Ranking-based Self-Supervised Representation Learning for Skeleton-based Action Recognition

Bizhu Wu, Junliang Chen, Jinheng Xie, Qiufu Li, Jianfeng Ren, Ruibin Bai, Rong Qu, Linlin Shen

Abstract*—Recently, researchers have achieved significant results in the skeleton-based action recognition. To better model the skeleton sequences, we drive the encoder to learn more discriminative representations in the self-supervised setting. We find that instead of clustering feature vectors to assign pseudo labels for samples as in DeepCluster, ranking them is a more reasonable, reliable, and efficient way to learn more effective feature representations. With this intuition, we propose a novel self-supervised learning framework, DeepRank. Specifically, we rank triplets of skeleton sequences with the ranking labels, obtained from the relative distances among them. Besides, to deeply mine complementary discriminative information that exists in different modalities of skeleton sequences, we further propose Multi-View DeepRank (MV-DeepRank) to enable encoders to comprehensively learn complementary features from multiple modalities. Extensive experimental results on the NTU RGB+D, NTU RGB+D 120, PKU-MMD I, and PKU-MMD II datasets under various evaluation settings demonstrate the generality, transferability, and superiority of our proposed self-supervised learning frameworks. Notably, our frameworks surpass the previous methods that employ the same backbone networks as ours by at least 1.8% (ST-GCN) and 2.1% (STTFormer) under the finetuning setting. Additionally, DeepRank gains a significant advantage on computational complexities, $O(1)$, over the contrastive learning-based methods, $O(\text{batch size})$, and the clustering-based methods, $O(\text{number of clusters})$.

***Index Terms*—Skeleton-Based Action Recognition, Self-Supervised Learning, Multi-modal Fusion.**

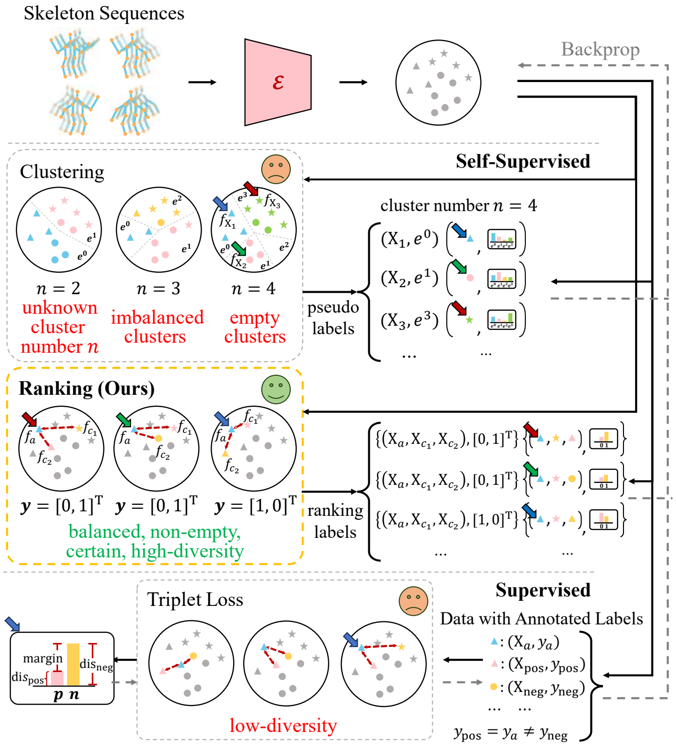
Skeleton Sequences
$\mathcal{E}$
Backprop
Self-Supervised
Clustering
e^0 e^1 e^2 e^3
f_{X_1} f_{X_2} f_{X_3}
$n = 2$
unknown cluster number n
$n = 3$
imbalanced clusters
$n = 4$
empty clusters
cluster number $n = 4$
pseudo labels
(X_1, e^0)
(X_2, e^1)
(X_3, e^3)
Ranking (Ours)
f_a f_{c_1} f_{c_2}
$\boldsymbol{y} = [0,1]^T$
$\boldsymbol{y} = [0,1]^T$
$\boldsymbol{y} = [1,0]^T$
balanced, non-empty, certain, high-diversity
ranking labels
$\{(X_a, X_{c_1}, X_{c_2}), [0,1]^T\}$
$\{(X_a, X_{c_1}, X_{c_2}), [0,1]^T\}$
$\{(X_a, X_{c_1}, X_{c_2}), [1,0]^T\}$
0 1
Triplet Loss
low-diversity
margin
dis_pos
dis_neg
$\boldsymbol{p}$ $\boldsymbol{n}$
Supervised
Data with Annotated Labels
▲: (X_a, y_a)
▲: $(X_\text{pos}, y_\text{pos})$
●: $(X_\text{neg}, y_\text{neg})$
$y_\text{pos} = y_a \neq y_\text{neg}$

Fig. 1. **DeepRank (ours) *vs.* DeepCluster *vs.* Triplet Loss. DeepCluster**: These methods usually generate pseudo labels by clustering the extracted features, and then improve the encoding by pseudo-labeled samples. **DeepRanking**: Our proposed self-supervised learning framework DeepRank ranks samples relatively by distances among extracted features to avoid the problem of empty clusters and imbalance clusters in DeepCluster. **Triplet Loss**: These methods construct triplets of samples but impose supervised constraints on both positive and negative samples.

I. Introduction

SKELETON-BASED action recognition has attracted widespread attention from researchers because eliminating background interference helps the effective learning of actions. Existing methods often focus on the data representation learning [1], [2] and the network architecture design [3]–[6] to perform this task in supervised setting, which still relies heavily on meticulously annotated training data. This is both labor-intensive and time-consuming to obtain. Additionally, limited supervision can lead to overfitting, particularly in models like transformers, which have weak inductive bias and high model capacity. These challenges highlight the need for self-supervised learning approaches for robust skeletal representations.

In the literature, prevalent pretext tasks originally designed for image data have been extended to 3D action representation learning, including reconstruction [7]–[9] and contrastive learning [10]–[12]. However, when adapting one of the most representative self-supervised learning approaches, DeepCluster [13], to 3D action representation learning, we found several limitations. Specifically, DeepCluster operates by first extracting features from all samples, clustering them using the K-means algorithm, assigning pseudo labels to the samples, and subsequently training the encoder using these *(sample, pseudo label)* pairs, as depicted at the top of Fig. 1. The clustering process introduces challenges, including the potential for empty clusters, cluster imbalance, and the cumbersome task of selecting an optimal number of clusters, denoted by n. When n is too large, reliably classifying samples into specific pseudo categories is challenging. Also, the encoder may capture irrelevant features by focusing on trivial differences between samples in the same categories. Conversely, when n is small, the pseudo-classification task is oversimplified. The encoder may be guided to identify commonalities among samples in different categories. These issues result in suboptimal pseudo labels, which, in turn, hinder the training of a discriminative encoder without additional techniques, such as explicitly constraining the number of samples per pseudo-class [10], [13].

Inspired by Gepshtein *et al.* [14] that people are far better at making relative judgments than absolute judgments, we transform the absolute judgment in DeepCluster, *i.e.,* classifying samples into specific pseudo categories, into a relative one and propose a self-supervised learning framework, **DeepRank**. Specifically, we rank triplets of samples and employ the distance between features as the ranking metric. As shown in Fig. 1, given a triplet of samples, we rank the two candidates $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ according to their feature distances to the anchor a.

During the ranking process, we have a closer candidate and a farther one in each triplet. So, there are two categories in this ranking task, either the first candidate c_1 or the second candidate c_2 is closer to the anchor a. The **certain** number of categories in this task saves us a lot of time and computations to carry out a lot of experiments to choose an optimal number of clusters n like DeepCluster (Table VI), and hence our proposed DeepRank is more efficient. Meanwhile, after converting the difficult absolute task into a relative ranking one, we offer a **simpler** solution. For oversimplified tasks like judging whether samples are in the same category, relative ranking them provides **nuanced insights**, revealing differences among samples in different categories. Besides, each triplet has its counterpart one, *i.e.*, switching the position of candidates, which is classified into the other category. So, we ensure that each ranking category is **non-empty** and **balanced**. Therefore, the ranking process can produce reliable and high-quality self-supervisions for our proposed self-supervised learning framework. With such self-supervisions, we can drive the encoder to learn more effective representations, and thus increase the performance of downstream tasks, such as skeleton-based action recognition.

It is worth mentioning that our DeepRank is different from the triplet loss [15] and the contrastive learning-based methods [12], [16], [17]. For the triplet loss, it indeed constructs triplets of samples, but the positive samples must be in the same categories as the anchors, while the negative ones must be in different categories from the anchors (as displayed at the bottom of Fig. 1). Thus, it requires ground-truth labels to train the encoder in a supervised manner. In contrast, the encoder of DeepRank is trained in a self-supervised manner. Besides, since we do not impose restrictions on samples, DeepRank has more combinations of triplets, *i.e.*, **high diversity**. For contrastive learning methods, they would wrongly consider samples that belong to the same category but are augmented from different samples as negative pairs, and thus push apart their features. In contrast, in our framework, samples in the same categories with the anchor are more likely regarded as positive ones (closer candidates in triplets) due to the closer feature distances. Thus, with more accurate self-supervisions, the pretrained encoders in our self-supervised learning frameworks can extract more discriminant representations.

Commonly used three modalities of skeleton sequences, joint, bone, and motion, may be complementary to each other. For example, both actions "pushing" and "pat on back" put the hand of one person behind the back of the other person, so these two actions are similar in the modality of joint. However, "pushing" is an urgent action while "pat on back" is a gentle one, so these two actions vary greatly in speed, which means they are distinct in the modality of motion. To simultaneously utilize complementary features of different modalities, we further propose **M**ulti-**V**iew **DeepRank** (**MV-DeepRank**). In this self-supervised learning framework, we design three different views of ranking tasks, *i.e.*, "self" view, "other" view, and "all" view, incorporating comprehensive information from various modalities in diverse ways. Meanwhile, we construct suitable network architectures for encoders to complementarily learn from the multi-view ranking task, driving encoders to not only retain the information hidden in the individual modality but also master the information across multiple modalities.

