3D Human Action Representation Learning via Cross-View Consistency Pursuit

Linguo Li[1,2*] Minsi Wang[1,2*] Bingbing Ni[1,2**] Hang Wang[1,2] Jiancheng Yang[1,2] Wenjun Zhang[1]
[1]Shanghai Jiao Tong University, Shanghai 200240, China
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
mswang1994@gmail.com, {LLG440982, nibingbing, zhangwenjun}@sjtu.edu.cn

Abstract

In this work, we propose a Cross-view Contrastive Learning framework for unsupervised 3D skeleton-based action Representation (CrosSCLR), by leveraging multi-view complementary supervision signal. CrosSCLR consists of both single-view contrastive learning (Skeleton-CLR) and cross-view consistent knowledge mining (CVC-KM) modules, integrated in a collaborative learning manner. It is noted that CVC-KM works in such a way that high-confidence positive/negative samples and their distributions are exchanged among views according to their embedding similarity, ensuring cross-view consistency in terms of contrastive context, i.e., similar distributions. Extensive experiments show that CrosSCLR achieves remarkable action recognition results on NTU-60 and NTU-120 datasets under unsupervised settings, with observed higher-quality action representations. Our code is available at `https://github.com/LinguoLi/CrosSCLR`.

1. Introduction

Human action recognition is an important but challenging task in computer vision research. Due to the lightweight and robust estimation algorithms [3, 56], 3D skeleton has become a popular feature representation to study human action dynamics. Many 3D action recognition works [6, 60, 17, 25, 44, 28, 16] use a fully-supervised manner and require massive labeled 3D skeleton data. However, annotating data is expensive and time-consuming, which prompts people to explore unsupervised methods [65, 26, 39, 47] on skeleton data. Some unsupervised methods exploit structure completeness within each sample based on pretext tasks, including reconstruction [9, 65], auto-regression [20, 47] and jigsaw puzzles [35, 54], but it is unsure that the designed pretext tasks generalize well for downstream tasks. Other unsupervised methods are based
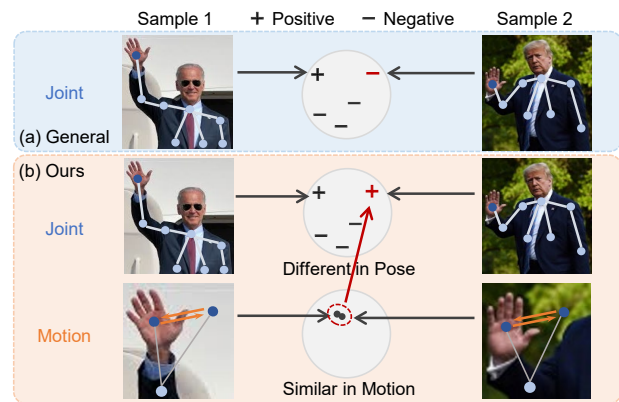

Figure 1. Hand waving in joint and motion form. Two samples are from the same action class. (a) usual contrastive learning methods regard them as negative pairs. (b) in a multi-view situation, considering their similar motion patterns, they can be positive pairs. This motivates us to introduce cross-view contrastive learning for skeleton representation.

on contrastive learning [55, 4, 11, 18], aiming to leverage the instance discrimination of samples in latent space.

Although the above approaches improve the skeleton representation capability to some extent, it is believed that the power of unsupervised methods is by far from fully explored. On the one hand, traditional contrastive learning uses only one positive pair generated by data augmentation and even similar samples are regarded as negative samples. Despite the high similarity, the negative samples are forced away in embedding space, which is unreasonable for clustering. On the other hand, current unsupervised methods [1, 65, 20, 26, 47] have not yet explored the rich intra-supervision information provided by different skeleton modalities. Considering that it is easy to obtain skeleton data in multiple "views", e.g., joint, motion and bone, complementary information preserved in different views can assist the operation to mine positive pairs from similar negative samples. As shown in Figure 1, the same hand waving actions are different in pose (joint), but similar in motion. Usual contrastive learning methods regard them as nega-

---

* Equal Contribution
** Corresponding Author: Bingbing Ni

tive pairs, keeping them away in embedding space. If such complementary information, i.e., different in joint but similar in motion, could be fully utilized and explored, the size of hidden positive pairs in joint can be boosted, enhancing training fidelity. Thus, the cross-view contrastive learning strategy takes advantage of multi-view knowledge, resulting in better-extracted skeleton features.

To this end, we propose a Cross-view Contrastive Learning framework for Skeleton-based action Representation (CrosSCLR), which exploits multi-view information for mining positive samples and pursuing cross-view consistency in unsupervised contrastive learning, enabling the model to extract more comprehensive cross-view features. First, parallel Contrastive Learning is evoked for each single-view Skeleton action Representation (SkeletonCLR), yielding multiple single-view embedding features. Second, inspired by the fact that the distance of samples in embedding space reflects the similarity of the samples in the original space, we refer to the extreme similarity of samples in one view to guide the learning process in another view, as shown in Figure 1. More specifically, Cross-View Consistent Knowledge Mining (CVC-KM) module is developed to exam the similarity of samples, and select the most similar pairs as positive ones to boost the positive set in complementary views, i.e., *embedding distance/similarity (confidence score) serves as the weight of corresponding mined sample in the contrastive loss*. In other words, CVC-KM conveys the most prominent knowledge from one view to others, introduces complementary pseudo-supervised constraint and promotes information sharing among views. The entire framework excavates positive pairs across views according to the distance between samples in the embedding space to promote knowledge exchange among views, so that the extracted skeleton features will contain multi-view knowledge and are more competitive for various downstream tasks. Extensive results on NTU-RGB+D [40, 27] datasets demonstrate that our method indeed boosts the 3D action representation learning benefiting from cross-view consistency. We summarize our contributions as follows:

- We propose CrosSCLR, a cross-view contrastive learning framework for skeleton-based action representation.

