Self-supervised 3D Skeleton Action Representation Learning with Motion Consistency and Continuity

Yukun Su[1,2], Guosheng Lin[3†], and Qingyao Wu[1,2†]

[1]School of Software and Engineering, South China University of Technology
[2]Key Laboratory of Big Data and Intelligent Robot, Ministry of Education
[3]School of Computer Science and Engineering, Nanyang Technological University
suyukun666@gmail.com, gslin@ntu.edu.sg, qyw@scut.edu.cn

Abstract

Recently, self-supervised learning (SSL) has been proved very effective and it can help boost the performance in learning representations from unlabeled data in the image domain. Yet, very little is explored about its usefulness in 3D skeleton-based action recognition understanding. Directly applying existing SSL techniques for 3D skeleton learning, however, suffers from trivial solutions and imprecise representations. To tackle these drawbacks, we consider perceiving the consistency and continuity of motion at different playback speeds are two critical issues. To this end, we propose a novel SSL method to learn the 3D skeleton representation in an efficacious way. Specifically, by constructing a positive clip (speed-changed) and a negative clip (motion-broken) of the sampled action sequence, we encourage the positive pairs closer while pushing the negative pairs to force the network to learn the intrinsic dynamic motion consistency information. Moreover, to enhance the learning features, skeleton interpolation is further exploited to model the continuity of human skeleton data. To validate the effectiveness of the proposed method, extensive experiments are conducted on Kinetics, NTU60, NTU120, and PKUMMD datasets with several alternative network architectures. Experimental evaluations demonstrate the superiority of our approach and through which, we can gain significant performance improvement without using extra labeled data.

1. Introduction

In recent years, 3D action recognition based on skeleton has made remarkable progress through learning discriminative features with deep learning networks [31, 33, 37, 49]. However, these methods rely heavily on supervision, and collecting such labels is very time-consuming and labor-intensive. This makes the development of unsupervised learning techniques and the use of a large amount of unlabeled data the urgent needs, and among them a powerful approach is self-supervised learning (SSL). In image domain, as images contain rich information that is beneficial to feature extraction, many effective SSL techniques [3, 6, 11, 38] are well exploited. Comparatively, for tasks over skeleton data which represent a person by 3D coordinate positions of key joints, it becomes very challenging to leverage SSL techniques to learn discriminative motion representation.

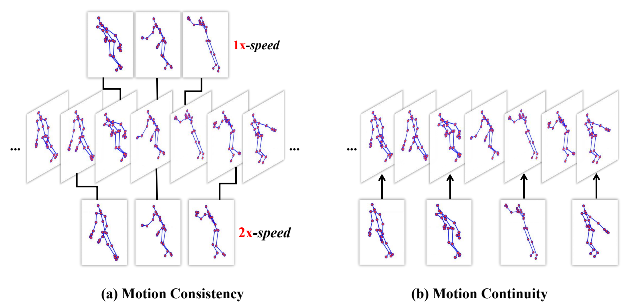

Figure 1. **An illustrative example (*i.e.*, "jump up") of our main idea**. **(a) Motion consistency.** Although the two clips are sampled consecutively (1x speed) and alternately (2x speed), respectively, we can easily tell they are similar because they share the same underlying skeletal movements and the consistent motion trends. **(b) Motion continuity.** When the sampling interval is set to 2 frames, the complemented motion of the interval frames should make the whole temporal motion look natural and coherent.

Some recent methods [53, 18] attempt to solve these challenges by directly adopting the existing video SSL tech-

[†]Corresponding authors.

niques on skeleton data such as using motion prediction [7], jigsaw puzzle recognition [26] and temporal clip orders prediction [48] as pretext tasks. As for sequence data, playback rate perception [1, 43] achieves great success and is the most common way to model spatial-temporal information, which can help networks to learn representative motion features. However, directly applying these methods to skeleton data suffer from two limitations: **(1)** Human skeleton motions in nature move at different speeds, and predicting different absolute playback speeds of the sequence is ambiguous, which will yield trivial solutions as mentioned in [11]. Namely, the network can easily predict the corresponding rates by simply remembering certain frames, this is harmful to features representation learning. **(2)** Unlike the video data, 3D skeleton only contains dynamic motion information but without appearance information. Such methods as in [46, 43] that explore instance appearance features are not suitable for the skeleton data, which will cause imprecise learning representations. Therefore, how to extend the existing SSL methods to the skeleton domain is a challenging task and has not been well explored.

Motivation. Inspired by human visual intuition, we observe that perceiving the motion consistency and continuity are two critical issues for learning motion representation. As shown in Figure 1(a), the same motion clips with different playback speeds look similar to each other since they share the intrinsic **motion consistency** (*i.e*., squat-down, leg-lifted). Further to say, we will not consider of an accelerating "walking" motion (*i.e*., 2x playback speed) as a "jumping" motion because they don't have the same underlying motion. In addition, as shown in Figure 1(b), we argue that it is possible for us to imagine the correlation between the missing frames when we have fully learned the motion since each clip has the property of **motion continuity**.

