Cross-View Nearest Neighbor Contrastive Learning of Human Skeleton Representation

Xuelian Zhang[1], Zengmin Xu[1,2(✉)], Lulu Wang[1], and Jiakun Chen[1]

[1] School of Mathematics and Computing Science, Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, Guilin, China

[2] Anview.AI, Guilin Anview Technology Co., Ltd, Guilin, China

xzm@guet.edu.cn

Abstract. Traditional self-supervised contrastive learning approaches regard different views of the **same** skeleton sequence as a positive pair for the contrastive loss. While existing methods exploit cross-modal retrieval algorithm of the **same** skeleton sequence to select positives. The common idea in these work is the following: ignore using **other** views after data augmentation to obtain more positives. Therefore, we propose a novel and generic Cross-View Nearest Neighbor Contrastive Learning framework for self-supervised action Representation (CrosNNCLR) at the view-level, which can be flexibly integrated into contrastive learning networks in a plug-and-play manner. CrosNNCLR utilizes different views of skeleton augmentation to obtain the nearest neighbors from features in latent space and consider them as positives embeddings. Extensive experiments on NTU RGB+D 60/120 and PKU-MMD datasets have shown that our CrosNNCLR can outperform previous state-of-the-art methods. Specifically, when equipped with CrosNNCLR, the performance of SkeletonCLR and AimCLR is improved by 0.4%∼12.3% and 0.3%∼1.9%, respectively.

Keywords: Self-supervised learning · Plug-and-play · Cross-view nearest neighbor contrastive learning · Action representation

1 Introduction

In the research field of robot action planning, the action of joints is an important factor to evaluate the goodness of robot products, which can be associated with human action analysis. As a research hotspot in computer vision, human action analysis plays an important part in video understanding [1]. Early researchers employ supervised learning methods [2–4] to study human action potential dynamics based on RGB frames, e.g., a two-stream network [5], spatial-temporal attention model [6], and LSTM network [7]. However, these visual representations are not robust to various backgrounds and appearances. Therefore, researchers focus on the human skeleton dataset, which offer light-weight

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
Z. Yu et al. (Eds.): CIRAC 2022, CCIS 1770, pp. 237–256, 2023.
https://doi.org/10.1007/978-981-99-0301-6_19

representations, attracting many people to research skeleton-based action recognition [8–10], e.g., Part-Aware LSTM [11] treats each joint point of each frame of skeleton sequence as an LSTM unit, and performs LSTM operations in both temporal and spatial dimensions. MS-G3D [12] proposes a multi-scale spatial-temporal aggregation scheme to solve the question of biased weighting. Although the recognition accuracy is improved, it requires large volumes of labeled robot body skeletons, which is time-consuming and labor-intensive to annotate in real life.

Given this, self-supervised learning methods are introduced [13,14], which can learn semantic content in large-scale unlabeled samples to provide supervised information for models and algorithms. The early emergence of various self-supervised model building strategies, e.g., jigsaw puzzles [15], colorization [16], prediction mask words [17], etc. With the emergence of the idea of contrastive learning, some self-supervised contrastive learning methods are constructed for 3D skeleton data, e.g., ISC [18] makes exploit of inter-skeleton contrastive learning methods to learn feature representations from skeleton inputs of multimode. Colorization [19] designs a skeleton cloud colorization technology to learn the feature representation of samples from unlabeled skeleton sequences. However, the above self-supervised learning works, ignoring the different views after skeleton augmentation can also be applied as an auxiliary tool to finding positive samples.

Therefore, we propose a self-supervised action representation approach based on Cross-View Nearest Neighbor Contrastive Learning (CrosNNCLR). Firstly, the framework introduces the nearest neighbor search algorithm to look for more semantically similar samples in the latent space by combining different views of the same skeleton sequence skeleton augmentation. Secondly, CrosNNCLR loss is proposed for the network to learn parameters more efficiently and minimize the embedding distribution between nearest-neighbor sample views. Finally, the proposed method is designed in a plug-and-play manner, which is integrated into the traditional and existing self-supervised contrastive learning models to form a new network structure. The main contributions of this paper are as follows:

- A plug-and-play block is designed using different views of each sample data after augmentation, and the close association between similar skeleton sequences, combined with the nearest neighbor search algorithm to find more positive sample pairs from the latent space.
- CrosNNCLR loss is proposed to learn the network model parameters, capture richer semantic information, enable better clustering of same category samples, and obtain a good feature space for 3D actions from unlabeled skeleton data.
- The plug-and-play block is integrated into an existing self-supervised contrastive learning network to form new model structures. We evaluated the model on three benchmark datasets, i.e., NTU RGB+D 60/120, PKU-MMD, and outperforms the mainstreaming methods.

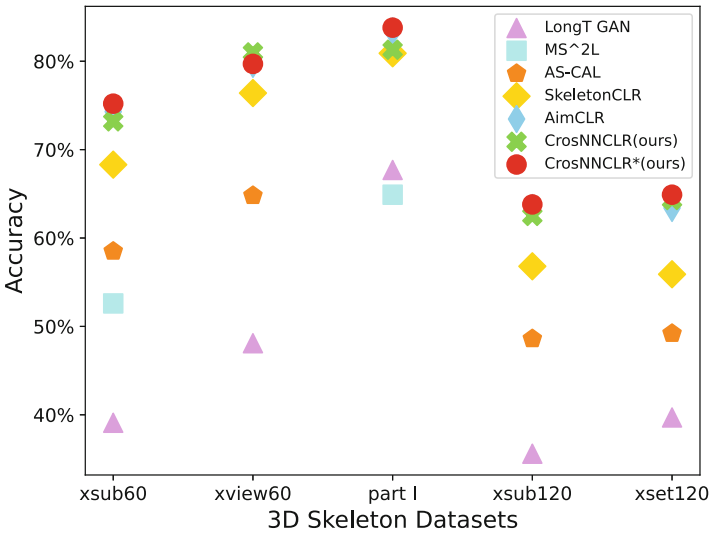

Fig. 1. Comparison of recognition accuracy of different models, which can visually observe the recognition accuracy of each model, where the green cross symbol and the red solid circle respectively indicate the proposed method CrosNNCLR and CrosNNCLR*.