To validate the effectiveness of the proposed self-supervised learning frameworks, we conduct extensive experiments on four benchmarks, *i.e.*, NTU RGB+D [18], NTU RGB+D 120 [19], and PKU-MMD I and II [20]. Our frameworks achieve state-of-the-art results under various evaluation settings.

Our contributions can be summarized as follows: 1) To alleviate the problems of DeepCluster, we propose a concise, effective and efficient self-supervised learning framework, DeepRank, to achieve more reliable and robust self-supervisions. 2) We further propose MV-DeepRank to merge information from different modalities of skeleton sequences in three distinct ways, and promote the encoders to capture information across various views. 3) Comprehensive experiments on four popular datasets show that the proposed self-supervised learning frameworks are effective and well-generalized.

II. Related Work

A. Skeleton-Based Action Recognition

Many algorithms have been developed for skeleton-based action recognition. According to [21], existing works can be summarized into two main groups: traditional methods based on hand-crafted features and deep learning-based methods. Traditional methods [22]–[24] often require a great deal of prior knowledge to represent skeleton sequences in the form of useful features like the 3D joint position feature and the local occupancy pattern [23]. After deep learning prevailed in computer vision, researchers drove models to automatically learn effective features from massive data.

Deep learning-based methods can be further divided into four categories by network architectures [25], *i.e.*, CNN-based methods, RNN-based methods, GCN-based methods, and transformer-based methods. CNN-based methods [1], [26], [27] often focus on the data representation and represent the skeleton sequences in the form of 2D grid-shape images, *e.g.*, Ke *et al.* [1] represented the relative positions between reference joints and the other joints over time in different channels of cylindrical coordinates as images, and utilized CNN to extract their features. Since the skeleton sequence is sequential data, it is natural to use RNN-based networks to model it [3], [28]. Du *et al.* [3] divided the human skeleton into five body parts in the spatial domain, concatenated the joints in a body part of a frame as an input timestep of RNN-based networks, and then hierarchically fused features from parts to the whole. GCN-based methods [4]–[6], [29]–[31] represent the skeleton sequences in the form of graphs. Typically, ST-GCN [4] used graphs to record whether the joints are connected or not and applied graph convolution both in the spatial and temporal dimensions to extract features. Recently, transformer-based networks [32], [33] popularized among the computer vision community and achieved state-of-the-art results over many kinds of tasks, including the skeleton-based action recognition task [34]–[36]. For instance, STTFormer [34] split the skeleton sequences into non-overlapping tuples along the temporal dimension, extracted multi-joint representations among adjacent frames by self-attention modules, and aggregated the features of tuples. In this paper, both ST-GCN and STTFormer are chosen as the backbone networks of our framework.

B. Self-Supervised Learning

Self-supervised learning attracts much attention since it enables feature learning without manual annotations, *i.e.*, it exploits the information hidden in data (such as context structure) via well-designed pretext tasks [37]–[39] to enhance the data representation power of the models. Among all the effective pretext tasks, generation-based methods and contrastive learning-based methods are two mainstreams [40], [41].

Generation-based methods [42]–[44] learn features by generating content for corresponding modalities. Some skeletal representation learning models leverage on pretext tasks in the same type to enhance their encoders, like predicting the future motion [7] and reconstructing the interval frames [8]. Among these methods, masked contents modeling [42] has been popular recently, which aims to predict specific contents for masked patches. SkeletonMAE [9] deployed the idea of Masked Auto-encoder (MAE) [42] on the skeleton sequences, predicting the joint coordinates of masked regions.

Contrastive learning-based methods [45]–[47] pull the representations of positive pairs closer while pushing apart those negative ones. Similarly, a few models [10]–[12], [17], [48]–[50] have employed this thought to train their skeletal encoders. CrosSCLR [51] trained its encoder on the MoCo v2 [16] framework and fused the information from other modalities to enhance the representation. Compared to deploying the extracted features of samples in the contrastive loss, CPM [52] utilized the similarity distribution of the given sample with regard to all samples in the contextual queue.

The contrastive learning-based methods have common ground with metric learning methods, *e.g.*, researchers tried to enhance the feature discrimination power in a self-supervised manner. Besides treating other samples as negative pairs, Fu *et al.* [53] and Pfister *et al.* [54] ranked images with different transformations by pairwise ranking loss to preserve intra-class variance. But these methods still have problems of wrongly considering samples in the same category but augmented from different samples as negative ones. TransRank [55] used a margin ranking loss to predict the confidence score of deciding the transformation to process a given clip. Carr *et al.* [56] simplified the difficult permutation task by ranking the relative position of the shuffled image patches. All these models rank among the views augmented from the same sample. On the contrary, in our frameworks, skeleton sequences in the same classes with the anchor, instead of those augmented from the same sample, are more likely regarded as positive ones (closer candidates in triplets) due to the closer relative distances of their features. Thus, with more accurate self-supervisions, encoders pretrained in our proposed self-supervised learning frameworks can extract more robust representations. To the best of our knowledge, our work is the first inter-sample ranking-based self-supervision approach for skeleton-based action recognition, which achieves better performance than existing clustering-based self-supervision methods.

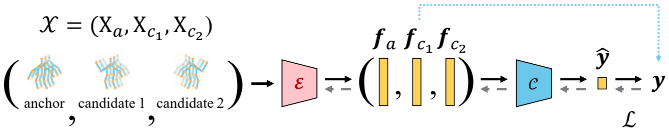
$\mathcal{X} = (\mathrm{X}_a, \mathrm{X}_{c_1}, \mathrm{X}_{c_2})$
anchor
candidate 1
candidate 2
$\mathcal{E}$
$\boldsymbol{f}_a$ $\boldsymbol{f}_{c_1}$ $\boldsymbol{f}_{c_2}$
$\mathcal{C}$
$\hat{\boldsymbol{y}}$
$\boldsymbol{y}$
$\mathcal{L}$

Fig. 2. **Details of DeepRank.** Given a triplet of skeleton sequences $\mathcal{X} = (\mathrm{X}_a, \mathrm{X}_{c_1}, \mathrm{X}_{c_2})$, we feed them into the encoder $\mathcal{E}$ to get their features. Then, we compare the distances between the two candidates and the anchor to get the ranking label for this triplet. To improve the encoder $\mathcal{E}$, we attach a classifier $\mathcal{C}$ after the encoder $\mathcal{E}$ and rank this triplet by predicting its ranking label. Best viewed in color.

III. Proposed DeepRank

A. Triplet Generation

Let $\mathcal{S}$ be the set of unlabeled skeleton sequences. We first construct a skeleton sequence triplet dataset $\mathcal{D}$ by randomly sampling skeleton sequences from $\mathcal{S}$,

$$\mathcal{D} = \{\mathcal{X} = (\mathrm{X}_a, \mathrm{X}_{c_1}, \mathrm{X}_{c_2}) : \forall \mathrm{X}_a, \mathrm{X}_{c_1}, \mathrm{X}_{c_2} \in \mathcal{S}\}, \qquad (1)$$

where $\mathrm{X}_a \in \mathbb{R}^{T_0 \times V_0 \times C_0}$ denotes an anchor sequence, and X_{c_1} and X_{c_2} are two candidate sequences. T_0, V_0, and C_0 are the temporal length, the number of joints, and the number of coordinate channels, respectively. Based on the distance between the anchor and the candidates, we can divide the triplets in $\mathcal{D}$ into two categories. Specifically, for any triplet of skeleton sequences $\mathcal{X} = (\mathrm{X}_a, \mathrm{X}_{c_1}, \mathrm{X}_{c_2}) \in \mathcal{D}$, we label it by,

$$\boldsymbol{y} = \begin{cases} [1, 0]^\top, & \text{if } d(\mathrm{X}_a, \mathrm{X}_{c_1}) < d(\mathrm{X}_a, \mathrm{X}_{c_2}), \\ [0, 1]^\top, & \text{otherwise}, \end{cases} \qquad (2)$$

$$d(\mathrm{X}_a, \mathrm{X}_{c_k}) = \mathcal{F}_{\mathrm{dis}}(\boldsymbol{f}_a, \boldsymbol{f}_{c_k}), \quad k \in \{1, 2\}, \qquad (3)$$

where $\mathcal{F}_{\mathrm{dis}}(\cdot, \cdot)$ is a distance metric function to measure the distance between two feature vectors,

$$\boldsymbol{f}_a = \mathcal{E}(\mathrm{X}_a) \in \mathbb{R}^{256 \times 1}, \qquad (4)$$

$$\boldsymbol{f}_{c_k} = \mathcal{E}(\mathrm{X}_{c_k}) \in \mathbb{R}^{256 \times 1}, \quad k \in \{1, 2\}, \qquad (5)$$

and $\mathcal{E}$ is the skeletal encoder. If the first candidate is closer to the anchor than the second one in a triplet $\mathcal{X}$ in terms of feature distance, we add $\mathcal{X}$ into the first category, and add it into the second category otherwise, indicating the second candidate is more similar to the anchor. One can easily find out that, in contrast with the uncertain number of clusters during the clustering process (see Fig. 1), there are two certain categories in our proposed ranking task, making the training