- We develop Contrastive Learning for Skeleton-based action Representation (SkeletonCLR) to learn the single-view representations of skeleton data.

- We use parallel SkeletonCLR models and CVC-KM to excavate useful samples across views, enabling the model to capture more comprehensive representation unsupervisedly.

- We evaluate our model on 3D skeleton datasets, e.g., NTU-RGB+D 60/120, and achieve remarkable results under unsupervised settings.

2. Related Work

Self-Supervised Representation Learning. Self-supervised learning is to learn feature representations from numerous unlabeled data, which usually generates supervision by pretext tasks, e.g., jigsaw puzzles [35, 36, 54], colorization [63], predicting rotation [8, 59]. For sequence data, supervision can be generated by frame orders [32, 7, 21], space-time cubic puzzles [19] and prediction [51, 26], but these methods highly rely on the quality of pretext tasks. Recently, contrastive methods [37, 55, 48, 18] based on instance discrimination have been proposed for representation learning. MoCo [11] introduces a memory bank to store the embeddings of negative samples, and SimCLR [4] uses a much larger mini-batch size to compute the embeddings in real time, but they can not capture the cross-view knowledge for 3D action representation. Concurrent work CoCLR [10] leverages multi-modal information for video representation via co-training, which doesn't consider the contrastive context. Our CrosSCLR simultaneously trains models in all views by encouraging cross-view consistency, leading to more representative embeddings.

Skeleton-based Action Recognition. To tackle skeleton-based action recognition tasks, early approaches are generally based on hand-craft features [33, 52, 49, 50]. Recent methods pay more attention to deep neural networks. For the sequence structure of skeleton data, many RNN-based methods [6, 40, 60, 45, 61] were carried out to effectively utilize the temporal feature. Since RNN suffers from gradient vanishing [14], CNN-based models [17, 22] attract researchers' attention, but they need to convert skeleton data to another form. Further, ST-GCN [57] was proposed to better model the graph structure of skeleton data. Then the attention mechanism [24, 42, 44, 64, 62] and multi-stream structure [25, 41, 42, 53] are applied to adaptively capture multi-stream features based on GCNs. We adopt the widely-used ST-GCN as the backbone to extract the skeleton features.

Unsupervised Skeleton Representation. Many unsupervised methods [46, 29, 31, 23] were proposed to capture action representations in videos. For skeleton data, previous works [58, 1] have achieved some progress in unsupervised representation learning without deep neural networks. Recent deep learning methods [9, 65, 20, 47] are based on the structure of encoder-decoder or generative adversarial network (GAN). LongT GAN [65] proposed an auto-encoder-based GAN for sequential reconstruction and evaluated it on the action recognition tasks. P&C [47] uses a weak decoder in the encoder-decoder model, forcing the encoder to learn more discriminative features. MS^{2}L [26] proposed a multi-task learning scheme for action representation learning. However, these methods highly depend on

reconstruction or prediction, and they do not exploit the natural multi-view knowledge of skeleton data. Thus, we introduce CrosSCLR for unsupervised 3D action representation.

3. CrosSCLR

Although 3D skeleton has shown its importance in action recognition, unsupervised skeleton representation has not been well exploited recently. Since the easily-obtained "multi-view" skeleton information plays a significant role in action recognition, we expect to exploit them to mine positive samples and pursue cross-view consistency in unsupervised contrastive learning, thus giving rise to a Cross-view Contrastive Learning (CrosSCLR) framework for Skeleton-based action Representation.

As shown in Figure 3, CrosSCLR contains two key modules: 1) SkeletonCLR (Section 3.1): a contrastive learning framework to unsupervisedly learn single-view representations, and 2) CVC-KM (Section 3.2): it conveys the most prominent knowledge from one view to others, introduces complementary pseudo-supervised constraint and promotes information sharing among views. Finally, the more discriminating representations can be obtained by cooperatively training (Section 3.2).

3.1. Single-View 3D Action Representation

Contrastive learning has been widely-used due to its instance discrimination capability, especially for images [4, 11] and videos [10]. Inspired by this, we develop *Skeleton-CLR* to learn single-view 3D action representations, based on the recent advanced practice, MoCo [11].

SkeletonCLR. It is a memory-augmented contrastive learning method for skeleton representation, which considers one sample's different augments as its positive samples and other samples as negative samples. In each training step, the batch embeddings are stored in first-in-first-out memory to get rid of redundant computation, serving as negative samples for the next steps. The positive samples are embedded close to each other while the embeddings of negative samples are pushed away. As shown in Figure 2, SkeletonCLR consists of the following major components:

- A data augmentation module $\mathcal{T}$ that randomly transforms the given skeleton sequence into different augments x, $\hat{x}$ that are considered as positive pairs. For skeleton data, we adopt *Shear* and *Crop* as the augmentation strategy (see Section 3.3 and Appendix).

- Two encoders f and $\hat{f}$ that embed x and $\hat{x}$ into hidden space: $h = f(x; \theta)$ and $\hat{h} = \hat{f}(\hat{x}; \hat{\theta})$, where $h, \hat{h} \in \mathbb{R}^{c_h}$. $\hat{f}$ is the momentum updated version of f: $\hat{\theta} \leftarrow \alpha\hat{\theta} + (1 - \alpha)\theta$, where α is a momentum coefficient. SkeletonCLR uses ST-GCN [57] as the backbone (Details are in Section 3.3).
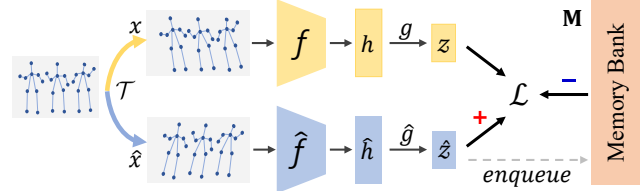

Figure 2. Architecture of single-view SkeletonCLR, which is a memory augmented contrastive learning framework.

