogram (a) and Temporal Cardinality Difference (b).

sure all points are covered by point tubes, r and l are set to maintain a minor overlap between adjacent point tubes.

Embedding. Following our baseline [9], each point tube is encoded into an embedding:

$$E_{\hat{p}_i} = \sum_{t=1}^l \sum_{p \in \text{Tube}_{\hat{p}_i}^t} f(p - \hat{p}_i), \quad (1)$$

where $\text{Tube}_{\hat{p}_i}^t$ is the t -th frame of the point tube centered at \hat{p}_i and $f(\cdot)$ is an MLP-based feature extractor. More details can be found in [9].

Masking. The design of the masking operation is related to the information redundancy of input [13, 15, 37]. Because of spatio-temporal coherence, the redundancy of the point cloud video is higher than low-dimensional data. Therefore, our MaST-Pre uses a high masking ratio on point cloud videos and the empirical result is 75%. A high ratio is helpful to alleviate information leakage and make reconstruction a meaningful self-supervised task. In addition, our MaST-Pre uses random masking of point tubes. As a spacetime-agnostic method, random masking is more effective than structure-aware strategies [13].

3.2. Autoencoding

Our autoencoder is based on vanilla Transformers of point cloud videos [9]. An asymmetric encoder-decoder design is introduced to MaST-Pre.

Encoder. To better capture the dynamics of point cloud videos, joint spatio-temporal attention is adopted [1, 25]. In addition, only the visible point tubes with spatio-temporal positional embeddings are fed into the encoder during pre-training.

Decoder. The decoder is similar to our encoder but a lightweight vanilla Transformer [9]. It takes both encoded point tubes and masked ones as input. By adding a full set of spatio-temporal positional embeddings to all tokens, location clues are provided for self-supervised learning. After decoding, only the embeddings of masked point tubes are fed to the following prediction heads.

3.3. Two-stream Prediction

It is demonstrated in [31, 36] that effective video representation integrates appearance and motion information. Therefore, two-stream self-supervised tasks are proposed to explicitly predict motions and reconstruct the appearance of masked cloud videos.

Appearance Stream. The reconstruction objectives are the point coordinates of masked point tubes. The l_2 Chamfer Distance loss is used between predictions $\mathbf{P}_{pre} \in \mathbb{R}^{l \times n \times 3}$ and the ground truth $\mathbf{P}_{gt} \in \mathbb{R}^{l \times n \times 3}$:

$$\mathcal{L}_{app} = \frac{1}{l} \sum_{i=1}^l \left\{ \frac{1}{|\mathbf{P}_{pre}^i|} \sum_{a \in \mathbf{P}_{pre}^i} \min_{b \in \mathbf{P}_{gt}^i} \|a - b\|_2^2 + \frac{1}{|\mathbf{P}_{gt}^i|} \sum_{b \in \mathbf{P}_{gt}^i} \min_{a \in \mathbf{P}_{pre}^i} \|b - a\|_2^2 \right\}. \quad (2)$$

Motion Stream. We propose the *temporal cardinality difference* as the target of the motion prediction stream. As shown in Fig. 3(a), the spherical support domain of the key point is divided into 8 octants. We follow the conventional rule that the area where the xyz -coordinates are greater than 0 is the first octant I and then increases counterclockwise.

Then, count the cardinality of each octant into the corresponding bin of a histogram. To alleviate the noise in real-world point clouds, a probabilistic approach is employed. Specifically, calculate the angular distance d of each point to the central line of its current octant, and divide it by 90° to normalize. For example, the current octant of **Point2** (the green one in Fig. 3(a)) is octant III. The angular difference $d2$ between **Point2** and the central dashed line of octant III is 30° . Consequently, the probabilities of **Point2** belonging to octant IV and III are $\frac{1}{3}$ (i.e., $\frac{30^\circ}{90^\circ}$) and $\frac{2}{3}$ (i.e., $\frac{60^\circ}{90^\circ}$), respectively. In particular, when a point falls on an axis (e.g., the $-y$ axis), its probabilities belonging to octant IV and III are both 0.5.

