Self-supervised 4D Spatio-temporal Feature Learning via Order Prediction of Sequential Point Cloud Clips

Haiyan Wang, Liang Yang, Xuejian Rong, Jinglun Feng, Yingli Tian
The City College of New York, New York, NY 10031
hwang005@citymail.cuny.edu, {lyang1,xrong,jfeng1,ytian}@ccny.cuny.edu

Abstract

Recently 3D scene understanding attracts attention for many applications, however, annotating a vast amount of 3D data for training is usually expensive and time consuming. To alleviate the needs of ground truth, we propose a self-supervised schema to learn 4D spatio-temporal features (i.e. 3 spatial dimensions plus 1 temporal dimension) from dynamic point cloud data by predicting the temporal order of sampled and shuffled point cloud clips. 3D sequential point cloud contains precious geometric and depth information to better recognize activities in 3D space compared to videos. To learn the 4D spatio-temporal features, we introduce 4D convolution neural networks to predict the temporal order on a self-created large scale dataset, NTU-PCLs, derived from the NTU-RGB+D dataset. The efficacy of the learned 4D spatio-temporal features is verified on two tasks: 1) Self-supervised 3D nearest neighbor retrieval; and 2) Self-supervised representation learning transferred for action recognition on smaller 3D dataset. Our extensive experiments prove the effectiveness of the proposed self-supervised learning method which achieves comparable results w.r.t. the fully-supervised methods on action recognition on MSRAction3D dataset.

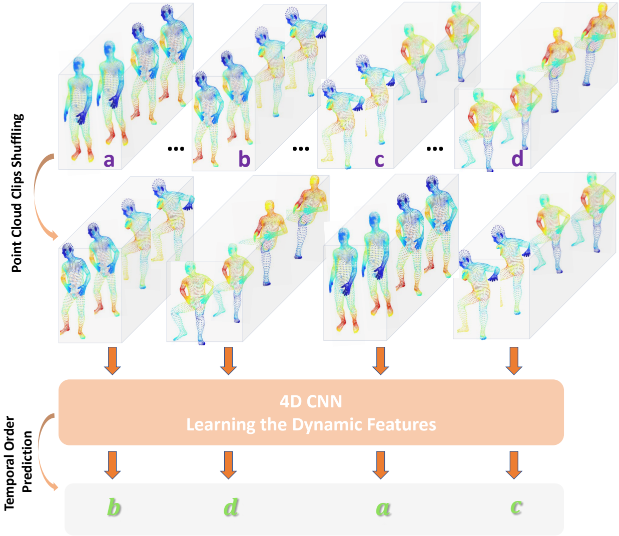

Point Cloud Clips Shuffling
a
...
b
...
c
...
d
4D CNN
Learning the Dynamic Features
Temporal Order Prediction
b
d
a
c

Figure 1. Illustration of our self-supervised method for 4D dynamic feature learning. First row is the sampled clips from point cloud sequences. Then in the second row, point cloud clips are shuffled and input to the 4D CNN network to learn the dynamic features of human action. Those features are further used to predict the correct permutations by the self-supervised manner learning.

1. Introduction

Understanding activities and motions in sequential 3D data, i.e. dynamic 4D data including 3 spatial dimensions and 1 time dimension, becomes more and more important in many applications such as autonomous driving and AR/VR techniques. Compared to images and videos, 4D data contain more information and features to describe our world. Recently, many deep learning methods were proposed to directly process static 3D data for different computer vision tasks, including VoxelNet[19], PointNet[23], PointNet++ [25], and DGCNN [36], etc. Compared to other grid data, point clouds are captured from the raw sensor such as LIDAR, RGBD cameras or laser scanners and have better and more complete representation for the 3D information compared to other data format.

Compared to static 3D data, it is crucial to explore dynamic 3D point clouds, especially for motion analysis and action recognition tasks which heavily depend on long-range temporal variations. A few methods were proposed to tackle 4D dynamic data [4, 17, 18]. The grid-based methods [18, 4] conduct 3D/4D convolution on both spatial and temporal dimensions. Liu et al. proposed a non-grid based method MeteorNet [17] which directly processes the dynamic point cloud data and finds the meaningful neighbor points along both spatial and temporal dimensions. However, these methods still heavily depend on point-wise annotated labels in training which require an immense amount of effort to generate. To utilize raw sequential point cloud data without requiring ground truth labels, we introduce a self-supervised learning approach to leverage the temporal order prediction to learn 4D spatiotemporal features. Existing point cloud-based self-supervised methods [27, 11] focus on only the static point cloud feature learning. They attempted to predict the spatial relationship for different point cloud parts or combine with the 2D shape and pose prediction. However, these methods might not be able to properly handle dynamic sequential point cloud.

Inspired by video self-supervised method [37], as shown in Figure 1, we first generate short point cloud clips from sequential data and shuffle them into different orders. Then a 4D CNN architecture is proposed to learn 4D spatiotemporal features by predicting the orders among these clips. The learned features can serve as pre-trained information for the transfer learning on other downstream tasks (e.g. action recognition) with small scale datasets. With the learned features by self-supervised schema from unannotated large scale datasets, the 4D CNN network for the downstream tasks with small datasets can be optimized and overcomes overfitting problems.