2 Related Work

Skeleton-based Action Recognition. To solve skeleton-based action recognition tasks, previous work is based on manual features [20–22]. In recent years, some approaches avail RNNs to process skeleton data in different time intervals, e.g., VA-RNN [23] proposes a view adaptive neural network to automatically transform skeleton sequences into observation viewpoints, eliminating the effect of viewpoint diversity. Although these approaches have achieved many significant results, researchers shifted their attention to CNNs due to the gradient disappearance question of RNNs, where VA-CNN [23] maps skeleton sequences as RGB frames. HCN [24] automatically learn hierarchical co-occurrence features of skeleton sequences. Considering that CNNs need to transform skeleton sequences into a specific form, which is not conducive to the feature representation of the original skeleton data itself, further proposes GCNs to model the graph structure of skeleton data, e.g., ST-GCN [25] proposes a spatial-temporal GCN to solve the problem of human action recognition based on skeleton data. In this work, we exploit ST-GCN [25] as the encoder.

Self-supervised Learning. Many self-supervised representation learning methods are employed to image and video classification. For RGB image data, MoCo [26] establishes a dynamic dictionary to do the self-supervised representation learning task by momentum contrast. OBow [27] incorporates knowledge distillation technology into the self-supervised contrastive learning model and reconstructs the bag-of-visual-words (BoW) representation of image features. For RGB

video data, CoCLR [28] avails complementary information from different modalities of the same data source to obtain positives from one view to another. Similarly, NNCLR [29] proposes the self-supervised image classification method, compatible with skeleton video data, RGB images have fewer actual practical application scenarios, as in real life, long videos are mainly used to record events.

Self-supervised Skeleton-Based Action Recognition. In recent years, researchers have proposed many self-supervised learning approaches of skeleton data, which are mainly divided into two types. The first one proposes encoder-decoder structures, e.g., LongT GAN [30] reconstructs masked 3D skeleton sequences by combining encoder, decoder, and generative adversarial networks. P&C [31] adopt a weak the decoder to discriminate their embedding similarity. The second is the contrastive learning network structures, e.g., CrosSCLR [32] roots a cross-view consistent knowledge mining method, which exploits the feature similarity of one modality to assist another modality for contrastive recognition learning. AimCLR [33] explores the different patterns of movement brought about by extreme augmentations to alleviate the irrationality of positive sample selection. However, these methods rely on obtaining positive samples from different modal data or views after adding skeleton augmentation, and ignoring the different views obtained after random skeleton augmentation can also be used as auxiliary tools to find positives. Therefore, we introduce CrosNNCLR to obtain positive sample pairs from nearest neighbors more concisely.

3 Approach

Although 3D skeleton data has made great progress in self-supervised contrastive learning representation, some algorithms regard different views of each skeleton as positive samples for the contrastive loss, i.e., only one positive sample exists. While other algorithms employ multimodal skeleton data to acquire positive samples of another modal skeleton view from one modal skeleton view, i.e., multiple positive samples exist. Unlike previous approaches [30–33], we apply different views of the same skeleton sequence after skeleton augmentation to increase the number of positive samples without using multimodal skeleton data. We first describe the main approaches of skeleton-based self-supervised representation learning, e.g., SkeletonCLR [32] and AimCLR [33]. Second, we base on the original model, establish a plug-and-play block, namely Cross-view Nearest Neighbor Contrastive Learning framework for self-supervised action Representation (CrosNNCLR). This module exploits the Nearest Neighbor (NN) of original positive samples to obtain more positives and enhance comparative instance discrimination.

3.1 Problem Setting

Given a skeleton sequence x is subjected to the following operations for obtain encoding features $z = \psi_\theta(aug(x))$, $\hat{z} = \psi_{\hat{\theta}}(aug(x))$, where $aug(\cdot)$ denotes the random data augmentation function and x generates different views q, k by

$aug(\cdot)$, ψ denotes the ST-GCN network function with nonlinear projection layer, ψ_θ enotes the Query encoder network whose network parameters are updated by gradient descent method, $\psi_{\hat{\theta}}$ denotes the Key encoder whose network parameters follow the momentum update.

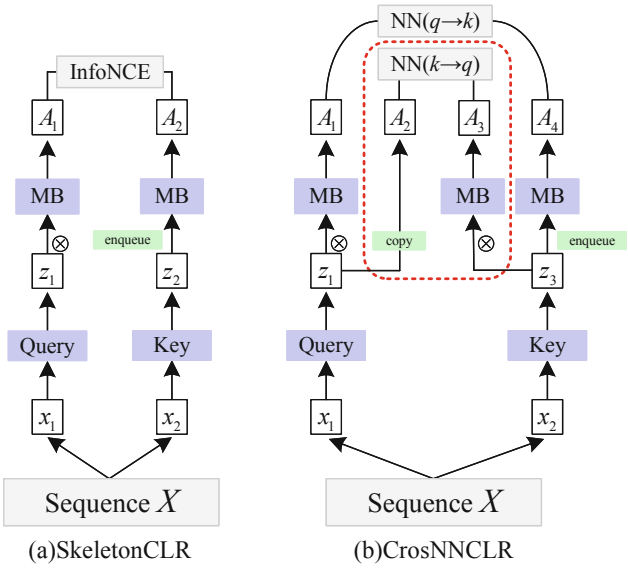

Fig. 2. Comparison between SkeletonCLR and CrosNNCLR, the left figure shows the SkeletonCLR framework and the right figure shows our CrosNNCLR framework. where $\boldsymbol{X}$ is the input N skeleton sequence data, $\boldsymbol{x_1}$, $\boldsymbol{x_2}$ are the different views obtained after random skeleton augmentations, **Query**, **Key** are the encoder network, $\boldsymbol{z_1}$, $\boldsymbol{z_2}$ are the encoding feature, **MB** is the memory bank, $\boldsymbol{A_i (i \in 1, 2, 3, 4)}$ is the positive sample set, **enqueue** indicates that encoding feature enters the MB to update negatives, $\otimes$ indicates the dot product, **copy** indicates the copy function. **InfoNCE**, $\mathbf{NN}_{q\rightarrow k}$ and $\mathbf{NN}_{k\rightarrow q}$ are the loss functions involved in each model.