Next, as shown in Fig. 3(b), the cardinality histograms $C \in \mathbb{R}^8$ between adjacent frames of a point tube are subtracted to obtain its temporal cardinality difference $CD \in \mathbb{R}^8$, which constitutes the ground truth $M_{gt} \in \mathbb{R}^{(l-1) \times 8}$ of the motion stream. The decoded embeddings of masked point tubes are passed through a linear layer to obtain the motion prediction $M_{pre} \in \mathbb{R}^{(l-1) \times 8}$. The smooth l_1 loss of i -th CD between prediction and ground truth is denoted as \mathcal{L}_m^i . The loss of our motion stream is denoted as \mathcal{L}_{motion} :

$$\mathcal{L}_m^i = \begin{cases} 0.5 \times (M_{pre}^i - M_{gt}^i)^2, & \text{if } |M_{pre}^i - M_{gt}^i| < 1 \\ |M_{pre}^i - M_{gt}^i| - 0.5, & \text{otherwise} \end{cases}, \quad (3)$$

$$\mathcal{L}_{motion} = \frac{1}{l-1} \sum_{i=1}^{l-1} \mathcal{L}_m^i. \quad (4)$$

Overall, the total loss of our MaST-Pre is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{app} + \mathcal{L}_{motion}. \quad (5)$$

With both loss terms, our MaST-Pre can simultaneously learn the geometry and dynamics of point cloud videos.

4. Experiments

Our experiments are conducted on four point cloud video datasets. Following [13, 15, 37], we implement end-to-end fine-tuning, semi-supervised learning, and transfer learning to evaluate the pre-trained MaST-Pre. Afterward, ablation studies are conducted to analyze the design of our MaST-Pre and show the visualization results.

4.1. Datasets

In this paper, to demonstrate the effectiveness of our spatio-temporal representations, we focus on long-term point cloud video tasks, including 4D action recognition and 4D gesture recognition.

MSRAction-3D [20] and **NTU-RGBD** [33] are utilized for the action recognition task. (1) MSRAction-3D is comprised of 567 videos in 20 daily actions. The average

Table 1. Action recognition accuracy on MSRAction-3D.

Methods		Accuracy (%)
Supervised Learning	MeteorNet [24]	88.50
	PSTNet [11]	91.20
	PSTNet++ [12]	92.68
	Kinet [50]	93.27
	PPTr [43]	92.33
	P4Transformer [9]	90.94
	PST-Transformer [10]	93.73
End-to-end	P4Transformer + MaST-Pre	91.29
Fine-tuning	PST-Transformer + MaST-Pre	94.08

frame number contained in each video is about 40. Following [9, 10], 270 videos are used as the training set and 297 videos are adopted as the test data. (2) NTU-RGBD consists of 56,880 videos with 60 fine-grained action categories. The frame number of each video is about 30 to 300. Under the cross-subject setting [33], 40,320 training videos and 16,560 test videos are used.

SHREC'17 [7] and **NvGesture** [28] are utilized for the gesture recognition task. (1) SHREC'17 is comprised of 2800 videos in 28 gestures. Following [7], this dataset is split into 1960 training videos and 840 test videos. (2) NvGesture consists of 1532 videos with 25 gesture classes. Following [27], 1050 videos are assigned to the training set and the remaining 482 videos to the test set.

4.2. Pre-training

During pre-training, given a point cloud video, 24 frames are densely sampled and 1024 points are selected for each frame. Following [9, 10], the frame sampling stride is set to 2/1 on NTU-RGBD/MSRAction-3D and only random scaling is employed for data augmentation. For division and embedding, the temporal downsampling rate is set to 2 and the temporal kernel size l of each point tube is set to 3. Meanwhile, the spatial downsampling rate is set to 32. The radius of each support domain r is set to 0.1/0.3 on NTU-RGBD/MSRAction-3D and the number of neighbor points n within the spherical query is set to 32. The masking ratio is set to 75% unless otherwise specified.