Self-supervised learning usually requires a large scale dataset to perform the pre-text task on it. However, most existing 3D or 4D action recognition datasets are relatively small-scale for the 3D action recognition task. Unlike the ImageNet [6] dataset for images or Kinetic [3] dataset for videos, there is no existing large-scale sequential point cloud action recognition dataset for self-supervised 4D spatiotemporal feature learning. Therefore, we derive a large scale sequential point cloud action recognition dataset named NTU-PCLs from the NTU-RGB+D [28] dataset. Although the NTU-RGB+D dataset contains both RGB and depth images, they are not aligned and cannot be directly used to obtain point cloud from the camera model. Therefore, we form the large-scale sequential point cloud dataset by registering the RGB and depth images based on the skeleton data.

The contributions of the proposed method are summarized as follows:

- We propose a self-supervised schema to learn 4D spatiotemporal features from dynamic point cloud clips by predicting their sequential orders without using any ground-truth point-wise annotations. To the best of our knowledge, this is the first work that explores self-supervised learning on dynamic 3D point cloud data.
- The generalization of our proposed self-supervised schema is evaluated on different networks including MinkowskliNet [4] and MeteorNet [17] and can effectively learn 4D spatiotemporal features from sequential point cloud data.
- We derive a new large-scale dataset, NTU-PCLs, by registering the color and depth images from the NTU-RGB+D dataset, which serves as the training dataset of the pre-text task.
- To evaluate the effectiveness of the learned 4D spatiotemporal features, two downstream tasks are conducted: 1) nearest neighbor retrieval and 2) action recognition. Experiment results demonstrate significant performance improvements compared to the models trained from scratch.

2. Related Work

2.1. 3D Action Recognition

Unlike 2D sequence-based action recognition, 3D action recognition that takes advantage of rich structure data which has the potential to improve the recognition accuracy with less singularity, especially with the widely available structure sensor, such as, Kinect, Intel Realsense, ASUS Xtion Pro, etc. Early studies mainly focused on aggregating 3D skeleton detection models [39, 43] or motion models [33, 8] to search in temporal and spatial alignment of action patterns. However, these earlier work heavily relied on human crafted features to conduct action recognition.

Recent deep learning-based methods for 3D action recognition can directly take the raw RGB-D data or learned 3D model. Direct depth segmented sequence-based action recognition has been tested to be a valid approach, which relies on either human segmentation [26] or flow detection [35]. Vieira et al. [32] first represented 3D data as an occupancy map which allows a 3D learning by preserving both spatial and temporal contextual information. You and Jiang [41] proposed a volume-map representation called 4D map as the input of a 3D CNN model which is robust to camera views in the complex indoor environment.

2.2. Deep Learning on Sequential Point Cloud

The raw representation of point clouds is a set of 3D coordinates. To understand the temporal embedding of a point cloud sequence, directly employing a 3 channel data structure representation of x, y, z is a common practice [24, 29]. For a sequential point cloud, usually a 2D model generates a feature vector output for each point cloud and stacks the entire sequences' output to aggregate a sequence feature for task learning [16]. However, point cloud sequences tend to be large, and which makes it difficult to directly learn from a sequence of point cloud input.

Volume metric representation is one widely adopted approach to describe the spatial distribution of a point cloud [10]. It has an advantage over the 3 channel approach by preserving the 3D structure of the target [5]. Thus, it allows further in-cooperating with a temporal model to learn spatial point cloud data [41]. Another kind of similar grid-based method such as FAF [18] proposed 3D convolution network structure that converts the input data to the bird's view representation and performs convolution over the spatial and temporal dimension. 4D MinkNet [4] also proposes a 4D convolutional network to learn the spatial-temporal representation in the 7D hyper-space. Sparse convolution and hybrid kernel are applied in this work to help the 4D convolution.

Meanwhile, Graph convolution [30] was first proposed to decouple 3D point clouds as voxel trees, thus alleviating the burden of computation for the sequential point cloud. [38] further proposed a 3D graph convolution model to directly learn from the 3D point cloud sequence, which allows direct temporal inferring and decreases the convolution operation size. Recently MeteorNet [17] was proposed to solve the sequential point cloud learning by directly input the raw 4D point cloud data. They adopt the meter module and aggregate the neighbor information by the chain-flow grouping strategy, leading to the improvement over both performance and the efficiency when processing the sequential data.

2.3. Self-Supervised Feature Learning

Deep CNNs in recent years have demonstrated impressive performance on large-scale image-based, video-based, and point cloud-based applications. However, the supervised training of these networks normally requires human-annotated labels. Recently, self-supervised learning methods for videos are proposed to reduce the dependence on data annotations. Ishan et al. and Basura et al. [20, 7] leverage the motion information to determine the distinguished frames and predict whether the shuffled frames in correct orders as a binary classification problem. [14] proposed an order prediction network that takes shuffled frames as input and leverages the learned features to sort the shuffled sequences as a multi-class classification problem to predict the permutations. [2] integrated the deep reinforcement learning method to sample the training images and then predict the permutations. [37] proposed a self-supervised video-based method that takes multiple video clips as input and outputs the exact sorted clip order.