3.1.1 SkeletonCLR

SkeletonCLR [32] is a single-view contrastive learning method based on skeleton representation. As shown in Fig. 2(a). Firstly, the given sequence of skeletons is randomly transformed into different views q, k that are regarded as positive pairs, the other samples in the memory bank $\mathrm{M} = \{m_i\}_{i=1}^{M}$ are considered negative samples, and q, k are embedded into the encoder ψ_θ, $\psi_{\hat{\theta}}$ to obtain the encoding features z, $\hat{z}$, θ and $\hat{\theta}$ are the required parameters for both encoders, $\hat{\theta}$ following the momentum update: $\hat{\theta} \leftarrow \alpha\hat{\theta} + (1-\alpha)\theta$, α is the momentum coefficient. Secondly, similar to the MoCo [26], SkeletonCLR exploits a FIFO strategy for updating negative samples in its memory bank to eliminate redun-

dant computation. In other words, a certain number of negative sample tensors are randomly generated when the model starts training, with the increase of training iterations, the N-dimensional tensor z is continuously transferred into the memory bank, replacing the previously randomly generated tensor as the negative samples for the next iteration. Finally, SkeletonCLR employs InfoNCE [26] loss to learn model parameter. The formula is as follows:

$$L_{\text{InfoNCE}} = -\log \frac{\exp(z \cdot \hat{z}/\tau)}{\exp(z \cdot \hat{z}/\tau) + \sum_{i=1}^{M} \exp(z \cdot m_i/\tau)}. \tag{1}$$

where $m_i \in \text{M}$, τ is a temperature hyperparameter, $z \cdot \hat{z}$ are normalized, and $z \cdot \hat{z}$ represents the dot product of two tensors to find the similarity between them.

SkeletonCLR relies only on one positive sample pair generated by the same sample under random data augmentation, while treating other samples as negative samples. This reduces the ability to discriminate intra-class variations, as semantically similar samples are hard to cluster with other positives in the embedding space.

3.1.2 AimCLR

The model framework of AimCLR [33] is shown in Fig. 3(a). Firstly, the model proposes an extreme skeleton augmentation method and adds a Query encoder branch to the dual branches of SkeletonCLR, which utilizes the method to data amplify for the same sample. Secondly, a data augmentation method based on the EADM is proposed, which can obtain different positives under the new branch after adding the Query encoder branch. Specifically, in the original branch, the same skeleton sequence is amplified by random normal skeleton augmentation, then the sample features z, $\tilde{z}$ are obtained through the encoder and projection layer (ψ_θ, $\psi_{\hat{\theta}}$), respectively. In the new branch, the same skeleton sequence is amplified by extreme skeleton augmentation to form two parallel branches, one of which passes through the encoder and projection layer ($\psi_{\hat{\theta}}$) to form the sample features $\tilde{z}$, then others passes through the encoder, projection layer (ψ_θ) and the EADM module to form the sample features $\tilde{z}_{drop}$. Finally, the Nearest Neighbor Mining (NNM) method with the multi-branch view is utilized to increase the number of positives, and updates the network parameters using D^3M loss function. The specific loss function involved in the work is as follows:

$$L_{d1} = -p(z \mid \hat{z}) \log p(z \mid \tilde{z}) - \sum_{i=1}^{M} p(m_i \mid \hat{z}) \log p(m_i \mid \tilde{z}). \tag{2}$$

$$L_{d2} = -p(z \mid \hat{z}) \log p(z \mid \tilde{z}_{drop}) - \sum_{i=1}^{M} p(m_i \mid \hat{z}) \log p(m_i \mid \tilde{z}_{drop}). \tag{3}$$

$$L_{\text{D}^3\text{M}} = 1/2(L_{d1} + L_{d2}). \tag{4}$$

$$L_N = -\log \frac{\exp(z \cdot \hat{z}/\tau) + \sum_{i \in N_+} \exp(\hat{z} \cdot m_i/\tau)}{\exp(z \cdot \hat{z}/\tau) + \sum_{i=1}^{M} \exp(\hat{z} \cdot m_i/\tau)}. \tag{5}$$

where $p(\cdot \mid \cdot)$ is the conditional probability and $m_i \in \mathrm{M}$, N_+ is the index value of the similar samples obtained by NNM method. The numerator of the loss L_N shows that the increase number of positive pairs, prompting a better clustering of more similar samples with high confidence.

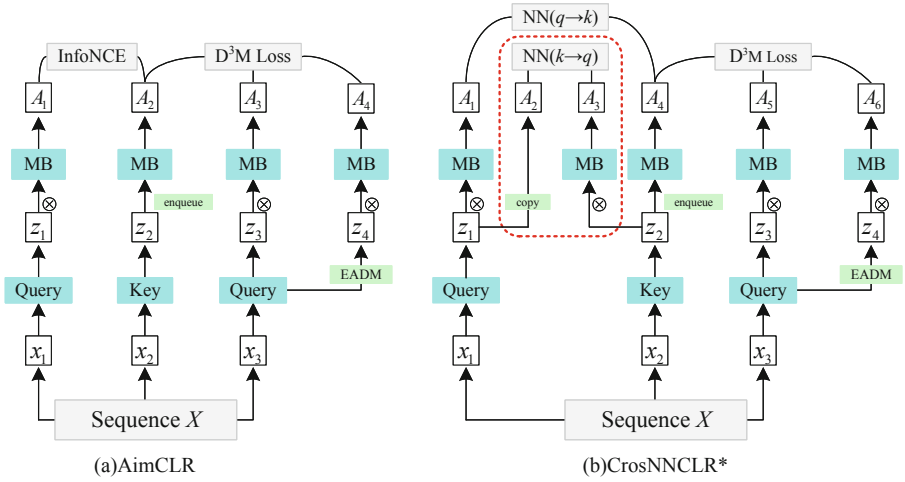

Fig. 3. Comparison between AimCLR and CrosNNCLR*, the left figure shows the AimCLR framework and the right figure shows our CrosNNCLR* framework. Where $\boldsymbol{x_i (i \in 1,2,3)}$ are the different views obtained after random skeleton augmentations, $\boldsymbol{z_i (i \in 1,2,3,4)}$ are the encoding feature, $\boldsymbol{A_i (i \in 1,2,3,4,5,6)}$ is the positive sample set, **EADM** is the new skeleton augmentation method. **InfoNCE**, **D^{3}M loss**, $\mathbf{NN}_{q \to k}$ and $\mathbf{NN}_{k \to q}$ are the loss functions involved in each model.

AimCLR employs embedding space that are most similar semantic information to q, achieves better clustering by different views of samples from the same category, but it ignores that view k also can be utilized for nearest neighbor search.

3.2 Our Approaches

To increase the richness of latent similarity representation, beyond single-instance positive samples and multi-instance positive samples generated by view q, we propose a plug-and-play block that combines the cross-view (q, k) nearest neighbor contrastive learning method to obtain more diverse positives. The

model construction is similar to MoCo [26], but it obtains not only negative set but also positive sample sets from the memory bank.

