Computing and Informatics, Vol. 44, 2025, 300–335, doi: 10.31577/cai_2025_2_300

SELF-SUPERVISED LEARNING FOR 3D ACTION PREDICTION BASED ON PAST COMPLETENESS AND FUTURE TREND

Yifan Wang[§]

Department of Computer Science and Technology
Xiamen University, Xiamen 361000, China
e-mail: wangyifan@stu.xmu.edu.cn

Tiancheng Zheng[§]

School of Law, Xiamen University
Xiamen 361000, China
e-mail: skycity1233@163.com

Hua Shi

School of Optoelectronic and Communication Engineering
Xiamen University of Technology, Xiamen 361024, Fujian, China
e-mail: shihua@xmut.edu.cn

Chong Zhao[*]

Department of Computer Science and Technology
Xiamen University, Xiamen 361000, China
e-mail: zhc@xmu.edu.cn

Eryong Wu[*]

Ocean Research Center of Zhoushan
Zhejiang University, Zhoushan, 316021, China
e-mail: wueryong@zju.edu.cn

[§] Co-first authors
[*] Corresponding authors

Abstract. The goal of the 3D action prediction task is to predict the action label corresponding to an incomplete 3D skeleton sequence. Existing studies are limited to the supervised framework. To eliminate the dependence of supervised learning on expensive labels, we propose a self-supervised learning method for 3D action prediction. We use three self-supervised tasks of action completeness perception, motion prediction, and global regularization to allow the network to learn the past and future information embedded in the sequence of unfinished actions, i.e., the action completeness that has occurred and the future motion trend, and to optimize the feature space learned by the model. Some models ignore the past and future information embedded in partial sequences, which is the key to action prediction by humans. Based on our self-supervised method, we design two modules, an action completeness perceptron, and a motion predictor, to complete missing information in partial inputs. And a novel network structure is proposed to fuse partial and complete prediction to achieve more reasonable action prediction. We have conducted extensive experiments on different datasets, and the results validate the effectiveness of our proposed method.

Keywords: 3D action prediction, self-supervised learning, multi-task, skeleton data, motion prediction

1 INTRODUCTION

With the development of computer vision, more and more researchers are focusing on the understanding of human actions. With the ability of understanding human actions, computers can play an important role in many fields such as intelligent surveillance, human-computer interaction, video understanding, etc. Human action understanding consists of many subtasks. Unlike action recognition, which has been widely studied, action prediction involves predicting classes of actions from incomplete sequences of actions. This is a very challenging task because partial sequences often contain insufficient discriminative information, so relatively little research has been done on action prediction. However, computers often need to recognize human actions before they are fully executed in many real-world scenarios, so action prediction has broader application scenarios than action recognition.

Depending on the form of input data, the mainstream action understanding methods can be divided into two categories: image action understanding based on video data and 3D action understanding based on skeleton data. Many studies on action prediction have focused on video data [1, 2]. Compared to video sequences, 3D skeleton sequences obtained using depth cameras [3] or pose estimation algorithms [4, 5] are robust to changes in background, appearance, and viewpoint, and can protect the privacy of the object, thus gaining widespread attention in recent years. From hand-crafted features to deep learning [6, 7], action recognition methods based on 3D skeleton data have been extensively studied and achieved good results. These action recognition methods all use supervised learning and they rely heavily on large-scale

labeled datasets, but manually labeling data is often expensive and time-consuming, so the question of how to train models using unlabeled data has attracted the interest of many researchers. A series of self-supervised or unsupervised methods that do not require labels are proposed [8, 9, 10]. Most of these methods use the idea of contrast learning, i.e., forcing the sample features to be similar to those of the corresponding positive samples and becoming distant from the negative samples.

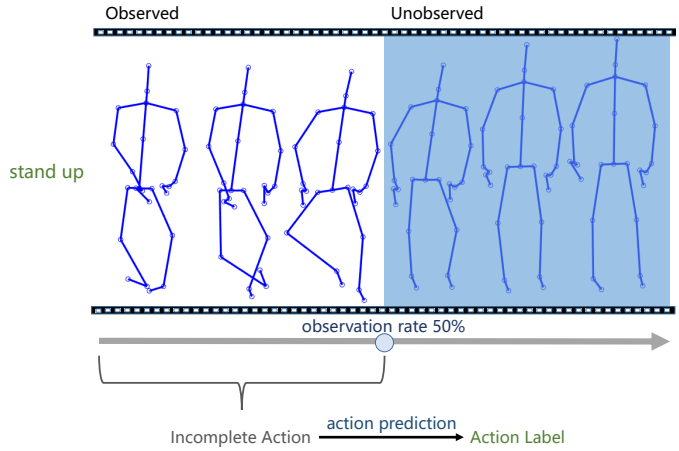


Figure 1. Goal of 3D action prediction is to infer the action label

Not many works focused on skeleton-based 3D action prediction, and only a few new papers have appeared in this area recently [11, 12, 13]. Depending on the feature extraction models, they can be broadly classified into three categories: RNN-based methods [12], CNN-based methods [13, 14, 15], and GCN-based methods [16, 17]. However, current studies on 3D action prediction are limited to a supervised framework, which requires a large amount of labeled data, while the labeling of the data is expensive and time-consuming. Therefore, in this paper, we propose a self-supervised framework for 3D action prediction to avoid the reliance on labels.

3D action prediction (or skeleton-based early action recognition) is shown in Figure 1, where the model needs to infer the action label of an incomplete action sequence with an unknown observation rate. This is an example on how a human would determine the action category (e.g. stand up) of this sequence. Firstly, the human would realize that the target has risen a little from the chair, and then the human would naturally imagine the target moving from a semi-sitting posture to a full standing posture sometime later. With this information, one can easily guess that the sequence is a "stand up" action.

Based on the fact that humans perform action prediction, we propose that there is rich information of past completeness and future trend contained in the sequence of uncompleted actions, and this information helps to predict the class of actions. We design three self-supervised tasks – action completeness perceptron, motion pre-

diction, and global regularization – to allow the network to learn the past and future information embedded in the skeleton sequence data to obtain discriminative feature representations. Different from the "completeness" of the action localization task [18, 19, 20], which describes the localization effect, the completeness proposed in this paper is characteristic of the action prediction task and corresponds to the observation rate of an incomplete action sequence.

In the action completeness perception task, the model is trained to perceive the observation rate of the current incomplete action sequence. Unlike previous research [21] in which the observation rate regression task was used to discern whether the samples were effectively enhanced, the proposed task perceives the observation rate directly from the features and serves as a self-supervised task to supervise the training of the feature extractor. Inspired by the researches on human motion trajectory prediction [22, 23], the motion prediction task is proposed as a self-supervised task to train the model to learn future trend information by reconstructing the complete action sequence from the features of partial sequences. In addition, contrast learning is a common unsupervised learning method. In this paper, the global regularization task based on contrast learning is used to optimize the feature space so that global information from the complete sequence is introduced into the feature representation. To the best of our knowledge, this is the first work to explore self-supervised learning on the 3D action prediction task.

Moreover, the action prediction task is fundamentally different from the action recognition task. The input to the action recognition task is a completed sequence, so the action recognition model only needs to capture information about what has happened in the past. However, the input to the action prediction task is incomplete sequence, and the model does not know not only what will happen later but also how much of the current action has been executed. Existing 3D action prediction methods all employ various strategies during training, such as using soft labels [11] or loss associated with observation rates to prevent overfitting [12, 13, 17], using regularization [36] or adversarial learning [15] to force the network to learn implicit global information, and storing indistinguishable instance pairs to allow the network to mine subtle discriminative information [16]. However, the design of these network structures all follow the idea of action recognition, i.e., the encoder-classifier structure: features of partial sequences are firstly extracted and then fed to the classifier to obtain the prediction results. These approaches ignore the past and future information about the action embedded in incomplete sequences, which has been shown by psychological studies [24] to be the key to action prediction by humans. In fact, when humans watch a certain action sequence, they already know what the target is currently doing and what the target will do in the future when they observe it for a certain time, i.e., when a certain observation ratio is reached, and this temporal completeness and trend information can help humans to accurately predict action categories. Therefore, we introduce two modules into the 3D action prediction network, an action completion perceptron and a motion predictor, and train their weights in the self-supervised process. Unlike previous video action prediction methods in [25] that predict future video features, we let

the network directly predict future motion sequences to generate complete action sequences due to the more compact and intuitive skeletal sequence representation of actions compared to video representations, as well as inspired by recent developments in motion prediction research [22, 23, 26]. Based on this, we propose a new network structure that fuses the prediction results of incomplete sequences with those of the complete sequences reconstructed by the network to achieve a better action prediction.

The contribution of this paper can be summarized as follows:

- We propose a novel multi-task self-supervised learning framework, which guides the 3D action prediction network to learn more discriminative feature representations by taking full advantage of past information, future information and co-occurrence information in sequences with different observation rates.
- Unlike previous methods which only extract features from incomplete sequence, we introduce two self-supervised trained modules into the 3D action prediction network to generate complete sequence, making and fusing predictions based on original incomplete sequence and generated complete sequence to obtain the final prediction.
- We have conducted sufficient experiments on different datasets, and the results validate the superiority of the multi-task self-supervised framework and network structure proposed in this paper.

2 RELATED WORK

2.1 Supervised 3D Action Recognition

Early skeleton-based action recognition use hand-crafted features. Wang et al. [6] proposed an ensemble model to represent each action and capture intra-class variance, then designed a new feature suitable to depth data. In [7], histogram-based 3D human pose representation was employed and an HMM classifier was used to identify the actions. Hussein et al. [27] introduced a novel descriptor for human action recognition based on covariance matrices and used multiple covariance matrices to encode the relationship between joint movement and time. Seidenari et al. [28] used joint positions to align multiple parts of the human body and proposed a multi-part bag-of-posed solution. Vemulapalli et al. [29] modeled human actions as curves in the Lie group using a new skeletal representation, and then mapped the action curves from the Lie group to its Lie algebra followed by a classifier.