P4Transformer [9] is utilized as our encoder, which consists of 10/5 layers of vanilla Transformers on NTU-RGBD/MSRAction-3D. The decoder is a 4-layer transformer. To verify the extensibility, PST-Transformer [10] is also used as an encoder on MSRAction-3D. Our MaST-Pre is pre-trained for 200 epochs and linear warmup is utilized for the first 10 epochs. The AdamW optimizer is used with a batch size of 128, and the initial learning rate is set to 0.001 with a cosine decay strategy.

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

Methods	Acc.
3DV-Motion [40]	84.5
3DV-PointNet++ [40]	88.8
PSTNet [11]	90.5
PSTNet++ [12]	91.4
Kinet [50]	92.3
P4Transformer [9]	90.2
PST-Transformer [10]	91.0
P4Transformer + MaST-Pre (End-to-end Fine-tuning)	90.8
P4Transformer + MaST-Pre (50% Semi-supervised)	87.8

4.3. End-to-end Fine-tuning

We first evaluate our MaST-Pre by fine-tuning the pre-trained encoder with a new classifier in a supervised manner. End-to-end fine-tuning experiments are conducted on NTU-RGBD and MSRAction-3D, respectively. In each experiment, the same dataset is used for pre-training and fine-tuning.

MSRAction-3D. During fine-tuning, 24 frames are densely sampled and 2048 points are selected in each frame. Following [11], the spatial search radius is set to 0.7. The model is trained for 50 epochs with a batch size of 12 on 4 GPUs. We use the AdamW optimizer and the initial learning rate is set to 0.001 with a cosine decay strategy. As shown in Table 1, compared with baselines trained in a fully supervised manner, our MaST-Pre introduces accuracy improvements for both P4Transformer and PST-Transformer. According to prior experience, the masked autoencoder needs to be fed a considerable amount of data in the pre-training stage to learn useful knowledge [13, 15]. However, the MSRAction-3D dataset is too small to bring significant improvement.

NTU-RGBD. The setup of fine-tuning is the same as pre-training, except that the pre-trained model is fine-tuned for 20 epochs with a batch size of 48 on 8 GPUs and the initial learning rate is set to 0.0005. From the end-to-end fine-tuning in Table 2, we can see that our pre-training method introduces an accuracy improvement compared with the baseline. By predicting the spatio-temporal structure, our MaST-Pre learns appearance and motion information during pre-training.

4.4. Semi-supervised Learning

We also evaluate the learned representations using a semi-supervised learning experiment. Specifically, the cross-subject training set of NTU-RGBD is used for pre-training, and then only a 50% training set is used for fine-tuning in a supervised manner. The setup of our semi-supervised learning experiment is the same as end-to-end fine-tuning on NTU-RGBD (Sec. 4.3).

Table 3. Gesture recognition accuracy (%) on NvGesture (NvG) and SHREC’17 (SHR).

Methods	NvG	SHR
FlickerNet [26]	86.3	-
PLSTM [27]	85.9	87.6
PLSTM-PSS [27]	87.3	93.1
Kinet [50]	89.1	95.2
P4Transformer [9] (30 Epochs)	84.8	87.5
P4Transformer [9] (50 Epochs)	87.7	91.2
P4Transformer + MaST-Pre (30 Epochs)	87.6	90.2
P4Transformer + MaST-Pre (50 Epochs)	89.3	92.4

From Table 2, we can see that the 50% semi-supervised result produced by our MaST-Pre achieves comparable performance to the fully supervised baseline even with only limited annotated data. This clearly demonstrates that MaST-Pre learns high-quality representations.

4.5. Transfer Learning

To evaluate the generalization ability of the representations learned by MaST-Pre, we conduct experiments by transferring the pre-trained encoder to other datasets. Specifically, the encoder is first pre-trained on NTU-RGBD following the setup in Sec. 4.2, and then fine-tuned with a new classifier on NvGesture and SHREC’17, respectively. Our transfer experiments are not only cross-dataset but also cross-task, *i.e.*, from action recognition to gesture recognition. We compare our fine-tuned results to the fully supervised baseline in Table 3.