There is no existing self-supervised method that works on the dynamic sequential point clouds. Previous self-supervised point cloud methods focus on the static point clouds [40, 11, 27, 42]. [40] proposed an autoencoder network which learns the folding process to fold the 2D images to the 3D object point cloud. During this process, more complete feature representations are learned by the network. [11] combined the point cloud and images to perform the self-supervised learning. It predicts the shape and pose from a single image and learns the discriminative 2D projection with the predicted point cloud. [27] focused on the raw point cloud data and trains the DGCNN [36] network to predict the spatial location relation of different point cloud parts. [42] proposed a ContrastNet in conjunction with the graph convolution network to predict whether two segments of a point cloud are from the same object as their pre-text task.

While all these methods focus on the static point cloud, in this paper, we propose a self-supervised learning framework to leverage the temporal order information of sequential point clouds and learn meaningful 4D spatiotemporal features.

3. Methodology

3.1. Overview

Existing supervised 4D dynamic point cloud action recognition methods based on ground truth labels [4, 17] achieved good performance on small datasets. However, these small datasets cannot take full use of the network capacity especially for 3D/4D CNN networks which have a huge amount of parameters. Currently there is no existing large scale 4D sequential point cloud dataset with pixel-level annotations. Inspired by the [37] we propose a self-supervised 4D dynamic feature learning method by predicting the temporal order from a unannotated large scale dataset and then fine-tune the learned models to recognize action recognition on small datasets with annotations.

Figure 2 demonstrates the pipeline of our proposed self-supervised learning schema which contains three main components: Tuple Clip Sampling (TCS), Feature Extraction (FE), and Recurrent Order Prediction (ROP). The TCS follows the similar strategy as [37] which uniformly samples and shuffles the point cloud clips comprised several static point clouds. In FE, we introduce and compare various 4D convolutional neural networks including 4D MinkNet [4] and MeteorNet-cls [17]. For each clip of one tuple, a 4D CNN is trained to extract features from the input clip. Weights are shared among all 4D CNNs which are used to extract features for one tuple clip. In the following ROP sub-network, we use LSTM and the fully connected layer to further process the extracted features and predict the permutations.

3.2. Tuple Clip Sampling

For a sequence with T frames of point clouds: $S = (S_1, S_2, ..., S_T)$, each frame (i.e. a set of point clouds) can be represented as $S_t = p_i^t \in \mathbb{R}^{n\times 6}, i = 1, 2, ..., n$ (n is the number of points and 6 stands for the total feature dimensions for each point including the XYZ and RGB, the RGB information comes from the registration of the 2D RGB and depth image). Several continuous point cloud frames form point cloud clips and we extract a total of N clips from each point cloud sequence. Every pair of two clips are not overlapped and has a total number of m interval frames.