Compared to traditional methods, deep learning has greatly improved the ability of models to extract features. The emergence of the NTU-RGBD dataset [30] has greatly facilitated the development of related research. Researchers initially tried to use RNN-based models for action recognition. Shahroudy et al. [30] proposed a novel part-aware extension of the LSTM model according to the physical characteristics of human body motion, which learns the long-term patterns specifically for

each body part by splitting the memory cell of the LSTM into part-based sub-cells. Then, a CNN-based method was introduced in [14], which used deep CNNs to learn hierarchical features from the generated images. Because of the natural adaptability of the human skeleton to graph data structures, Yan et al. [31] applied graph-based neural networks to action recognition for the first time. Most of the subsequent work was based on the improvement of ST-GCN. Shi et al. [32] proposed a two-stream framework to adaptively learn the topology of the graph for different GCN layers and skeleton samples, with explicitly formulating and combining the first-order information and the second-order information of the skeleton data. A novel channel-wise topology refinement graph convolution network [33] was proposed to dynamically learn different topologies in different channels, that is, learn a shared topology as a generic prior for all channels and refine it with channel-specific correlations for each channel.

2.2 Unsupervised 3D Action Recognition

In recent years, self-supervised and unsupervised learning methods that can learn feature representations from unlabeled data have attracted the interest of many researchers. Zheng et al. [8] first explored unsupervised learning approaches for skeleton-based action recognition, they proposed a framework consisting of three sub-networks: an encoder extracts feature, a decoder reconstructs the randomly masked input sequence, and a discriminator learns to distinguish the original from the reconstructed sequence. Subsequent studies have mostly used contrastive learning strategies. MS^{2}L [9] designed three tasks to learn skeleton dynamics, temporal evolution, and regularized feature space respectively, with a new training strategy. Si et al. [10] proposed a novel framework for semi-supervised 3D action recognition, which tightly couples self-supervised learning into a semi-supervised algorithm via neighbor relation exploration and adversarial learning.

There have been many unsupervised learning studies using contrast learning [34, 35, 36] in recent years, Li et al. [34] proposed that the rows and columns of the feature matrix of the data correspond to the instance and clustering representations. Li et al. [35] unified deep clustering into a framework of representation learning, Lin et al. [36] maximized the mutual information of different views through contrastive learning. In the field of 3D action recognition, more and more researchers pay attention to contrast learning. The key to contrast learning is the construction of positive and negative samples, and a series of works have been done to investigate this. Gao et al. [37] enhanced the sample through the composition of view transformation and distance transformation. CrosSCLR [38] leveraged multi-view complementary supervision signal to examine the similarity of samples, and mined positive pairs from similar negative samples. Su et al. [39] proposed a self-supervised approach to drive the network to learn the discriminative motion representation features by constructing speed-changed and motion-broken clips. Wang et al. [40] proposed a novel unsupervised representation learning framework that simultaneously captures skeletal postures and motion dynamics by performing contrastive learning between the two

representations which are learned from skeleton coordinate sequences and velocity sequences respectively. Thoker et al. [41] proposed inter-skeleton contrasting to learn from a pair of skeleton representations in a cross-contrastive fashion and introduced several skeleton-specific spatial and temporal augmentations. Guo et al. [42] performed contrastive learning with distributional divergence minimization loss based on extreme augmentations and NNM.

2.3 3D Action Prediction

Compared to video-based action prediction, 3D action prediction using skeletal coordinate sequences as input has gained the attention of researchers in recent years. Hu et al. [11] used RGB-D sequences as input to fuse 3D skeleton information, used local cumulative frame features to represent RGB-D sequences, and also used soft labels to mitigate the interference caused by different action sequences having the same sub-actions in the early stages of training. Jain et al. [12] introduced a sensory-fusion architecture that jointly learns to anticipate and fuse information from multiple sensory streams with LSTM to capture temporal dependencies. Ke et al. [13] proposed a global regularizer to force the network to extract global information from incomplete partial action sequences and assign different weights to sequences with different observation rates during training. Liu et al. [43] stacked multiple dilated convolutional layers with different perceptual ranges and regressed the starting position of the current action to select the appropriate causal convolutional kernel while encoding spatial features using different levels of dilated tree convolutional kernels. Ke et al. [15] proposed to use adversarial learning to minimize the difference between partial and complete sequences in the feature space. Weng et al. [44] introduced category exclusion into action prediction by using reinforcement learning to train an agent to generate a series of masks to exclude interfering negative categories, thereby improving prediction accuracy. Because many actions have small differences at early stages and can be easily misclassified as another action, Li et al. [16] used Hard Instance-Interference Class (HI-IC) Bank to dynamically store similar pairs of indistinguishable action samples and enhanced the network's ability to mine subtle discriminative information through adversarial learning. Li et al. [17] introduced an adaptive graph convolutional network to the action prediction task and used adversarial learning to make the features of partial and complete sequences as similar as possible, and also introduced a temporal-dependent loss function to prevent the network from over-focusing on partial sequences with small observation rates.

3 METHOD

In this section, we present our self-supervised learning framework and network in details. We firstly describe the problem formulation and symbols in Section 3.1, then we outline the general framework of self-supervised learning in Section 3.2, specify

each self-supervised task in Section 3.3, and provide a description of our 3D action prediction network with its training strategy in Section 3.4. The main symbols are summarized in Table 1.

Notations	Definitions
X	complete 3D skeleton sequence
x_i	i^{th} frame in X
O	the observation ratio
X_O	the partial skeleton sequence under the observation ratio O corresponding to X
X_P	the reconstructed complete skeleton sequence corresponding to X
E	the feature encoder
Cls	the action classifier
H_O	the action completeness perceptron
H_M	the motion predictor
H_R	the global regularization head
F	the feature extracted from X_O
$F_{original}$	the feature extracted from X
$F_{complete}$	the feature extracted from X_P
Cat	the number of action categories X_P
L_O	the loss of action completeness perceptron
L_M	the loss of motion prediction
L_R	the loss of global regularization
L_{self}	total loss of the self-supervised training phase
L_{action}	the loss of the action prediction training phase

Table 1. Notations and definitions

3.1 Problem Formulation

3D action prediction requires predicting the action category before the action is fully executed, as shown in Figure 1. A skeleton sequence with a total of T frames is given as $X = \{x_1, x_2, x_3, \ldots, x_{T-1}, x_T\}$, where x_t denotes the skeleton joint coordinates at frame t. For each sample X_O, the value of the observation rate O is between 0.1 and 0.9 ($O \in (0, 1)$). The input for 3D action prediction is a partial sequence $X_O = \left\{x_1, x_2, x_3, \ldots, x_{\lfloor O \times T \rfloor - 1}, x_{\lfloor O \times T \rfloor}\right\}$, i.e., only the first $\lfloor O \times T \rfloor$ frames of the complete action are known. The goal of 3D action prediction is to learn a predictor P, which predicts the action label of an incomplete action sequence, i.e., $Y = P(X_O)$, where Y is the action class probability distribution of the sequence. The predictor usually consists of two parts, the feature encoder E and the classifier Cls. The skeleton sequence is firstly fed into E to extract the features $F = E(X_O)$, and then the features are input to Cls to obtain the prediction result $Y = Cls(F)$, and the overall process can be formulated as

$$Y = P(X_O) = Cls(E(X_O)). \tag{1}$$

Unlike previous approaches to 3D action prediction that focus on supervised learning, we use self-supervised learning to train the encoder E, and propose a 3D action prediction network P_S using past and future information in the skeleton. To achieve this, we sample each sequence in the action recognition dataset at different observation rates (0.1–0.9) to obtain data with different lengths of the observed parts.

3.2 Multi-Task Self-Supervised Learning Framework for 3D Action Prediction

In order to extract the past and future information embedded in the skeleton sequence, while making the learned feature representations more general, we use several self-supervised tasks to train the network together: an action completeness perception task corresponding to past information in the skeleton sequence, a motion prediction task corresponding to future information in the skeleton sequence, and a global regularization task to optimize the feature space.

The overall framework of multi-task self-supervised learning is shown in Figure 2. Different tasks have a shared feature encoder E and respective task head $H \in \{H_O, H_M, H_R\}$. For the input samples, after extracting features using the shared feature encoder, the features are fed into different task heads to achieve different tasks. The gradients of multiple task loss functions $L \in \{L_O, L_M, L_R\}$ are back-propagated simultaneously to supervise the training of the network. Here, we use a bidirectional GRU network [45, 46] as the feature encoder. The dimension of incomplete skeleton sequence X_O is $T \times (C \times V \times M)$, and after feature encoding, the dimension of feature F is $T \times C\_\text{mid}$, where C, V, M are the coordinate dimension of each joint, the number of joints, and the number of human bodies, respectively, and $C\_\text{mid} = 2 \times \text{hidden\_size}$, and hidden_size is the number of hidden units of bidirectional GRU.

With the above self-supervised learning framework, it is possible to train a feature extractor suitable for action prediction tasks without action labels, while the multi-task design can improve the generalizability of the feature extractor. In addition, the action completeness perceptron and motion predictor trained by the self-supervised tasks are the basis of the 3D action prediction network based on past and future.