During fine-tuning, an AdamW optimizer with a batch size of 24 is used, and the initial learning rate is set to 0.002 with a cosine decay strategy. The pre-trained model is fine-tuned for 50 epochs on NvGesture and SHREC’17. As shown in Table 3, our MaST-Pre pre-training facilitates the P4Transformer to produce superior accuracy compared to the fully supervised baseline. Moreover, our MaST-Pre also performs faster convergence. Compared with the baseline without pre-training, significant improvements are achieved after fine-tuning for only 30 epochs (*e.g.*, 84.8% \rightarrow 87.6% on NvGesture and 87.5% \rightarrow 90.2% on SHREC’17). This demonstrates that our MaST-Pre has excellent generalization ability across different tasks, facilitating the accuracy improvement of downstream tasks.

4.6. Ablation Studies

In order to balance authority and efficiency, the experiments of ablation studies are conducted on 10% NTU-RGBD, which contains 4032 training videos and 1656 test videos with category balance.

Architecture Design. Our MaST-Pre utilizes a two-stream prediction to jointly learn both appearance and motion information. To demonstrate the effectiveness of this

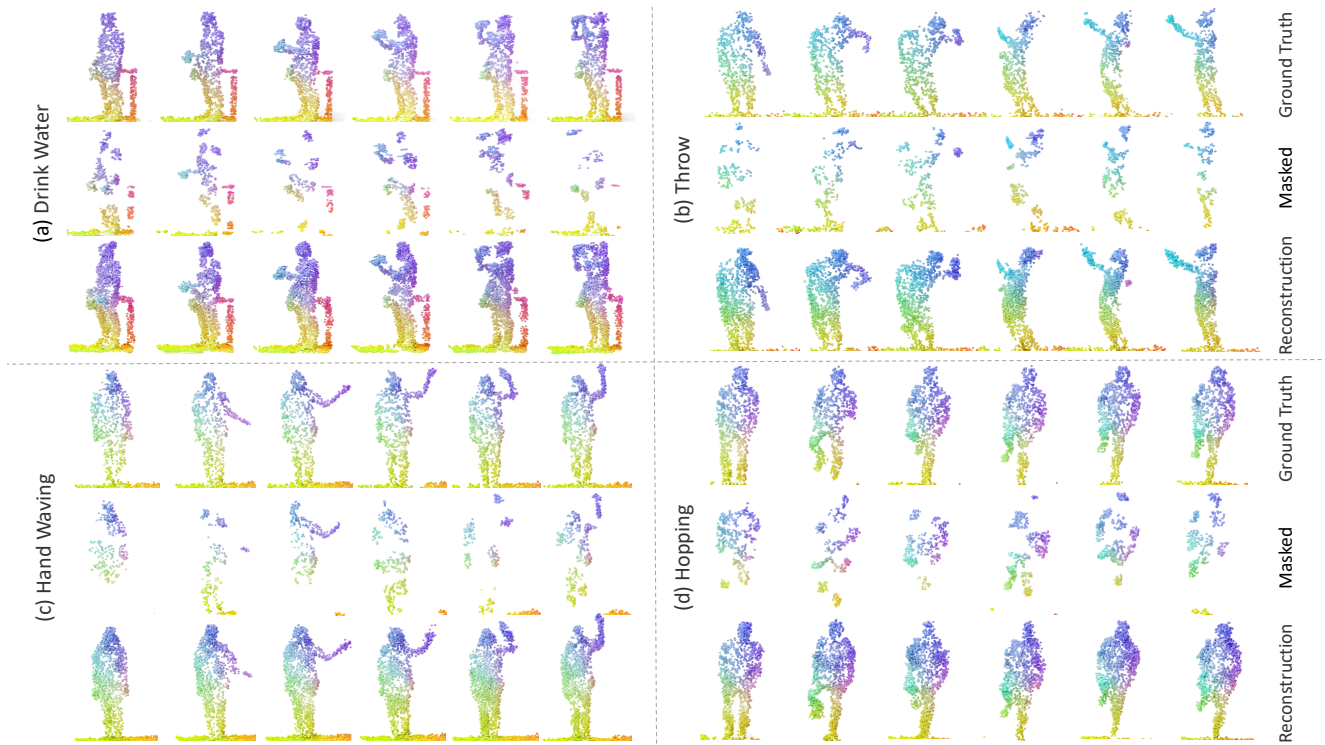


Figure 4. Visualization of reconstruction results. For each action sample, the ground truth lies in the first row, the masked video lies in the second row, and the result lies in the third row.