SkeletonCLR takes different views obtained by random skeleton augmentations as positive pairs, denoted as $(z, \hat{z})$. AimCLR utilizes the extreme skeleton augmentation, EADM, and NNM approaches to generate three skeleton augmentation pairs, which are paired to form positives, denoted as $(z, \hat{z}), (z, \tilde{z}), (z, \tilde{z}_{drop})$. Instead, we root the nearest neighbor search algorithm to find out multiple positives from the memory bank that is similar to z and $\hat{z}$ semantic information, denoted as $(z, \hat{z}), (z, \hat{z}_1), (z_1, \hat{z})$. In Figs. 2(b) and 3(b), similar to SkeletonCLR and AimCLR, we acquire the negative samples from the memory bank and use the idea of InfoNCE loss (1), we define the loss function as follows:

$$L_{q\to k} = -\log \frac{\exp(z\cdot\hat{z}/\tau) + \exp(\mathrm{NN}(z, M)\cdot\hat{z}/\tau)}{\exp(z\cdot\hat{z}/\tau) + \sum_{i=1}^{N}\exp(z\cdot m_i/\tau)}. \tag{6}$$

where $\hat{z}_1 = \mathrm{NN}(z, M) = \underset{m_i\in M}{\operatorname{argmin}} \parallel z - m_i \parallel_2$ is the nearest neighbor operation, τ is the temperature hyperparameter, the numerator contains 2 positive samples. The denominator contains $N+1$ samples in total, including 2 positives and $N-1$ negatives.

Similarly, the nearest neighbor instances in the feature space of view k can be used as pseudo-labels. Therefore, the loss function equation is as follows:

$$L_{k\to q} = -\log \frac{\exp(z\cdot\hat{z}/\tau) + \exp(\mathrm{NN}(\hat{z}, M)\cdot z/\tau)}{\exp(z\cdot\hat{z}/\tau) + \sum_{i=1}^{N}\exp(\hat{z}\cdot m_i/\tau)}. \tag{7}$$

where $z_1 = \mathrm{NN}(\hat{z}, M)$, and the parameters are expressed the same as Eq. (6). The two views take positive samples from each other to enhance the network model performance and obtain better clustering results.

$$L_{\mathrm{CrosNNCLR}} = 1/2(L_{q\to k} + L_{k\to q}). \tag{8}$$

The cross-view loss function $L_{\mathrm{CrosNNCLR}}$ pulls in more high-confidence positive samples than the single-view loss function L_{InfoNCE}, making it easier to aggregate same category sample features in the embedding space.

3.2.1 CrosNNCLR Based on SkeletonCLR

This section focuses on using Nearest Neighbors (NN) to find out semantically similar samples with view q or view k, and improve contrastive instance discrimination methods. The concrete implementation framework of CrosNNCLR is shown in Fig. 2(b). Firstly, CrosNNCLR is inserted into SkeletonCLR, which compensates for the shortcoming that the original model treats same category samples as negative samples. Secondly, CrosNNCLR loss is proposed to enhance the InfoNCE loss-based approach. Finally, we evaluate the proposed approach

Algorithm 1: CrosNNCLR(plug-and-play block) Pseudocode.

```
# Query, Key: encoder network
# N: batch size
# MB: memory bank(queue)
# t: temperature

for x in loader:
   x1, x2 = aug(x), aug(x) # random augmentation
   z1, z2 = Query(x1), Key(x2) # obtain the encoded features
   h1, h2, mb = normalize(z1), normalize(z2), normalize(MB) # l2-normalize

   NN1 = NN(h1, mb) # cross-view the nearest neighbor index
   NN2 = NN(h2, mb) # cross-view the nearest neighbor index

   loss = L(NN1, h2, h1)/2 + L(NN2, h1, h2)/2 # Loss_CrosNNCLR
   loss.backward() # back-propagate
   update([Query.params, Key.params]) # SGD update
   update_queue(MB, z2)

def L(nn, c, d, t=0.07):
   logits_cd = mm(c, d.T)/t # mm: Matrix multiplication
   logits_nnc = mm(nn, c.T)/t # mm: Matrix multiplication
   logits = concat([logits_cd, logits_nnc], axis=1) # logits_qk, logits_kq
   labels = range(N)
   loss = CrossEntropyLoss(logits, labels)
   return loss

def NN(h, mb):
   simi = mm(h, mb.T) # mm: Matrix multiplication
   nn = simi.argmax(dim=1) # Top-1 NN indices
   return mb[nn]
```

with the linear evaluation protocol, and carry out relevant experimental verification on three benchmark datasets. Algorithm 1 provides the pseudo-code for the pre-training task of CosNNCLR.

3.2.2 CrosNNCLR* Based on AimCLR

This section focuses on using Nearest Neighbors (NN) to find samples that are semantically similar to view k. The feature representation is learned by pulling the distance between different views of each sample and the nearest neighbor samples in the embedding space. The specific implementation framework of CrosNNCLR* is shown in Fig. 3(b). Firstly, CrosNNCLR branch is integrated into the AimCLR, which alleviates the reliance of self-supervised learning on the data augmentation approach. Secondly, CrosNNCLR loss is added to the D^3M loss to improve contrastive instance discrimination methods. Finally, we evaluate the proposed approach with the linear evaluation protocol, and carry out relevant experimental verification on three benchmark datasets.

4 Experiments

In this section, our CrosNNCLR and CrosNNCLR* are compared with other self-supervised skeleton representation learning approaches. The datasets (Subsect. 4.1) and experimental settings (Subsect. 4.2) for CrosNNCLR and

CrosNNCLR* are described. In Subsect. 4.3, the model performance is compared. In Subsect. 4.4, ablation experiments are performed to demonstrate the effectiveness of the proposed methods.

4.1 Datasets

NTU RGB+D 60 (NTU-60) [34]**.** It consists of 56,880 action sequences. The dataset is captured from different viewpoints, with action samples performed by 40 actors, and contains 60 action categories. Two evaluation benchmarks for this dataset are utilized: Cross-Subject (xsub) and the Cross-View (xview).

PKU-MMD (PKU) [35] **.** It is a human action analysis benchmark dataset with good annotation information. It specifically consists of 28,000 action sequences, with 51 action classes. PKU-MMD is divided into two parts, the first part (part I) is a large-amplitude action detection task and the second part (part II) is a small-amplitude action detection task.