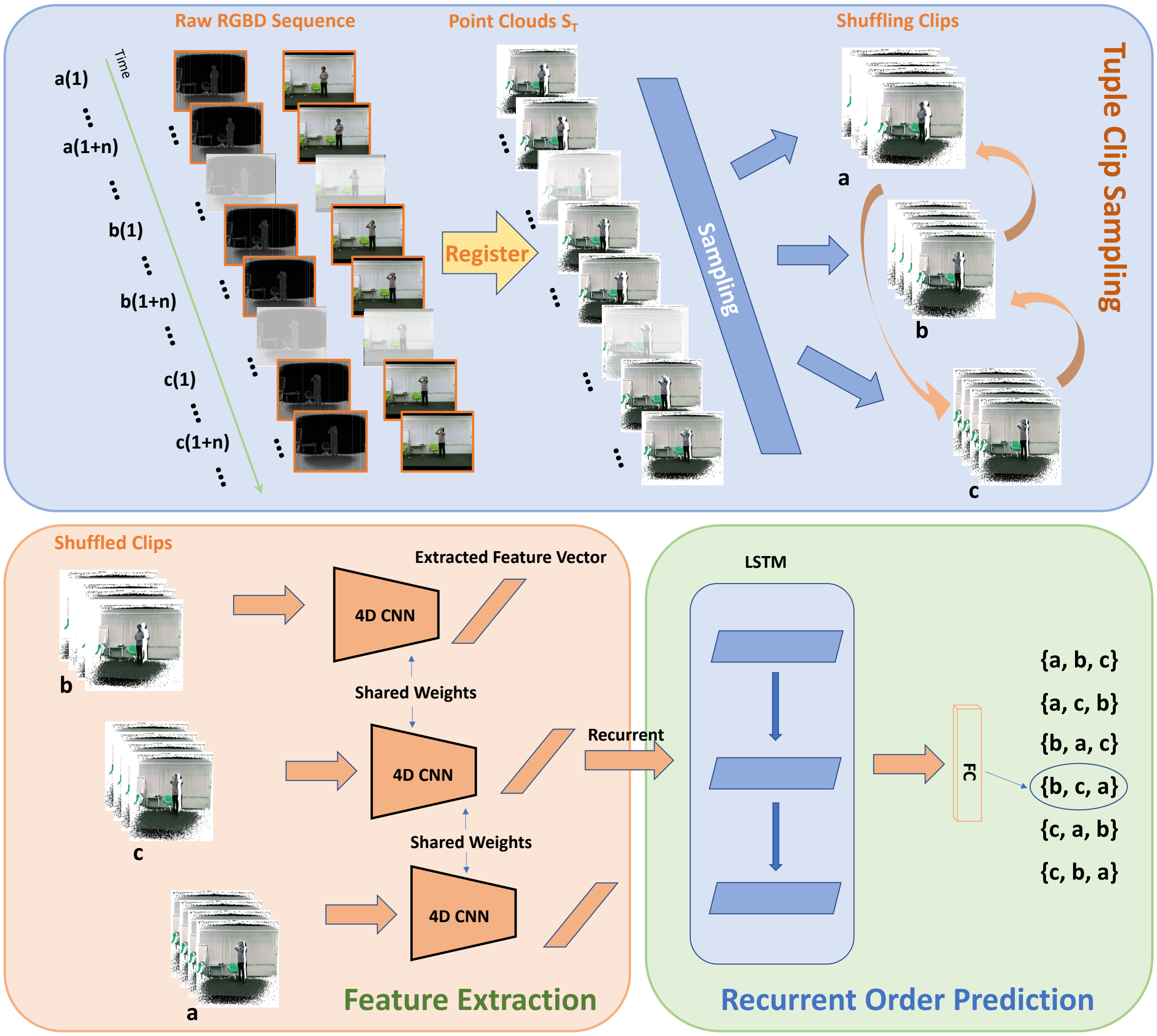


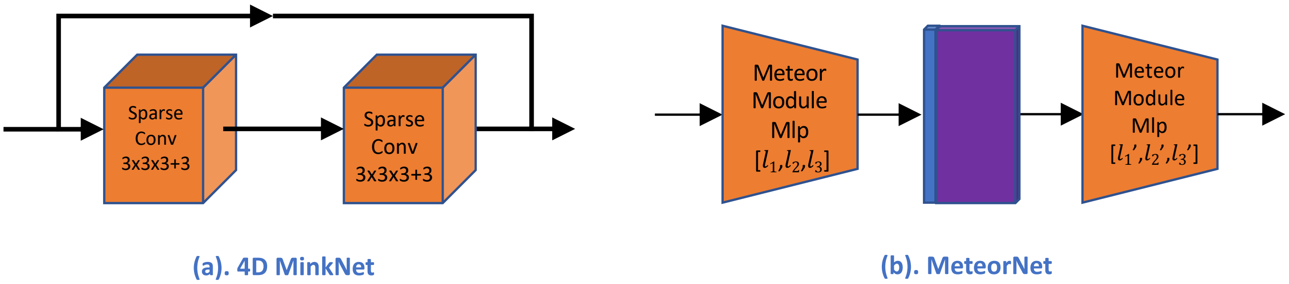
Figure 2. The main pipeline of the proposed self-supervised 4D spatiotemporal feature learning from sequential point clouds. Through the Tuple Clip Sampling network, the sampled and shuffled clips form a tuple of clips. Then the generated point cloud clips are input to the feature extraction network to learn the 4D dynamic features, which will be further fed to recurrent network to predict temporal order in a self-supervised manner.

When considering the strategy of extracting the clips, we follow the rules in [21] which make the task neither too simple nor too ambiguous. After the clips are sampled, they are randomly shuffled to different permutations and input to the network. The different permutations have already been predefined according to the number of clips which should be $N!$, so the target of the convolution is learning the multi-class classification between those permutations to predict the correct order of sampled clips. For example, in Figure 2, the three point cloud clips $\{a, b, c\}$ are sampled from one point cloud sequence. They are shuffled to different permutations such as $\{b, a, c\}$ and input the 4D CNN networks to extract the dynamic features.

3.3. Feature Extraction

After preparing the shuffled data as input, the 4D convolution neural networks are trained to recognize the orders with the given input clips. As shown in Figure 2, each clip is processed by a weights-shared 4D CNN network. In this paper, the 4D MinkNet [4] and MeteorNet [17] are adopted as two types of our dynamic feature learning networks. In addition, we can compare the proposed self-supervised learning method with the grid-based method and non-grid based method according to the experiments on the MinkNet and MeteorNet.

4D MinkNet [4] The 4D MinkNet model is 4D dimensional convolution neural network that directly processes



(a). 4D MinkNet

(b). MeteorNet

Figure 3. The two types of 4D CNNs. (a) 4D MinkNet sparse convolution blocks which adopts the hybrid convolution kernel. " $\times$ " means the hypercubic kernel and "+" means the hypercross kernel. (b) Stacked Meteor Modules form the MeteorNet. Each Meteor Module consists of multiple MLP layers.

the sequential data in a high dimensional space. To solve the discrete problem for the point cloud data, it introduces a sparse tensor and generalize sparse convolution network as shown in Figure 3(a). Due to the cost of convolution in the 4D space, the hybrid kernel and a trilateral-stationary conditional random field are proposed to handle the problem. For the hybrid kernel, it combines the advantages of the cross-shaped kernel and traditional cubic kernel so that extracting both the spatial and temporal features. For the trilateral-stationary CRF which aims to tackle the segmentation loss inconsistency problem, it extends the traditional CRF to the 7D hyper-space including the 3D space, 1D time and 3D chromatic space.