Based on the above observation, we propose a novel SSL method to learn the 3D skeleton representation in an efficacious way. Specifically, we construct two clips from the same sampled motion sequence as positive and negative pairs, respectively. Then we train the networks to distinguish their intrinsic motion consistency instead of predicting the specific playback speed of each video clip. The positive pairs are with the same motion but different playback speeds, while the negative pairs are with the same playback speeds but motion-broken. Our objective is to pull the positive closer while pushing the negative farther to the original clip in the latent space. In this sense, the networks can pay more attention to the skeleton dynamic motion information so as to learn discriminative feature representation.

Moreover, to encourage the networks to learn the enhanced motion features, we design a skeleton interpolation module, which aims to model the motion continuity of human skeleton data. In this task, the input actions at different playback speeds are reconstructed to the actions of a particular interpolation rate. Namely, some accelerating motion can complement the dynamic information of the missing frames (*e.g*., a 2x playback speed motion can be interpolated into a 1x playback speed motion) to establish the learning of motion coherence, so as to have better representation of the underlying motion features.

In the proposed self-supervised framework, we utilize different deep neural networks as our backbones to learn skeleton representation. To validate the effectiveness of our approach in deep learning for 3D skeleton-based action understanding, we conduct massive experiments covering different settings, including self-supervised pre-training, fine-tuning on downstream tasks and semi-supervised training. Experimental results show the superiority of our proposed method and we can significantly boost the performance without using any extra labeled data. The main contributions of our paper can be summarized as follows:

- We propose a novel approach for self-supervised skeleton representation learning by perceiving motion consistency and continuity, through which, we can drive the network to learn the discriminative motion representation features.
- By constructing speed-changed and motion-broken clips, we encourage the positive pairs closer while pushing the negative pairs to force the network to learn the intrinsic motion consistency information. Moreover, skeleton interpolation is further exploited to model the continuity of human skeleton data to enhance the learning features.
- Extensive experimental evaluations on three network architectures under several settings show the effectiveness of our proposed approach powered by self-supervised pre-training. We consider these findings will encourage more research on unsupervised pretext task design for 3D skeleton action understanding.

2. Related Work

2.1. Skeleton-based Action Recognition

Human skeletons can well reflect the nature of human activities. Some early work [40, 41, 44] identified actions by using the geometric relationship between bones and joints. However, the performance of these handcrafted-feature-based methods is unsatisfactory. Benefiting from the deep neural networks, data-driven methods have become the mainstream methods. CNN-based methods [19, 13, 22] converted skeletal data into pseudo-image data by designing transformation rules, and then perform convolution operation. Leveraging the merits of recurrent layers, many works [54, 50, 51] utilized Recurrent Neural Networks (RNN) to model long-short term temporal evolution of different actions. However, both RNNs and CNNs fail to fully represent the structure of the skeleton data because the skeleton data are naturally embedded in the form of graphs rather than a vector sequence or 2D grids. Recently, Graph CNNs [5, 25] showed advantages of graph representation in many tasks for non-Euclidean data, as it can naturally deal with these irregular structures. ST-GCN [49] first proposed the spatial-temporal graph convolution aim at modeling dynamic skeletons sequences. Subsequently, Shi et al. [32] employed the two-stream method to add an adaptive dynamic learning module to improve the action recognition accuracy. In [17], Li et al. explored the A-links and S-links from input data for capturing actional dependencies and then refine them during training. Also, there are some other graph-based approaches [52, 4] with lower computational complexity.

2.2. Self-Supervised Learning

Image: Self-supervised learning aims to learn feature representations from a large amount of unlabeled data, which is usually achieved by setting different pretext tasks and utilize easy-to-obtain automatically generated supervision. In the image domain, [16] performed image colorization pretext to establish a mapping from objects to colors. In recent studies, some works [26, 45] tried to solve jigsaw problems to learn the information of different patches in the images. Komodakis *et al.* [15] proposed a simple rotation transformation to make the network to predict different rotation degrees of the images to identify object's features. Later, such transformations as scaling, warping and inpainting have been applied to the latest work [11]. With the birth of the contrastive learning paradigm [3, 9], most of the current works [47, 8] explored to construct positive pairs and negative pairs for feature learning.

Video: In terms of the video domain, many methods in the field of 2D are still applicable to 3D field. Some previous video self-supervised learning methods focused on learning features from static images [42] and from segmenting objects using optical flow [27]. Recently, some works paid much more attention to model the temporal information from videos. Xu *et al.* [48] shuffled the order of video clips and force the network to predict different orders. Luo *et al.* [23] generated blanks by withholding video clips and created options by applying spatial-temporal operations on the withheld clips for features learning. More recently, many works [1, 43] have been proposed to learn features through discriminating playback speeds.

Skeleton: As aforementioned, there is little previous investigation on skeleton self-supervised learning. Although [53] proposed a skeleton inpainting architecture to learn the long-term dynamics and [36] utilized Predict & Cluster manner to learn features. However, they ignored the high-level semantic and spatial-temporal information of the skeleton and thus may yield less discriminative feature representations. Besides, they only measure their capability under the limited settings. Si *et al.* [34] proposed the adversarial SSL learning for only the semi-supervised setting. Lin *et al.* [18] applied the existing SSL techniques to skeleton data, which we have discussed it may suffer from some limitations.

Hence, we propose an effective self-supervised strategy to learn the representation that is beneficial for 3D skeleton-based action recognition. Meanwhile, we hope to unify the evaluation standards (*e.g.*, use certain networks as the backbones and evaluate on self-supervised pre-training, fine-tuning on downstream tasks like in 2D image domain) to facilitate more follow-up researches in this field.