NTU RGB+D 120 (NTU-120) [36] **.** It is an extension of the NTU-60 dataset, which contains 120 actions performed by 106 actors, and the total number of action skeleton sequences expanded to 114,480. Similarly, two evaluation benchmarks for this dataset are utilized: Cross-Subject (xsub) and Cross-Setup (xset).

4.2 Experimental Settings

The hardware platform in this experiment includes four TITAN XP graphics cards with 128 GB memory, the software platform includes python 3.6 and the PyTorch 1.2.0 framework. The parameter configuration of CrosNNCLR is consistent with the SkeletonCLR, where the models run 800 epochs and the linear evaluations run 100 epochs. The parameter configuration of CrosNNCLR* is consistent with the AimCLR, where the models run 300 epochs and the linear evaluations run 100 epochs. Specifically, during the training of these models, the Query and Key encoders mainly use the ST-GCN network with a hidden layer dimension of 256, a feature dimension of 128, the batch size is 128, the momentum coefficient $\alpha = 0.999$, $M = 32768$, and the initial lr $= 0.1$, which becomes 0.01 after 250 epochs, the weight decay is 0.0001, the initial value of the learning rate for linear evaluation is 0.3, which became 0.03 after 80 epochs of evaluation.

4.3 Analysis of Experimental Results

We design a plug-and-play block for enhancing positives, which utilizes CrosNNCLR to identify sample instances in the latent space. It means that sample instances with semantic information more similar to different views will be mined, and take them as positive samples. This subsection mainly gives the experimental results of CrosNNCLR and CrosNNCLR*.

Experimental studies are conducted on different datasets to compare the model performance. As shown in Table 1, on the single mode of skeleton dataset (joint, motion, bone), the action recognition of our CrosNNCLR is higher than

the SkeletonCLR, except for the motion modality under the xset benchmark on the NTU-120 dataset, and the bone modality under the xsub evaluation benchmark on the NTU-60 dataset. The performance of our 3s-CrosNNCLR is better than the 3s-SkeletonCLR, when the evaluation effects of these three modalities are fused.

Table 1. Comparison of SkeletonCLR and CrosNNCLR linearity evaluation results on the NTU-60/120 and PKU datasets. "3s" means three stream fusion.

Method	Stream	NTU-60(%)		PKU(%)	NTU-120(%)	
		xsub	xview	part I	xview	xset
SkeletonCLR	joint	68.3	76.4	80.9	56.8	55.9
CrosNNCLR	joint	**73.2**	**81.0**	**81.3**	**62.5**	**64.3**
SkeletonCLR	motion	53.3	50.8	63.4	39.6	**40.2**
CrosNNCLR	motion	**56.7**	**62.0**	**67.4**	**41.4**	36.4
SkeletonCLR	bone	**69.4**	67.4	72.6	48.4	52.0
CrosNNCLR	bone	64.3	**72.9**	**77.0**	**60.4**	**64.3**
3s-SkeletonCLR	joint+motion+bone	75.0	79.8	85.3	60.7	62.6
3s-CrosNNCLR	joint+motion+bone	**76.0**	**83.4**	**86.2**	**67.4**	**68.3**

As shown in Table 2, our CrosNNCLR* is compared with the original AimCLR on three modal datasets. On the skeleton single-modal dataset (joint), our CrosNNCLR* is better than the AimCLR. On the other skeleton modal datasets (motion, bone), the effect of our presented model is similar to the original model's performance. When three modalities are fused, the recognition performance of our 3s-CrosNNCLR* is superior to the 3s-AimCLR on the xsub60, PKU part I, and xset120, our model's recognition is similar to the original model on the xview60 and xsub120.

Table 2. Comparison of AimCLR and CrosNNCLR* linearity evaluation results on the NTU-60/120 and PKU datasets. "3s" means three stream fusion.

Method	Stream	NTU-60(%)		PKU(%)	NTU-120(%)	
		xsub	xview	part I	xview	xset
AimCLR	joint	74.3	79.7	83.4	63.4	63.4
CrosNNCLR*	joint	**75.2**	**79.7**	**83.8**	**63.8**	**64.9**
AimCLR	motion	**66.8**	**70.6**	72.0	**57.3**	54.4
CrosNNCLR*	motion	66.1	69.8	**72.8**	56.4	**54.9**
AimCLR	bone	**73.2**	77.0	82.0	62.9	63.4
CrosNNCLR*	bone	72.8	**77.2**	**83.9**	**63.1**	**65.7**
3s-AimCLR	joint+motion+bone	78.9	**83.8**	87.8	**68.2**	68.8
3s-CrosNNCLR*	joint+motion+bone	**79.2**	83.7	**88.6**	68.0	**69.9**

Through the above experiments, the overall results show the model can not only enhance the expression of semantic relevance among different views of the

same skeleton sequence, but also learn the low-level semantic information of the skeleton samples, and the recognition of the proposed method with different modalities of the skeleton data is verified through comparative experiments. We explain this result by that the model construction can take advantage of the close correlation among the views of the skeleton data, integrate the idea of CrosNNCLR, find out the semantically similar samples with view q, k as positive sample pairs, change the original model's method of obtaining more positive pairs, and capture richer view information.

4.4 Analysis of Ablation Experiment Results

To verify the effectiveness of our CrosNNCLR and CrosNNCLR* model for representation learning, we compare them with the latest unsupervised action recognition works, including AS-CAL, ISC, Colorization, LongT GAN, MS^{2}L, P&C, SkeletonCLR, CrosSCLR and AimCLR, etc. It is also compared with a small number of fully-supervised action recognition models, including Part-Aware LSTM, VA-RNN, Soft RNN, and ST-GCN, etc. The ablation experiments are mainly carried out on the NTU-60/120 and PKU datasets.

Table 3. Comparison of experimental accuracy on the NTU-60 dataset (joint).

Method	xsub(%)	xview(%)
LongT GAN(AAAI 18)	39.1	48.1
MS^{2}L(ACM MM 20)	52.6	–
P&C(CVPR 20)	50.7	76.3
AS-CAL(Information Sciences 21)	58.5	64.8
SkeletonCLR(CVPR 21)	68.3	76.4
CrosSCLR(CVPR 21)	72.9	79.9
AimCLR(AAAI 22)	74.3	79.7
CrosNNCLR(ours)	73.2	**81.0**
CrosNNCLR*(ours)	**75.2**	79.7

Table 4. Comparison of experimental accuracy on the NTU-60 dataset (joint+ motion+bone).