MeteorNet [17] As shown in the right of Figure 3, MeteorNet module is proposed for the 4D sequential point cloud tasks in a directly processing manner. Instead of suffering from the quantization loss, this module takes the point cloud sequence as input, and output the learned features with the aggregation information from the spatio-temporal metric space. Meanwhile, they keep adopting the Multi-layer Perceptron (MLP) and Max-pooling function which are the same as PointNet++ [25] to train the network.

$$h(p_i^{(t)}) = \underset{p_j^{(t')} \in \mathcal{N}(p_i^{(t)})}{MAX} \{\zeta(f_j^{(t')}, f_i^{(t)}, \mathbf{x}_j^{(t')} - \mathbf{x}_i^{(t)}, t' - t)\} \tag{1}$$

where N represent the neighbours for the point $p_i^{(t)}$, $p_j^{(t')}$ is neighbours of $p_i^{(t)}$, $f_j^{(t')}, f_i^{(t)}$ are the related features for point respectively, such as RGB. ζ is the MLP layer function. Here, two ways of grouping methods are introduced to aggregate the neighbour's feature information and find a better N, direct grouping method and chain-flow based grouping method. Direct grouping method just uses a monotonically increased radius for the neighborhood range, while chained-flow grouping method estimates point flow to track the motion trajectories to improve effective and efficient. Here, in order to compare the generalization capability of different methods, we adopt the direct grouping based method.

3.4. Recurrent Order prediction

The feature vectors generated by feature extractor are used to predict the temporal order of these clips. Unlike the method in [20] which treated temporal order prediction as a binary classification problem, we handle the temporal order prediction as multi-classes classification problem as [37]. As mentioned before, all possible permutations have already been pre-defined. For instance, if we take a total of N point cloud clips as input, the network will output $N!$ probabilities and the order with the maximum probability will be selected as the predicted order.

However, instead of simply using the several shuffled and combination of fully-connected layers, the temporal order is predicted recurrently. A recurrent structure is introduced here to better capture the long term memory of 4D sequential data and context information to improve the order classification accuracy. RNN has been proved to be capable of learning meaningful feature representations from sequential data. And long short term memory (LSTM) [9] is introduced to solve the vanishing gradient problem in the RNN model. It contains three additional gates in the hidden layer: the input gate, forget gate and output gate, which helps to address the sequential context information.

$$f_t = h_\theta(C_t), G_t = \gamma(G_{t-1}, f_t), t = 1, 2, ..., N, \tag{2}$$

where the C_t is the input point cloud clip which has N independent point clouds obtained at t independent timestamps. h_θ is the convolution neural networks. $f_t \in \mathbb{R}^{1024}$ denotes the output of the 4D CNN model which acts as input sequence feature of the LSTM. G_t is the current state of the LSTM which is jointly computed based on input f_t and previous state G_{t-1}. γ is the non-linear function used in the LSTM to model the recurrent context feature.

$$p_i = exp(\zeta(G_t))/\sum_{j=1}^{N} exp(\zeta(G_t). \tag{3}$$

The hidden state G_t is further connected to the fully-connected layers ζ to conduct the order classification and the highest probability p_i of the prediction is selected as the predicted order.

3.5. Loss Function

The loss function is calculated as the cross entropy loss between the predicted $K!$ probabilities with the pre-defined permutations to optimize the network h_θ.

$$L_{order} = -\frac{1}{K!}\sum_{i=1}^{N}\left[q_i \log \hat{q}_i + (1-q_i)\log(1-\hat{q}_i)\right] , \tag{4}$$

where q_i is the prediction of different orders, $\hat{q}_i$ is considered to be ground truth of pre-defined permutations.

4. Experiment

4.1. Datasets

NTU-RGB+D NTU-RGB+D [28] is currently the largest 3D human action recognition dataset with 3D annotations. It contains total of 56,880 video action samples of 60 action classes captured by three Microsoft Kinect V2 cameras simultaneously from three different viewpoints. There are 4 different data modalities for each sample including RGB video, depth map sequence, 3D skeletal data, and infrared (IR) video.

New NTU-PCLs point cloud dataset Since the raw RGB images and depth images in NTU-RGB+D dataset are not aligned, where RGB frames with frame size 1,920×1,080 and depth frames are 512×424, we cannot obtain registered 3D point cloud sequences from the raw RGB-D videos. In order to obtain color registered point cloud sequences and encode 3D features for better temporal order learning, we employ using quadratic relation to describe the spatial alignment between depth frame and RGB frame. The RGB to depth registration model is:

$$\begin{aligned} {}^{D}u =& c_{0,0} * Sign(({}^{C}u - {}^{C}u_0)^2) + c_{0,1} * Sign(({}^{C}v - {}^{C}v_0)^2) \\ &+ c_{0,2} * ({}^{C}u - {}^{C}u_0) + c_{0,3} * ({}^{C}v - {}^{C}v_0) + c_{0,4} \\ {}^{D}v =& c_{1,0} * Sign(({}^{C}u - {}^{C}u_0)^2) + c_{1,1} * Sign(({}^{C}v - {}^{C}v_0)^2) \\ &+ c_{1,2} * ({}^{C}u - {}^{C}u_0) + c_{1,3} * ({}^{C}v - {}^{C}v_0) + c_{1,4}, \end{aligned} \tag{5}$$