3. Method

Problem definition. Let $\mathcal{M}$=$\{m_i\}_{i=1}^N$ be a skeleton motion set containing N sequences. We sample a clip $c_i \in m_i$ from the action set with r_i playback speed. Our goal task is to learn an encoder $f(\cdot;\theta)$ in a self-supervised manner that models the skeleton clips c_i to its corresponding features x_i that best represents the spatial-temporal features of the motion in the latent space.

3.1. Spatial-Motion Consistency

Given a skeleton action sequence, we first sample 3 clips c_i, c_j and c_k with playback speeds r_i, r_j and r_k, respectively. Consider the temporal ambiguity among action sequences, we sample a fixed length of 32 frames of each clip as a learning sample, and the start frame of each sample is randomly chosen. Typically, we consider 4 playback speed candidates, where the corresponding speeds r are $1\times$, $2\times$, $4\times$, $8\times$, respectively. For example, when r = 2 and starting from the 10^{th} frame, it contains frame $\{10, 12, 14, ..., 72, 74\}$ in total length of 32 frames. If the desired training clip is longer than the original skeleton sequence sample, we will loop over it from the start.

As is shown in Figure 2, the core idea of the motion modeling module is to maintain the spatial-motion consistency of the positive pairs while breaking the spatial regions of the negative pairs. To this end, we apply different transformations on the three sampled input clips respectively to construct a triplet, *e.g.*, basic $b = S(r_i, c_i)$, positive sample $p = S(r_j, c_j)$ and negative sample $n = B(r_k, c_k)$, where $r_j \neq r_i = r_k$ and $S(r,)$ indicates the operation of uniformly sampling with the same interval frames r, $B(r,)$ denotes the operation of randomly breaking the subsampling skeleton (*i.e.*, shuffle data). We observe that compared to b, negative n shuffles the skeleton sequences destroying the underlying content of the motion and it breaks the motion semantics of the original movement. As for positive p, it changes the speed but retains the spatial and structural information keeping the intrinsic motion consistency as b.

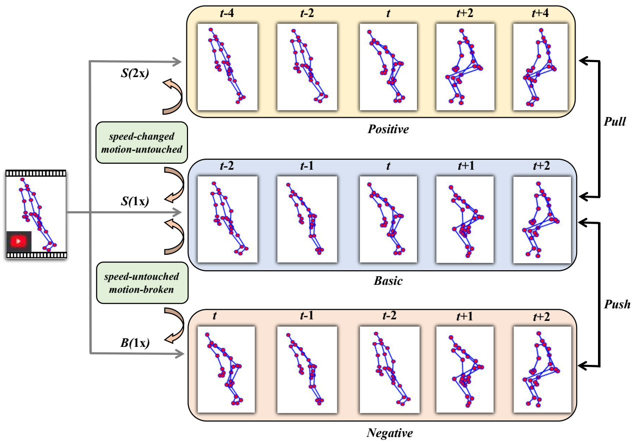

Figure 2. Motion consistency modeling module. The positive pairs are constructed by speed-changed operation while the negative pairs are constructed by motion-broken operation. Then we map the triplet by the encoder into a feature space. The objective is to pull the positive pairs closer while pushing away the negative pairs.

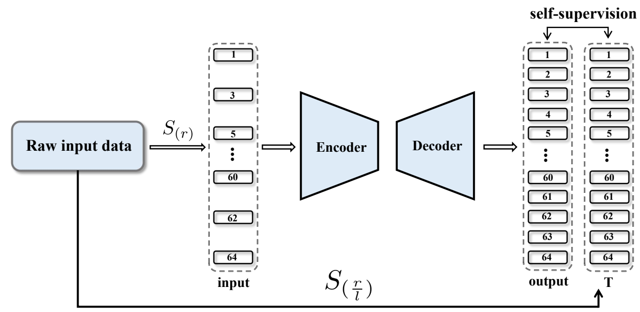

Figure 3. Motion continuity modeling module. The encoder is shared with the motion consistency modeling module. The input is sampled with the r playback speed and the output is the upsampling-interpolated motion by the decoder with the interpolation rate of l. The ground-truth of input $S(r)$ can be sampled online from the raw input data by operation $S(\frac{r}{l})$.

Afterwards, we train the network encoder $f(\cdot;\theta)$ and project the triplet (c_i, c_j, c_k) to an embedding feature space, and term them as x_i, x_j and x_k, respectively. We expect the features of positive pairs to be closer compared with the negative pairs. The assumption behind this is that the networks must first learn to understand the underlying content of skeleton motion before they can distinguish the triplet. Formally, we can achieve this goal by using a triplet loss [29] as follows:

$$\mathcal{L}_{triplet} = \max(0, \gamma - (d(x_i, x_j) - d(x_i, x_k))), \quad (1)$$

where $\gamma > 0$ is a margin hyperparameter, $d(x_i, x_j) = ||x_i - x_j||_2$ and $d(x_i, x_k) = ||x_i - x_k||_2$.