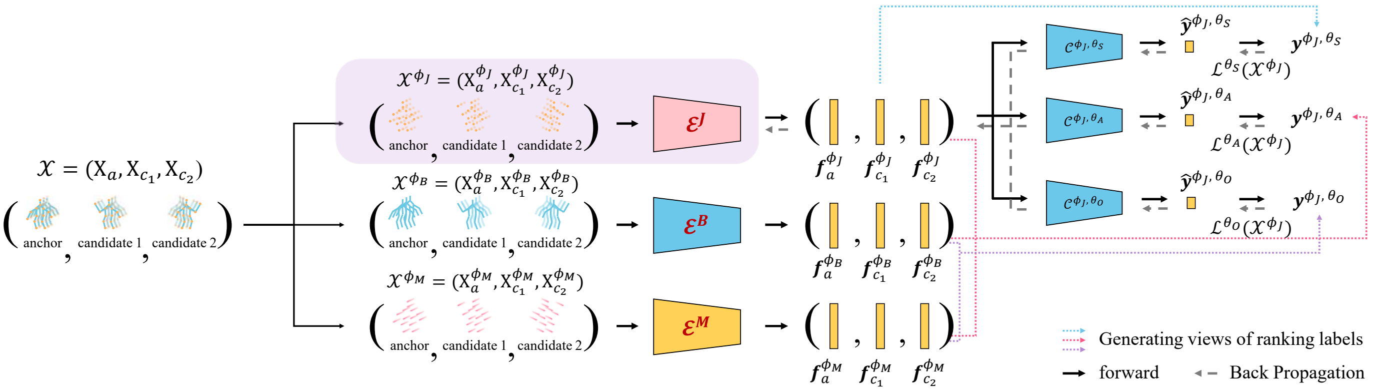
$\mathcal{X} = (X_a, X_{c_1}, X_{c_2})$
anchor
candidate 1
candidate 2
$\mathcal{X}^{\phi_J} = (X_a^{\phi_J}, X_{c_1}^{\phi_J}, X_{c_2}^{\phi_J})$
$\mathcal{X}^{\phi_B} = (X_a^{\phi_B}, X_{c_1}^{\phi_B}, X_{c_2}^{\phi_B})$
$\mathcal{X}^{\phi_M} = (X_a^{\phi_M}, X_{c_1}^{\phi_M}, X_{c_2}^{\phi_M})$
$\mathcal{E}^J$
$\mathcal{E}^B$
$\mathcal{E}^M$
$f_a^{\phi_J}$ $f_{c_1}^{\phi_J}$ $f_{c_2}^{\phi_J}$
$f_a^{\phi_B}$ $f_{c_1}^{\phi_B}$ $f_{c_2}^{\phi_B}$
$f_a^{\phi_M}$ $f_{c_1}^{\phi_M}$ $f_{c_2}^{\phi_M}$
$\mathcal{C}^{\phi_J,\theta_S}$
$\mathcal{C}^{\phi_J,\theta_A}$
$\mathcal{C}^{\phi_J,\theta_O}$
$\hat{y}^{\phi_J,\theta_S}$
$\hat{y}^{\phi_J,\theta_A}$
$\hat{y}^{\phi_J,\theta_O}$
y^{ϕ_J,θ_S}
y^{ϕ_J,θ_A}
y^{ϕ_J,θ_O}
$\mathcal{L}^{\theta_S}(\mathcal{X}^{\phi_J})$
$\mathcal{L}^{\theta_A}(\mathcal{X}^{\phi_J})$
$\mathcal{L}^{\theta_O}(\mathcal{X}^{\phi_J})$
Generating views of ranking labels
forward
Back Propagation

Fig. 3. **Details of MV-DeepRank in the modality of joint.** Given a triplet of skeleton sequences $\mathcal{X} = (X_a, X_{c_1}, X_{c_2})$, we compute its corresponding three modalities and separately feed them into their respective encoders. Then, we compare the distances between the two candidates and the anchor to get three views of the ranking label for this triplet. The "self" view utilizes the information in the self modality. The "other" view combines the information from the other two modalities. The "all" view integrates relationships among skeleton sequences from all three modalities. To improve the joint encoder $\mathcal{E}^J$, we attach three classifiers after the joint encoder $\mathcal{E}^J$ and rank this triplet by predicting the ranking labels of the three views. Best viewed in color.

process more **certain**. Besides, the numbers of triplets in two categories are the same (**balanced** and **non-empty**), as that $\mathcal{X} = (X_a, X_{c_1}, X_{c_2})$ belongs to one category if and only if its counterpart $\mathcal{X}' = (X_a, X_{c_2}, X_{c_1})$ belongs to the other category.

B. Ranking Learning

With the skeleton sequence triplet dataset $\mathcal{D}$, the skeleton sequence ranking problem is transformed into a binary classification task. As Fig. 2 shows, in the ranking learning, a triplet $\mathcal{X}$ is first extracted its feature by the encoder,

$$\begin{aligned}\mathcal{E}(\mathcal{X}) = \mathcal{E}(X_a, X_{c_1}, X_{c_2}) &= (\mathcal{E}(X_a), \mathcal{E}(X_{c_1}), \mathcal{E}(X_{c_2})) \\ &= (\boldsymbol{f}_a, \boldsymbol{f}_{c_1}, \boldsymbol{f}_{c_2}),\end{aligned} \tag{6}$$

which is then transformed into the predicted label $\hat{\boldsymbol{y}}$ by a classifier $\mathcal{C}$ containing two fully-connected (FC) layers. The classifier $\mathcal{C}$ concatenates two candidate features with the anchor feature respectively and feeds them into the first FC layer. Then, the subtraction of the output intermediate features is fed into the second FC layer, and finally, the predicted label is generated by applying a Softmax function, *i.e.*,

$$\tilde{\boldsymbol{f}}_k = \mathcal{F}_{\text{FC}_0}(\mathcal{F}_{\text{Concat}}(\boldsymbol{f}_a, \boldsymbol{f}_{c_k})) \in \mathbb{R}^{256}, \quad k \in \{1, 2\}, \tag{7}$$

$$\hat{\boldsymbol{y}} = \mathcal{F}_{\text{Softmax}}(\mathcal{F}_{\text{FC}_1}(\tilde{\boldsymbol{f}}_1 - \tilde{\boldsymbol{f}}_2)) \in \mathbb{R}^2, \tag{8}$$

where $\mathcal{F}_{\text{FC}_0}$ and $\mathcal{F}_{\text{FC}_1}$ are two FC layers with the size of 512×256 and 256×2. $\mathcal{F}_{\text{Concat}}$ denotes the concatenation operation.

In the training, we adopt the cross-entropy loss,

$$\mathcal{L}(\mathcal{X}) = -\boldsymbol{y}^\top \log \hat{\boldsymbol{y}}. \tag{9}$$

One can easily find that the proposed framework has a time complexity of $O(1)$, with a huge advantage over $O(n)$ for clustering-based methods and $O(\text{batch size})$ for contrastive learning-based methods, where n is the number of clusters.

IV. Proposed Multi-View DeepRank

Skeleton sequences have three common modalities, *i.e.*,

$$\mathbb{M} = \{\phi_J, \phi_B, \phi_M\}, \tag{10}$$

where ϕ_J denotes the joint modality, which records the 3D coordinates of human keypoints over time; ϕ_B represents the bone modality, capturing the length and direction of bones by computing the differences between adjacent joints in each frame; and ϕ_M refers to the motion modality, which describes the temporal displacement of joints and is obtained by applying a temporal difference to joint coordinates. These three modalities explicitly record the skeletal information from different perspectives.

We employ three encoders with identical architectures, denoted as $\{\mathcal{E}^J, \mathcal{E}^B, \mathcal{E}^M\}$, to extract features from different modalities and perform ranking learning separately. It is worth noting that both the bone and motion modalities are directly derived from the joint data through simple subtraction operations. As a result, all three modalities share consistent data shape, enabling a unified processing pipeline across modalities.

To reflect the various relationships among the skeleton sequences, for each modality $\phi \in \mathbb{M}$, we design three views, *i.e.*,

$$\mathbb{V} = \{\theta_S, \theta_O, \theta_A\}, \tag{11}$$

where θ_S refers to the "self" view, reflecting the implicit relationships among skeleton sequences in the self modality; θ_O refers to the "other" view, integrating the implicit relationships among skeleton sequences in the other two modalities; And θ_A refers to the "all" view, merging the implicit relationships among skeleton sequences in all three modalities. To comprehensively explore the information from all modalities and learn a better representation, we propose **Multi-View DeepRank** (**MV-DeepRank**) to explicitly incorporate information from different modalities and various views. Similarly, it iterates between two steps: Multi-View Triplet Generation and Multi-View Ranking Learning.

A. Multi-View Triplet Generation

Given a triplet of skeleton sequences $\mathcal{X} = (X_a, X_{c_1}, X_{c_2})$, we can compute its corresponding three modalities, *i.e.*,

$$\mathcal{X}^\phi = (x_a^\phi, x_{c_1}^\phi, x_{c_2}^\phi), \quad \phi \in \mathbb{M}. \tag{12}$$

In any view $\theta \in \mathbb{V}$, we label the triplet $\mathcal{X}^\phi$ using the ranking rule similar to that defined in Eq. (2), *i.e.*,

$$\boldsymbol{y}^{\phi,\theta} = \begin{cases} [1,0]^\top, & \text{if } d^\theta(\mathrm{X}_a,\mathrm{X}_{c_1}) < d^\theta(\mathrm{X}_a,\mathrm{X}_{c_2}), \\ [0,1]^\top, & \text{otherwise,} \end{cases} \tag{13}$$

where d^θ is metric function to measure the distances between candidate features and the anchor feature in the view of θ. For any triplet $\mathcal{X}^\phi = (x_a^\phi, x_{c_1}^\phi, x_{c_2}^\phi)$ with modality $\phi \in \mathbb{M}$, in "self" view, we generate its ranking label $\boldsymbol{y}^{\phi,\theta_S}$ using

$$d^{\theta_S}(\mathrm{X}_a^\phi, \mathrm{X}_{c_k}^\phi) = \mathcal{F}_{\mathrm{dis}}(\boldsymbol{f}_a^\phi, \boldsymbol{f}_{c_k}^\phi), \tag{14}$$

$$\boldsymbol{f}_a^\phi = \mathcal{E}^\phi(\mathrm{X}_a^\phi), \tag{15}$$

$$\boldsymbol{f}_{c_k}^\phi = \mathcal{E}^\phi(\mathrm{X}_{c_k}^\phi), \quad k \in \{1,2\}; \tag{16}$$

In "other" view, we generate its ranking label $\boldsymbol{y}^{\phi,\theta_O}$, using

$$d^{\theta_O}(\mathrm{X}_a^\phi, \mathrm{X}_{c_k}^\phi) = \sum_{\substack{\phi' \in \mathbb{M} \\ \phi' \neq \phi}} \mathcal{F}_{\mathrm{dis}}(\boldsymbol{f}_a^{\phi'}, \boldsymbol{f}_{c_k}^{\phi'}), \quad k \in \{1,2\}, \tag{17}$$

to further widen the gap between the distances of two candidates and the anchor, or correct the ranking results from a single modality, producing a more confident ranking result; In "all" view, we generate its ranking label $\boldsymbol{y}^{\phi,\theta_A}$ using

$$d^{\theta_A}(\mathrm{X}_a^\phi, \mathrm{X}_{c_k}^\phi) = \sum_{\phi \in \mathbb{M}} \mathcal{F}_{\mathrm{dis}}(\boldsymbol{f}_a^\phi, \boldsymbol{f}_{c_k}^\phi), \tag{18}$$

to comprehensively integrate clues from all three modalities.