where $({}^{D}u, {}^{D}v)$ and $({}^{C}u, {}^{C}v)$ denote the pixel coordinates in a depth image and RGB image respectively. $Sign()$ represents the sign value of the expression to keep the polarity of a pixel relative to the center of RGB image. $c_{0,i}$ and $c_{1,i}$ ($i = 0, 1, 2, 3, 4$) are the coefficients of the registration model. We first learn the model using 'Linear Regression' based on the body joints matching between depth and RGB frames, and then register the RGB information to each depth image using the learned parameters. The point cloud can be obtained by back-projecting the RGB and depth information to the 3D space using the pin-hole camera model. In this manner, we can easily generate a 4D point cloud dataset based on NTU-RGB+D dataset, which contains total of $56,880$ point cloud sequences of 60 human action categories. The generated point cloud sequences are used to extract the sampled and shuffled point cloud clips for temporal order prediction.

Table 1. Temporal order prediction by using MinkNet and MeteorNet as 4D encoder respectively, the results show the average accuracy.

Traning Dataset	4D MinkNet (%)	MeteorNet (%)
NTU-PCLs (train)	85.3	79.2

MSRAction3D [15] This dataset is a relatively small which contains a total of 567 depth map sequences. There are 20 action types for 10 subjects such as *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up & throw*. Each subject performs each action 2 or 3 times. The spatial size of depth sequences is 320×240. Following the way in MeteorNet [17] processing the data, we obtain the 3D point cloud sequences from the depth maps and use cross-subject test setting [15] to split the training and testing data. The nearest neighbor retrieval and action recognition experiments are conducted on this dataset.

4.2. Implementation Details

The NTU-PCLs point cloud dataset is generated to train the proposed self-supervised learning framework. The sequential point clouds of camera ID 1&2 (total of 50,000) are used as self-supervised training set, 6,000 samples of camera ID 3 are chosen as the verification training set while other 888 samples of camera ID 3 are chosen as the verification testing set (named as NTU-PCLs (val) for the following experiments). For each point cloud sequence, we select 3 clips to predict the temporal order with the skip frames at 10. Each clip consists of 8 frames.

The 4D MinkNet is implemented by Minkowsi Engine [4] in PyTorch [22]. The backbone which combines the STResNetBase and ResNet50 is adopted as our 4D feature extraction network for MinkNet. Batch normalization, relu and max-pooling layers for sparse convolution are added to the network. The output vector after 6 residual blocks is fed in the recurrent network as inputs. The 4D MinkNet is trained with SGD optimizer, 400,000 iterations and learning rate of 0.001.

The MeteorNet [17] is implemented by adopting the direct grouping strategy and monotonically increasing radius.

Table 2. Comparative demonstration of clip retrieval accuracy over NTU-RGB+D dataset, we compare the average accuracy by increment the size of retrieval.

Methods	Top1	Top5	Top10	Top20	Top50
4D MinkNet [4] (scratch)	49.6	51.3	58.7	63.2	69.9
4D MinkNet [4] (Ours)	51.1	55.8	59.1	69.5	80.4
MeteorNet [17] (scratch)	42.5	49.9	55.6	64.8	71.3
MeteorNet [17] (Ours)	49.1	56.7	60.4	68.1	**82.5**

Table 3. Action recognition results on the NTU-PCLs (val) and MSRAaction 3D through transfer learning.

Method	Input	#of Frames	Accuracy (%)	
			MSRAaction 3D	NTU-PCLs (val)
Vieria et al. [31]	depth	20	78.20	-
Kiaser et al. [13]	depth	18	81.43	-
Actionlet [34]	groundtruth skeleton	full	88.21	-
PointNet++ [25]	static point cloud	1	61.61	63.10
4D MinkNet [4] (scratch)	point cloud sequences	8	75.62	85.60
4D MinkNet (Ours)		8	**86.31**	**88.40**
MeteorNet-cls [17] (scratch)	point cloud sequences	8	81.14	84.32
MeteorNet-cls (Ours)		8	85.40	86.75

One fully connected layer is appended after the set abstraction layer which is the last layer of Meteor Module to obtain a 512 dimension feature vector for order prediction. This network is trained with the Adam optimizer, 300 epochs and learning rate of 0.05.

4.3. Temporal Order Prediction

In order to evaluate the effectiveness of proposed framework, we conduct a straightforward testing by predicting the temporal orders on NTU-PCLs (val) dataset. As shown in Table 1, the average accuracy of two networks are both higher than 79%, which shows our method is able to leverage the long term memory information and predicts the correct temporal order through learning the 4D dynamic features.