It is worth mentioning that we only consider to construct the negative pairs within the same skeleton action sequence. In the training process, we can also use other action sequences as negative samples to train our network to learn more deep motion representation features. Specifically, we maintain (c_i, c_j) as the positive pairs and sample K clips $\{c_n\}_{n=1}^{K}$ from other samples to form in (c_i, c_n) as the negative pairs. We then apply the InfoNCE loss [10] as the training loss to fulfill this objective:

$$\mathcal{L}_{NCE} = -log\frac{exp(d(x_i, x_j)/\tau)}{exp(d(x_i, x_j)/\tau) + \sum_{n=1}^{K} exp(d(x_i, x_n)/\tau)}, \quad (2)$$

where τ is a temperature hyper-parameter which affects the concentration level of distribution. We use a memory bank with size K to save features proposed in [9].

3.2. Temporal-Motion Continuity

The motion continuity modeling module is performed with a feature decoder network, as shown in Figure 3. More specifically, as for the decoder, we conduct 4 convolution blocks using spatial-temporal convolution operation [49] and add a simply modified spatial-temporal deconvolution in the last layer (the details of the decoder can be found in the supplementary material). Unlike the previous work [53], we do not directly reconstruct the input skeleton action sequences, but we set a specific interpolation rate to conduct upsampling-interpolation of high-semantic skeleton sequences. Compared to reconstruct the original skeleton data, we aim to interpolate and complement the motion of the missing interval frames to recover the whole action, making the whole temporal motion look coherent and natural, which can drive the networks better capture the differences in dynamics among adjacent frames and understand the essence of the motions.

To predict the interpolated motions, we generate the self-supervision ground-truth as shown in Figure 3 black arrow part. We assume that the interpolation rate is set to l, which means that the interpolated ground-truth can be sampled online from the raw input skeleton data across $\frac{r}{l}$ frames. Namely, the total length of the interpolated frames are l times of the input skeleton samples. When the input skeleton clip is sampled in $1\times$ rate in its original pace, we repeat the clip and splice these l segments together. Note that we only consider up-interpolating the output of the speed-changed clips from the encoder and ignore the motion-broken clips, because the motion-broken data lose the continuity of the original action, and interpolating them will destroy the learning ability of the networks. Formally, denote the interpolation ground-truth $\mathcal{X} \in \mathbb{R}^{n \times 3 \times T'}$, where n is the number of joints and T' represents the number of frames. When we obtain the predicted 3D interpolation skeleton $\hat{\mathcal{X}}$, the training loss function can be defined as:

$$\mathcal{L}_{Itep} = \frac{1}{nT'} \sum_{i=1}^{n} \sum_{t=1}^{T'} \|\hat{\mathcal{X}}_{i,:,t} - \mathcal{X}_{i,:,t}\|_2^2, \tag{3}$$

Finally, we train the networks on two tasks jointly (motion consistency and continuity). The total objective function can be formulated as follows:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{triplet} + \lambda_2 \mathcal{L}_{NCE} + \lambda_3 \mathcal{L}_{Itep}, \tag{4}$$

where λ_1, λ_2 and λ_3 are three weight hyper-parameters.

4. Experiments

To verify our approach, we perform extensive experimental evaluations of our formulation on four datasets. First, since the NTU60-RGBD dataset [30] is the most commonly used dataset, we conduct several ablation studies on it to examine the contributions of the proposed method based on the recognition performance. Then, to find out whether the encoder $f(\cdot;\theta)$ can learn good representation features for skeleton sequences with self-supervision, we complete other experiments under different settings.

4.1. Datasets

NTU60-RGBD [30]: This dataset contains 56,000 action clips in 60 action classes. There are 25 joints for each subject in the skeleton sequences. The original paper of the dataset recommends two benchmarks: 1) cross-subject (X-Sub) benchmark with 40,320 and 16, 560 clips for training and evaluation; 2) cross-view (X-View) benchmark 37, 920 and 18, 960 clips. Training clips in this setting come from the camera views 2 and 3, and the evaluation clips are all from the camera view 1.

NTU120-RGBD [20]: The dataset contains 114,480 action samples in 120 action classes. The original paper of the dataset recommends two benchmarks: (1) cross-subject (X-sub) benchmark: the 106 subjects are split into training and testing groups. Each group contains 53 subjects. (2) cross-setup (X-setup) benchmark: training data comes from samples with even setup IDs, and testing data comes from samples with odd setup IDs.

Kinetics-Skeleton [12]: It is a large data set for human action analysis in 400 classes. The dataset is divided into a training set (240,000 clips) and a validation set (20,000 clips). Since only raw video clips are provided, skeleton data can be obtained by estimating joint locations on certain pixels with OpenPose toolbox [2] and each sample consists of 18 body joints.

PKUMMD [21]: PKU Multi-Modality dataset is a new large scale benchmark for human action understanding. It contains almost 20, 000 action instances and 5.4 million frames in 52 action categories. Each sample consists of 25 body joints. This dataset consists of two parts and it is also split into cross-subject (X-sub) and cross-view (X-view) subset.

4.2. Implementation Details

Training. Our network is built upon the PyTorch library. We use stochastic gradient descent (SGD) as the optimization strategy. The learning rate is initially set to 0.1 with momentum of 0.9, and the weight decay is set to 0.0001. The parameters in our method are set by experience as λ_1 = 1, λ_2 = 1 and λ_3 = 1. The temperature factor τ is set as 0.5 and the interpolation rate l is 2. We set γ = 0.15 and K = 6536 for memory bank size. For the Kinetics-Skeleton dataset, the batch-size is 256 and for the other three datasets, the batch-size is 64. Since we adopt three different network architectures [49, 32, 17] to conduct the experiments, we strictly follow other settings in the original paper including the total training epochs, the decline of learning rate in a different epoch, and the data pre-processing. All the experiments are conducted with 4 TITANX GPUs.