Table 4. Ablation studies on pre-training architectures.

	Appearance Stream	Motion Stream	Acc. (%)
A0			65.85
A1	✓		70.13
A2 (Ours)	✓	✓	78.25

Table 5. Ablation studies on masking ratios.

	B1	B2 (Ours)	B3
Masking Ratio	65%	75%	85%
Accuracy (%)	76.91	78.25	77.20

architecture, we first present the performance of model A0 as the 10% NTU-RGBD baseline in a fully supervised manner (65.85%). Then, model A1 is developed by removing the motion prediction stream. Quantitative results are presented in Table 4. It shows that with solely the appearance stream, the performance gain introduced by model A1 pre-training is limited. When these two streams are combined, comprehensive information can be learned and superior accuracy is achieved by our model A2 (65.85% \rightarrow 78.25%). This demonstrates the effectiveness of our mask-based pre-training on point cloud videos and the necessity of the task of explicitly predicting motions.

Masking Ratio. The masking operation plays a critical role in our MaST-Pre. Therefore, we conduct experiments

Table 6. Ablation studies on the appearance stream.

	# Frames	Temporal & Spatial	Accuracy (%)
D1	1	-	77.34
D2	3	Coupling	58.60
D3 (Ours)	3	Decoupling	78.25

to study different masking ratios, and the results are presented in Table 5. It shows that a high masking ratio is beneficial to our MaST-Pre and the highest accuracy is achieved at 75% masking ratio (model B2).

Appearance Stream. Right reconstruction targets in the appearance stream contribute to the performance of our MaST-Pre. For model D3, in each point tube, the reconstruction loss (Eq. 2) is first calculated in each frame separately and then aggregated over l frames, which is a decoupled manner. In contrast, model D2 calculates the reconstruction loss in a coupled manner by considering all points together. We also develop model D1 to reconstruct only the middle frame of each point tube.

As shown in Table 6, model D2 brings no accuracy gain and is even worse than the baseline model A0 (58.60% vs. 65.85%). In addition, D3 outperforms D1 and D2. This is because D3 implicitly learns spatio-temporal information during the process of reconstructing decoupled point tubes. We further visualize the reconstruction results in Fig. 4.