Method	xsub(%)	xview(%)
3s-Colorization(ICCV 21)	75.2	83.1
3s-SkeletonCLR(CVPR 21)	75.0	79.8
3s-CrosSCLR(CVPR 21)	77.8	83.4
3s-AimCLR(AAAI 22)	78.9	**83.8**
3s-CrosNNCLR(ours)	76.0	83.4
3s-CrosNNCLR*(ours)	**79.2**	83.7

Results of Linear Evaluation on the NTU-60 Dataset. As shown in Table 3, for the skeleton single-modal data (joint), the recognition accuracy of our CrosNNCLR respectively increases 4.9% and 4.6% over the original SkeletonCLR on the xsub and xview benchmarks dataset, then by 0.3% and 0.1% over the cross-modal contrastive learning CrosSCLR on the xview benchmark dataset, respectively. On the xsub benchmark dataset, the recognition accuracy of our CrosNNCLR* is 0.3% higher than the AimCLR. On the xview benchmark dataset, the recognition effect of our CrosNNCLR* is equal to the original AimCLR.

In Table 4, the performance of skeleton multimodal data (joint+motion +bone) is given. We can see that our 3s-CrosNNCLR respectively obtains 76.0% and 83.4% recognition accuracy, the presented 3s-CrosNNCLR* obtains 79.2% and 83.7% recognition accuracy respectively. Compared with the other models, these results further demonstrate the effectiveness of CrosNNCLR.

Table 5. Comparison of experimental accuracy on the PKU dataset (joint).

Method	part I(%)	part II(%)
Supervised:		
ST-GCN(AAAI 18)	84.1	48.2
VA-RNN(TPAMI 19)	84.1	**50.0**
Self-supervised:		
LongT GAN(AAAI 18)	67.7	26.0
MS^{2}L(ACM MM 20)	64.9	27.6
3s-CrosSCLR(CVPR 21)	84.9	21.2
ISC(ACM MM 21)	80.9	36.0
3s-AimCLR(AAAI 22)	87.8	38.5
3s-CrosNNCLR*(ours)	**88.6**	**44.7**

Table 6. Comparison of experimental accuracy on the NTU-120 dataset (joint).

Method	xsub(%)	xset(%)
Supervised:		
Part-Aware LSTM(CVPR 16)	25.5	26.3
Soft RNN(TPAMI 18)	36.3	44.9
Self-supervised:		
P&C(CVPR 20)	42.7	41.7
AS-CAL(Information Sciences 21)	48.6	49.2
3s-CrosSCLR(CVPR 21)	67.9	66.7
ISC(ACM MM 21)	67.9	67.1
3s-AimCLR(AAAI 22)	**68.2**	68.8
3s-CrosNNCLR*(ours)	68.0	**69.9**

Results of Linear Evaluation on the PKU Dataset. Table 5 gives a comparison of the current state-of-the-art approaches. Firstly, our 3s-CrosNNCLR* respectively achieves 88.6% and 44.7% recognition accuracy on the part I and part II datasets, which gains 0.8 and 6.2% points better than the original 3s-AimCLR, respectively, and higher than other algorithms. Secondly, the self-supervised learning approaches achieve higher recognition than some fully-supervised models, e.g., ST-GCN and VA-RNN, demonstrating that CrosNNCLR has a strong discriminatory ability to distinguish the motion pattern caused by skeleton noise.

Table 7. Comparison of linear evaluation results of SkeletonCLR and CrosNNCLR at different epochs on the NTU-60/120 datasets (joint).

Method	Datasets	300ep	400ep	500ep	600ep	700ep	800ep	900ep	1000ep
SkeletonCLR	xsub60	68.3	70.0	70.0	70.4	70.4	69.7	70.6	70.6
CrosNNCLR	xsub60	**70.8**	**71.4**	**71.9**	**72.7**	**73.4**	**73.2**	**73.5**	**73.7**
SkeletonCLR	xview60	76.4	74.9	74.3	74.1	74.0	73.7	73.6	73.1
CrosNNCLR	xview60	**76.5**	**78.4**	**79.4**	**80.0**	**80.7**	**81.0**	**80.7**	**80.9**
SkeletonCLR	xsub120	56.8	56.0	56.1	56.1	56.1	56.3	55.8	55.5
CrosNNCLR	xsub120	**60.7**	**61.8**	**62.0**	**62.1**	**62.4**	**62.5**	**62.7**	**62.7**
SkeletonCLR	xset120	55.9	54.7	54.9	55.2	54.9	54.1	54.6	54.7
CrosNNCLR	xset120	**62.2**	**62.8**	**63.4**	**63.8**	**64.0**	**64.3**	**64.5**	**64.8**

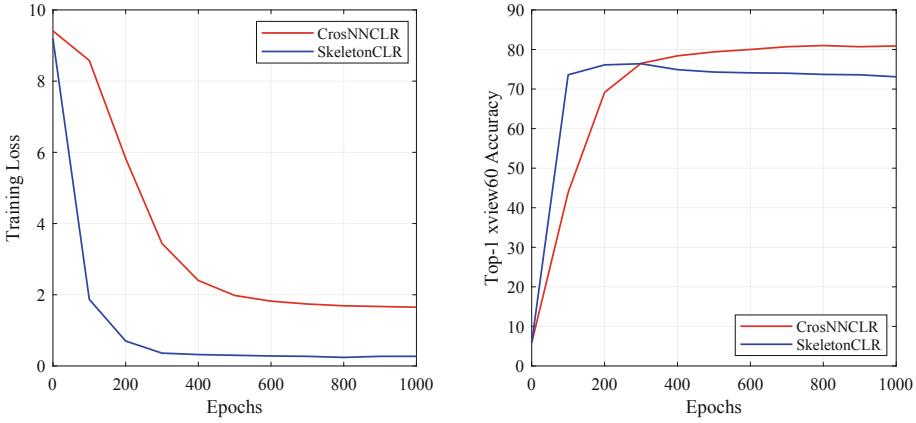

Fig. 4. CrosNNCLR vs SkeletonCLR Training curves and linear evaluation curves for xview60 linear evaluation.

Results of Linear Evaluation on the NTU-120 Dataset. As shown in Table 6, on the xsub benchmark dataset, our 3s-CrosNNCLR* achieves 68.0% recognition accuracy, which is similar to the 3s-AimCLR. On the xset benchmark

dataset, our 3s-CrosNNCLR* achieves 69.9% action recognition accuracy, which gains 1.1% improvement over the original 3s-AimCLR. In fully-supervision, the accuracy of the proposed algorithm is better than some methods, e.g., Part-Aware LSTM and Soft RNN, which verifies the validity of the proposed model again.