- A simple projector g and its momentum updated version $\hat{g}$ that project the hidden vector to a lower dimension space: $z = g(h)$, $\hat{z} = \hat{g}(\hat{h})$, where $z, \hat{z} \in \mathbb{R}^{c_z}$. The projector is a fully-connected (FC) layer with ReLU.

- A memory bank $\mathbf{M} = \{m_i\}_{i=1}^M$ that stores negative samples to avoid redundant computation of the embeddings. It is a first-in-first-out queue updated per iteration by $\hat{z}$. After each inference step, $\hat{z}$ will enqueue while the earliest embedding in $\mathbf{M}$ will dequeue. During contrastive training, $\mathbf{M}$ provides numerous negative embeddings while the new calculated $\hat{z}$ is the positive embedding.

- An InfoNCE [37] loss for instance discrimination:

$$\mathcal{L} = -\log \frac{\exp(z \cdot \hat{z}/\tau)}{\exp(z \cdot \hat{z}/\tau) + \sum_{i=1}^M \exp(z \cdot m_i/\tau)} \quad (1)$$

where $m_i \in \mathbf{M}$, τ is the temperature hyper-parameter [12], and dot product $z \cdot \hat{z}$ is to compute their similarity where $z, \hat{z}$ are normalized.

Constrained by contrastive loss $\mathcal{L}$, the model is unsupervisedly trained to discriminate each sample in the training set. At last, we can obtain a strong encoder f that is beneficial to extract single-view distinguishing representations.

Limitations of Single-View Contrastive Learning. The above SkeletonCLR still suffers the following limitations:

1) *Embedding distribution can provide more reliable information.* We expect samples from the same category are embedded closely. However, instance discrimination in SkeletonCLR uses only one positive pair and even similar samples are regarded as negative samples. It is unreasonable that the negative samples are forced away in embedding space despite their high embedding similarity. In other words, one positive pair cannot fully describe the relationships of samples, and a more reliable embedding distribution is needed, i,e., positive/negative setting plus embedding similarity. We aim to mine more representative knowledge to facilitate contrastive learning, which is also the knowledge we want to exchange across views. Thus, we introduce the *contrastive context* in Section 3.2.

2) *Multi-view data can benefit representation learning.* SkeletonCLR only relies on single-view data. As shown in Figure 1, since we don't have any annotations, different samples of the same class are inevitably embedded

into distinct places far from each other, i.e., they distribute sparsely/irregularly, bringing much difficulty for linear classification. Considering the readily generated multi-view data of 3D skeleton (see Section 3.3), if such complementary information in Figure 1, i.e., different in joint but similar in motion, could be fully utilized and explored, the size of hidden positive pairs in joint can be boosted, enhancing training fidelity. To this end, we inject this consideration into unsupervised contrastive learning framework.

3.2. Cross-View Consistent Knowledge Mining

Motivated by the situation in Figure 1 that complementary knowledge is preserved in multiple views, we propose the Cross-View Consistent Knowledge Mining (CKC-KM), leveraging the high similarity of samples in one view to guide the learning process in another view. It excavates positive pairs across views according to the embedding similarity to promote knowledge exchange among views, then the size of hidden positive pairs in each view can be boosted and the extracted skeleton features will contain multi-view knowledge, resulting in a more regular embedding space.

In this section, we first clarify *contrastive context* as the consistent knowledge across views, and then show how to mine high-confidence knowledge, and finally inject its cross-view consistency into single-view SkeletonCLR to further benefit the cross-view unsupervised representation.

Contrastive Context as Consistent Knowledge. As discussed above, the knowledge we want to exchange across views is one sample's *contrastive context*, which describes this sample's relationships with others (distribution) in embedding space under the settings of contrastive learning. Notice that SkeletonCLR uses a memory bank to store necessary embeddings. Given one sample's embedding z and corresponding memory bank $\mathbf{M}$, its contrastive context is a similarity set $\mathcal{S}$ among z and $\mathbf{M}$ conditioned on specific knowledge miner Γ that generates index set N_+ of positive samples,

$$\mathcal{S} = \{s_i\}_{i \in N} = \{z \cdot m_i\}_{i \in N} \tag{2}$$

$$(\mathcal{S}_+, N_+) = \Gamma(\mathcal{S}) \tag{3}$$

where $\mathcal{S}_+ = \{s_i\}_{i \in N_+}$ and dot product "·" is to compute the similarity s_i among embeddings z and m_i. N is the index set of embeddings in memory bank and N_+ is the index set of positive samples selected by knowledge miner Γ. Thus contrastive context $\mathcal{C}(\mathcal{S}|N_+)$ consists of following two aspects:

- *Embedding Context* $\mathcal{S}$: it is the relationship between one sample and others in embedding space, i.e., distribution;
- *Contrastive Setting* N_+: it is the positive setting mined by Γ according to the embedding similarity $\mathcal{S}$;

thus $\mathcal{C}(\mathcal{S}|N_+) = \{\mathcal{S}_+, \mathcal{S}_-\}$ has *positive* context $\mathcal{S}_+$ and *negative* context $\mathcal{S}_-$, where $\mathcal{S} = \mathcal{S}_+ \cup \mathcal{S}_-$. The contrastive context contains not only the information of the most similar samples but the detailed relationships of samples (distribution).

In Equation (1), the embedding z has positive context $\mathcal{S}_+ = \{z \cdot \hat{z}\}$ which does not consider any of neighbors in embedding space except for the augments. Despite the high similarity, the negative samples are forced away in embedding space, and then samples belonging to the same category are difficultly embedded into the same cluster, which is not efficient to build a "regular" embedding space for downstream classification tasks.

High-confidence Knowledge Mining. To solve the above issue, we develop the high-confidence Knowledge Mining mechanism (KM), which selects the most similar pairs as positive ones to boost the positive sets. It shares similar high-level spirit with neighborhood embedding [13] but performs differently in an unsupervised contrastive manner.