Settings. (1) Self-supervised pre-training: compared to train from scratch and randomly initialize the weights of network, we initialize the encoder with the learned weights from self-supervised tasks and then learn the classifier for action recognition. (2) Semi-supervised: The encoder is pretrained with unlabeled data, then trained with the classifier using a very small percentage (*i.e.* 5%-10%) of training labeled data. (3) Fine-tuning: The encoder is pretrained with unlabeled data on a larger dataset, where the pretrained weights are used as the initialization and are further refined on the target downstream task (small dataset).

4.3. Ablation Study

To explore the learning features of our proposed method, we apply them to three different backbone networks under the setting of self-supervised pre-training to study the effectiveness. More details are illustrated in the following.

The effect of pretext losses. As shown in Table 1, compared with training from scratch, using different pretext tasks for self-pretraining can help to boost the action recognition performance. Specifically, as for motion consistency pretext task, using $\mathcal{L}_{triplet}$ only works better than $\mathcal{L}_{NCE}$ only. This because we use other video clips as negative pairs, there still exists many artificial cues [14] to distinguish two videos for the networks to solve the task, which will lead to poor learning representations. When we combine these losses, we can further promote the networks performance, which verifies that it can learn more deep motion representations as we mentioned in Sec.3.1. As for motion continuity pretext task, by only employing $\mathcal{L}_{Itep}$ also can help the three backbone networks to improve different de-

Pre-training settings	NTU60 X-sub			NTU60 X-view		
	ST-GCN	2S-AGCN	AS-GCN	ST-GCN	2S-AGCN	AS-GCN
w/o pre-training	81.5	88.5	86.8	88.3	95.1	94.2
w/ $\mathcal{L}_{triplet}$ only	$82.3_{+0.8}$	$89.2_{+0.7}$	$87.6_{+0.8}$	$89.2_{+0.9}$	$95.9_{+0.8}$	$94.9_{+0.7}$
w/ $\mathcal{L}_{NCE}$ only	$81.8_{+0.3}$	$89.0_{+0.5}$	$87.2_{+0.4}$	$88.7_{+0.4}$	$95.7_{+0.6}$	$94.5_{+0.3}$
w/ $\mathcal{L}_{Itep}$ only	$82.5_{+1.0}$	$89.3_{+0.8}$	$87.6_{+0.7}$	$89.4_{+1.1}$	$95.9_{+0.8}$	$95.0_{+0.8}$
$\mathcal{L}_{triplet}$ + $\mathcal{L}_{NCE}$	$82.6_{+1.1}$	$89.4_{+0.9}$	$88.0_{+1.2}$	$89.5_{+1.2}$	$96.0_{+0.9}$	$95.1_{+0.9}$
$\mathcal{L}_{triplet}$ + $\mathcal{L}_{NCE}$ + $\mathcal{L}_{Itep}$ **(ours)**	$83.0_{+1.5}$	$89.7_{+1.2}$	$88.4_{+1.6}$	$89.7_{+1.4}$	$96.3_{+1.2}$	$95.5_{+1.3}$

Table 1. Exploration of different pre-training settings on the NTU60-RGBD dataset. All models are pre-trained on the NTU60-RGBD dataset itself except for the w/o pre-training setting.

Positive	Negative	NTU60 X-sub	NTU60 X-view
S-changed	M-jittered	81.9	88.6
S-changed	V-transformed	81.7	88.7
-	M-shuffled	81.8	88.6
S-changed	M-shuffled	**82.3**	**89.2**

Table 2. Exploration of different operations of constructing motion pairs on the NTU60-RGBD dataset. All the methods are pre-trained on NTU60-RGBD itself with ST-GCN [49] backbone. **S**, **M** and **V** denote speed, motion and view, respectively.

Methods	NTU60 X-sub	NTU60 X-view
Direct reconstruction	82.0	88.8
Specific rate interpolation	**82.5**	**89.4**

Table 3. Exploration of different motion continuity modeling operations on the NTU60-RGBD dataset. All the methods are pre-trained on NTU60-RGBD itself with ST-GCN [49] backbone.

grees in terms of accuracy. When we jointly train the networks using three losses together, all three backbone networks can achieve the best performance.

The effect of motion consistency pairs. We also reveal the different operations to break the motion consistency to conduct negative pairs for learning. Among them, M-jittered and V-transformed mean we randomly jitter the skeleton (*e.g.*, add some noise to disturb the skeleton joints) and transform the coordinates frames of the skeleton points, respectively. As shown in Table 2, from which we can conclude that S-changed positive pairs and M-shuffled negative pairs are effective for the networks to learn the intrinsic motion representations.