B. Multi-View Ranking Learning

In the multi-view ranking learning, for any triplet $\mathcal{X}^\phi = (x_a^\phi, x_{c_1}^\phi, x_{c_2}^\phi)$ with modality $\phi \in \mathbb{M}$, we predict its three labels using three classifiers $\{\mathcal{C}^{\phi,\theta} : \phi \in \mathbb{M}, \theta \in \mathbb{V}\}$ in different views (see Fig. 3). Similarly, each classifier contains two FC layers and generates the predicted label from the anchor feature and candidate features, *i.e.*,

$$\tilde{\boldsymbol{f}}_k^{\phi,\theta} = \mathcal{F}_{\mathrm{FC}_0}^{\phi,\theta}(\mathcal{F}_{\mathrm{Concat}}(\boldsymbol{f}_a^\phi, \boldsymbol{f}_{c_k}^\phi)), \tag{19}$$

$$\hat{\boldsymbol{y}}^{\phi,\theta} = \mathcal{F}_{\mathrm{Softmax}}(\mathcal{F}_{\mathrm{FC}_1}^{\phi,\theta}(\tilde{\boldsymbol{f}}_1^{\phi,\theta} - \tilde{\boldsymbol{f}}_2^{\phi,\theta})). \tag{20}$$

The loss is:

$$\mathcal{L}(\mathcal{X}^\phi) = \sum_{\theta \in \mathbb{V}} \lambda^\theta \cdot \mathcal{L}^\theta(\mathcal{X}^\phi) = -\sum_{\theta \in \mathbb{V}} \lambda^\theta \cdot (\boldsymbol{y}^{\phi,\theta})^\top \log \hat{\boldsymbol{y}}^{\phi,\theta}, \tag{21}$$

where λ^θ is the weight hyper-parameters of different views.

In our work, DeepRank refers to the joint encoder trained with ranking labels in the self view. 3s-DeepRank indicates that the encoders of three modalities are separately trained with ranking labels in the self view. MV-DeepRank refers to the joint encoder trained with ranking labels in three views. 3s-MV-DeepRank indicates that the encoders of three modalities are trained with ranking labels in three respective views.

C. Discussions on Ranking Labels from Three Views

The ranking labels $\boldsymbol{y}^{\phi,\theta_O}$ and $\boldsymbol{y}^{\phi,\theta_A}$ are generated by incorporating information from multiple modalities, and hence they are different from $\boldsymbol{y}^{\phi,\theta_S}$ which only considers hidden clues from the individual modality. To quantify the differences, we analyze the proportion of ranking labels $\boldsymbol{y}^{\phi_J,\theta_O}$ and $\boldsymbol{y}^{\phi_J,\theta_A}$ that are not equal to $\boldsymbol{y}^{\phi_J,\theta_S}$ in the modality of joint. The results were obtained from randomly sampled 124,992,000 triplets from NTU RGB+D [18]. As shown in Fig. 4, nearly 28.9% and 1.1% of ranking labels $\boldsymbol{y}^{\phi_J,\theta_O}$ and $\boldsymbol{y}^{\phi_J,\theta_A}$ are unequal to $\boldsymbol{y}^{\phi_J,\theta_S}$ for the same triplet. It is the differences among views that provide complementary information for each modality. Otherwise, MV-DeepRank will degrade to DeepRank.

The way of learning complementary information in MV-DeepRank is similar to Multi-Head Deep Clustering [57], which separately clusters two modalities of RGB videos, *i.e.*, audio and RGB frames, and trains their encoders with clustering self-supervisions in two modalities. Similarly, the audio and RGB frames from the same video may be clustered into totally different groups. They believe that the semantic correlation and the differences between the two modalities enrich the self-supervised task. Their experimental results show that this way of self-supervised learning further enhances the encoders in two modalities. The differences among the ranking labels from three views indeed help encoders exploit more complementary and comprehensive information.

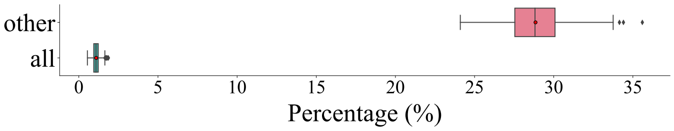

Fig. 4. The proportion of ranking labels $\boldsymbol{y}^{\phi_J,\theta_O}$ and $\boldsymbol{y}^{\phi_J,\theta_A}$ that are not equal to $\boldsymbol{y}^{\phi_J,\theta_S}$ in the modality of joint.

D. Finetuning for Action Recognition

After achieving the rankings, the encoder $\mathcal{E}$ could extract discriminative features for different skeleton sequences. To perform action recognition, we attach a randomly initialized linear action classifier $\mathcal{C}_{\mathrm{AR}}$ on top of the self-supervised pretrained encoder $\mathcal{E}$, with the output dimension matching the number of action categories in the target datasets. This task is trained using the standard cross-entropy loss:

$$\mathcal{L}_{\mathrm{AR}}(\mathrm{X}) = -\boldsymbol{y}_{\mathrm{AR}}^\top \log \mathcal{C}_{\mathrm{AR}}(\mathcal{E}(\mathrm{X})), \tag{22}$$

where $\boldsymbol{y}_{\mathrm{AR}}$ denotes the ground-truth action label. We explore both semi-supervised and fully-supervised settings for action recognition training, where both the encoder $\mathcal{E}$ and classifier $\mathcal{C}_{\mathrm{AR}}$ are trained jointly.

During inference, the softmax score of the output of the action classifier $\mathcal{C}_{\mathrm{AR}}$ is utilized to predict the action label. In the case of multiple modalities, we first compute bone and motion data from the given skeleton sequence (i.e., joint data). Then, the three modalities of data are fed into their respective well-tuned encoders and action classifiers. Finally, the softmax scores of all three modalities are summed to generate a fused score, which is used to predict the final action label.

V. Experimental Results

A. Datasets

Experiments are conducted on four popular skeleton-based action recognition datasets, *i.e.*, NTU RGB+D [18], NTU RGB+D 120 [19], PKU-MMD I, and PKU-MMD II [20].

NTU RGB+D [18]. It is a large-scale skeleton-based action recognition dataset, including 56,578 videos from 40 subjects and 3 cameras, with 60 action categories. Each human body is represented by 25 joints. Owing to the way of data collection, the recognition performance is usually evaluated by two protocols, *i.e.*, 1) Cross-Subject (X-Sub) that splits the data of 40 subjects into training and test set and 2) Cross-View (X-View) that uses data recorded by two of the cameras as the training set, while the left one is kept as the test set.

NTU RGB+D 120 [19]. This dataset is extended from the NTU RGB+D, adding 57,357 more videos and 60 more actions. So, it contains 113,945 videos with 120 action labels in total. In addition to different viewpoints and subjects, this dataset also takes different locations and backgrounds into account. Similarly, it has two performance evaluation protocols, Cross-Subject (X-Sub) and Cross-Setup (X-Setup).

PKU-MMD [20]. It is a large-scale dataset for both action detection and action recognition tasks. The human body is represented by 25 joints. The dataset consists of two parts, *i.e.*, PKU-MMD I containing 51 action categories with a total of about 20000 action instances; and PKU-MMD II containing 41 action categories with a total of about 7000 action instances. The dramatic changes in the view of PKU-MMD II result in significant skeleton noise, so it is more challenging to train and evaluate than PKU-MMD I. Both parts have the same evaluation protocols as NTU RGB+D, *i.e.*, X-Sub and X-View. As the previous works did, we carry the experiments under the X-Sub evaluation protocols on both parts.

B. Implementation Details

We evaluate the performance of the proposed frameworks on both ST-GCN [4] and STTFormer [34] (following SkeletonMAE [9] for fair comparisons) as encoders. For data preprocessing and data augmentations, we follow that of [9], [34]. The code is based on PyTorch and all the experiments can be conducted on a single A100 40GB GPU.

Self-supervised Pretraining. When pretraining the proposed frameworks, we follow XDC [57] to fix the ranking labels of the triplets in $\mathcal{D}$ and train the encoders until the validation loss becomes stable. Then, we re-extract the features from the well-trained encoders to form the new ranking labels in $\mathcal{D}$. The learning rate is set as 0.1 and 0.01 when separately employing ST-GCN and STTFormer as encoders. SGD with momentum of 0.9 and weight decay of 0.0001 is used to optimize our self-supervised learning frameworks. The batch size is 64, but it is worth noting that a batch of skeleton sequences can be combined into $P(64, 3) = 249,984$ triplets for the ranking learning process. When training MV-DeepRank, we set $\lambda_S^\theta = \lambda_O^\theta = \lambda_A^\theta = 1$ in our method by experience.