Selection of Epoch Values for CrosNNCLR. Given the skeleton single-modal data (joint), corresponding experiments are conducted on four datasets from the NTU-60/120 to select the epoch values suitable for our CrosNNCLR. Firstly, Table 7 and Fig. 4 show the performance of CrosNNCLR is superior to SkeletonCLR after 300 epochs on single-modal (joint), while the model does not converge. In contrast SkeletonCLR no longer has a significant increase in loss and accuracy after 300 epoch, proving that the model has reached convergence. Furthermore, the performance of our CrosNNCLR continues to improve when the loss value continues to decrease. At 1000 epochs, the proposed method obtains 73.7%, 80.9%, 62.7%, and 64.8% recognition accuracy respectively, further increasing the gap to 4∼8% points. Finally, we select the experimental results at 800 epochs training as the final evaluation results, mainly due to the weak feature expression of the model at the beginning of training, it is unable to learn deeper semantic information. As the number of iterations increases, the network will learn richer semantic representation and promote the convergence of the network model.

Table 8. Performance with only crop augmentation (joint) for xview60 linear evaluation.

Method	SkeletonCLR	CrosNNCLR	AimCLR	CrosNNCLR*
Full aug.	76.4	**81.0**	79.7	79.7
Only crop	51.5(↓ 24.9)	**63.0(↓ 18)**	53.3(↓ 26.4)	54.6(↓ 25.1)

Table 9. Embedding size (joint) for xview60 linear evaluation.

Embedding size	128	256	512	1024
Top-1	76.8	81.0	77.6	**80.7**
Top-5	96.4	**97.2**	96.8	**97.2**

Table 10. Memory bank size (joint) for xview60 linear evaluation.

Queue size	256	512	1024	2048	4096	8192	16384	32768
Top-1	75.5	74.5	78.2	78.7	74.2	79.0	77.6	**81.0**
Top-5	96.3	95.7	96.8	96.8	96.0	96.9	96.7	**97.2**

Data Augmentation. Both SkeletonCLR and AimCLR rely on multiple data augmentation methods to obtain the best performance. However, CrosNNCLR and CrosNNCLR* do not rely too much on complex augmentation approaches, because a richer real column of similar samples can be obtained from the cross-view nearest neighbors. As shown in Table 8, we remove the complex data augmentation methods and keep only one data augmentation method, random crops. Although the method proposed in this paper also benefits from complex data augmentation operations, CrosNNCLR relies much less on its removed data augmentation operations in comparison.

Embedding Size. As shown in Table 9, four embedding sizes have been selected for comparison, i.e., 128, 256, 512 and 1024, from which we can see that our CrosNNCLR is more robust and finds similar recognition results for different embedding sizes.

Memory Bank Size. Enhancing the number of samples in the memory bank usually improves the model performance, and the experimental results are shown in Table 10, which has a peak value of 32768. Overall, Using a larger memory bank in the cross-view nearest neighbor method increases the probability of capturing similar samples.

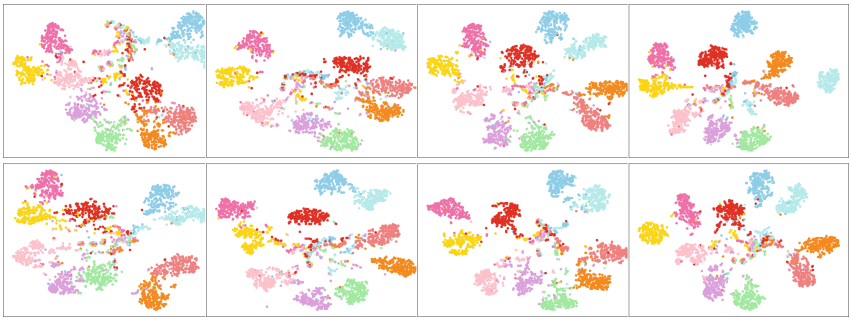

Fig. 5. The t-SNE visualization of the embedding features of SkeletonCLR, AimCLR, CrosNNCLR, and CrosNNCLR* on the xsub and xview dataset of NTU-60, where the first-row visualization results include: t-SNE(SkeletonCLR) and t-SNE(AimCLR) on the xsub and xview datasets. The second-row visualization results include: t-SNE(CrosNNCLR) and t-SNE(CrosNNCLR*) on the xsub and xview datasets.

Qualitative Analysis Results. To verify the effectiveness of inserting CrosNNCLR modules into existing models, the t-SNE [37] dimensionality reduction algorithm visualizes the embedded features distribution of SkeletonCLR, AimCLR, CrosNNCLR, and CrosNNCLR*. As shown in Fig. 5, 10 classes from the xsub and xview datasets of NTU-60 are selected for embedding comparisons. Compared to SkeletonCLR and AimCLR, the proposed method CrosNNCLR and CrosNNCLR* can cluster the embedding features of the same class more compactly, and separate the embedding features of different classes.

Quantitative Analysis Results. To more clearly and intuitively compare the action classification results of SkeletonCLR and CrosNNCLR, we plot the test results into a confusion matrix on the NTU-60 dataset. As shown in Fig. 6, we compare the 10 kinds of actions single-modal (joint) of xsub and xview datasets. In general, our CrosNNCLR is more accurate than SkeletonCLR in most of the actions, e.g., the classification of "eat meal" increased from 48% and 63% to 55% and 73%, respectively. The classification accuracy of CrosNNCLR is similar to SkeletonCLR for a few actions, e.g., 48% and 46% for "clapping" under the xsub evaluation benchmark, respectively. Thus, the conclusion is validated that our CrosNNCLR model can improve the feature representation of each type of action.