The effect of specific interpolation. The results in Table 3 demonstrate that compared to direct reconstruction, specific rate interpolation achieves better performance and it can drive the networks learn more critical representations. As aforementioned, up-sampling interpolation can help the networks predict and simulate the relationship between adjacent frames, which can model the temporal continuity of the

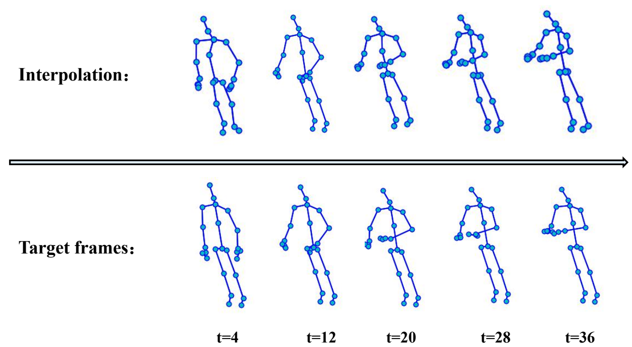

Figure 4. The interpolation skeleton action sample from the motion continuity modeling module with ST-GCN [49] backbone. We present the action "play with phone / tablet" in the NTU60-RGBD dataset. Both the interpolated motions and the ground-truth target are shown.

motions. Although our self-supervised learning method is not specifically designed for interpolating the human skeleton, however, as shown in Figure 4, our interpolation results are basically in line with expectations. Through this self-learning strategy, we can make the encoder extract higher semantic representation features.

4.4. Evaluation

Self-supervised pre-training. We compare our method termed as **MCC** (**M**otion **C**onsistency and **C**ontinuity) with the state-of-the-art unsupervised learning methods. Besides, we directly apply the latest existing SSL techniques in the video domain to skeleton data. As shown in Table 4, our MCC achieves the best results on all backbone networks over three datasets. This shows that our proposed method allows the network to learn the latent feature representations of the motions and it can boost the performance of skeleton action recognition without using additional labeled data for training. In addition, the results also reveal that the existing SSL strategies are not suitable enough for skeleton data, which is in line with the limitations we mention in Sec.1. Meanwhile, during the training process, we find

Method	Architecture	NTU60		NTU120		Kinetics	
		X-sub	X-view	X-sub	X-setup	top-1	top-5
LongT GAN [53] $_{\text{AAAI'2018}}$	Unidirectional GRUs	-	49.6*	-	-	-	-
MS^{2}L [18]$_{\text{ACMMM'2020}}$	BiGRU	78.8*	81.8*	-	-	-	-
VPD [24]$_{\text{ECCV'2020}}$	SeBiReNet	-	81.4*	-	-	-	-
Clip Order prediction [48]$_{\text{CVPR'2019}}$	ST-GCN	82.1	88.6	76.0	76.8	31.3	53.5
	2S-AGCN	89.0	95.8	80.6	82.5	36.8	59.7
	As-GCN	87.5	94.9	78.4	80.0	35.6	57.4
Jigsaw puzzle recognition [14]$_{\text{AAAI'2019}}$	ST-GCN	81.8	89.0	76.3	77.1	31.7	53.8
	2S-AGCN	88.8	95.4	80.8	82.4	36.6	59.4
	As-GCN	87.1	94.6	78.6	79.9	35.8	57.7
pace prediction [1]$_{\text{CVPR'2020}}$	ST-GCN	81.5	88.8	75.8	75.9	31.3	53.6
	2S-AGCN	89.2	95.6	80.3	82.1	36.3	59.1
	As-GCN	87.3	95.0	78.0	79.8	35.2	57.0
MCC (ours)	ST-GCN	**83.0**	**89.7**	**77.0**	**77.8**	**32.3**	**54.6**
	2S-AGCN	**89.7**	**96.3**	**81.3**	**83.3**	**38.1**	**60.8**
	As-GCN	**88.4**	**95.5**	**79.4**	**80.8**	**36.4**	**58.6**

Table 4. Comparison with other self-supervised methods on NTU60, NTU120, and Kinetics datasets. (* means our reproduced results.)

Network	NTU-60				NTU-120				Kinetics	
	5%∼data		10%∼data		5%∼data		10%∼data		10%∼data	
	X-sub	X-view	X-sub	X-view	X-sub	X-setup	X-sub	X-setup	top-1	top-5
ST-GCN [49]	38.2	40.4	52.4	56.9	25.3	27.1	37.6	40.1	11.9	28.6
+MCC (ours)	$42.4_{+4.2}$	$44.7_{+4.3}$	$55.6_{+3.2}$	$59.9_{+3.0}$	$29.7_{+4.4}$	$31.3_{+4.2}$	$40.7_{+3.1}$	$43.4_{+3.3}$	$14.8_{+2.9}$	$32.2_{+3.6}$
2s-AGCN [32]	43.5	49.1	57.2	62.0	29.2	30.8	44.1	48.7	18.6	34.8
+MCC (ours)	$47.4_{+3.9}$	$53.3_{+4.2}$	$60.8_{+3.6}$	$65.8_{+3.8}$	$33.8_{+4.6}$	$35.1_{+4.3}$	$47.0_{+2.9}$	$51.8_{+3.1}$	$21.3_{+2.7}$	$37.9_{+3.1}$
AS-GCN [17]	41.1	44.7	55.7	59.5	27.4	28.9	41.2	44.6	17.1	33.7
+MCC (ours)	$45.5_{+4.4}$	$49.2_{+4.5}$	$59.2_{+3.5}$	$63.1_{+3.6}$	$31.6_{+4.2}$	$32.9_{+4.0}$	$44.9_{+3.7}$	$47.8_{+3.2}$	$20.2_{+3.1}$	$37.5_{+3.8}$

Table 5. Evaluation of semi-supervised results on NTU60, NTU120 and Kinetics dataset with 5%, 10% labels of training data. "+ MCC" indicates training the network by initializing the self-supervised pre-trained weights of our proposed method.

that the network initialized with self-supervised pre-trained weights can speed up the convergence to reach the desired accuracy, which can help us train our models in the limited time. It's worth mentioning that we evaluate, for the first time, the learned representations of the 3D skeleton on 3 mainstream and challengeable datasets (NTU60, NTU120, Kinetics), which demonstrates the effectiveness and generality of our method.