3.3 Self-Supervised Tasks Based on Past and Future

3.3.1 Action Completeness Perception

In the 3D action prediction task, the only input to the network is the sequence of uncompleted skeletal actions X_O. The degree of completeness of the action, i.e., the observation rate O, is unknown to the network. For an action that is being executed, it is possible to perceive how much it has been executed by observing the part that has been completed. Perceiving the degree of completeness of the

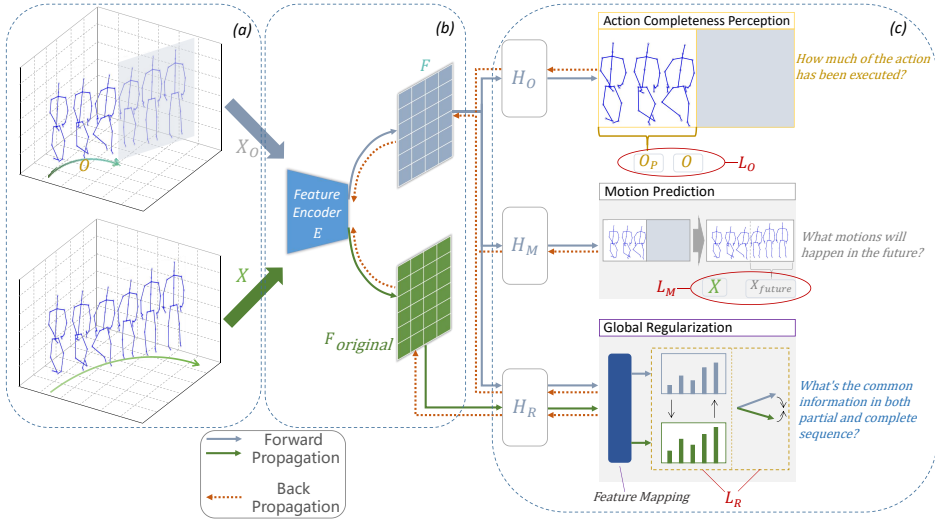


Figure 2. Multi-task self-supervised learning framework for 3D action prediction. The network contains a shared feature encoder and different downstream task heads. a) The partial sequence X_O under the observation ratio O and its corresponding complete sequence X are fed into the network separately. b) F and $F_{original}$ extracted by the feature extractor are the features of partial and original complete sequences, respectively. c) The network is trained to perform three different self-supervised tasks of the action completeness perception task, the motion prediction task and the global regularization task.

current action can help the model better understand the action. This is the past information embedded in the sequence of unfinished skeletons, and we propose to design the action completeness perception task based on it. As shown in Figure 2, in order for the feature extractor E to learn the past information contained in the action sequence, i.e., to gain the ability to perceive the degree of action completeness, this self-supervised task takes an unfinished partial skeleton action sequence as input, and after applying a shared encoder to obtain the feature representation $F = E(X_O)$ for the partial sequence, F is fed into the action completeness-aware network H_O to predict the observation rate of the sequence. We use a two-layer multilayer perceptron as the action completeness perceptron H_O. The last layer of the multilayer perceptron converts the feature dimension to 1 dimension and uses a sigmoid function as the activation function to obtain the predicted observation rate

$$O_P = H_O(F) = \text{sigmoid}(FC(F)). \tag{2}$$

The loss is obtained based on the predicted observation rate from the network and the actual observation rate. We use the mean square error loss as the loss function

$$L_O = \frac{\sum_{i=1}^{N} \|O_{P,i} - O_i\|_2^2}{N}, \tag{3}$$

where N is the size of the batch, O_P, i and O_i refers to the predicted observation rate and real observation rate of the i^{th} sample in the batch. The weights of the shared feature encoder E and the action completeness perceptron H_O are trained by back-propagation using this loss.

3.3.2 Motion Prediction

Action incompleteness is a characteristic of action prediction input, which leads to missing information in the feature. It is possible for humans to infer the trajectory of the subsequent human joints when they observe only the skeletal movement that has occurred (the observed portion). As shown in Figure 1, for the skeletal sequence of the "stand up" action, the target has already stood up halfway, and we can infer that the target will continue to stand up in the future until fully standing. That is, the incomplete action sequence already contains certain trend information, and this trend information about the future can help humans predict and determine the class of the action. We therefore propose the motion prediction task to make the features extracted by the feature encoder E contain such information. As shown in Figure 2, for an unfinished partial action sequence X_O with observation rate O, which corresponds to a complete action with frame number T, its valid frames are the first $\lfloor O \times T \rfloor$ frames, and we use the skeleton coordinates of the $\lfloor O \times T \rfloor^{\text{th}}$ frame to fill the $T - \lfloor O \times T \rfloor$ blank frames that follow. After applying the shared encoder to obtain the feature representation F of the partial sequence X_O, F is fed to the motion predictor to generate future movements X_{future}. We adopt a residual design that allows the motion predictor to predict the relative motion of the human body that will occur, i.e., the motion of the human joints in the future relative to the $\lfloor O \times T \rfloor^{\text{th}}$ frame, and then add it to X_O to generate the complete skeleton sequence. The motion predictor consists of a GRU and a fully connected layer, where the GRU works as a feature decoder to transform the feature distribution and the FC layer transforms the dimension of the feature from $T \times C\_\text{mid}$ to $T \times (C \times V \times M)$. The motion predictor H_M is formulated as following

$$X_P = H_M(F, X_O) = FC(GRU(F)) + X_O, \tag{4}$$

where X_P is the complete action sequence generated by the motion predictor.

After generating the reconstructed sequence X_P through the network, we use the mean square error loss between it and the original complete sequence X as the reconstruction loss for the motion prediction task

$$L_M = \frac{\sum_{i=1}^{N} \|X_{P,i} - X_i\|_2^2}{N}, \tag{5}$$

where N is the batch size. The weights of the shared feature encoder E and the motion predictor H_M are trained by back-propagation using this loss.

3.3.3 Global Regularization

As shown in Figure 2, we propose the global regularization task to optimize the feature space, which takes the partial action sequence X_O and its corresponding complete action sequence X as input, and feeds their features extracted by the feature encoder E into the global regularization head H_R for feature mapping to obtain $F_{observed}$ and F_{full}, respectively

$$\begin{cases} F_{observed} = H_R\left(E\left(X_O\right)\right), \\ F_{full} = H_R\left(E\left(X\right)\right) \end{cases} \tag{6}$$

expecting the mapped features $F_{observed}$ and F_{full} to be similar in distance. H_R is a two-layer multilayer perceptron.

To measure the feature similarity, we consider both the distance of features in terms of direction and numerical values. We use the cosine function to measure the similarity between two feature vectors in terms of direction.

$$\text{similarity}\left(x_1, x_2\right) = \frac{x_1 \cdot x_2}{\max\left(\left\|x_1\right\|_2 \cdot \left\|x_2\right\|_2, \epsilon\right)}, \tag{7}$$

where x_1 and x_2 are two feature vectors and ϵ is a constant. For the difference between the features in terms of values, we measure them using statistical measurements. Then the loss of global regularization can be formulated as

$$\begin{aligned} L_R = & \frac{\sum_{i=1}^{N}\left(1 - \text{similarity}\left(F_{observed,i}, F_{full,i}\right)\right)}{N} \\ & + \frac{\sum_{i=1}^{N}\left\|\text{mean}\left(F_{observed,i}\right) - \text{mean}\left(F_{full,i}\right)\right\|_2^2}{N} \\ & + \frac{\sum_{i=1}^{N}\left\|\text{var}\left(F_{observed,i}\right) - \text{var}\left(F_{full,i}\right)\right\|_2^2}{N}, \end{aligned} \tag{8}$$

where N is the batch size, mean $(\cdot)$ is the mean of the feature vector, and var $(\cdot)$ is the variance of the feature vector. The first term in (8) is the cosine loss, which forces the partial and complete features to be similar in direction, and the second and third terms supervise the partial and complete features to be similar in mean and variance, respectively. The weights of the shared feature encoder E and the global regularizer H_R are trained by back-propagation of the global regularization loss, thus learning a feature space in which incomplete sequences and their corresponding complete sequences have similar feature representations.

When GCN, which is more sensitive to spatial information, is used as the backbone of our method, we employ various data augmentation methods to transform the input sequence X_O and its corresponding complete sequence X. We introduce the key encoder E_k and the key contrastive head H_R_k. The weights of them are updated using momentum. We also introduce a two-layer MLP to map queries to

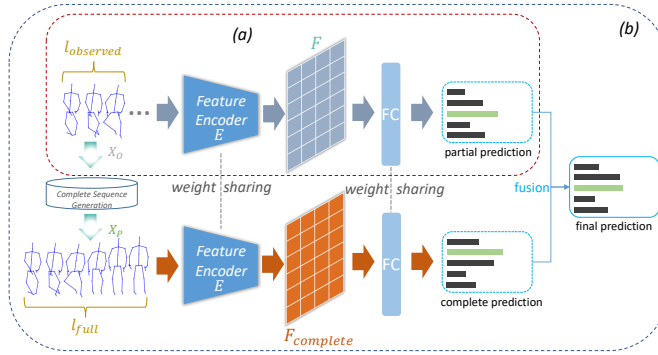


Figure 3. The overall structure of the proposed 3D action prediction network based on past and future. a) Previous methods for action prediction derived results based only on the features of incomplete sequences. b) Our network generates complete sequence from incomplete sequence and combines both partial and complete predictions.

keys. The loss of the mapped query and key is formulated as

$$L\left(q,k\right)=-2\cdot\frac{\langle MLP(q),k\rangle}{\left\|MLP(q)\right\|_2\left\|k\right\|_2},\tag{9}$$

where q is the feature of the input sequence encoded by E and H_R in turn, k is the feature of the input sequence encoded by E_k and H_R_k in turn, and MLP is a two-layer multilayer perceptron.

By performing data augmentation on the input sequence X_O and its corresponding complete sequence X, we are able to obtain four samples. Their corresponding queries and keys can be formulated as

$$\begin{cases} q_O = H_R\left(E\left(X_O\right)\right), \\ q = H_R\left(E\left(X\right)\right), \\ q_{O,aug} = H_R\left(E\left(aug\left(X_O\right)\right)\right), \\ q_{aug} = H_R\left(E\left(aug\left(X\right)\right)\right), \\ k_O = H_R\_k\left(E\_k\left(X_O\right)\right), \\ k = H_R\_k\left(E\_k\left(X\right)\right), \\ k_{O,aug} = H_R\_k\left(E\_k\left(aug\left(X_O\right)\right)\right), \\ k_{aug} = H_R\_k\left(E\_k\left(aug\left(X\right)\right)\right), \end{cases}\tag{10}$$

where aug is data augmentation.

We use the loss in (9) to measure the differences between X_O and X, X_O and $aug(X_O)$, X and $aug(X)$, $aug(X_O)$ and $aug(X)$, respectively. The global regular-

ization loss L_R can be written as

$$\begin{aligned} L_R = & L(q_O, k) + L(q, k_O) + L(q_O, k_{O,aug}) + L(q_{O,aug}, k_O) \\ & + L(q, k_{aug}) + L(q_{aug}, k) + L(q_{O,aug}, k_{aug}) \\ & + L(q_{aug}, k_{O,aug}) \end{aligned} \tag{11}$$

The graph convolutional network can be better trained to optimize the distribution of features through the contrast learning between data augmented sequences.