Specifically, it is based on the following observation in Figure 4 that after single-view contrastive learning, two embeddings most likely belong to the same category if they are embedded closely enough; on the contrary, two embeddings hardly belong to the same class if they locate extremely far from each other in embedding space. Therefore, we can facilitate contrastive learning by setting the most similar embeddings as positive to make it more clustered:

$$\Gamma(\mathcal{S}) = \mathrm{Topk}(\mathcal{S}) \tag{4}$$

$$\mathcal{L}_{\mathrm{KM}} = -\log \frac{\exp(z \cdot \hat{z}/\tau) + \sum_{i \in N_+} \exp(z \cdot m_i/\tau)}{\exp(z \cdot \hat{z}/\tau) + \sum_{i \in N} \exp(z \cdot m_i/\tau)} \tag{5}$$

where $\Gamma = \mathrm{Topk}$ is the function to select the index of top-K similar embeddings and N_+ is their index set in memory bank. Compared to Equation (1), Equation (5) will leads to a more regular space by pulling close more high-confidence positive samples. Additionally, since we don't have any labels, a larger K may harm the contrastive performance (see Section 4.3).

Cross-View Consistency Learning. Considering easily-obtained multi-view skeleton data, complementary information preserved in different views can assist the operation to mine positive pairs from similar negative samples in Figure 1. Then the size of hidden positive pairs can be boosted by cross-view knowledge communication, resulting in better-extracted skeleton features. To this end, we design the cross-view consistency learning which not only mines the high-confidence positive samples from complementary view but also lets the embedding context be consistent in multiple views. Its two-view case is illustrated in Figure 3 for example.

Specifically, samples x^u and x^v are generated from the same raw data x by the view generation method in Section 3.3, where u and v indicate two types of data views. After single-view contrastive learning, two SkeletonCLR
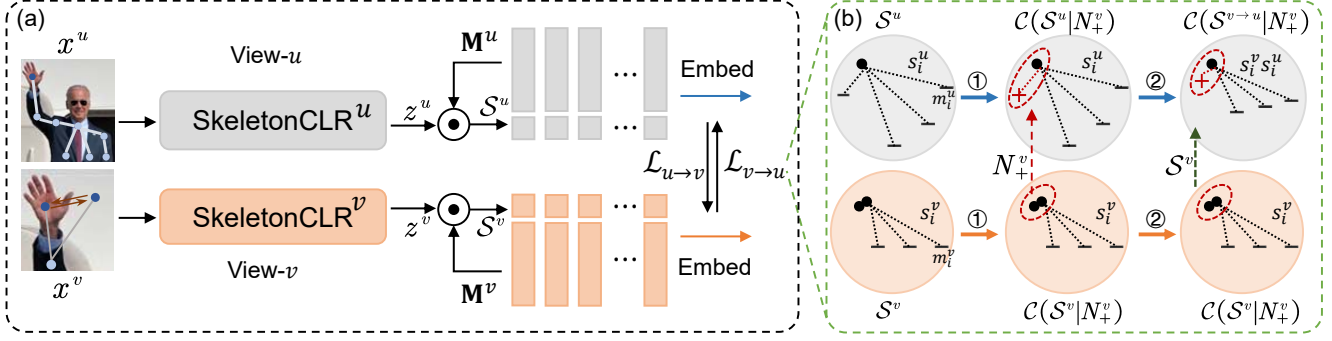
Figure 3. (a) CrosSCLR. Given two samples x^u, x^v generated from the same raw data, e.g., joint and motion, SkeletonCLR models produce single-view embeddings while cross-view consistent knowledge mining (CVC-KM) exchanges multi-view complementary knowledge. (b) how $\mathcal{L}_{v \to u}$ works in embedding space. In step 1, we mine high-confidence knowledge N_+^v from similarities $\mathcal{S}^v$ to boost the positive set of view u, i.e., z^u shares z^v's neighbors; In step 2, we use the similarities $\mathcal{S}^v$ to supervise the embedding distribution in view u. z^u, z^v share similar relationships with others. Thus, two embedding spaces become similar under the constraint of $\mathcal{L}_{\text{cross}}$.

modules obtain the embeddings z^u, z^v and corresponding memory bank $\mathbf{M}^u$, $\mathbf{M}^v$ respectively. We can mine the high-confidence knowledge from two views by $(\mathcal{S}_+^u, N_+^u) = \text{Topk}(\mathcal{S}^u)$ and $(\mathcal{S}_+^v, N_+^v) = \text{Topk}(\mathcal{S}^v)$, where $\mathcal{S}_+^u, \mathcal{S}_+^v$ are the positive context of z^u, z^v respectively.

CrosSCLR aims to learn the consistent embedding distribution in different views by encouraging the similarity of contrastive context, i.e., exchanging high-confidence knowledge across views. In Figure 3 (b), if we want to use the knowledge of view v to guide view u's contrastive learning, it contains two aspects: 1) step 1: we select the most similar pairs (positive) in view v as the positive sets in view u, i.e., $\mathcal{S}^u, N_+^v \to \mathcal{C}(\mathcal{S}^u|N_+^v)$. Thus the sample z^u shares z^v's positive neighbors; 2) step 2: we use the embedding similarity $s_i^v = z^v \cdot m_i^v$ in view v as the weight of corresponding embedding m_i^u in view u to provide the detailed relational information, i.e., $m_i^{v \to u} = s_i^v m_i^u$. Then the similarity is computed by $z^u \cdot m^{v \to u} = s_i^v(z^u \cdot m_i^u) = s_i^u s_i^v$ and z^u has embedding context $\mathcal{S}^{v \to u} = \{z^u \cdot m_i^{v \to u}\}_{i \in N} = \{s_i^u s_i^v\}_{i \in N}$. Finally, the overall loss is conducted as:

$$\mathcal{L}_{v \to u} = -\log \frac{\exp(z^u \cdot \hat{z^u}/\tau) + \sum_{i \in N_+^v} \exp(s_i^u s_i^v)/\tau)}{\exp(z^u \cdot \hat{z^u}/\tau) + \sum_{i \in N} \exp(s_i^u s_i^v)/\tau)} \tag{6}$$

$$\mathcal{L}_{\text{cross}} = \mathcal{L}_{u \to v} + \mathcal{L}_{v \to u} \tag{7}$$

where $\mathcal{L}_{v \to u}$ means we transfer the contrastive context of z^v to that of z^u. Since s_i^u, s_i^v are the embedding context of z_i^u, z_i^v, we call Equation (7) as *cross-view contrastive context learning*, which constrains the similar distribution of two views (see Section 4.3, the results of t-SNE). Compared to Equation (5), Equation (6) considers the cross-view information, cooperatively using one view's high-confidence positive samples and its distribution to instruct the other view's contrastive learning, resulting in more regular space and better extracted skeleton features.

Learning CrosSCLR. For more views, CrosSCLR has fol-

lowing objective:

$$\mathcal{L}_{\text{cross}} = \sum_u^U \sum_v^U \mathcal{L}_{u \to v} \tag{8}$$

where U is the number of views and $v \neq u$.

In the early training process, the model is not stable and strong enough to provide reliable cross-view knowledge without the supervision of labels. As the unreliable information may lead astray, it is not encouraged to enable cross-view communication too early. We perform two-stage training for CrosSCLR: 1) each view of the model is individually trained with Equation (1) without cross-view communication. 2) then the model can supply high-confidence knowledge, so the loss function is replaced with Equation (8), starting cross-view knowledge mining.

3.3. Model Details

View Generation of 3D Skeleton. Generally, 3D human skeleton sequence has T frames with V joints, and each joint has $C = 3$ coordinate feature, which can be noted as $x \in \mathbb{R}^{C \times T \times V}$. Different from videos, the views [41, 42] of skeleton, e.g., joint, motion, bone, motion of bone, can be easily obtained, which is a natural advantage for skeleton-based representation learning. Motion is represented as the temporal displacement between frames: $x_{:,t+1,:} - x_{:,t,:}$, and bone is the distance between two neighboring joints in the same frame: $x_{:,:,v_2} - x_{:,:,v_1}$. For simplicity, we use three views: joint, motion and bone in experiments.

Encoder f. We adopt ST-GCN [57] as encoder, which is suitable for modeling graph-structure skeleton data by exploiting the spatial and temporal relations. After a series of ST-GCN blocks, the output feature $x_{out} \in \mathbb{R}^{c_{out} \times t_{out} \times V}$ is applied by an average pooling operation on spatial and temporal dimensions, obtaining final representation $h \in \mathbb{R}^{c_h}$.

4. Experiments

4.1. Datasets

NTU-RGB+D 60. NTU-RGB+D 60 (NTU-60) dataset [40] is a large-scale dataset of 3D joint coordinate sequences for skeleton-based action recognition, containing $56,578$ skeleton sequences in 60 action categories. Each skeleton graph contains $V = 25$ body joints as nodes, and their 3D coordinates are initial features. There are two protocols [40] recommended. 1) Cross-Subject (xsub): training data and validation data are collected from different subjects. 2) Cross-View (xview): training data and validation data are collected from different camera views.

NTU-RGB+D 120. NTU-RGB+D 120 (NTU-120) dataset [27] is an extended version of NTU-60, containing $113,945$ skeleton sequences in 120 action categories. Two protocols [27] are recommended. 1) Cross-Subject (xsub): training data and validation data are collected from different subjects. 2) Cross-Setup (xset): training data and validation data are collected from different setup IDs.

NTU-RGB+D 61-120. NTU-RGB+D 61-120 (NTU-61-120) dataset is a subset of NTU-120 dataset, containing $57,367$ skeleton sequences in the last 60 action categories in NTU-120. The categories in NTU-61-120 do not intersect with those in NTU-60. This dataset is used as external dataset to evaluate the transfer capability of our method.

4.2. Experimental Settings

All the experiments are conducted on the PyTorch [38] framework. For data pre-processing, we remove the invalid frames of each skeleton sequence and then resize them to the length of 50 frames by linear interpolation. For optimization, we use SGD with momentum (0.9) and weight decay (0.0001). The mini-batch size is set to 128.

Data Augmentation $\mathcal{T}$. For skeleton sequence, we choose Shear [39] and Crop [43] as the augmentation strategy.

Shear is a linear transformation on the spatial dimension. The transformation matrix is defined as:

$$\mathbf{A} = \begin{bmatrix} 1 & a_{12} & a_{13} \\ a_{21} & 1 & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} \qquad (9)$$

where a_{12}, a_{13}, a_{21}, a_{23}, a_{31}, a_{32} are shear factors randomly sampled from $[-\beta, \beta]$. β is the shear amplitude. The sequence x is multiplied by the transformation matrix $\mathbf{A}$ on the channel dimension. Then, the human pose in 3D coordinate is inclined at a random angle.

Crop is an augmentation on the temporal dimension that symmetrically pads some frames to the sequence and then randomly crops it to the original length. The padding length is defined as T/γ, γ is noted as padding ratio. The padding operation uses the reflection of the original boundary.

		NTU-60 (%)	
Method	View	xsub	xview
SkeletonCLR	Joint	68.3	76.4
SkeletonCLR	Motion	53.3	50.8
SkeletonCLR	Bone	69.4	67.4
2s-SkeletonCLR	Joint + Motion	70.5	77.9
3s-SkeletonCLR	Joint + Motion + Bone	75.0	79.8
CrosSCLR	Joint	72.9	79.9
CrosSCLR	Motion	72.7	77.6
CrosSCLR	Bone	75.2	78.8
2s-CrosSCLR	Joint + Motion	74.5	82.1
3s-CrosSCLR	Joint + Motion + Bone	**77.8**	**83.4**

Table 1. Comparisons of SkeletonCLR and CrosSCLR on each view and their ensembles. SkeletonCLR models are trained independently and "+" means the ensemble model.