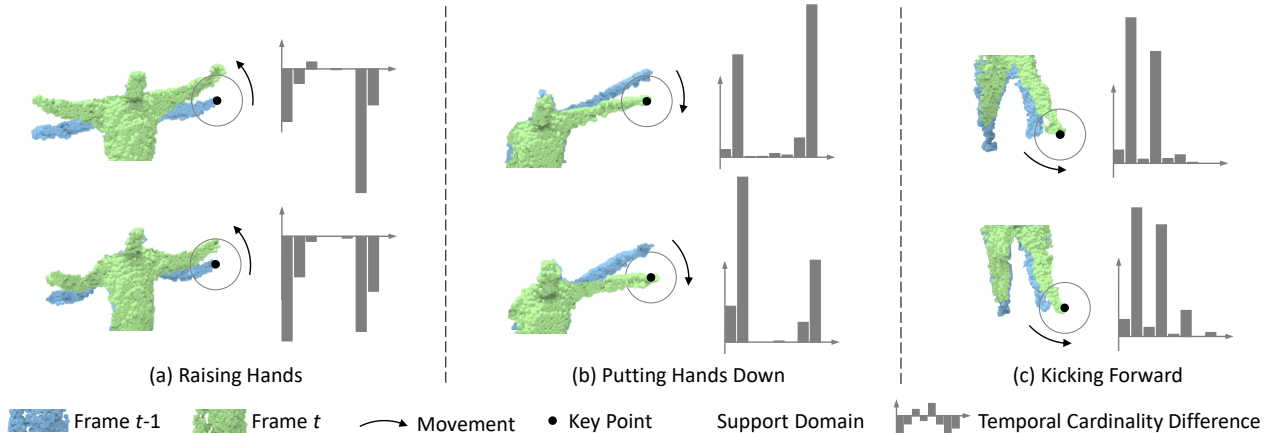


Figure 5. Samples of temporal cardinality difference, computed by subtracting cardinality histograms of frame $t-1$ from frame t .

Table 7. Ablation studies on the temporal cardinality difference.

	E1	E2 (Ours)	E3		F1	F2 (Ours)		G1 (Ours)	G2
# Section	4	8	16	Interpolation	✗	✓	# Stride	1	2
Accuracy (%)	77.85	78.25	69.50	Accuracy (%)	76.69	78.25	Accuracy (%)	78.25	75.85

Temporal Cardinality Difference. In order to investigate the temporal cardinality difference, we present the results under different sections, interpolations, and strides in Table 7. We develop the model E1/E3 to divide the support domain into 4/16 sections, while their accuracy is lower than E2 with 8 sections. This is because small space resolution will introduce noises and large resolution makes temporal cardinality difference insensitive to motions.

Next, we develop model F1 by removing interpolation. While F2 outperforms F1 because interpolation improves its robustness. Finally, we develop model G2 to calculate the cardinality difference with temporal stride 2, but its performance is worse than G1 with stride 1. This is because a large temporal stride cannot capture fine-grained motions.

We further visualize multiple samples of temporal cardinality difference in Fig. 5 to demonstrate its effectiveness in modeling motions. We present three typical actions, each consisting of two samples. As shown in Fig. 5(a), the temporal cardinality differences of the two *raising hands* actions project extremely similar motion patterns. Points in the first and seventh octants flow out heavily over time. Meanwhile, temporal cardinality differences within *putting hands down* (Fig. 5(b)) also display similarities, as well as in *kicking forward* (Fig. 5(c)). In particular, the temporal cardinality differences between *raising hands* and *putting hands down* are approximately reversible, which reflects its effectiveness in modeling dynamics.

Computational Complexity. Table 8 shows the pre-training complexity and corresponding fine-tuning accuracy of the two models. After adding the motion prediction stream, model H2 achieves much higher accuracy than

Table 8. Time (mins/epoch) and memory (MiB) complexities.

Architectures		Encoder	Time	Memory	Acc. (%)
H1	Only Appearance	w/o [M]	5.4	6414	70.13
H2 (Ours)	Two Streams	w/o [M]	6.2	6418	78.25

H1 with only a minor increase in pre-training complexities (70.13% \rightarrow 78.25%).

5. Conclusion

In this paper, we introduce a masked spatio-temporal structure prediction method for point cloud video pre-training, termed as MaST-Pre. For modeling subtle dynamics, the temporal cardinality difference is proposed, which can be calculated online directly from inputs. Based on point-tube masking, MaST-Pre jointly conducts point cloud video reconstruction and temporal cardinality difference prediction to learn both appearance and motion information. Experiments on four benchmarks show that our MaST-Pre is an effective pre-training framework to boost the performance of point cloud video understanding.

Acknowledgments. This work was partially supported by the Fundamental Research Funds for the Central Universities (No. 226-2023-00048), the National Natural Science Foundation of China (No. U20A20185, 61972435), the Guangdong Basic and Applied Basic Research Foundation (2022B1515020103), the Shenzhen Science and Technology Program (No. RCYX20200714114641140), and the Chongqing Technology Innovation and Application Development Special Key Project(cstc2021jscx-cylhX0006).