3.4 3D Action Prediction Network Based on Past and Future

Based on the proposed self-supervised tasks, we design a novel 3D action prediction network using past and future information without additional training. The overall network structure is shown in Figure 3.

3.4.1 Network Structure

We use the action completeness perceptron and motion predictor trained by the self-supervised tasks in Section 3.3 to generate complete sequences based on partial sequences. As shown in Figure 4, the observation rate O of an incomplete action sequence is unknown, and the length l_{full} of its corresponding complete action sequence is also unknown. At first, we input the incomplete sequence X_O into the feature encoder E to get the feature $F = E(X_O)$ of the partial sequence. The observation rate O_P of the sequence is then estimated by the action completeness perceptron H_O, and the length of the complete action is obtained by dividing the number of frames of the observed sequence $l_{observed}$ with the predicted observation rate

$$l_{full} = \lfloor l_{observed} / O_P \rfloor . \tag{12}$$

At the same time, feature F of the partial sequence is sent to the motion predictor H_M to generate the human motion in the future period, and the complete skeleton sequence X_P can be obtained by intercepting the first L_{full} frames of the generated sequence. Similarly, we feed the generated complete sequence X_P into the feature encoder to obtain the complete feature $F_{complete}$, which can be seen in Figure 3. We use a fully connected layer as the action classifier Cls, which converts the dimensions of features into the number of action categories. The partial and global feature are fed into Cls to obtain the prediction result *predict* based on the partial sequence and the prediction result $\text{predict}_{\text{complete}}$ based on the generated full sequence, respectively. As the observation ratio increases, the partial sequence contains more and more information, and the prediction based on it becomes more and more reliable. We fuse these two together to obtain the final prediction.

$$\begin{aligned} \text{predict}_{\text{final}} &= \left(\text{predict} + \text{predict}_{\text{complete}}\right)/2 \\ &= \left(Cls\left(E\left(X_O\right)\right) + Cls\left(E\left(H_M\left(E\left(X_O\right)\right)\right)\right)\right)/2. \end{aligned} \tag{13}$$

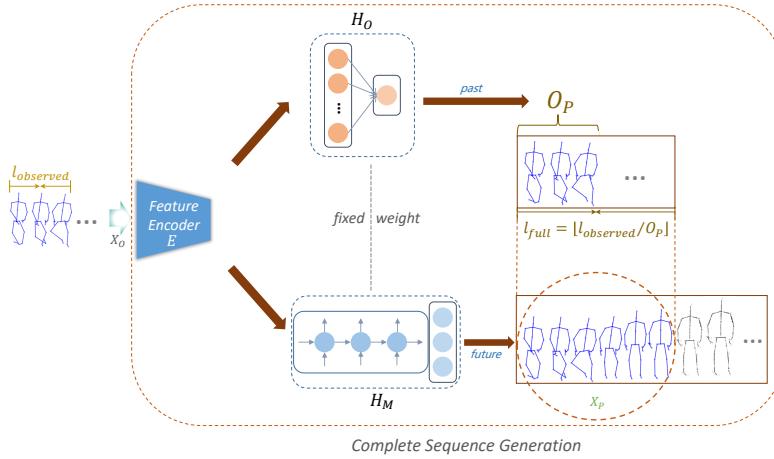


Figure 4. Our sequence generation strategy

3.4.2 Network Training

The training of this network consists of two phases: self-supervised training and action prediction training.

In the self-supervised training phase, we use the self-supervised task in Section 3.3 to train the feature encoder E, the action completeness perceptron H_O, and the motion predictor H_M. The total loss in this phase is

$$L_{self} = L_O + L_M + L_R, \tag{14}$$

where the first term supervises the training of E and H_O, the second term supervises the training of E and H_M, and the third term supervises the training of E and H_R. E, H_O, and H_M are applied to the proposed 3D action prediction network.

In the action prediction training phase, the network includes the feature encoder E, the action completeness perceptron H_O, the motion predictor H_M, and the action classifier Cls. For K-nearest neighbor (KNN) evaluation, a KNN classifier without training is used as Cls to classify the output features of the network. For linear evaluation, according to the network structure shown in Figure 3, we get the local prediction result *predict* and the complete prediction result $\text{predict}_{\text{complete}}$. Since both of them are important, we calculate the losses of these two predictions separately and then add them together, for the total loss function below

$$L_{\text{action}} = \sum_{i=1}^{N} \left(CE\left(\text{predict}_i, \text{label}_i\right)/2\right)/N + \sum_{i=1}^{N} \left(CE\left(\text{predict}_{\text{complete},i}, \text{label}_i\right)/2\right)/N, \tag{15}$$

where N is the batch size, Cat is the number of action categories, and *label* is the action label. The weights of E, H_O, and H_M are frozen during the training, and only the weights of Cls (a single fully connected layer) are supervised by the action labels.

4 EXPERIMENTS

4.1 Datasets

NTU RGB + D dataset [30] is a large-scale multimodal human action recognition dataset containing 60 action categories and 56 800 skeleton sequences. The recordings were performed by 40 volunteers and captured with the Microsoft Kinect v2 sensor. Each action is captured by 3 cameras at the same time, those have the same height but different horizontal angles: $-45°$, $0°$ and $45°$. Two evaluation benchmarks are provided for this dataset:

1. Cross-Subject (CS): The dataset is divided into a training set and a testing set by subject, where the training set and the testing set each contains 20 subjects. For this evaluation, the training and testing sets have 40 320 and 16 560 samples, respectively.
2. Cross-View (CV): The dataset is divided by camera, where samples from cameras 2 and 3 are used for the training set while samples from camera 1 are used for the testing set. For this evaluation, the training and testing sets have 37 920 and 18 960 samples, respectively.

SYSU 3D HOI dataset [47] contains 12 categories of actions performed by 40 volunteers, with a total of 480 samples. All these actions are human-object interactions, captured by a Kinect camera. Since the skeleton data cannot represent the manipulated objects and some actions have the same manipulated objects and motions, it is more difficult to predict 3D actions on this dataset. We adopt the same-subject and cross-subject criterion provided by the authors to evaluate the proposed method. In the first setting, for each activity class, half of the samples are selected for training and the rest for testing. In the second setting, samples performed by half of the subjects are used as the training set and the remaining half as the testing set. The authors provided 30 random splits, we evaluate the model under each split separately, and finally report the average accuracy.

NTU RGB + D 120 dataset [48] is an extension of the NTU RGB + D dataset, expanding its size to 120 action categories with 113 945 samples. This dataset provides two evaluation criteria:

1. Cross-Subjects (X-Sub): samples from 53 subjects are used for training, and samples from other 53 subjects for testing.

2. Cross-Setup (X-Set): samples with even collection setup IDs are used as training set and samples with odd IDs are used as test set, i.e., 16 settings are used for training and the remaining 16 settings are left for testing.

UAV-Human dataset [49] is a large-scale human action understanding dataset collected via UAV, including video and skeleton multiple modalities. The dataset was collected by UAVs in urban and rural areas in daytime and nighttime scenarios, respectively. The dataset contains 23 031 skeleton samples and 155 action categories for action understanding. The dataset provides two rubrics, v1 and v2, both of which divide the dataset based on subjects, with samples from 89 subjects under each division as the training set and samples from the remaining 30 subjects as the testing set.

After processing, all four datasets are expanded to 9 times of their original size, with 511 200, 4 320, 1 025 505 and 207 279 samples, respectively.

4.2 Implementation Details

Because the action prediction dataset needs to generate samples at each observation ratio, the number of total samples is 9 times larger than that of the original action recognition dataset. To speed up the training, we downsample all samples to shorten the length of individual sequences. All experiments are done on a single Nvidia GTX 1080Ti GPU, using PyTorch 1.1.0.

The output of the network trained by the self-supervised tasks and the target task do not match. In order to evaluate the performance of our proposed self-supervised learning method, we need to convert the output of the network. Referring to [42], we conducted experiments under both KNN and linear evaluation protocols.

KNN Evaluation Protocol. The process of KNN evaluation is to apply the KNN classifier directly on the features of the encoder trained by self-supervised learning where no ground-truth action labels are needed.

Linear Evaluation Protocol. The process of linear evaluation is to train the feature encoder by self-supervised learning, fix its weights and then train a single layer FC to match the output of the model in the action prediction task and dataset.

For NTU RGB + D dataset, we adopt the data pre-processing method in [50]. After pre-processing, we downsample all samples uniformly to half of the original sequence, and truncate the part that exceeds 64 frames. Then each sequence is divided into 9 samples with different observation rates from 0.1 to 0.9. We fill the blank frames with the last frame for samples less than 64 frames. Adam optimizer is used to train the network with the batch size of 256. In the self-supervised training phase, the training epoch is set to 10, the initial learning rate is set to $5 \times e^{-4}$, and decays with a factor of 0.1 at epoch 6. For the KNN evaluation criterion, partial feature F and complete feature $F_{complete}$ are averaged and fed directly into a KNN

classifier with K-value set to 25 to obtain the prediction results. For the linear evaluation criterion, the training epoch is set to 30, the initial learning rate is set to $1 \times e^{-2}$, and with a decay factor of 0.1 every 10 epochs.

For SYSU 3D HOI dataset, we apply the same data generation approach as on NTU RGB + D. We evaluate the proposed method under the same-subject and cross-subject protocols respectively, both of which provide 30 different dataset divisions. In the self-supervised training phase, the training epoch is set to 10, the initial learning rate is set to $1 \times e^{-3}$, and decays with a factor of 0.1 every 3 epochs. For KNN evaluation criterion, the value of K is set to 6. For linear evaluation criterion, in the action prediction training phase, the training epoch is set to 300, the initial learning rate is set to $5 \times e^{-2}$, and decays with a factor of 0.1 at epoch 200.