Unsupervised Pre-training. We generate three views of skeleton sequences, i.e., joint, motion and bone. For the encoder, we adopt ST-GCN [57], but the number of channels in each layer is reduced to $1/4$ of the original setting. For contrastive settings, we follow that in MOCOv2 [5] but reduce the size of memory bank $\mathbf{M}$ to 30k. For data augmentation, We set shear amplitude $\beta = 0.5$ and the padding ratio $\gamma = 6$. The model is trained for 300 epochs with the learning rate 0.1 (multiplied by 0.1 at epoch 250). InfoNCE loss in Equation (1) is used in the first 150 epochs, and then replaced with $\mathcal{L}_{cross}$ in Equation (8) after 150-th epoch. We set $K = 1$ as the default in the knowledge mining mechanism.

Linear Evaluation Protocol. The models are verified by linear evaluation for action recognition task, i.e., attaching the frozen encoder to a linear classifier (a *fully-connected* layer followed by a *softmax* layer), and then training the classifier supervisedly. We train models for 100 epochs with learning rate 3.0 (multiplied by 0.1 at epoch 80).

Finetune Protocol. We append a linear classifier to the *learnable* encoder, and then train the whole model for the action recognition task, to compare it with fully-supervised methods. We train for 100 epochs with learning rate 0.1 (multiplied by 0.1 at epoch 80).

4.3. Ablation Study

All experiments in this section are conducted on NTU-60 dataset and follow the unsupervised pre-training and linear evaluation protocol in Section 4.2.

Effectiveness of CrosSCLR. In Table 1, we separately pretrain SkeletonCLR and jointly pre-train CrosSCLR models on different skeleton views, e.g., joint, motion and bone. We adopt linear evaluation on each view of the models. Table 1 reports that 1) CrosSCLR improves the capability of each single SkeletonCLR model, e.g., CrosSCLR-joint (79.88) *v.s* SkeletonCLR-joint (76.44) on xview protocol; 2)
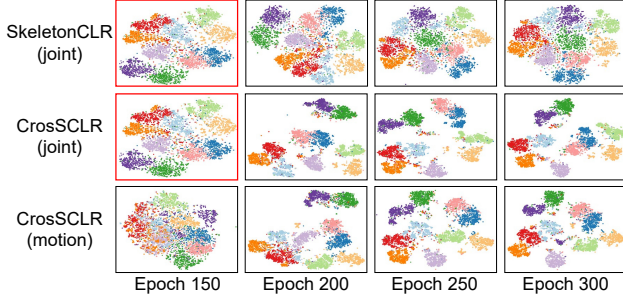
Figure 4. The t-SNE visualization of embeddings at different epochs during pre-training. Embeddings from 10 categories are sampled and visualized with different colors. For CrosSCLR, $\mathcal{L}_{cross}$ starts to be available at epoch 150, so its distribution has no difference from that of SkeletonCLR before epoch 150, shown in red boxes. More visual results are shown in Appendix.

CrosSCLR bridges the performance gap of two views and jointly improves their accuracy, e.g., for SkeletonCLR, joint (76.44) *v.s* motion (50.82) but for CrosSCLR, joint (79.88) *v.s* motion (77.59); 3) CrosSCLR improves the multi-view ensemble results via cross-view training. In summary, the cross-view high-confidence knowledge does help the model extract more discriminating representations.

Qualitative Results. We apply t-SNE [30] with fix settings to show the embedding distribution of SkeletonCLR and CrosSCLR on $150, 200, 250, 300$ epochs during pre-training in Figure 4. Note that cross-view loss, Equation (8), is available only after epoch 150. From the visual results, we can draw a similar conclusion to that in Table 1. Embeddings of CrosSCLR are clustered more closely than that of SkeletonCLR, which is more discriminating. For CrosSCLR, the distributions of joint and motion are distinct at 150-th epoch but look very similar at 300-th epoch, i.e., consistent distribution. Especially, they both build a more "regular" space than SkeletonCLR, proving the effectiveness of CrosSCLR.

Effects of Contrastive Setting top-K. As hyper-parameter K determines the number of mined samples, influencing the depth of knowledge exchange, we study how K impacts the performance in cross-view learning. Table 2 shows that K has a great influence on the performance and achieves the best result when $K = 1$. However, a larger K decreases the performance, because the not so confident information may lead the model astray in an unsupervised case.

Contrastive Setting N_+ and Embedding Context $\mathcal{S}$. We develop following models in Table 4 for comparison: 1) SkeletonCLR + $\mathcal{L}_{KM}$ is a model with single-view knowledge mining. 2) CrosSCLR *w/o.* embedding context (EC) is the model only using the contrastive setting N_+ for cross-view learning, which ignores the embedding context/distribution, i.e., $S_i^v = 1, \forall i \in N$ in Equation (6). The results of SkeletonCLR + $\mathcal{L}_{KM}$ show that KM improves

top-K	NTU-60 (%)	
	xsub	xview
0	70.5	77.4
1	**74.5**	**82.1**
3	73.7	79.9
5	72.4	79.2
7	73.0	78.6
10	64.4	69.9

Table 2. Results of pre-training 2s-CrosSCLR with various K in knowledge miner Γ.

Augmentation		NTU-60 (%)	
Shear β	Crop γ	xsub	xview
0	0	33.3	26.2
0.2	0	62.7	67.7
0.5	0	**66.3**	**68.8**
1.0	0	62.0	66.8
0.5	4	67.6	76.3
0.5	6	68.3	**76.4**
0.5	8	**69.1**	74.7

Table 3. Ablation study on different data augmentations for SkeletonCLR (joint).