Semi-supervised training. In some cases, there is very little labeled data that we can use, which makes it difficult for us to train the data-driven network models. As shown in Tabel 5, when we use a small amount of data (*i.e.*, 5%, 10% of data) to train from scratch, the accuracy of the model will drop sharply. After we adopt the self-supervised pre-trained weights, it is noticeable that we can gain a significant boost among all the network structures compared with the random initialized models. Specifically, with 5% labeled data, the accuracy increases by about 4.3%, and with 10% labeled data, the accuracy increases by approximate 3.5% among the three backbones. Figure 5 below compares

Backbone	Pre-train Dataset	PKUMMD (Acc.)
ST-GCN	w/o pre-training	48.2
	PKUMMD	$49.6_{+1.4}$
	NTU60 X-view	$51.8_{+3.6}$
	NTU60 X-sub	$52.7_{+4.5}$
	NTU120 X-setup	$50.5_{+2.3}$
	NTU120 X-sub	$54.5_{+6.3}$

Table 6. Exploration of different pre-train datasets for fine-tuning on PKUMMD Part-II subset.

the skeleton response by using self-supervised learning and training from scratch when has only 10% Kinetics data. It can be shown that the model after self-supervised training learns the connection between each skeleton point more respectable, instead of just remembering a certain skeleton point or feature for reasoning.

Fine-tuning on downstream tasks. As is common practice in the image and video fields, they perform self-supervised pre-training on the large scale ImageNet [28],

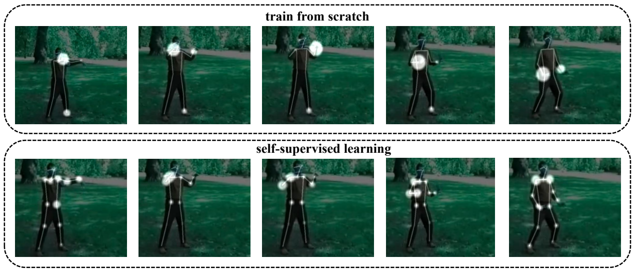

Figure 5. The response magnitude of all the joints (white dots) in a motion sequence in the last layer of ST-GCN [49] backbone. The Tai-Chi video sequence is selected from a clip of UCF101 [35] dataset with an interval of 30 frames.

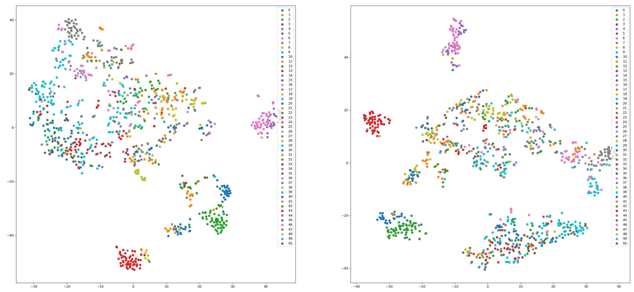

Figure 6. The t-SNE [39] visualization of the last layer features of ST-GCN [49] backbone on PKUMMD dataset. (a) **Sup** is trained with the supervised objective for the labeled samples from scratch. (b) **Sup + fine-tune** is trained by fine-tuning the learned weight from upstream dataset through self-supervised pre-training.

Kinetics [12] datasets, then initialize the network with the learned weights, and finally train on the small dataset to validate the transferability of learned representation. First, we explore the effect of different pre-train datasets for fine-tuning on downstream tasks (for simplicity, we conduct comparative experiments on ST-GCN backbone). As shown in Table 6, when training from scratch, the accuracy of the network is 48.2%. When we pretrain on the PKUMMD dataset itself, we can gain the improvement of 1.4%. When applying the NTU dataset for self-supervised pre-training, we can significantly improve the accuracy by a large margin. Among them, the NTU120 X-sub subset brings us a **6.3%** boost, which illustrates the benefits of the transferability of learned representation in the 3D skeleton.

Next, we compare the networks performance with other methods on the PKUMMD dataset (all the networks are pre-trained on NTU dataset except for the w/o pre-training setting). As shown in Table 7, MCC increases the accuracy by **6.3%**, **6.2%** and **5.6%** compared with the random initialized models on three backbone architectures, respectively. Moreover, although the backbones are different, our method can gain more relative performance boost compared to LongT GAN [53] (43.1% $\rightarrow$ 44.8%) and MS^{2}L [18] (45.7% $\rightarrow$ 45.8%). By using manual annotation, fully supervised method for fine-tuning can achieve the best performance, however, the ground-truth labels are hard to collect and the results of our SSL method is close to those of the fully-supervised manner, which shows the benefits of the discriminative features learned from the proposed method. Finally, as shown in Figure 6, with the benefit of fine-tuning, features of **Sup + fine-tune** presents a more discriminative distribution than **Sup**, which shows the compact intra-class distance and more distinguishable inter-class distance.