4.3 Ablation Study

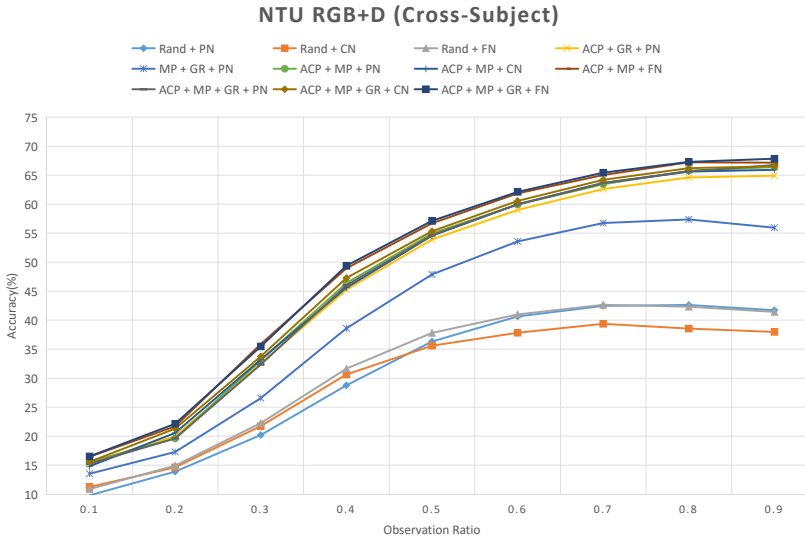
We verify the effectiveness of each self-supervised task and different network structure in ablation study. We use "ACP" for the action completeness perception task, "MP" for the motion prediction task, "GR" for the global regularization task, and "PN", "CN" and "FN" for the partial prediction network, complete prediction network and final prediction network, respectively.

We evaluate the performance of each of the above methods following the CS and CV criteria respectively. The experimental results are shown in Table 2 and Figure 5 a), Table 3 and Figure 5 b).

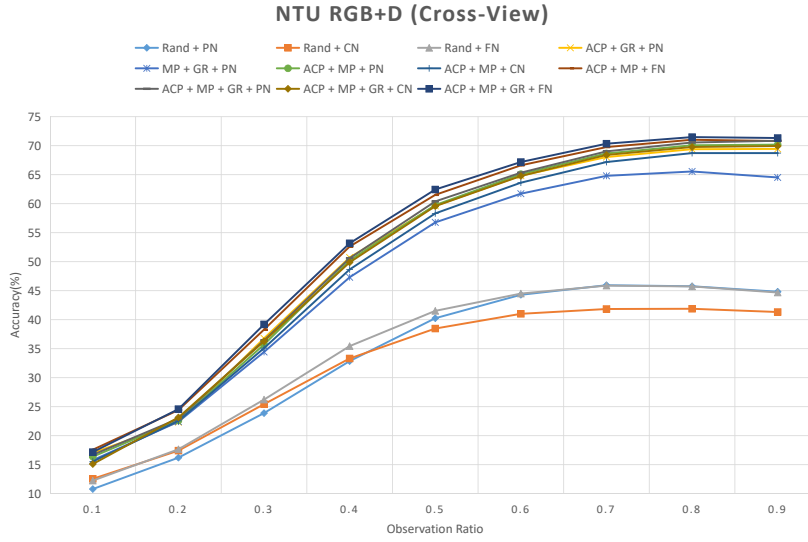
As seen in the tables, while the model parameters are self-supervised trained, the average accuracy of the prediction has been improved by 16.37 % and 18.58 % for the two criteria of cross-subject and cross-view, respectively, when the partial prediction network is used, compared to the random initialization. When the complete prediction network is used, the average accuracy of the prediction is improved by 18.14 % and 19.76 % for the cross-subject and cross-view criteria, respectively. When the fusion prediction network was used, the average accuracy of the prediction improved by 17.56 % and 18.89 % under the two criteria of cross-subjects and cross-views, respectively. This indicates that the multi-task self-supervised learning framework proposed in this paper can effectively train the model to extract 3D action feature representations suitable for action prediction.

As seen in Table 2 and Table 3, when using the partial prediction network, the average prediction accuracy of the model is decreased by 6.26 % and 4.3 % under cross-subject and cross-view, respectively, by removing the action completeness perception task. Without the motion prediction task, the average prediction accuracy of the model is decreased by 0.61 % and 1.53 % under the two divisions, respectively. This indicates that the two self-supervised tasks of action completeness perception based on past completeness and motion prediction based on future trend proposed in this paper can effectively train the model to learn discriminative feature representations.

When both the action completeness perception and the motion prediction tasks are included in the multi-task self-supervised learning framework, the weights of the



a) Ablation experiment results (%) on NTU RGB + D (Cross-Subject)



b) Ablation experiment results (%) on NTU RGB + D (Cross-View)

Figure 5. Ablation experiment results on NTU RGB + D dataset

Self-Supervised Tasks			Network	Average
Action Completeness Perception	Motion Prediction	Global Regularization		
×	×	×	Partial Prediction	30.75
×	×	×	Complete Prediction	29.75
×	×	×	Final Prediction	31.70
√	×	√	Partial Prediction	46.51
×	√	√	Partial Prediction	40.86
√	√	×	Partial Prediction	47.25
√	√	×	Complete Prediction	47.15
√	√	×	Final Prediction	48.97
√	√	√	Partial Prediction	47.12
√	√	√	Complete Prediction	47.89
√	√	√	Final Prediction	**49.26**

Table 2. Ablation experiment results (%) on NTU RGB + D (Cross-Subject)

Self-Supervised Tasks			Network	Average
Action Completeness Perception	Motion Prediction	Global Regularization		
×	×	×	Partial Prediction	33.88
×	×	×	Complete Prediction	32.58
×	×	×	Final Prediction	34.88
√	×	√	Partial Prediction	50.93
×	√	√	Partial Prediction	48.16
√	√	×	Partial Prediction	50.98
√	√	×	Complete Prediction	49.93
√	√	×	Final Prediction	52.55
√	√	√	Partial Prediction	52.46
√	√	√	Complete Prediction	52.34
√	√	√	Final Prediction	**53.77**

Table 3. Ablation experiment results (%) on NTU RGB + D (Cross-View)

action completeness perceptron H_O and the motion predictor H_M can be obtained, and thus the complete sequence generation module is constructed as described in Section 3.4.1 to obtain the complete and fused predictions. Therefore, to verify the usefulness of the global regularization task, after removing this task from the multi-task self-supervised learning framework, comparison experiments are conducted on three networks, i.e., partial prediction, complete prediction, and final prediction, respectively. It can be seen in Table 2 and Table 3 that, after removing the global regularization task, the average prediction accuracy of the partial prediction network is decreased by 1.48 % under the cross-view, the complete prediction network

is decreased by 0.74 % and 2.41 % under the cross-subject and cross-view, respectively, and the final prediction network is decreased by 0.29 % and 1.22 % under the two divisions, respectively. This fully validates the effectiveness of the global regularization task proposed in this paper.

Self-Supervised Tasks	Cross-Subject	Cross-View
Action Completeness Perceptron	0.00968	0.00902
Motion Prediction	0.00559	0.00506

Table 4. Self-supervised tasks loss on NTU RGB + D

The detailed prediction accuracies of the models under different combinations of the various self-supervised tasks at different observation rates are shown in Figure 5 a) and Figure 5 b). It can be visualized from the figures that the accuracy of the partial prediction model at different observation rates all show a significant decrease when there is no supervision of the action completeness perception task or the motion prediction task. In the absence of a global regularization task to optimize the feature space, the accuracy of all three prediction models yielded degradation. This further illustrates the role of each self-supervised task in the multi-task self-supervised learning framework proposed in this paper.

In addition, compared to partial prediction based on incomplete sequence only and complete prediction based on generated sequence only, the average accuracy of the final prediction is improved considerably under both cross-subject and cross-view divisions. This is because the partial prediction is based on incomplete sequence with missing discriminative information, and the complete prediction based on generated sequence can make up for the missing information to a certain extent in the early stage of the action, while the partial prediction can provide increasingly reliable predictions as the action proceeds and more information is contained in the observed incomplete sequences. Therefore, final prediction that combines partial prediction and complete prediction can achieve higher prediction accuracy.

We also provide in Table 4 the difference between the experimental results of the proposed self-supervised tasks and the ground truth.

4.4 Comparison with Other Methods

To evaluate the ability of the learned feature representation capturing action information, we evaluate our model on the action prediction task. In addition, the action prediction task becomes an action recognition task when the observation ratio of a sequence reaches 1.0, so we also evaluate the performance of the ACP + MP + GR + PN proposed in this paper which discriminates the category of action based on the observed segment only on the action recognition task as a complementary experiment to illustrate the effectiveness of our self-supervised training.

Methods		Observation Ratios									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Supervised Methods											
RNN	F-RNN-EL (ICRA 16) [12]	–	7.07	–	18.98	–	44.55	–	63.84	–	71.09
	Weng et al. (TCSVT 20) [44]	29.39	35.56	45.25	54.63	62.07	67.08	70.63	72.91	74.54	75.53
CNN	Ke et al. (CVPR 17) [14]	–	8.34	–	26.97	–	56.78	–	75.13	–	80.43
	EARN + GR + TCE (ACCV 18) [13]	–	–	33.46	–	54.28	–	–	66.21	–	–
	Local + LGN (TIP 19) [15]	–	32.12	–	63.82	–	77.02	–	82.45	–	83.19
GCN	Local + AGCN-AL (TCDS 21) [17]	–	38.18	59.02	71.19	78.29	82.25	–	86.33	–	87.20
	Li et al. (ECCV 20) [16]	–	42.39	–	72.24	–	82.99	–	86.75	–	87.54
Self-Supervised Methods (Linear Evaluation Protocol)											
RNN	ACP + MP + GR + PN (Ours)	15.24	19.61	32.35	45.99	54.64	59.97	63.65	65.66	66.73	70.10
	ACP + MP + GR + CN (Ours)	15.57	21.29	33.74	47.26	55.33	60.55	64.19	66.20	66.53	–
	ACP + MP + GR + FN (Ours)	16.50	22.13	35.46	49.38	57.12	62.11	65.42	67.28	67.83	–
GCN	ACP + MP + GR + PN (Ours)	15.04	23.69	38.67	49.53	57.85	62.51	65.62	67.70	68.50	71.72
	ACP + MP + GR + CN (Ours)	15.54	23.93	38.72	48.64	54.93	58.52	61.51	63.57	64.33	–
	ACP + MP + GR + FN (Ours)	16.95	25.98	41.38	51.84	58.73	62.73	65.55	67.67	68.79	–