Method	Views of Pre-training	NTU-60 (%)	
		xsub	xview
SkeletonCLR	Joint	68.3	76.4
SkeletonCLR + $\mathcal{L}_{KM}$	Joint	69.3	77.4
CrosSCLR *w/o.* EC	Joint + Motion	71.4	78.5
CrosSCLR	Joint + Motion	**72.9**	**79.9**

Table 4. Ablation study on contrastive settings N_+ and embedding context (EC). The models are linear evaluated on **only joint**.

the representation capability of SkeletonCLR. Additionally, CrosSCLR achieves worse performance without using embedding context (EC), proving the significance of similarity/distribution among samples.

Effects of Augmentations. SkeletonCLR and CrosSCLR are based on contrastive learning, but the data augmentation strategy used on skeleton data is rarely explored, especially for the GCN encoder. We verify the effectiveness of data augmentation and the impact of different augmented intensities in skeleton-based contrastive learning by conducting experiments on SkeletonCLR, as shown in Table 3. It indicates the importance of data augmentation in SkeletonCLR. We choose $\beta = 0.5$ and $\gamma = 6$ as default settings according to the mean accuracy on xsub and xview protocols.

4.4. Comparison

We compare CrosSCLR with other methods under *linear evaluation* and *finetune* protocols. Since the backbone in many methods is an RNN-based model, e.g., GRU or LSTM, we additionally use LSTM (following the setting in [39]) as the encoder for a fair comparison, i.e., CrosSCLR (LSTM).

Unsupervised Results on NTU-60. In Table 5, LongT GAN [65] adversarially trains the model by skeleton inpainting pretext task, MS^{2}L [26] trains the model by multi-task scheme, i.e, prediction, jiasaw puzzle and instance discrimination, AS-CAL [39] uses momentum LSTM encoder for contrastive learning with single-view skeleton sequence, P&C [47] trains a stronger encoder by weakening decoder, and SeBiReNet [34] constructs a human-like GRU

Method	Encoder	Classifier	NTU-60 (%) xsub	xview
LongT GAN [65]	GRU	FC	39.1	48.1
MS^{2}L [26]	GRU	GRU	52.6	-
AS-CAL [39]	LSTM	FC	58.5	64.8
P&C [47]	GRU	KNN	50.7	76.3
SeBiReNet [34]	GRU	LSTM	-	79.7
3s-CrosSCLR (LSTM)	LSTM	FC	62.8	69.2
3s-CrosSCLR (LSTM)	LSTM	LSTM	70.4	79.9
3s-CrosSCLR‡	ST-GCN	FC	72.8	80.7
3s-CrosSCLR	ST-GCN	FC	**77.8**	**83.4**

Table 5. Unsupervised results on NTU-60. These methods are pre-trained to learn encoder and then follow the linear evaluation protocol to learn the classifiers. "‡" indicates the model pre-trained on NTU-61-120.

Method	Supervision	NTU-120 (%) xsub	xset
Part-Aware LSTM [40]	Supervised	25.5	26.3
Soft RNN [15]	Supervised	36.3	44.9
TSRJI [2]	Supervised	67.9	62.8
ST-GCN [57]	Supervised	**79.7**	**81.3**
AS-CAL [39]	Unsupervised	48.6	49.2
3s-CrosSCLR (LSTM)	Unsupervised	53.9	53.2
3s-CrosSCLR	Unsupervised	**67.9**	**66.7**

Table 6. Unsupervised results on NTU-120. We show and compare our method with unsupervised and supervised methods.

network to utilize view-independent and pose-independent feature. Our CrosSCLR exploits the multi-view knowledge by cross-view consistent knowledge mining. Taking a fully-connected layer (FC) as the classifier, our model outperforms other methods with the same classifier. With LSTM classifier and LSTM encoder, our model outperforms the above methods on both xsub and xview protocols.

Results on NTU-120. As few unsupervised results are reported on NTU-120 dataset, we compare our method with unsupervised and supervised methods. As shown in Table 6, TSRJI [2] supervisedly utilizes attention LSTM, AS-CAL [39] adopts LSTM for skeleton modeling, and our method defeats the other unsupervised method and some of the supervised methods.

Linear Classification with Fewer Labels. We follow the same protocol as that of MS^{2}L [26], i.e., pre-training with all training data and then finetuning the classifier with only 1% and 10% randomly-selected labeled data respectively. As shown in Table 7, CrosSCLR achieves higher performance than other methods.

Finetuned Results on NTU-60 and NTU-120. We first unsupervisedly pre-train our model and follow the *finetune* protocol for evaluation. For fair comparison, ST-

Method	Label Fraction	NTU-60 (%) xsub	xview
LongT GAN [65]	1%	35.2	-
MS^{2}L [26]	1%	33.1	-
3s-CrosSCLR	1%	**51.1**	**50.0**
LongT GAN [65]	10%	62.0	-
MS^{2}L [26]	10%	65.2	-
3s-CrosSCLR	10%	**74.4**	**77.8**

Table 7. Linear classification with fewer labels on NTU-60.

Method	NTU-60 (%) xsub	xview	NTU-120 (%) xsub	xset
3s-ST-GCN* [57]	85.2	91.4	77.2	77.1
3s-CrosSCLR‡ (FT)	85.6	92.0	-	-
3s-CrosSCLR (FT)	**86.2**	**92.5**	**80.5**	**80.4**

Table 8. Finetuned results on NTU-60 and NTU-120. ST-GCN* is the method reproduced by released code. "‡" indicates the model pre-trained on NTU-61-120. "FT" means finetune protocol.

GCN* [57] in Table 8 has the same number of parameters as 3s-CrosSCLR ($1/4$ channel with three streams). It shows that the finetuned model, CrosSCLR (FT) outperforms the supervised ST-GCN on both NTU-60 and NTU-120 datasets, indicating the effectiveness of cross-view pretraining.