4.4. Nearest Neighbor Retrieval

Neural network feature encoding has been proved to achieve high accuracy for image retrieval task [1]. Here, we attempt to retrieve similar action clips given a category-known database. Unlike using a bag-of-words dictionary [12] based on human-crafted features, we directly search the closet k similar clips using a similar metric defined in Eq. 6. Given two 3D point-cloud clips and the learned feature vectors f_1 and f_2, the similarity score is defined as:

$$s(f_1, f_2) = \frac{1}{||f_1|| \cdot ||f_2||} \sum_{i=0}^{n} f_1(i) \cdot f_2(i) \tag{6}$$

$s(f_1, f_2)$ is the similarity score between two clips. For each clip, the similarity score between itself and the known-action dataset is calculated. Then, the k clips with the highest score are selected as the most similar k actions. Following the similar setting in [37], we extract the features of the last layer in the feature extraction network for both the testing and training datasets. The clips in the testing data are adopted to retrieval the training counterpart. The clips extracted from the NTU-RGB+D (val) dataset are used as the query clips. Thereafter the similarity score is calculated between the NTU-RGB+D (val) dataset and the NTU-RGB+D verification training dataset. Table 2 reports the accuracy of the Topk ($k = 1, 5, 10, 20, 50$) results. It can be seen that with our pre-trained model, both 4D MinkNet and MeteorNet are able to achieve higher performances compared to the versions that are trained from scratch. Also, the qualitative results are demonstrated in Figure 4. The center frame is selected to represent the whole point cloud sequence. As we can see, both the 4D MinkNet and MeteorNet can successfully find the similar features in the training dataset.

4.5. Transfer Learning to Action Recognition

We further conduct the transfer learning to verify the effectiveness of our proposed self-supervised method on relatively small datasets. The experiments are tested on MSRAction3D and a subset of NTU-PCLs dataset (the 3rd camera viewpoint with 30 actions.) The 4D convolution networks are first trained with our proposed self-supervised learning method to predict the temporal order. The learned models are employed as pretrained models and then finetuned for 200 epochs on the two small scale datasets. The action recognition accuracy are reported in the Table 3 using the average accuracy of sampled clips as the point cloud sequence prediction accuracy.

On the MSRAaction 3D datase, the action recognition accuracy for 4D MinkNet and MeteorNet trained from scratch is 75.62% and 81.14% respectively. There might some overfitting problem during the training process due to the limited size of training data, especially for the 4D

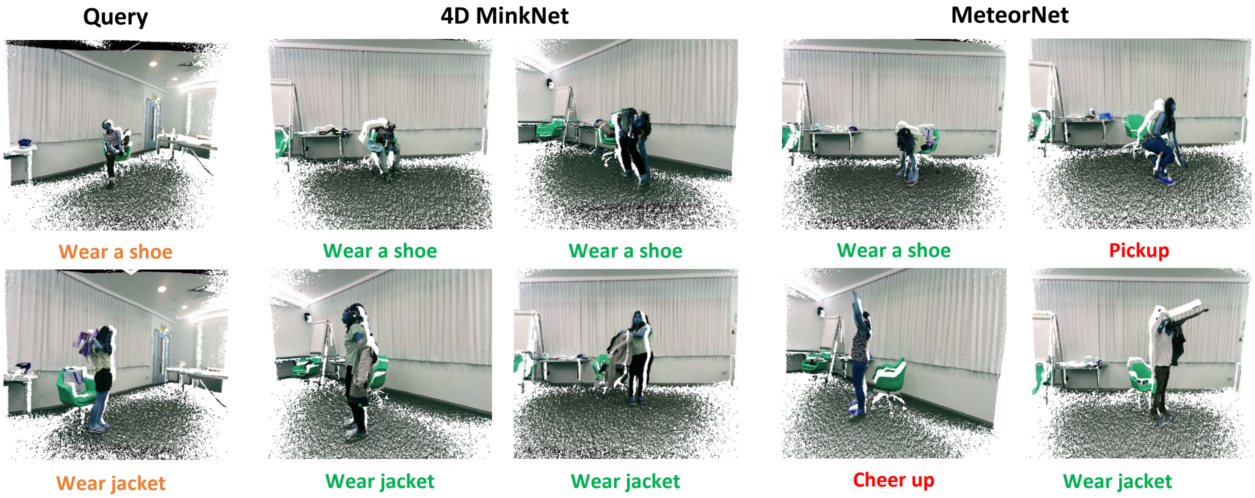


Figure 4. Demostraton of top2 accuracy for nearest neighbor retrieval. The center frame is used to represent the point cloud sequence. The first column represent query clips from the testing split,the middle and last two columns show the result with 4D MinkNnet and the MeteorNet respecively.

Table 4. The relation of action recognition accuracy and different amount training data for the self-supervised learning on both the NTU-PCLs (val) and MSRAaction 3D dataset. The performance keeps increasing when the size of data becomes larger.

Amount of Data	NTU-PCLs (val)	MSRAaction 3D
0	85.60%	75.62%
10,000	86.51%	78.91%
20,000	87.22%	80.46%
40,000	88.19%	84.82%
All	88.40%	86.31%

MinkNet, which has immense number of parameters. However, after applying our pre-trained model obtained from the self-supervised learning, the performance of 4D MinkNet and MeteorNet are boosted to 86.3% and 85.40%. For the NTU-PCLs (val) dataset, the 4D MinkNet and MeteorNet achieve 85.60% and 84.32% when training from scratch. With the pretrained models, the performance has been improved to 88.40% and 86.75% respectively. The experiment results demonstrate the effectiveness of our proposed self-supervised learning method on the 3D point sequences.