Finetuning. When employing ST-GCN as the encoder, the finetuning process lasts for 100 epochs with an initial learning rate of 0.1 (multiplied by 0.1 at epoch 80). When employing STTFormer as the encoder, the finetuning process lasts for 90 epochs with an initial learning rate of 0.1 (respectively multiplied by 0.1 at epochs 60 and 80) and 5 linearly warm-up epochs. The skeleton sequences are randomly cropped with a sampled $p \in [0.5, 1]$ (fixed to 0.95 during testing), and then resized to 120 frames. Other hyperparameters follow the papers of corresponding backbones without any modification.

TABLE I
Ablation study of normalization for feature vectors in distance metric function $\mathcal{F}_{\text{DIS}}(\cdot, \cdot)$. The performance (%) is evaluated on the NTU RGB+D X-Sub dataset under the finetuning setting. We **bold** the better results in each column.

Normalization	Joint	Bone	Motion	Ensemble
w/o Normalization	88.6	89.4	88.0	91.8
Min-Max Normalization	89.4	**89.5**	88.0	91.7
L1 Normalization	89.1	89.2	87.7	92.0
L2 Normalization	**89.5**	89.0	**88.5**	**92.3**

C. Ablation Study

We first ablate different types of normalization applied to the feature vectors when calculating the distance between them in Equation 3, and present the results in Table I. Specifically, we use the Euclidean distance as the distance metric functions $\mathcal{F}_{\text{dis}}(\cdot, \cdot)$. We apply ST-GCN as the encoder and finetune it on the NTU RGB+D dataset with the X-Sub protocol. The ensemble results are obtained by averaging the predicted results of the three modalities. The results demonstrate that applying L2 Normalization to the feature vectors yields the best performance. Specifically, the results for the joint, motion, and ensemble modalities show improvements of 0.9%, 0.5%, and 0.5%, respectively, compared to the models without normalization. This indicates that L2 Normalization effectively captures the similarity between samples. By making the feature vectors less sensitive to vector magnitudes, L2 Normalization is particularly well-suited for high-dimensional spaces, such as those used for skeletal representations. We also visualize the joint features extracted by the encoders trained in DeepRank without normalization (left) and with L2 normalization (right), on NTU RGB+D in Fig. 5. Obviously, incorporating L2 Normalization results in more discriminative features. Hence, L2 normalization is adopted as the default for distance calculations in the subsequent experiments.

Table II explores two more sophisticated classifiers C during ranking learning. In the first row (denoted as "MLP"), the classifier consists of two MLP heads, following HiCLR [58]. Specifically, we utilize a shared 2-layer MLP head with ReLU activation to project the concatenation of the anchor feature and the candidate features into a latent space. The two resulting features are then concatenated along the channel dimension and passed through another shared 2-layer MLP head (also with ReLU activation) to predict the ranking label. In the second row (denoted as "ViT"), we follow MAE [42] and build the classifier as a lightweight Vision Transformer (ViT) network [33]. Specifically, the anchor and candidate features are treated as input tokens, each linearly embedded and augmented with sine-cosine positional embeddings. These

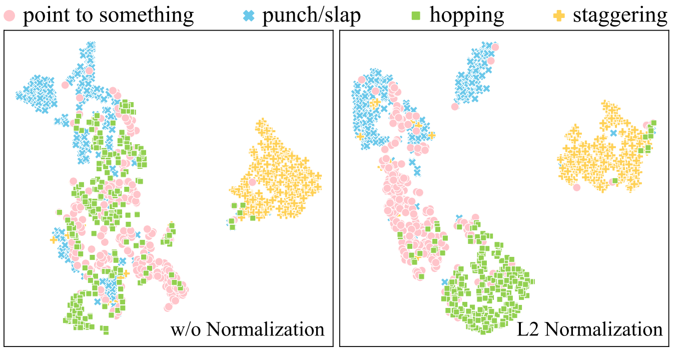

Fig. 5. T-SNE visualizations of joint features from encoders trained in DeepRank without L2 Normalization (left) and with L2 Normalization (right) for distance calculations.

embeddings are then processed by a standard Transformer encoder consisting of two identical blocks. Each block has an embedding dimension of 256, two attention heads in the multi-head self-attention module, and a hidden dimension of 128 in the feed-forward network. The output token representations are averaged and passed through a fully connected layer to predict the ranking label. Results show that our simple FC classifier introduced in Section III-B achieves the best performance. We attribute this to the fact that our task primarily aims to assist the encoder in learning discriminative representations of skeleton sequences. Introducing more complex classifiers appears to introduce additional learning complexity, which may distract the encoder from its primary objective and potentially degrade performance.

To separately quantify the improvements brought by the self-supervisions from different views in MV-DeepRank, we apply ST-GCN as the encoder and conduct the semi-supervised learning experiments on the 1% training data of NTU RGB+D with the X-Sub protocol, following the finetuning setting. The average performance over five trials is reported in Table III. We observe that when using self-supervisions from **one single view** (the first block), integrating more modalities of information (the “other” and “all” views) brings improvements for the joint and bone modalities, compared with only exploiting information from the “self” view. When combining self-supervisions from **multiple views** (the second and third block), the performances of all three modalities and the final ensemble are further improved, which means self-supervisions from different views indeed provide complementary information. Therefore, it is advantageous to train encoders in the multi-view setting.

TABLE II
Ablation study of different classifier C. The performance (%) is evaluated on the NTU RGB+D X-Sub dataset under the finetuning setting. We **bold** the best results in each column.

Classifier C	Joint	Bone	Motion	Ensemble
MLP	88.2	**89.2**	88.1	92.0
ViT	88.8	88.5	87.7	92.0
simple FC	**89.5**	89.0	**88.5**	**92.3**

TABLE III
Ablation study on self-supervisions from different views. The performance (%) is evaluated on the NTU RGB+D X-Sub under semi-supervised setting (1% training data). We report the average of five runs as the final performance. The best in each column is **Bolded**, and the second is underlined.

View			Modality			
self	other	all	Joint	Bone	Motion	Ensemble
✓			38.8	32.0	42.8	52.4
	✓		40.0	35.3	38.7	50.8
		✓	40.7	35.3	40.5	52.0
✓	✓		42.3	39.1	43.9	54.3
✓		✓	43.0	36.7	43.6	53.6
✓	✓	✓	**47.7**	**40.0**	**47.2**	**55.6**

Fig. 6 visualizes the average cosine distances of all the skeletal features extracted from different encoders on the NTU RGB+D dataset. Specifically, we calculate the cosine distance between every two extracted skeletal features, group them according to their ground truth action labels, and obtain the average cosine distance, *i.e.*, each entry (i, j) in the matrix represents the average cosine distance between skeletal features in category y_i and y_j. We have the following observations. 1) The first matrix shows that the bone encoder can well distinguish two-person interactions (the last 11 actions) from single-person actions, showing that even a randomly initialized encoder exploits some information and reflects the similarities among the input data. Thus, the initial ranking labels, derived from com paring feature similarities, are reasonably reliable. After pretraining the encoders in DeepRank, the bone encoder pretrained on DeepRank (the third one) pushes further apart different single-person actions (the first 49 actions) compared to the randomly initialized encoder. It demonstrates that DeepRank can explicitly utilize the different distances among actions, further enlarge this information, and thus enhance the discriminative power of the encoder. 2) The similarity distributions of modality-specific encoders vary after DeepRank training, as skeleton sequences from different modalities have different information. For example, the action “walking towards” is extremely different from other actions in the modality of joint (the second one), while the other two modalities (the third and the fourth one) do not display such a huge difference. This discrepancy among modalities provides complementary information, enhancing the robustness of our self-supervision. 3) As shown in the last matrix, the bone encoder pretrained in MV-DeepRank merges the similarity distributions of all three modalities, which makes the diagonal similarity (features in the same categories) more apparent, *i.e.*, skeleton sequences in the same categories are encoded to more similar feature vectors. Thus, it is significant to utilize information from multiple modalities to train the encoders.

D. Comparison with State-of-the-Art Methods

As shown in Table IV and Table V, we conduct **finetuning** evaluations on NTU RGB+D and NTU RGB+D 120, respectively. The rows in the table are grouped into five blocks

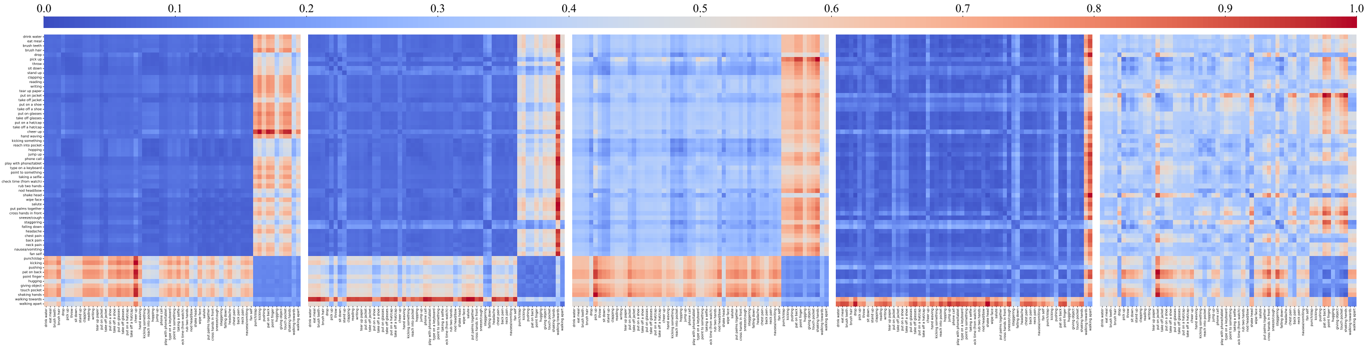

Fig. 6. The average cosine distance matrices of all skeletal features on NTU RGB+D. From left to right, the skeletal features are extracted by: a randomly initialized encoder (in the modality of bone), DeepRank pretrained encoders (separately in the modality of joint, bone, and motion), and MV-DeepRank pretrained encoder (in the modality of bone). Even a randomly initialized encoder reflects data similarities. DeepRank helps the encoder enlarge these similarities, and these similarities vary with modalities. MV-DeepRank merges similarities across multiple modalities. Best viewed in color and zoom in.