Transfer Ability. We first pre-train CrosSCLR on NTU-61-120, and then transfer it to NTU-60 for linear evaluation, noted as CrosSCLR‡. The model trained under xsub protocol is transferred to the xsub protocol of NTU-60; the model trained under xset protocol is transferred to the xview protocol of NTU-60. In Table 5, it achieves better results than the other unsupervised methods, and its supervisedly finetuning result is higher than ST-GCN as shown in Table 8.

5. Conclusion

In this work, we propose a Cross-view Contrastive Learning framework for unsupervised 3D skeleton-based action representation to exploit multi-view high-confidence knowledge as complementary supervision. It integrates single-view contrastive learning with cross-view consistent knowledge mining modules which convey the contrastive settings and embedding context among views by high-confidence sample mining. Experiments show remarkable results of CrosSCLR for action recognition on NTU datasets.

Acknowledgement

References

[1] Amor Ben Tanfous, Hassen Drira, and Boulbaba Ben Amor. Coding kendall's shape trajectories for 3d action recognition. In *CVPR*, pages 2840–2849, 2018.

[2] Carlos Caetano, François Brémond, and William Robson Schwartz. Skeleton image representation for 3d action recognition based on tree structure and reference joints. In *SIBGRAPI*, pages 16–23. IEEE, 2019.

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *TPAMI*, 43(1):172–186, 2019.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv:2002.05709*, 2020.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv:2003.04297*, 2020.

[6] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, pages 1110–1118, 2015.

[7] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, pages 3636–3645, 2017.

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv:1803.07728*, 2018.

[9] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial geometry-aware human motion prediction. In *ECCV*, pages 786–803, 2018.

[10] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *NeurIPS*, 33, 2020.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

[13] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *NIPS*, 15:857–864, 2002.

[14] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[15] Jian-Fang Hu, Wei-Shi Zheng, Lianyang Ma, Gang Wang, Jianhuang Lai, and Jianguo Zhang. Early action prediction by soft regression. *TPAMI*, 41(11):2568–2583, 2018.

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

[17] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *CVPR*, pages 3288–3297, 2017.

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv:2004.11362*, 2020.

[19] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, volume 33, pages 8545–8552, 2019.

[20] Jogendra Nath Kundu, Maharshi Gor, Phani Krishna Uppala, and Venkatesh Babu Radhakrishnan. Unsupervised feature learning of human actions as trajectories in pose embedding manifold. In *WACV*, pages 1459–1467. IEEE, 2019.

[21] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, pages 667–676, 2017.

[22] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Skeleton-based action recognition with convolutional neural networks. In *ICME Workshops*, pages 597–600. IEEE, 2017.

[23] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Unsupervised learning of view-invariant action representations. In *NeurIPS*, pages 1254–1264, 2018.

[24] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*, pages 3595–3603, 2019.

[25] Duohan Liang, Guoliang Fan, Guangfeng Lin, Wanjun Chen, Xiaorong Pan, and Hong Zhu. Three-stream convolutional neural network with multi-task and ensemble learning for 3d action recognition. In *CVPR Workshops*, 2019.

[26] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In *ACMMM*, pages 2490–2498, 2020.

[27] Jun Liu, Amir Shahroudy, Mauricio Lisboa Perez, Gang Wang, Ling-Yu Duan, and Alex Kot Chichung. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *TPAMI*, 2019.

[28] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *CVPR*, pages 143–152, 2020.

[29] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *CVPR*, pages 2203–2212, 2017.

[30] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[31] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *CVPR*, pages 2891–2900, 2017.

[32] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, pages 527–544. Springer, 2016.

[33] Bingbing Ni, Gang Wang, and Pierre Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *ICCV Workshops*, pages 1147–1153. IEEE, 2011.

[34] Qiang Nie, Ziwei Liu, and Yunhui Liu. Unsupervised 3d human pose representation with viewpoint and pose disentanglement. *ECCV*, 2020.

[35] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016.

[36] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, pages 9359–9367, 2018.

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018.

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

[39] Haocong Rao, Shihao Xu, Xiping Hu, Jun Cheng, and Bin Hu. Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. *arXiv:2008.00188*, 2020.

[40] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016.

[41] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *CVPR*, pages 7912–7921, 2019.

[42] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, pages 12026–12035, 2019.

[43] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

[44] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *CVPR*, pages 1227–1236, 2019.

[45] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. Spatio-temporal attention-based lstm networks for 3d action recognition and detection. *TIP*, 27(7):3459–3471, 2018.

[46] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, pages 843–852, 2015.

[47] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *CVPR*, pages 9631–9640, 2020.

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

[49] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, pages 588–595, 2014.

[50] Raviteja Vemulapalli and Rama Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *CVPR*, pages 4471–4479, 2016.

[51] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, pages 4006–4015, 2019.

[52] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, pages 1290–1297. IEEE, 2012.

[53] Minsi Wang, Bingbing Ni, and Xiaokang Yang. Learning multi-view interactional skeleton graph for action recognition. *TPAMI*, 2020.

[54] Chen Wei, Lingxi Xie, Xutong Ren, Yingda Xia, Chi Su, Jiaying Liu, Qi Tian, and Alan L Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *CVPR*, pages 1910–1919, 2019.

[55] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018.

[56] Jingwei Xu, Zhenbo Yu, Bingbing Ni, Jiancheng Yang, Xiaokang Yang, and Wenjun Zhang. Deep kinematics analysis for monocular 3d human pose estimation. In *CVPR*, pages 899–908, 2020.

[57] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[58] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *ICCV*, pages 2752–2759, 2013.

[59] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, pages 1476–1485, 2019.

[60] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *ICCV*, pages 2117–2126, 2017.

[61] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *TPAMI*, 41(8):1963–1978, 2019.

[62] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *CVPR*, pages 1112–1121, 2020.

[63] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666. Springer, 2016.

[64] Xikun Zhang, Chang Xu, and Dacheng Tao. Context aware graph convolution for skeleton-based action recognition. In *CVPR*, pages 14333–14342, 2020.

[65] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In *AAAI*, 2018.