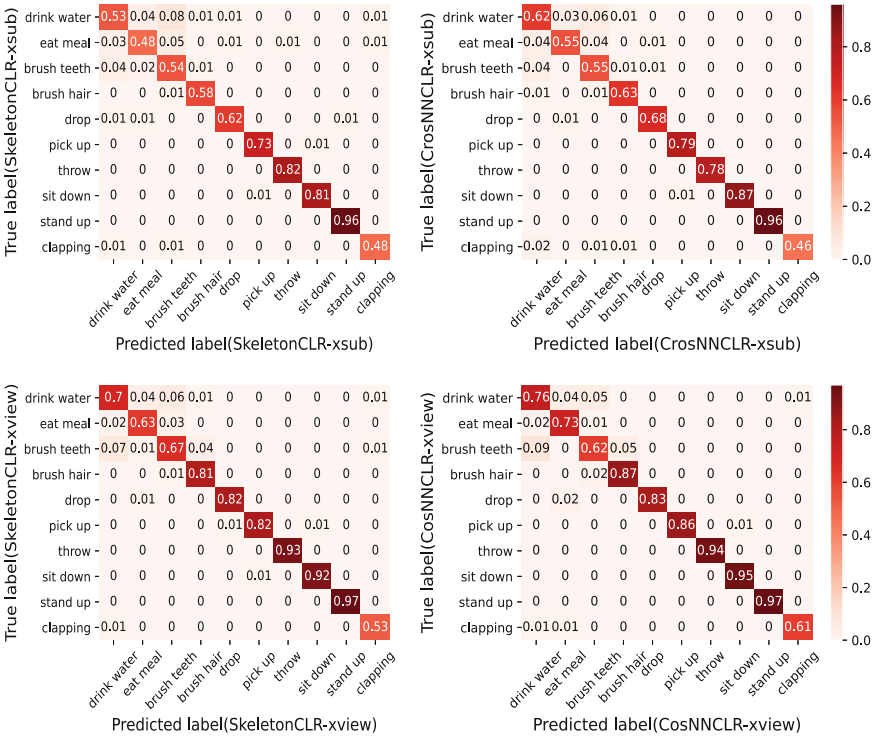

Fig. 6. Comparison of the confusion matrix on NTU-60 datasets

5 Conclusion

In this paper, a generic module called Cross-View Nearest Neighbor Contrastive Learning framework for self-supervised action Representation is proposed to obtain more positives. Our CrosNNCLR, as a view-level block, can be applied to

existing contrastive learning architectures in the plug-and-play manner, bringing consistent improvements. Moreover, under various linear evaluation protocols, our method outperforms previous state-of-the-art methods on the NTU-60/120 and PKU datasets, demonstrating that the model has good generalization in low-probability learning scenarios. In future work, we will study the self-supervised action recognition based on robot body skeletons, aiming at active human-robot collaboration and lightweight recognition model with knowledge distillation techniques.

Acknowledgements. This research was supported by the National Nature Science Foundation of China (61862015), the Science and Technology Project of Guangxi (AD21220114), the Guangxi Key Research and Development Program (AB17195025).

References

1. Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action representation, segmentation and recognition. Comput. Vis. Image Underst. **115**(2), 224–241 (2011)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
3. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference On Computer Vision, pp. 1440–1448 (2015)
4. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Patt. Anal. Mach. Intell. **40**(4), 834–848 (2017)
5. Li, C., Zhong, Q., Xie, D., Pu, S.: Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. arXiv preprint arXiv:1804.06055, 2018
6. Song, S., Lan, C., Xing, J., Zeng, W., Liu, J.: Spatio-temporal attention-based LSTM networks for 3D action recognition and detection. IEEE Trans. Image Process. **27**(7), 3459–3471 (2018)
7. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-temporal LSTM With trust gates for 3D human action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 816–833. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_50
8. Kay, W., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017
9. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3D action recognition. arXiv e-prints (2017)
10. Liang, D., Fan, G., Lin, G., Chen, W., Zhu, H.: Three-stream convolutional neural network with multi-task and ensemble learning for 3D action recognition. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2019)
11. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using LSTMs. In: International Conference on Machine Learning, pp. 843–852. PMLR (2015)

12. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 143–152 (2020)
13. Gui, L.-Y., Wang, Y.-X., Liang, X., Moura, J.M.F.: Adversarial geometry-aware human motion prediction. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11208, pp. 823–842. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01225-0_48
14. Kundu, J.N., Gor, M., Uppala, P.K., Radhakrishnan, V.B.: Unsupervised feature learning of human actions as trajectories in pose embedding manifold. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1459–1467. IEEE (2019)
15. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 69–84. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_5
16. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_40
17. Zhang, J., Zhao, Y., Saleh, M., Liu, P.: PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning, pp. 11328–11339. PMLR (2020)
18. Thoker, F.M., Doughty, H., Snoek, C.G.M.: Skeleton-contrastive 3D action representation learning. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 1655–1663 (2021)
19. Ni, B., Wang, G., Moulin, P.: RGBD-HuDaAct: a color-depth video database for human daily activity recognition. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV workshops), pp. 1147–1153. IEEE (2011)
20. Vemulapalli, R., Arrate, F., Chellappa, R.: 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, pp. 588–595 (2014)
21. Vemulapalli, R., Chellapa, R.: Rolling rotations for recognizing human actions from 3D skeletal data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4471–4479 (2016)
22. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining Actionlet ensemble for action recognition with depth cameras. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1290–1297. IEEE (2012)
23. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive neural networks for high performance skeleton-based human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **41**(8), 1963–1978 (2019)
24. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1110–1118 (2015)
25. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI Conference on Artificial Intelligence (2018)
26. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)

27. Gidaris, S., Bursuc, A., Puy, G., Komodakis, N., Cord, M., Perez, P.: OBoW: online bag-of-visual-words generation for self-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6830–6840 (2021)
28. Han, T., Xie, W., Zisserman, A.: Self-supervised co-training for video representation learning. Adv. Neural. Inf. Process. Syst. **33**, 5679–5690 (2020)
29. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9588–9597 (2021)
30. Zheng, N., Wen, J., Liu, R., Long, L., Gong, Z.: Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In: AAAI-18 (2018)
31. Su, K., Liu, X., Shlizerman, E.: Predict & cluster: Unsupervised skeleton based action recognition (2019)
32. Li, L., Wang, M., Ni, B., Wang, H., Yang, J., Zhang, W.: 3D human action representation learning via cross-view consistency pursuit (2021)
33. Guo, T., Liu, H., Chen, Z., Liu, M., Wang, T., Ding, R.: Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. arXiv e-prints (2021)
34. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: NTU RGB+D: a large scale dataset for 3D human activity analysis. In: IEEE Computer Society, pp. 1010–1019 (2016)
35. Liu, J., Song, S., Liu, C., Li, Y., Hu, Y.: A benchmark dataset and comparison study for multi-modal human action analytics. ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) **16**(2), 1–24 (2020)
36. 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 Trans. Pattern Anal. Mach. Intell. **42**(10), 2684–2701 (2020)
37. Shi, S.: Visualizing data using GTSNE (2021)