TABLE IV
COMPARISON OF THE ACTION RECOGNITION ACCURACY (%) WITH OTHER SELF-SUPERVISED METHODS ON NTU RGB+D. WE **BOLD** THE BEST RESULT AND UNDERLINE THE SECOND-BEST ONE IN EACH BLOCK. 'GEN.' REFERS TO GENERATIVE PRETRAINING-BASED METHODS. 'DIS.' REFERS TO DISCRIMINATIVE PRETRAINING-BASED METHODS.

Method	Gen.	Dis.	Architecture	X-Sub	X-View
MS^{2}L [7] ACMMM'20	✓	✓	BiGRU	78.6	-
VPD [59] ECCV'20	✓	-	SeBiReNet	-	79.7
MCC [8] ICCV'21	✓	✓	2s-AGCN	89.7	**96.3**
Colorization [60] ICCV'21	✓	-	3s-DGCNN	88.0	94.9
Hi-TRS [61] ECCV'22	-	✓	3s-Transformer	90.0	95.7
HiCLR [58] AAAI'23	-	✓	3s-Transformer	**90.4**	95.7
MCC [8] ICCV'21	✓	✓	ST-GCN	83.0	89.7
PSTL [62] AAAI'23	-	✓	ST-GCN	84.5	92.0
CPM [52] ECCV'22	-	✓	ST-GCN	84.8	91.1
DeepCluster [13] ECCV'18	-	✓	ST-GCN	89.4	93.0
DeepRank (ours)	-	✓	ST-GCN	89.5	93.8
MV-DeepRank (ours)	-	✓	ST-GCN	**89.6**	**94.4**
CrosSCLR [51] CVPR'21	-	✓	3s-ST-GCN	86.2	92.5
AimCLR [63] AAAI'22	-	✓	3s-ST-GCN	86.9	92.8
PSTL [62] AAAI'23	-	✓	3s-ST-GCN	87.1	93.9
HiCLR [58] AAAI'23	-	✓	3s-ST-GCN	88.3	93.2
DeepCluster [13] ECCV'18	-	✓	3s-ST-GCN	92.1	95.5
DeepRank (ours)	-	✓	3s-ST-GCN	**92.3**	95.9
MV-DeepRank (ours)	-	✓	3s-ST-GCN	**92.3**	**96.2**
AimCLR [63] AAAI'22	-	✓	STTFormer	83.9	90.4
CrosSCLR [51] CVPR'21	-	✓	STTFormer	84.6	90.5
SkeletonMAE [9] ICMEW'23	✓	-	STTFormer	86.6	92.9
DeepCluster [13] ECCV'18	-	✓	STTFormer	87.9	92.4
DeepRank (ours)	-	✓	STTFormer	89.9	94.9
MV-DeepRank (ours)	-	✓	STTFormer	**90.0**	**95.0**
DeepCluster [13] ECCV'18	-	✓	3s-STTFormer	91.8	95.6
DeepRank (ours)	-	✓	3s-STTFormer	**92.4**	96.2
MV-DeepRank (ours)	-	✓	3s-STTFormer	**92.4**	**96.4**

TABLE V
COMPARISON OF THE ACTION RECOGNITION ACCURACY (%) WITH OTHER SELF-SUPERVISED METHODS ON NTU RGB+D 120. WE **BOLD** THE BEST RESULT AND UNDERLINE THE SECOND-BEST ONE IN EACH BLOCK.

Method	Gen.	Dis.	Architecture	X-Sub	X-Setup
MCC [8] ICCV'21	✓	✓	2s-AGCN	81.3	83.3
Hi-TRS [61] ECCV'22	-	✓	3s-Transformer	85.3	87.4
HiCLR [58] AAAI'23	-	✓	3s-Transformer	**85.6**	**87.5**
MCC [8] ICCV'21	✓	✓	ST-GCN	77.0	77.8
CPM [52] ECCV'22	-	✓	ST-GCN	78.4	78.9
PSTL [62] AAAI'23	-	✓	ST-GCN	78.6	78.9
DeepCluster [13] ECCV'18	-	✓	ST-GCN	83.0	84.0
DeepRank (ours)	-	✓	ST-GCN	83.8	85.0
MV-DeepRank (ours)	-	✓	ST-GCN	**84.0**	**85.1**
CrosSCLR [51] CVPR'21	-	✓	3s-ST-GCN	80.5	80.4
AimCLR [63] AAAI'22	-	✓	3s-ST-GCN	80.1	80.9
PSTL [62] AAAI'23	-	✓	3s-ST-GCN	81.3	82.6
HiCLR [58] AAAI'23	-	✓	3s-ST-GCN	82.1	83.7
DeepCluster [13] ECCV'18	-	✓	3s-ST-GCN	87.3	89.4
DeepRank (ours)	-	✓	3s-ST-GCN	**88.3**	89.7
MV-DeepRank (ours)	-	✓	3s-ST-GCN	88.2	**90.0**
AimCLR [63] AAAI'22	-	✓	STTFormer	74.6	77.2
CrosSCLR [51] CVPR'21	-	✓	STTFormer	75.0	77.9
SkeletonMAE [9] ICMEW'23	✓	-	STTFormer	76.8	79.1
DeepCluster [13] ECCV'18	-	✓	STTFormer	83.2	85.6
DeepRank (ours)	-	✓	STTFormer	85.1	86.3
MV-DeepRank (ours)	-	✓	STTFormer	**85.4**	**86.5**
DeepCluster [13] ECCV'18	-	✓	3s-STTFormer	88.1	90.1
DeepRank (ours)	-	✓	3s-STTFormer	88.7	**90.4**
MV-DeepRank (ours)	-	✓	3s-STTFormer	**88.8**	90.2

according to different backbone networks. The second and fourth blocks summarize the performance of ST-GCN and STTFormer using only the joint modality, while the third and fifth blocks report the ensemble results on these two backbones across all three modalities. Methods employing other backbone networks are listed in the first block. Compared to all the previous methods that employ the same backbone network, ST-GCN, both DeepRank and MV-DeepRank achieve **state-of-the-art** performances. Meanwhile, a consistent improvement is observed when using STTFormer as the encoder owing to its large number of parameters compared to ST-GCN-based frameworks. Besides, our STTFormer-based frameworks surpass SkeletonMAE [9], which also employed STTFormer as the backbone network, by 2.1%~8.6% on all four protocols. Notably, our DeepRank consistently outperforms DeepCluster (with n = 256, which is the optimal number of clusters according to Table VI) across all four evaluation protocols on two datasets. In particular, DeepRank achieves an improvement of 2% compared to DeepCluster on the NTU RGB+D X-Sub protocol with STTFormer as the backbone. This further demonstrates the superior effectiveness of DeepRank.

TABLE VI
Comparison with DeepCluster on NTU RGB+D X-Sub in the finetuning setting. DeepCluster needs intensive tuning on n. Besides, the best n varies with modalities, so the ensemble are hard to achieve the best. In contrast, DeepRank obtains better results efficiently.

Method	Modality			Ensemble
	Joint	Bone	Motion	
DeepCluster (n = 8)	89.6	88.7	87.8	91.5
DeepCluster (n = 16)	89.1	88.9	87.8	90.7
DeepCluster (n = 32)	89.6	88.6	87.9	92.1
DeepCluster (n = 64)	88.6	88.4	87.9	92.0
DeepCluster (n = 128)	88.9	89.0	88.1	90.8
DeepCluster (n = 256)	89.4	89.0	87.8	92.1
DeepCluster (Avg.)	89.2	88.8	87.8	91.5
DeepRank (ours)	89.5	89.0	88.5	**92.3**

TABLE VII
Comparison of semi-supervised (10% training data) performance (%) on the NTU RGB+D dataset. * indicates the re-implemented results in [9].

Method	Backbone	X-Sub	X-View
MCC [8]	ST-GCN	55.6	59.9
CPM [52]		73.0	77.1
3s-AimCLR [63]		78.2	81.6
3s-HiCLR [58]		79.6	84.0
3s-DeepRank (ours)		80.1	83.2
3s-MV-DeepRank (ours)		**80.7**	**84.5**
CrosSCLR* [51]	STTFormer	71.0	75.1
AimCLR* [63]		70.2	76.2
SkeletonMAE [9]		73.0	76.9
DeepRank (ours)		74.6	78.5
MV-DeepRank (ours)		74.9	78.9
3s-DeepRank (ours)		79.4	83.5
3s-MV-DeepRank (ours)		**79.8**	**83.8**

We also group and compare methods in Tables IV and V by their self-supervised learning strategy. Specifically, they are categorized into two types: (1) Generative pretraining-based methods (*e.g.*, SkeletonMAE [9], Colorization [60]) often mask parts of the input and train the model to reconstruct the missing information (**generating content**). While these methods capture the underlying spatio-temporal dynamics of sequences, they struggle to effectively separate different actions compared to discriminative methods. (2) Discriminative pretraining-based methods often **generate pseudo-labels** and train models to classify samples. Among them, contrastive learning approaches [51], [58], [61], [63] define positive and negative pairs to facilitate action discrimination. However, they are sensitive to false negatives and prone to overfitting on spurious correlations in the absence of strong data augmentations. DeepCluster, in particular, relies on clustering for pseudo-label generation but suffers from issues mentioned in Section I, resulting in inferior pseudo-labels and less robust discriminative representations. In contrast, our proposed method achieves better performance and generates more accurate and balanced pseudo-labels.