References

- [1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 4
- [2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022. 2
- [3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 2
- [4] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pre-training from pixels. In *ICML*, 2020. 2
- [5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1
- [6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 1
- [7] 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
- [8] 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. 2
- [9] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4D transformer networks for spatio-temporal modeling in point cloud videos. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6
- [10] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point spatio-temporal transformer networks for point cloud video modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2181–2192, 2022. 1, 3, 5, 6
- [11] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. PSTNet: Point spatio-temporal convolution on point cloud sequences. In *ICLR*, 2021. 1, 2, 3, 5, 6
- [12] Hehe Fan, Xin Yu, Yi Yang, and Mohan Kankanhalli. Deep hierarchical representation of point cloud videos via spatio-temporal decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9918–9930, 2021. 1, 3, 5, 6
- [13] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *NeurIPS*, 2022. 2, 4, 5, 6
- [14] Basura Fernando, Efstratios Gavves, José Oramas, Amir Ghodrati, and Tinne Tuytelaars. Rank pooling for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):773–787, 2016. 3
- [15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 2, 4, 5, 6
- [16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1
- [17] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3D scene understanding with contrastive scene contexts. In *CVPR*, 2021. 2
- [18] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3D point clouds. In *ICCV*, 2021. 2
- [19] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 1, 2
- [20] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3D points. In *CVPRW*, 2010. 5
- [21] Hanxue Liang, Hehe Fan, Zhiwen Fan, Yi Wang, Tianlong Chen, Yu Cheng, and Zhangyang Wang. Point cloud domain adaptation via masked local 3D structure prediction. In *ECCV*, 2022. 2
- [22] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *ECCV*, 2022. 2
- [23] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3D: Learning scene flow in 3D point clouds. In *CVPR*, 2019. 3
- [24] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. MeteorNet: Deep learning on dynamic 3D point cloud sequences. In *ICCV*, 2019. 1, 3, 5
- [25] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, 2022. 4
- [26] Yuecong Min, Xiujuan Chai, Lei Zhao, and Xilin Chen. FlickerNet: Adaptive 3D gesture recognition from sparse point clouds. In *BMVC*, 2019. 6
- [27] Yuecong Min, Yanxiao Zhang, Xiujuan Chai, and Xilin Chen. An efficient PointLSTM for point clouds based gesture recognition. In *CVPR*, 2020. 5, 6
- [28] 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
- [29] 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. 2, 3
- [30] Helmut Pottmann and Johannes Wallner. *Computational line geometry*. Springer, 2001. 3
- [31] Rui Qian, Shuangrui Ding, Xian Liu, and Dahua Lin. Static and dynamic concepts for self-supervised video representation learning. In *ECCV*, 2022. 4
- [32] Yongming Rao, Jiwen Lu, and Jie Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3D point clouds. In *CVPR*, 2020. 2

- [33] 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
- [34] Zhiqiang Shen, Xiaoxiao Sheng, Longguang Wang, Yulan Guo, Qiong Liu, and Xi Zhou. Pointcmp: Contrastive mask prediction for self-supervised learning on point cloud videos. In *CVPR*, 2023. 3
- [35] Xiaoxiao Sheng, Zhiqiang Shen, and Gang Xiao. Contrastive predictive autoencoders for dynamic point cloud self-supervised learning. In *AAAI*, 2023. 3
- [36] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 4
- [37] 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, 4, 5
- [38] 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
- [39] 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. 2
- [40] 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. 1, 3, 6
- [41] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, 2022. 2
- [42] Yimin Wei, Hao Liu, Tingting Xie, Qiuhong Ke, and Yulan Guo. Spatial-temporal transformer for 3D point cloud sequences. In *WACV*, 2022. 1, 3
- [43] 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
- [44] 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. 2
- [45] 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. 2
- [46] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, 2018. 2
- [47] 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. 2, 3
- [48] Zhuoyang Zhang, Yuhao Dong, Yunze Liu, and Li Yi. Complete-to-partial 4D distillation for self-supervised point cloud sequence representation learning. *arXiv preprint arXiv:2212.05330*, 2022. 3
- [49] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3D features on any point-cloud. In *ICCV*, 2021. 2
- [50] 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. 1, 3, 5, 6