Table 5. Action prediction accuracy (%) on NTU RGB + D (Cross-Subject)

Methods		Observation Ratios									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Supervised Methods											
RNN	Weng et al. (TCSVT 20) [44]	30.55	37.22	46.95	57.18	64.97	69.92	73.13	75.41	76.88	77.99
GCN	Li et al. (ECCV 20) [16]	–	53.15	–	82.87	–	91.34	–	93.71	–	94.03
Self-Supervised Methods (Linear Evaluation Protocol)											
RNN	ACP + MP + GR + FN (Ours)	17.10	26.53	42.11	54.26	63.03	67.33	70.08	71.53	71.82	76.30

Table 6. Action prediction accuracy (%) on NTU RGB + D (Cross-View)

Methods		Observation Ratios									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Hand-Crafted Methods											
Hu et al. (ECCV 16) [11]		–	29.58	35.83	35.42	45.83	53.33	–	58.75	–	54.17
Supervised Methods											
RNN	F-RNN-EL (ICRA 16) [12]	–	31.61	–	53.37	–	68.71	–	73.96	–	75.53
CNN	Ke et al. (CVPR 17) [14]	–	26.76	–	52.86	–	72.32	–	79.40	–	80.71
	EARN + GR + TCE (ACCV 18) [13]	–	–	56.06	–	64.35	–	–	69.94	–	–
	Local + LGN (TIP 19) [15]	–	58.81	–	74.21	–	82.18	–	84.42	–	83.14
GCN	Local + AGCN-AL (TCDS 21) [17]	–	63.46	73.65	80.93	85.60	87.92	–	90.38	–	90.47
Self-Supervised Methods (Linear Evaluation Protocol)											
RNN	ACP + MP + GR + PN (Ours)	35.50	46.22	53.06	56.52	60.29	63.31	65.66	66.80	66.24	65.62
	ACP + MP + GR + CN (Ours)	39.74	48.27	54.08	56.83	60.22	64.94	65.91	65.15	65.26	–
	ACP + MP + GR + FN (Ours)	38.59	48.87	54.75	58.00	62.23	65.59	66.76	66.66	66.30	–

Table 7. Action prediction accuracy (%) on SYSU 3D HOI (Cross-Subject)

Methods		Observation Ratios									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Self-Supervised Methods (Linear Evaluation Protocol)											
RNN	ACP + MP + GR + PN (Ours)	35.06	44.41	50.65	53.34	57.33	60.18	62.25	63.52	62.95	63.44
	ACP + MP + GR + CN (Ours)	38.25	46.25	50.76	53.80	56.30	60.47	61.80	61.77	61.34	–
	ACP + MP + GR + FN (Ours)	37.76	47.36	52.26	55.51	58.69	61.81	62.75	62.90	62.38	–

Table 8. Action prediction accuracy (%) on SYSU 3D HOI (Same-Subject)

Datasets	Methods	Observation Ratios								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
NTU RGB + D 120 (X-Sub)	ACP + MP + GR + PN	6.22	9.44	15.04	21.41	27.46	32.24	36.13	38.43	**39.25**
	ACP + MP + GR + CN	6.76	10.14	17.17	23.72	29.55	33.80	36.91	38.56	38.78
	ACP + MP + GR + FN	**7.32**	**10.97**	**17.82**	**24.66**	**30.38**	**34.34**	**37.15**	**38.97**	39.20
NTU RGB + D 120 (X-Set)	ACP + MP + GR + PN	8.17	12.09	19.43	27.28	34.07	39.16	42.41	44.68	45.68
	ACP + MP + GR + CN	8.10	12.83	21.34	29.22	35.54	39.92	43.05	44.82	45.73
	ACP + MP + GR + FN	**9.11**	**13.78**	**22.37**	**30.17**	**36.41**	**40.94**	**43.65**	**45.15**	**45.88**
UAV-Human (v1)	ACP + MP + GR + PN	12.56	16.51	**19.66**	20.15	21.13	**21.64**	21.22	21.17	**21.05**
	ACP + MP + GR + CN	11.37	14.96	17.01	17.93	19.07	19.47	19.76	20.31	19.94
	ACP + MP + GR + FN	**12.88**	**16.95**	19.15	**20.37**	**21.26**	21.15	**21.45**	**21.49**	20.70
UAV-Human (v2)	ACP + MP + GR + PN	22.07	29.35	34.02	36.54	38.71	**38.69**	**39.26**	**38.92**	37.79
	ACP + MP + GR + CN	21.57	27.65	31.32	32.76	34.99	35.96	36.23	36.64	36.32
	ACP + MP + GR + FN	**23.20**	**30.75**	**34.95**	**37.16**	**38.83**	38.60	38.80	38.53	**37.82**

Table 9. Action prediction accuracy (%) on NTU RGB + D 120 and UAV-Human datasets

Datasets	Methods	Observation Ratios								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
NTU RGB + D (Cross-Subject)	ACP + MP + GR + PN	13.09	14.10	18.16	22.94	30.21	34.67	37.63	39.15	39.35
	ACP + MP + GR + CN	11.37	15.11	23.82	30.95	36.47	39.50	39.97	39.97	39.69
	ACP + MP + GR + FN	**13.16**	**15.86**	**24.00**	**31.11**	**37.23**	**40.53**	**41.32**	**40.96**	**40.63**
NTU RGB + D (Cross-View)	ACP + MP + GR + PN	11.55	13.11	16.77	20.87	26.09	29.23	33.15	34.19	33.92
	ACP + MP + GR + CN	10.52	14.56	**22.07**	**28.75**	**33.47**	**34.73**	36.74	35.08	34.95
	ACP + MP + GR + FN	**11.61**	**14.95**	21.62	28.04	33.46	34.68	**37.18**	**36.07**	**35.24**
SYSU 3D HOI (Same-Subject)	ACP + MP + GR + PN	24.44	27.72	30.63	32.05	34.18	35.81	36.58	36.70	36.83
	ACP + MP + GR + CN	25.33	28.13	34.02	**35.84**	**36.87**	**38.36**	**37.62**	37.45	37.36
	ACP + MP + GR + FN	**26.68**	**29.76**	**34.27**	35.65	36.19	37.72	37.52	**37.97**	**37.91**
SYSU 3D HOI (Cross-Subject)	ACP + MP + GR + PN	27.59	32.76	35.98	39.01	41.47	43.47	44.11	44.38	43.91
	ACP + MP + GR + CN	30.13	34.79	39.77	41.95	43.84	45.40	45.38	44.01	42.77
	ACP + MP + GR + FN	**31.59**	**36.97**	**40.62**	**42.84**	**44.40**	**45.41**	**45.69**	**45.30**	**44.59**

Table 10. Action prediction accuracy (%) on NTU RGB + D and SYSU 3D HOI datasets under KNN evaluation protocol

4.4.1 Action Prediction

There are still relatively few studies on 3D action prediction (or skeleton-based early action recognition). Therefore we compare with other supervised 3D action prediction methods as a reference for the performance of our proposed self-supervised method.

The experimental results on the NTU RGB + D dataset are shown in Tables 5 and 6. We follow the Cross-Subject criteria in Table 5 as other methods do. We present the performance of ACP + MP + GR + FN under linear evaluation protocol. It can be seen that our proposed RNN-based self-supervised ACP + MP + GR + FN has outperformed some supervised methods [12, 14, 13] which require a large number of action labels to train the entire model, while we have only trained a single fully connected layer with labels. The weights of the feature extractor E, action completeness perceptron H_O, and motion predictor H_M of our model are obtained and fixed by self-supervised training without labels. More specifically, the accuracy of our method is on average 16.61 % higher than [12], 8.42 % higher than [14], and 1.97 % higher than [13]. Compared to the state-of-the-art GCN-based supervised approach [16], our method is only 20.86 % lower on average at each observation ratio. Note that our approach uses only the simple GRU as the baseline, which has a faster speed and fewer parameters than GCN. And the performance of our method has greatly outperformed the supervised method [12] that also uses RNN as the baseline, which is on average 16.61 % lower than ours at each observation ratio. Compared to supervised RNN method using reinforcement learning for category exclusion [44], our self-supervised approach remains competitive, with an average accuracy of only 7.63 % lower than that at all observation rates. The performance of our method is further improved with GCN as the backbone. It can be seen that our GCN-based ACP + MP + GR + FN achieves a large performance improvement when the observation rate is small. Compared to our RNN-based approach, it improves the accuracy by 3.85 % at an observation rate of 0.2, 5.92 % at 0.3, and 2.46 % at 0.4.

The experimental results on the NTU RGB + D dataset under the Cross-View criteria are shown in Table 6. It can be seen that our self-supervised method is only 5.38 % lower than the best supervised RNN-based method [44] at each observation rate. Compared to the state-of-the-art GCN-based supervised approach [16], our method is only 25.35 % lower on average at each observation ratio.

The experimental results on the SYSU 3D HOI dataset under cross-subject protocol are shown in Table 7 where performance of ACP + MP + GR + FN under linear evaluation protocol is presented. The good performance in the early stage of the action demonstrates the effectiveness of our proposal to capture past and future information in the action sequence for the action prediction task. Compared with the state-of-the-art GCN-based supervised approach [17], our method is only 20.97 % lower on average at each observation ratio. Note that in [17], the authors use parameters trained on the NTU RGB + D dataset as initialization, while we only perform self-supervised training on the SYSU dataset.

Currently no methods provide experimental results on the SYSU 3D HOI dataset under the same-subject protocol. In Table 8, we present the performance of our method to provide a baseline for the community.

In addition, to provide more sufficient validation, we have conducted experiments on NTU RGB + D 120 and UAV-Human, two large-scale datasets with action category numbers of 120 and 155, respectively, and the results are shown in Table 9.