Since the original work of DeepCluster was trained on RGB images, we re-implement it on the skeleton sequences. For comprehensive comparisons, we train it by setting various numbers of clusters n during the clustering process. All the hyperparameters and the training process are set the same as ours. From the results in Table VI, DeepRank exceeds DeepCluster by 0.3%, 0.2%, 0.7%, and 0.8% in the modalities of joint, bone, motion, and ensemble on average, proving the effectiveness of DeepRank. One can find that DeepCluster is sensitive to n and needs careful and intensive tuning. Notably, large n does not improve performance on this dataset, and increases the time required for the clustering process. Besides, the optimal cluster number n varies with modalities, so the ensemble results of DeepCluster are hard to achieve the best. That explains why studies like [64], [65] perform tedious experiments to determine the minimum while insensitive number of clusters on their datasets, which in contrast, underscores the advantages of ours. Instead, DeepRank does not need such a parameter tuning and is thus much more **efficient**.

Besides, we visualize the features extracted by the encoder well-trained in DeepCluster with $n = 8$ and DeepRank on NTU RGB+D in the modality of bone (Fig. 7). We randomly choose one of the clusters grouped by DeepCluster and get the corresponding ground-truth labels for each sample inside it. Statistical analysis shows that the two categories with the highest number of samples in this cluster are "falling down" and "stand up". Then, we visualize the samples with these two ground-truth labels in this cluster using t-SNE. Obviously, since these samples are clustered into the same group and assigned the same pseudo label in DeepCluster, their features are pulled together during the pretraining stage and thus become indistinguishable. Meanwhile, the features learned by our framework are still distinguishable across different categories and more suitable for downstream tasks.

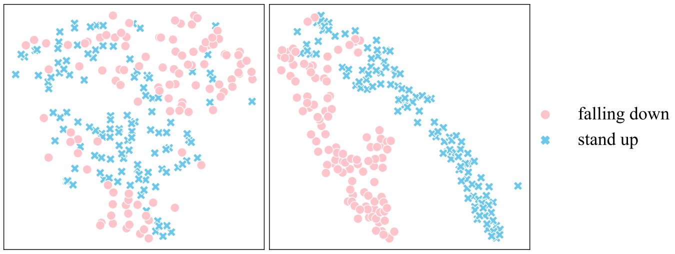

Fig. 7. **t-SNE visualizations** of learned features on NTU RGB+D in the modality of bone from the encoders trained in DeepCluster (**left**) and DeepRank (**right**). For better visualization, we only visualize the samples (**left**) from the two categories with the highest number of samples in a randomly chosen cluster from DeepCluster (n = 8). The right figure plots the corresponding features learned by DeepRank.

The **semi-supervised** results that employ the same backbone networks as ours are presented in Table VII. Specifically, we finetune our self-supervised pretrained model with only 10% training data. When deploying ST-GCN as the encoder, the proposed MV-DeepRank significantly outperforms the state-of-the-art models while DeepRank performs nearly the same as the second. Meanwhile, a large gain is achieved when deploying STTFormer as the encoder. Compared with the previous methods [9], [51], [63], both DeepRank and MV-

TABLE VIII
COMPARISON OF TRANSFER LEARNING PERFORMANCE (%) ON THE PKU-MMD II DATASET.

Method	Pretraining Dataset	Backbone	
		ST-GCN	STTFormer
Supervised	-	48.2	55.5
DeepRank	NTU-60	54.7	58.8
	NTU-120	56.4	58.9
	PKUMMD I	55.0	58.5
MV-DeepRank	NTU-60	56.0	59.8
	NTU-120	55.1	59.1
	PKUMMD I	60.1	60.3

TABLE IX
GENERALIZING DEEPCLUSTER AND DEEPRANK TO THE CIFAR-100 DATASET [67].

Method	CIFAR-100 Top-1 Accuracy (%)
DeepCluster (n = 16)	65.4
DeepCluster (n = 32)	65.3
DeepCluster (n = 64)	64.3
DeepCluster (n = 128)	64.8
DeepCluster (n = 256)	64.8
DeepCluster (n = 512)	64.6
DeepCluster (Avg.)	64.9
DeepRank (ours)	**66.0**

DeepRank largely improve the semi-supervised results, verifying the effective representation of our proposed frameworks.

We also evaluate the **transferability** of the proposed frameworks in Table VIII. Concretely, we first pretrain the encoder on the source datasets, *i.e.*, NTU RGB+D and NTU RGB+D 120, and then finetune it on the target dataset, *i.e.*, PKU-MMD II. Compared to training from scratch, pretraining brings performance improvements ranging from 6.5% to 11.9% for the downstream task on PKU-MMD II when using ST-GCN as the encoder. Moreover, compared to our ST-GCN-based frameworks, STTFormer-based ones exhibit better performance (+0.2%~4.1%) after large-scale pre-training on three datasets. The learned representation by our frameworks is shown transferable and versatile across datasets.

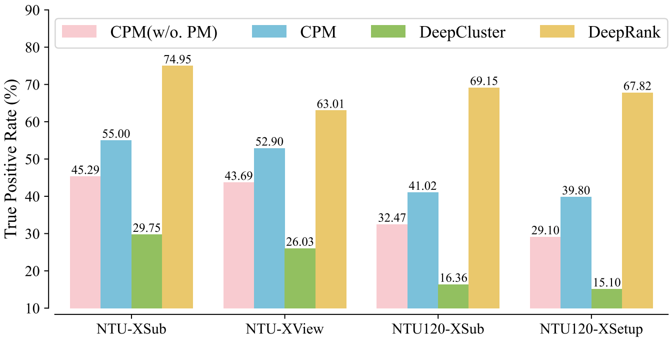

Fig. 8. Precision of positive instances.

Moreover, following CPM [52], a contrastive learning-based method with positive sample mining, we evaluate how well the non-self positives are recognized in DeepRank and DeepCluster. Specifically, for DeepRank, we calculate the precision of true positives in triplets during one epoch. As for DeepCluster (with n=256), the most frequent ground-truth label within each cluster is assigned as its representative label, and we calculate the proportion of correctly clustered samples accordingly. As shown in Fig. 8, DeepCluster exhibits a low true-positive rate, suggesting that the clusters are highly mixed and contain samples from different categories. Consequently, the pseudo-labels generated by DeepCluster are of poor quality, which undermines the effectiveness of self-supervised learning. In contrast, DeepRank identifies much more true positives than others, indicating that our self-supervisions are more accurate, so the learned representations for different classes are discriminative.

Additionally, we demonstrate the generalization capacity of the proposed self-supervised framework, DeepRank, on image data. In particular, we pretrain ResNet-18 [66] encoders using both DeepRank and DeepCluster on the CIFAR-100 dataset [67]. The pretraining and finetuning settings are consistent with those described in Section V-B. The only difference is in the fine-tuning phase, where the initial learning rate is set to 0.1 and decayed by a factor of 10 at 50% and 75% of the total 100 training epochs. The experimental results, presented in Table IX, demonstrate that DeepCluster requires extensive tuning of the number of clusters, n, to achieve competitive performance. In contrast, DeepRank achieves superior results with greater efficiency.

VI. CONCLUSION

In this paper, we propose a novel self-supervised learning framework, DeepRank, to better model the skeleton sequences. Rather than classifying samples into specific pseudo categories, we rank skeleton sequences according to the distances among their features. Meanwhile, since different modalities of skeleton sequences own distinct similarity distributions, we further devise three diverse views of self-supervisions, incorporating information from multiple modalities in three different ways. Then, we explicitly and concurrently supervise the encoders with them, encouraging the encoders to learn the hidden similarity clues more complementarily and comprehensively. We conduct extensive experiments on four various benchmarks under three evaluation settings. Visual and quantitative results show that the proposed frameworks are robust, general, and effective. We hope this simple but effective framework can bring more inspiration for future research.

REFERENCES

[1] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 3288–3297.

[2] T. S. Kim and A. Reiter, "Interpretable 3D human action analysis with temporal convolutional networks," in *CVPRW*, 2017, pp. 1623–1631.

[3] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1110–1118.

[4] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.

[5] H. Tian, X. Ma, X. Li, and Y. Li, "Skeleton-based action recognition with select-assemble-normalize graph convolutional networks," *IEEE Trans. Multimedia*, vol. 25, pp. 8527–8538, 2023.

[6] J. Cheng, D. Shi, C. Li, Y. Li, H. Ni, L. Jin, and X. Zhang, "Skeleton-based gesture recognition with learnable paths and signature features," *IEEE Trans. Multimedia*, vol. 26, pp. 3951–3961, 2024.

[7] L. Lin, S. Song, W. Yang, and J. Liu, "MS^{2}L: Multi-task self-supervised learning for skeleton based action recognition," in *ACM Int. Conf. Multimedia*, 2020, pp. 2490–2498.

[8] Y. Su, G. Lin, and Q. Wu, "Self-supervised 3D skeleton action representation learning with motion consistency and continuity," in *Int. Conf. Comput. Vis.*, October 2021, pp. 13 328–13 338.

[9] W. Wu, Y. Hua, C. Zheng, S. Wu, C. Chen, and A. Lu, "SkeletonMAE: Spatial-temporal masked autoencoders for self-supervised skeleton action recognition," in *Int. Conf. Multimedia and Expo Worksh.* IEEE, 2023, pp. 224–229.

[10] M. Wang, X. Li, S. Chen, X. Zhang, L. Ma, and Y. Zhang, "Learning representations by contrastive spatio-temporal clustering for skeleton-based action recognition," *IEEE Trans. Multimedia*, vol. 26, pp. 3207–3220, 2024.