Method	Architecture	PKUMMD (Acc.)
LongT GAN [53]	unidirectional GRUs	44.8
MS^{2}L [18]	BiGRU	45.8
w/o pre-training	LongT GAN	43.1*
	MS^{2}L	45.7
	ST-GCN	48.2
	2S-AGCN	54.6
	AS-GCN	52.8
Fully supervised	LongT GAN	48.4*
	MS^{2}L	49.8*
	ST-GCN	60.5
	2S-AGCN	66.8
	AS-GCN	65.4
MCC (ours)	ST-GCN	**54.5**
	2S-AGCN	**60.8**
	AS-GCN	**58.4**

Table 7. Comparison of action recognition transfer learning results on PKUMMD Part-II subset. (* means our reproduced results.)

5. Conclusion

In this paper, we propose a novel self-supervised learning method for skeleton-based action recognition. By constructing positive and negative pair clips, we encourage the network to separate them to learn the intrinsic dynamic motion consistency information. Skeleton interpolation is further exploited to model the continuity of human skeleton data. Extensive evaluations demonstrate the effectiveness of our approach. We hope these findings will encourage more research on 3D skeleton representation learning.

Acknowledgement

This work was supported by National Natural Science Foundation of China (NSFC) 61876208, Key-Area Research and Development Program of Guangdong Province 2018B010108002, Central Universities of China under Grant D2192860, and the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-003), and the MOE Tier-1 research grants: RG28/18 (S), RG22/19 (S) and RG95/20.

References

[1] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9922–9931, 2020. 2, 3, 7

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 5

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 3

[4] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192, 2020. 3

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016. 3

[6] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10364–10374, 2019. 1

[7] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[8] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. *arXiv preprint arXiv:2006.05582*, 2020. 3

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 3, 4

[10] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 4

[11] Simon Jenni, Hailin Jin, and Paolo Favaro. Steering self-supervised feature learning beyond local pixel statistics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6408–6417, 2020. 1, 2, 3

[12] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5, 8

[13] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3288–3297, 2017. 2

[14] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8545–8552, 2019. 5, 7

[15] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018. 3

[16] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016. 3

[17] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3595–3603, 2019. 3, 5, 7

[18] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2490–2498, 2020. 1, 3, 7, 8

[19] Hong Liu, Juanhui Tu, and Mengyuan Liu. Two-stream 3d convolutional neural network for skeleton-based action recognition. *arXiv preprint arXiv:1705.08106*, 2017. 2

[20] Jun Liu, Amir Shahroudy, Mauricio Lisboa Perez, Gang Wang, Ling-Yu Duan, and Alex Kot Chichung. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 5

[21] Jiaying Liu, Sijie Song, Chunhui Liu, Yanghao Li, and Yueyu Hu. A benchmark dataset and comparison study for multi-modal human action analytics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(2):1–24, 2020. 5

[22] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017. 2

[23] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. *arXiv preprint arXiv:2001.00294*, 2020. 3

[24] Qiang Nie, Ziwei Liu, and Yunhui Liu. Unsupervised 3d human pose representation with viewpoint and pose disentanglement. In *European Conference on Computer Vision*, pages 102–118. Springer, 2020. 7

[25] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016. 3

[26] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 2, 3

[27] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2701–2710, 2017. 3

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 7

[29] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 4

[30] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016. 5

[31] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019. 1

[32] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019. 3, 5, 7

[33] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018. 1

[34] Chenyang Si, Xuecheng Nie, Wei Wang, Liang Wang, Tieniu Tan, and Jiashi Feng. Adversarial self-supervised learning for semi-supervised 3d action recognition. In *European Conference on Computer Vision*, pages 35–51. Springer, 2020. 3

[35] Khurram Soomro, Amir Roshan Zamir, and M Shah. A dataset of 101 human action classes from videos in the wild. *Center for Research in Computer Vision*, 2(11), 2012. 8

[36] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9631–9640, 2020. 3

[37] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5323–5332, 2018. 1

[38] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 1

[39] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8

[40] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014. 2

[41] Raviteja Vemulapalli and Rama Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4471–4479, 2016. 2

[42] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2443–2451, 2015. 3

[43] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *European Conference on Computer Vision*, pages 504–521. Springer, 2020. 2, 3

[44] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297. IEEE, 2012. 2

[45] Chen Wei, Lingxi Xie, Xutong Ren, Yingda Xia, Chi Su, Jiaying Liu, Qi Tian, and Alan L Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1910–1919, 2019. 3

[46] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2

[47] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *arXiv preprint arXiv:2007.10985*, 2020. 3

[48] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. 2, 3, 7

[49] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 1, 3, 4, 5, 6, 7, 8

[50] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2117–2126, 2017. 2

[51] Pengfei Zhang, Jianru Xue, Cuiling Lan, Wenjun Zeng, Zhanning Gao, and Nanning Zheng. Adding attentiveness to the neurons in recurrent neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–151, 2018. 2

[52] Xikun Zhang, Chang Xu, and Dacheng Tao. Context aware graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14333–14342, 2020. 3

[53] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In *Thirty-Second AAAI conference on artificial intelligence*, 2018. 1, 3, 4, 7, 8

[54] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. *arXiv preprint arXiv:1603.07772*, 2016. 2