The KNN evaluation results on the NTU RGB + D dataset and SYSU 3D HOI dataset are shown in Table 10. Compared to ACP + MP + GR + PN, which uses only partial sequence features, ACP + MP + GR + CN, which uses complete sequence features, and ACP + MP + GR + FN, which combines both features, have substantially improved their performance. This indicates that our complete sequence generation module alleviates the missing information in incomplete sequences to some extent and encodes features with more discrimination and integrity.

4.4.2 Action Recognition

Unlike 3D action prediction, there have been many self-supervised or unsupervised methods of 3D action recognition. For fairness of comparison, we show in Table 11 the performance comparison with other self-supervised or unsupervised RNN-based action recognition methods proposed in recent years using the same encoder as ours.

Methods	X-Sub	X-View
LongTGAN (AAAI 18) [8]	39.1	48.1
MS^{2}L (ACM MM 20) [9]	52.6	–
* Gao et al. (NeurIPS Workshop 20) [37]	52.3	62.1
PCRP (TMM 21) [51]	53.9	63.5
CAE + (Information Sciences 21) [52]	58.5	64.8
P & C FW-AEC (CVPR 20) [53]	50.7	76.1
3s-CrosSCLR (LSTM) (CVPR 21) [38]	62.8	69.2
CRRL (IEEE TIP 22) [40]	67.6	73.8
ACP + MP + GR + PN (Ours)	**70.1**	**76.3**

Table 11. Comparison of self-supervised or unsupervised RNN-based action recognition methods on NTU RGB + D. "*" represents depth image based methods.

Protocols	Methods	Accuracy
Same-Subject	ACP + MP + GR + PN	63.44
	ACP + MP + GR + CN	63.01
	ACP + MP + GR + FN	63.48
Cross-Subject	ACP + MP + GR + PN	65.62
	ACP + MP + GR + CN	65.40
	ACP + MP + GR + FN	65.63

Table 12. Action recognition accuracy (%) on SYSU 3D HOI

For the action recognition task, the input to the model is the complete sequence and we do not need to predict the future motion of the human body. Therefore, ACP + MP + GR + PN is adopted to accomplish this task. The encoder is trained by self-supervised tasks (ACP + MP + GR) and then fixes its weights to train an FC layer to perform action classification. As shown in the Table 11, applying our ACP + MP + GR + PN directly to the skeleton-based action recognition task yields superior performance. Our method has outperformed all the advanced unsupervised RNN-based action recognition methods. Our method outperforms the state-of-the-art RNN-based unsupervised method CRRL [40] by 2.5 % in Cross-Subject criteria and is 0.2 % higher than P & C FW-AEC [53] in Cross-View criteria. Note that 3s-CrosSCLR (LSTM) [38] utilizes views from three data modalities and CRRL [40] simultaneously uses information from two streams, while our ACP + MP + GR + PN is a single stream network using only joint information. Most of these methods employ complex contrastive learning strategies to construct positive and negative samples for feature learning, while our method still achieves excellent performance with only self-supervised tasks using past and future information. This indicates that the feature representation learned by the self-supervised tasks based on past and future information proposed in this paper has a strong ability to describe the action.

In addition, to provide a baseline for the community to refer to, we also show the performance of the proposed method for action recognition on the SYSU dataset in Table 12.

5 CONCLUSION

In this paper, we have proposed a self-supervised learning approach for 3D action prediction. We believe the unfinished skeleton sequence contains past and future information contributing to action prediction. We designed three self-supervised tasks to simultaneously guide the training of the network based on past completeness and future trends, while optimizing the feature space. In addition, we proposed a novel 3D action prediction network that employed the feature encoder and modules trained by self-supervised tasks as components to fuse both partial prediction and complete prediction. Through ablation experiments and the performance comparisons on two datasets with two tasks, we demonstrated the effectiveness of our proposed method. We hoped our work could inspire future researchers to conduct more studies on label-free learning for 3D action prediction.

Acknowledgements

This work was supported by the Shenzhen Science and Technology Projects (No. JCYJ20200109143035495), Natural Science Foundation of Fujian Province (No. 2022J011275 and No. 2023J01003) and Zhejiang Province's R & D Key Project (No. 2024C03036).

REFERENCES

[1] Ryoo, M. S.: Human Activity Prediction: Early Recognition of Ongoing Activities from Streaming Videos. 2011 International Conference on Computer Vision, IEEE, 2011, pp. 1036–1043, doi: 10.1109/ICCV.2011.6126349.

[2] Cao, Y.—Barrett, D.—Barbu, A.—Narayanaswamy, S.—Yu, H.—Michaux, A.—Lin, Y.—Dickinson, S.—Siskind, J. M.—Wang, S.: Recognize Human Activities from Partially Observed Videos. 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2658–2665, doi: 10.1109/CVPR.2013.343.

[3] Han, J.—Shao, L.—Xu, D.—Shotton, J.: Enhanced Computer Vision with Microsoft Kinect Sensor: A Review. IEEE Transactions on Cybernetics, Vol. 43, 2013, No. 5, pp. 1318–1334, doi: 10.1109/TCYB.2013.2265378.

[4] Martinez, J.—Hossain, R.—Romero, J.—Little, J. J.: A Simple Yet Effective Baseline for 3D Human Pose Estimation. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2640–2649, doi: 10.1109/ICCV.2017.288.

[5] Xu, J.—Yu, Z.—Ni, B.—Yang, J.—Yang, X.—Zhang, W.: Deep Kinematics Analysis for Monocular 3D Human Pose Estimation. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 896–905, doi: 10.1109/CVPR42600.2020.00098.

[6] Wang, J.—Liu, Z.—Wu, Y.—Yuan, J.: Mining Actionlet Ensemble for Action Recognition with Depth Cameras. 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1290–1297, doi: 10.1109/CVPR.2012.6247813.

[7] Xia, L.—Chen, C. C.—Aggarwal, J. K.: View Invariant Human Action Recognition Using Histograms of 3D Joints. 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 20–27, doi: 10.1109/CVPRW.2012.6239233.

[8] Zheng, N.—Wen, J.—Liu, R.—Long, L.—Dai, J.—Gong, Z.: Unsupervised Representation Learning with Long-Term Dynamics for Skeleton Based Action Recognition. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018, No. 1, pp. 2644–2651, doi: 10.1609/aaai.v32i1.11853.

[9] Lin, L.—Song, S.—Yang, W.—Liu, J.: MS2L: Multi-Task Self-Supervised Learning for Skeleton Based Action Recognition. Proceedings of the 28th ACM International Conference on Multimedia (MM '20), 2020, pp. 2490–2498, doi: 10.1145/3394171.3413548.

[10] Si, C.—Nie, X.—Wang, W.—Wang, L.—Tan, T.—Feng, J.: Adversarial Self-Supervised Learning for Semi-Supervised 3D Action Recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. M. (Eds.): Computer Vision – ECCV 2020. Springer, Cham, Lecture Notes in Computer Science, Vol. 12352, 2020, pp. 35–51, doi: 10.1007/978-3-030-58571-6_3.

[11] Hu, J. F.—Zheng, W. S.—Ma, L.—Wang, G.—Lai, J.: Real-Time RGB-D Activity Prediction by Soft Regression. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9905, 2016, pp. 280–296, doi: 10.1007/978-3-319-46448-0_17.

[12] Jain, A.—Singh, A.—Koppula, H. S.—Soh, S.—Saxena, A.: Recurrent Neural Networks for Driver Activity Anticipation via Sensory-Fusion Architecture. 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 3118–3125, doi: 10.1109/ICRA.2016.7487478.

[13] Ke, Q.—Liu, J.—Bennamoun, M.—Rahmani, H.—An, S.—Sohel, F.—Boussaid, F.: Global Regularizer and Temporal-Aware Cross-Entropy for Skeleton-Based Early Action Recognition. In: Jawahar, C. V., Li, H., Mori, G., Schindler, K. (Eds.): Computer Vision – ACCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11364, 2018, pp. 729–745, doi: 10.1007/978-3-030-20870-7_45.

[14] Ke, Q.—Bennamoun, M.—An, S.—Sohel, F.—Boussaid, F.: A New Representation of Skeleton Sequences for 3D Action Recognition. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4570–4579, doi: 10.1109/CVPR.2017.486.

[15] Ke, Q.—Bennamoun, M.—Rahmani, H.—An, S.—Sohel, F.—Boussaid, F.: Learning Latent Global Network for Skeleton-Based Action Prediction. IEEE Transactions on Image Processing, Vol. 29, 2019, pp. 959–970, doi: 10.1109/TIP.2019.2937757.

[16] Li, T.—Liu, J.—Zhang, W.—Duan, L.: HARD-Net: Hardness-AwaRe Discrimination Network for 3D Early Activity Prediction. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. M. (Eds.): Computer Vision – ECCV 2020. Springer, Cham, Lecture Notes in Computer Science, Vol. 12356, 2020, pp. 420–436, doi: 10.1007/978-3-030-58621-8_25.

[17] Li, G.—Li, N.—Chang, F.—Liu, C.: Adaptive Graph Convolutional Network with Adversarial Learning for Skeleton-Based Action Prediction. IEEE Transactions on Cognitive and Developmental Systems, Vol. 14, 2022, No. 3, doi: 10.1109/TCDS.2021.3103960.

[18] Luo, W.—Zhang, T.—Yang, W.—Liu, J.—Mei, T.—Wu, F.—Zhang, Y.: Action Unit Memory Network for Weakly Supervised Temporal Action Localization. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 9964–9974, doi: 10.1109/CVPR46437.2021.00984.

[19] Lee, P.—Byun, H.: Learning Action Completeness from Points for Weakly-Supervised Temporal Action Localization. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 13628–13637, doi: 10.1109/ICCV48922.2021.01339.

[20] Li, G.—Cheng, D.—Ding, X.—Wang, N.—Li, J.—Gao, X.: Weakly Supervised Temporal Action Localization with Bidirectional Semantic Consistency Constraint. IEEE Transactions on Neural Networks and Learning Systems, Vol. 35, 2024, No. 9, pp. 13032–13045, doi: 10.1109/TNNLS.2023.3266062.

[21] Liu, C.—Gao, Y.—Li, Z.—Du, C.—Liu, F.—Shi, X.: Action Prediction Network with Auxiliary Observation Ratio Regression. 2021 IEEE International Conference on Multimedia and Expo (ICME), 2021, pp. 1–6, doi: 10.1109/ICME51207.2021.9428266.