[11] X. Gao, Y. Yang, Y. Zhang, M. Li, J.-G. Yu, and S. Du, "Efficient spatio-temporal contrastive learning for skeleton-based 3D action recognition," *IEEE Trans. Multimedia*, vol. 25, pp. 405–417, 2023.

[12] S. Xu, H. Rao, X. Hu, J. Cheng, and B. Hu, "Prototypical contrast and reverse prediction: Unsupervised skeleton based action recognition," *IEEE Trans. Multimedia*, vol. 25, pp. 624–634, 2023.

[13] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[14] S. Gepshtein, Y. Wang, F. He, D. Diep, and T. D. Albright, "A perceptual scaling approach to eyewitness identification," *Nat. Commun.*, vol. 11, no. 1, pp. 1–10, 2020.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2015.

[16] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[17] Q. Zeng, C. Liu, M. Liu, and Q. Chen, "Contrastive 3D human skeleton action representation learning via crossmoco with spatiotemporal occlusion mask data augmentation," *IEEE Trans. Multimedia*, vol. 25, pp. 1564–1574, 2023.

[18] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 1010–1019.

[19] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, 2019.

[20] J. Liu, S. Song, C. Liu, Y. Li, and Y. Hu, "A benchmark dataset and comparison study for multi-modal human action analytics," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 16, no. 2, pp. 1–24, 2020.

[21] W. Xin, R. Liu, Y. Liu, Y. Chen, W. Yu, and Q. Miao, "Transformer for skeleton-based action recognition: A review of recent advances," *Neurocomput.*, 2023.

[22] Y. Goutsu, W. Takano, and Y. Nakamura, "Motion recognition employing multiple kernel learning of fisher vectors using local skeleton features," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2015, pp. 79–86.

[23] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1290–1297.

[24] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5378–5387.

[25] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human action recognition from various data modalities: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

[26] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *Int. Conf. Multimedia and Expo Worksh.*, 2017, pp. 597–600.

[27] M. Liu and J. Yuan, "Recognizing human actions as the evolution of pose estimation maps," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1159–1168.

[28] W. Ng, M. Zhang, and T. Wang, "Multi-localized sensitive autoencoder-attention-lstm for skeleton-based action recognition," *IEEE Trans. Multimedia*, vol. 24, pp. 1678–1690, 2022.

[29] J. Liu, X. Wang, C. Wang, Y. Gao, and M. Liu, "Temporal decoupling graph convolutional network for skeleton-based gesture recognition," *IEEE Trans. Multimedia*, vol. 26, pp. 811–823, 2024.

[30] X. Wang, W. Zhang, C. Wang, Y. Gao, and M. Liu, "Dynamic dense graph convolutional network for skeleton-based human motion prediction," *IEEE Trans. Image Process.*, vol. 33, pp. 1–15, 2024.

[31] W. Myung, N. Su, J.-H. Xue, and G. Wang, "DeGCN: Deformable Graph Convolutional Networks for Skeleton-Based Action Recognition," *IEEE Trans. Image Process.*, vol. 33, pp. 2477–2490, 2024.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. Neural Inform. Process. Syst.*, vol. 30, 2017.

[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[34] H. Qiu, B. Hou, B. Ren, and X. Zhang, "Spatio-temporal tuples transformer for skeleton-based action recognition," *arXiv preprint arXiv:2201.02849*, 2022.

[35] W. Xin, Q. Miao, Y. Liu, R. Liu, C.-M. Pun, and C. Shi, "Skeleton Mixformer: Multivariate topology representation for skeleton-based action recognition," in *ACM Int. Conf. Multimedia*, 2023, pp. 2211–2220.

[36] Y. Zhang, B. Wu, W. Li, L. Duan, and C. Gan, "STST: Spatial-temporal specialized transformer for skeleton-based action recognition," in *ACM Int. Conf. Multimedia*, 2021, pp. 3229–3237.

[37] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification," in *Eur. Conf. Comput. Vis.*, 2016, pp. 527–544.

[38] L. Meng, Q. Zhang, R. Yang, and Y. Huang, "Unsupervised deep triplet hashing for image retrieval," *IEEE Sign. Process. Letters*, vol. 31, pp. 1489–1493, 2024.

[39] F. Zhang and H. Che, "Separable consistency and diversity feature learning for multi-view clustering," *IEEE Sign. Process. Letters*, vol. 31, pp. 1595–1599, 2024.

[40] C. Tao, X. Zhu, W. Su, G. Huang, B. Li, J. Zhou, Y. Qiao, X. Wang, and J. Dai, "Siamese image modeling for self-supervised vision representation learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 2132–2141.

[41] H. Wang, J. Fan, Y. Wang, K. Song, T. Wang, and Z. Zhang, "DropPos: pre-training vision transformers by reconstructing dropped positions," in *Adv. Neural Inform. Process. Syst.*, ser. Adv. Neural Inform. Process. Syst. Red Hook, NY, USA: Curran Associates Inc., 2024.

[42] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[43] Z. Miao, H. Luo, D. Liu, and J. Zhang, "Improving visual representations of masked autoencoders with artifacts suppression," *IEEE Sign. Process. Letters*, vol. 31, pp. 2615–2619, 2024.

[44] J. Zhu, H. Ma, J. Chen, and J. Yuan, "High-quality and diverse few-shot image generation via masked discrimination," *IEEE Trans. Image Process.*, vol. 33, pp. 2950–2965, 2024.

[45] Q. Zhou, Q. Wang, Q. Gao, M. Yang, and X. Gao, "Unsupervised discriminative feature selection via contrastive graph learning," *IEEE Trans. Image Process.*, vol. 33, pp. 972–986, 2024.

[46] Z. Liu, X. Wu, S. Wang, and Y. Shang, "Violent video recognition based on global-local visual and audio contrastive learning," *IEEE Sign. Process. Letters*, vol. 31, pp. 476–480, 2024.

[47] J. Zhu, G. Luo, B. Duan, and Y. Zhu, "Class incremental learning with deep contrastive learning and attention distillation," *IEEE Sign. Process. Letters*, vol. 31, pp. 1224–1228, 2024.

[48] C. Pang, X. Lu, and L. Lyu, "Skeleton-based action recognition through contrasting two-stream spatial-temporal networks," *IEEE Trans. Multimedia*, vol. 25, pp. 8699–8711, 2023.

[49] S. Guan, X. Yu, W. Huang, G. Fang, and H. Lu, "DMMG: Dual min-max games for self-supervised skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 33, pp. 395–407, 2024.

[50] L. Lin, J. Zhang, and J. Liu, "Mutual information driven equivariant contrastive learning for 3D action representation learning," *IEEE Trans. Image Process.*, vol. 33, pp. 1883–1897, 2024.

[51] L. Li, M. Wang, B. Ni, H. Wang, J. Yang, and W. Zhang, "3D human action representation learning via cross-view consistency pursuit," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 4741–4750.

[52] H. Zhang, Y. Hou, W. Zhang, and W. Li, "Contrastive positive mining for unsupervised 3D action representation learning," in *Eur. Conf. Comput. Vis.* Springer, 2022, pp. 36–51.

[53] Z. Fu, Y. Li, Z. Mao, Q. Wang, and Y. Zhang, "Deep metric learning with self-supervised ranking," in *AAAI Conf. Artif. Intell.*, vol. 35, no. 2, 2021, pp. 1370–1378.

[54] J. Pfister, K. Kobs, and A. Hotho, "Self-supervised multi-task pretraining improves image aesthetic assessment," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 816–825.

[55] H. Duan, N. Zhao, K. Chen, and D. Lin, "TransRank: Self-supervised video representation learning via ranking-based transformation recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 3000–3010.

[56] A. N. Carr, Q. Berthet, M. Blondel, O. Teboul, and N. Zeghidour, "Self-supervised learning of audio representations from permutations with differentiable ranking," *IEEE Sign. Process. Letters*, vol. 28, pp. 708–712, 2021.

[57] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran, "Self-supervised learning by cross-modal audio-video clustering," *Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 9758–9770, 2020.

[58] J. Zhang, L. Lin, and J. Liu, "Hierarchical consistent contrastive learning for skeleton-based action recognition with growing augmentations," in *AAAI Conf. Artif. Intell.*, vol. 37, no. 3, 2023, pp. 3427–3435.

[59] Q. Nie, Z. Liu, and Y. Liu, "Unsupervised 3D human pose representation with viewpoint and pose disentanglement," in *Eur. Conf. Comput. Vis.*, 2020, pp. 102–118.

[60] S. Yang, J. Liu, S. Lu, M. H. Er, and A. C. Kot, "Skeleton cloud colorization for unsupervised 3D action representation learning," in *Int. Conf. Comput. Vis.*, 2021, pp. 13 423–13 433.

[61] Y. Chen, L. Zhao, J. Yuan, Y. Tian, Z. Xia, S. Geng, L. Han, and D. N. Metaxas, "Hierarchically self-supervised transformer for human skeleton representation learning," in *Eur. Conf. Comput. Vis.* Springer, 2022, pp. 185–202.

[62] Y. Zhou, H. Duan, A. Rao, B. Su, and J. Wang, "Self-supervised action representation learning from partial spatio-temporal skeleton sequences," in *AAAI Conf. Artif. Intell.*, vol. 37, no. 3, 2023, pp. 3825–3833.

[63] G. Tianyu, L. Hong, C. Zhan, L. Mengyuan, W. Tao, and D. Runwei, "Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition," in *AAAI Conf. Artif. Intell.*, 2022.

[64] B. Pang, Y. Zhang, Y. Li, J. Cai, and C. Lu, "Unsupervised visual representation learning by synchronous momentum grouping," in *Eur. Conf. Comput. Vis.* Springer, 2022, pp. 265–282.

[65] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 9912–9924, 2020.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[67] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.