4.6. Ablation Study of Training Data Amount

In this section, we study the impact of different amount of training data on NTU-PCLs to the transfer learning. Since the MeteorNet is not sensitive enough for the different amount of data because of the limited network capacity, here we only report the ablation study of data amount for the 4D MinkNet. The learned feature extraction models are then used to perform the action recognition task on the NTU-PCLs (val) and MSRAaction 3D datasets.

As shown in Table 4, with the increasing of the data amount for self-supervised training, the performance of the learned features for transfer learning keeps increasing and achieves highest accuracy when using all of the training data on both the NTU-PCLs (val) and MSRAaction 3D datasets.

5. Conclusions

In this paper, we have proposed a self-supervised method to learn the 4D dynamic features on the point cloud sequences by predicting the temporal order of clips. Besides, we further introduce a recurrent model to learn the temporal embeddings of point cloud clips, which has an advantage of inferring inter-relation between clips. To validate the proposed method, we further created a new sequential point cloud dataset, namely NTU-PCL, based on NTU-RGB+D data set by proposed an RGB to depth alignment model. We employ two different 4D CNN encoders to conduct self-supervised learning and do comprehensive ablation experiments to demonstrate the advantage of our proposed method. Moreover, we introduce the action clip retrieval using a 4D feature similarity metric to search the nearest action on different data sets. Results show that our self-supervised approach is capable of providing state-of-art model initialization and infer the action similarity. The further direction includes learning 3D point cloud flow using a 2D optical flow supervision, which is also in a self-supervised manner.

6. Acknowledgement

This work was supported in part by NSF under awards IIS-1400802 and IIS-2041307.

References

[1] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014.

[2] Uta Büchler, Biagio Brattoli, and Björn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *ECCV*, 2018.

[3] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[4] Christopher Bongsoo Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3070–3079, 2019.

[5] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[7] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5729–5738, 2016.

[8] Ankur Gupta, Julieta Martinez, James J Little, and Robert J Woodham. 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2601–2608, 2014.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[10] Jing Huang and Suya You. Point cloud labeling using 3d convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2670–2675. IEEE, 2016.

[11] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised Learning of Shape and Pose with Differentiable Point Clouds. In *NeurIPS*, 2018.

[12] Hervé Jégou and Ondřej Chum. Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In *European conference on computer vision*, pages 774–787. Springer, 2012.

[13] Alexander Kläser, Marcin Marszalek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[14] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 667–676, 2017.

[15] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 9–14, 2010.

[16] Mengyuan Liu, Chen Chen, and Hong Liu. 3d action recognition using data visualization and convolutional neural networks. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 925–930. IEEE, 2017.

[17] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. Meteornet: Deep learning on dynamic 3d point cloud sequences. *ArXiv*, abs/1910.09165, 2019.

[18] 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. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018.

[19] Daniel Maturana and Sebastian A. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.

[20] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016.

[21] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *ArXiv*, abs/1603.09246, 2016.

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[23] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, 2017.

[24] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for

3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[25] Charles R Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NIPS*, 2017.

[26] Hossein Rahmani and Ajmal Mian. 3d action recognition from novel viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1506–1515, 2016.

[27] Jonathan Sauder and Bjarne Sievers. Context prediction for unsupervised deep learning on point clouds. *ArXiv*, abs/1901.08396, 2019.

[28] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, 2016.

[29] Yang Tan, Hongxin Lin, Zelin Xiao, Shengyong Ding, and Hongyang Chao. Face recognition from sequential sparse 3d data via deep registration. In *2019 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2019.

[30] Dorina Thanou, Philip A Chou, and Pascal Frossard. Graph-based compression of dynamic 3d point cloud sequences. *IEEE Transactions on Image Processing*, 25(4):1765–1778, 2016.

[31] Antônio Wilson Vieira, Erickson Rangel do Nascimento, Gabriel L. Oliveira, Zicheng Liu, and Mario Fernando Montenegro Campos. Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In *CIARP*, 2012.

[32] Antonio W Vieira, Erickson R Nascimento, Gabriel L Oliveira, Zicheng Liu, and Mario FM Campos. Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In *Iberoamerican congress on pattern recognition*, pages 252–259. Springer, 2012.

[33] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[34] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297, 2012.

[35] Pichao Wang, Wanqing Li, Zhimin Gao, Chang Tang, and Philip O Ogunbona. Depth pooling based large-scale 3-d action recognition with convolutional neural networks. *IEEE Transactions on Multimedia*, 20(5):1051–1061, 2018.

[36] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *CoRR*, abs/1801.07829, 2018.

[37] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Bo Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10326–10335, 2019.

[38] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[39] Xiaodong Yang and YingLi Tian. Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, 25(1):2–11, 2014.

[40] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2017.

[41] Quanzeng You and Hao Jiang. Action4d: Online action recognition in the crowd and clutter. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11857–11866, 2019.

[42] Ling Zhang and Zhigang Zhu. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. *2019 International Conference on 3D Vision (3DV)*, pages 395–404, 2019.

[43] Yu Zhu, Wenbin Chen, and Guodong Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 486–491, 2013.