[22] Cui, Q.—Sun, H.: Towards Accurate 3D Human Motion Prediction from Incomplete Observations. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 4799–4808, doi: 10.1109/CVPR46437.2021.00477.

[23] Dang, L.—Nie, Y.—Long, C.—Zhang, Q.—Li, G.: MSR-GCN: Multi-Scale Residual Graph Convolution Networks for Human Motion Prediction. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 11447–11456, doi: 10.1109/ICCV48922.2021.01127.

[24] Elsner, B.—Hommel, B.: Effect Anticipation and Action Control. Journal of Experimental Psychology: Human Perception and Performance, Vol. 27, 2001, No. 1, pp. 229–240, doi: 10.1037/0096-1523.27.1.229.

[25] Gammulle, H.—Denman, S.—Sridharan, S.—Fookes, C.: Predicting the Future: A Jointly Learnt Model for Action Anticipation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 5561–5570, doi: 10.1109/ICCV.2019.00566.

[26] Dong, M.—Xu, C.: Skeleton-Based Human Motion Prediction with Privileged Supervision. IEEE Transactions on Neural Networks and Learning Systems, Vol. 34, 2023, No. 12, pp. 10419–10432, doi: 10.1109/TNNLS.2022.3166861.

[27] Hussein, M. E.—Torki, M.—Gowayyed, M. A.—El-Saban, M.: Human Action Recognition Using a Temporal Hierarchy of Covariance Descriptors on 3D Joint Locations. Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI'13), 2013, pp. 2466–2472, https://www.ijcai.org/Proceedings/13/Papers/363.pdf.

[28] Seidenari, L.—Varano, V.—Berretti, S.—Bimbo, A. D.—Pala, P.: Weakly Aligned Multi-Part Bag-of-Poses for Action Recognition from Depth Cameras. In: Petrosino, A., Maddalena, L., Pala, P. (Eds.): New Trends in Image Analysis and Processing – ICIAP 2013. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 8158, 2013, pp. 446–455, doi: 10.1007/978-3-642-41190-8_48.

[29] Vemulapalli, R.—Arrate, F.—Chellappa, R.: Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 588–595, doi: 10.1109/CVPR.2014.82.

[30] Shahroudy, A.—Liu, J.—Ng, T. T.—Wang, G.: NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1010–1019, doi: 10.1109/CVPR.2016.115.

[31] Yan, S.—Xiong, Y.—Lin, D.: Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018, No. 1, pp. 7444–7452, doi: 10.1609/aaai.v32i1.12328.

[32] Shi, L.—Zhang, Y.—Cheng, J.—Lu, H.: Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12018–12027, doi: 10.1109/CVPR.2019.01230.

[33] Chen, Y.—Zhang, Z.—Yuan, C.—Li, B.—Deng, Y.—Hu, W.: Channel-Wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 13339–13348, doi: 10.1109/ICCV48922.2021.01311.

[34] Li, Y.—Yang, M.—Peng, D.—Li, T.—Huang, J.—Peng, X.: Twin Con-

trastive Learning for Online Clustering. International Journal of Computer Vision, Vol. 130, 2022, No. 9, pp. 2205–2221, doi: 10.1007/s11263-022-01639-z.

[35] Li, Y.—Hu, P.—Liu, Z.—Peng, D.—Zhou, J. T.—Peng, X.: Contrastive Clustering. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, No. 10, pp. 8547–8555, doi: 10.1609/aaai.v35i10.17037.

[36] Lin, Y.—Gou, Y.—Liu, X.—Bai, J.—Lv, J.—Peng, X.: Dual Contrastive Prediction for Incomplete Multi-View Representation Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 45, 2023, No. 4, pp. 4447–4461, doi: 10.1109/TPAMI.2022.3197238.

[37] Gao, X.—Yang, Y.—Du, S.: Contrastive Self-Supervised Learning for Skeleton Action Recognition. In: Bertinetto, L., Henriques, J. F., Albanie, S., Paganini, M., Varol, G. (Eds.): NeurIPS 2020 Workshop on Pre-Registration in Machine Learning. Proceedings of Machine Learning Research (PMLR), Vol. 148, 2021, pp. 51–61, `https://proceedings.mlr.press/v148/gao21a.html`.

[38] Li, L.—Wang, M.—Ni, B.—Wang, H.—Yang, J.—Zhang, W.: 3D Human Action Representation Learning via Cross-View Consistency Pursuit. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 4739–4748, doi: 10.1109/CVPR46437.2021.00471.

[39] Su, Y.—Lin, G.—Wu, Q.: Self-Supervised 3D Skeleton Action Representation Learning with Motion Consistency and Continuity. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 13308–13318, doi: 10.1109/ICCV48922.2021.01308.

[40] Wang, P.—Wen, J.—Si, C.—Qian, Y.—Wang, L.: Contrast-Reconstruction Representation Learning for Self-Supervised Skeleton-Based Action Recognition. IEEE Transactions on Image Processing, Vol. 31, 2022, pp. 6224–6238, doi: 10.1109/TIP.2022.3207577.

[41] Thoker, F. M.—Doughty, H.—Snoek, C. G. M.: Skeleton-Contrastive 3D Action Representation Learning. Proceedings of the 29th ACM International Conference on Multimedia (MM '21), 2021, pp. 1655–1663, doi: 10.1145/3474085.3475307.

[42] Guo, T.—Liu, H.—Chen, Z.—Liu, M.—Wang, T.—Ding, R.: Contrastive Learning from Extremely Augmented Skeleton Sequences for Self-Supervised Action Recognition. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, No. 1, pp. 762–770, doi: 10.1609/aaai.v36i1.19957.

[43] Liu, J.—Shahroudy, A.—Wang, G.—Duan, L. Y.—Kot, A. C.: Skeleton-Based Online Action Prediction Using Scale Selection Network. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 42, 2020, No. 6, pp. 1453–1467, doi: 10.1109/TPAMI.2019.2898954.

[44] Weng, J.—Jiang, X.—Zheng, W. L.—Yuan, J.: Early Action Recognition with Category Exclusion Using Policy-Based Reinforcement Learning. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 30, 2020, No. 12, pp. 4626–4638, doi: 10.1109/TCSVT.2020.2976789.

[45] Cho, K.—Van Merriënboer, B.—Gulcehre, C.—Bahdanau, D.—Bougares, F.—Schwenk, H.—Bengio, Y.: Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. 2014,

pp. 1724–1734, doi: 10.3115/V1/D14-1179.

[46] Schuster, M.—Paliwal, K. K.: Bidirectional Recurrent Neural Networks. IEEE Transactions on Signal Processing, Vol. 45, 1997, No. 11, pp. 2673–2681, doi: 10.1109/78.650093.

[47] Hu, J. F.—Zheng, W. S.—Lai, J.—Zhang, J.: Jointly Learning Heterogeneous Features for RGB-D Activity Recognition. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5344–5352, doi: 10.1109/CVPR.2015.7299172.

[48] Liu, J.—Shahroudy, A.—Perez, M.—Wang, G.—Duan, L. Y.—Kot, A. C.: NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 42, 2020, No. 10, pp. 2684–2701, doi: 10.1109/TPAMI.2019.2916873.

[49] Li, T.—Liu, J.—Zhang, W.—Ni, Y.—Wang, W.—Li, Z.: UAV-Human: A Large Benchmark for Human Behavior Understanding with Unmanned Aerial Vehicles. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 16261–16270, doi: 10.1109/CVPR46437.2021.01600.

[50] Zhang, P.—Lan, C.—Zeng, W.—Xing, J.—Xue, J.—Zheng, N.: Semantics-Guided Neural Networks for Efficient Skeleton-Based Human Action Recognition. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 1109–1118, doi: 10.1109/CVPR42600.2020.00119.

[51] Xu, S.—Rao, H.—Hu, X.—Cheng, J.—Hu, B.: Prototypical Contrast and Reverse Prediction: Unsupervised Skeleton Based Action Recognition. IEEE Transactions on Multimedia, Vol. 25, 2021, pp. 624–634, doi: 10.1109/TMM.2021.3129616.

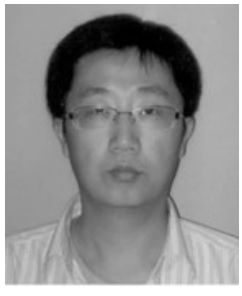
[52] Rao, H.—Xu, S.—Hu, X.—Cheng, J.—Hu, B.: Augmented Skeleton Based Contrastive Action Learning with Momentum LSTM for Unsupervised Action Recognition. Information Sciences, Vol. 569, 2021, pp. 90–109, doi: 10.1016/j.ins.2021.04.023.

[53] Su, K.—Liu, X.—Shlizerman, E.: PREDICT & CLUSTER: Unsupervised Skeleton Based Action Recognition. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9628–9637, doi: 10.1109/CVPR42600.2020.00965.

Yifan Wang received his M.Sc. degree in computer technology from the Xiamen University, Fujian, China. His current research interests include machine learning and computer vision.

Tiancheng Zheng is currently studying at the Xiamen University in China. He is currently a Ph.D. candidate in law at the Xiamen University. His main research interest is data method.

Hua Shi is a Lecturer of the School of Opto-Electronic and Communication Engineering, Xiamen University of Technology in China. He received his Ph.D. from the Xiamen University, P.R. China in 2014. His research is in the areas of machine learning, computer vision, and artificial intelligence.



Chong Zhao received his B.Sc. in the Department of Computer Science from the Jilin University, his M.Sc. in the Academy of Mathematics and Systems Science from the Chinese Academy of Sciences, and his Ph.D. in the Department of Computer Science and Engineering from the Chinese University of Hong Kong. He is currently Assistant Professor in the Department of Computer Science, Xiamen University. His research interests include geometry processing, computer graphics and computer vision.

Eryong Wu received his Ph.D. degree in information and communication engineering from the Zhejiang University, China in 2007. He is currently a Research Faculty Member at the Zhejiang University. His research interests include ultrasonic imaging testing, computer vision, and robot navigation.