Pattern Recognition 150 (2024) 110333



Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr





Improving self-supervised action recognition from extremely augmented skeleton sequences

Tianyu Guo, Mengyuan Liu *, Hong Liu, Guoquan Wang, Wenhao Li

National Key Laboratory of General Artificial Intelligence, Peking University, Shenzhen Graduate School, China
Ultra-High Definition Immersive Media Technology Laboratory of Guangdong Province, China

ARTICLE INFO

Keywords:
Self-supervised skeleton-based action recognition
Contrastive learning

ABSTRACT

Self-supervised contrastive learning has been widely applied to skeleton-based action recognition due to its ability to learn discriminative features. However, directly applying the existing contrastive learning framework for 3D skeleton learning is limited by the well-designed augmentations and the simple multi-stream decision-level fusion. To deal with these drawbacks, we propose a three-stream contrastive learning framework utilizing abundant information mining for self-supervised action representation (3s-AimCLR++). For single-stream contrastive learning, extreme augmentation is first proposed to generate more movement patterns, which can introduce more movement patterns to improve the universality of the learned representations. Since directly using extreme augmentation can barely boost the performance due to the drastic changes in original identity, the Distributional Divergence Minimization (DDM) loss is proposed to utilize the extreme augmentation more gently. Moreover, the Single-Stream Nearest Neighbors Mining (SNNM) is proposed to expand positive samples to make the learning process more reasonable. For multi-stream, existing methods simply ensemble the results. Yet, considering the complementarity of information between different streams, we propose Multi-Stream Aggregation and Interaction (MSAI) strategy to better fuse multi-stream information. Extensive experiments on NTU-60, NTU-120, and PKU-MMD datasets have verified that our 3s-AimCLR++ can significantly perform favorably against state-of-the-art methods under a variety of evaluation protocols. The code and models are available at https://github.com/Levigty/AimCLR-v2.

1. Introduction

Human action recognition is an important research topic in computer vision, which has been widely applied in many applications such as intelligent video surveillance, human–computer interaction, and video content analysis [1]. Methods focused on recognizing actions from RGB data achieve promising results, however, they still face the challenges such as background clutter, illumination changes, and appearance variations. With the development of depth sensors and the human pose estimation algorithms [2], the acquisition of skeleton data is easier. In addition, skeleton data is more robust to background clutter, illumination changes, and appearance variation. Therefore, skeleton-based action recognition has gradually become a significant branch of action recognition in recent years.

For skeleton-based action recognition, one of the main challenges is to learn discriminative spatial and temporal features of skeleton sequences. There are some efforts have been made in previous works. Traditional skeleton-based action recognition methods [3,4] mainly focus on designing hand-crafted features to model the pattern of the skeleton sequences. With the increasing development of deep learning technologies, Recurrent Neural Networks (RNNs) and Long-Short Term Memory (LSTM) have been exploited to model the temporal dynamics of skeleton sequences [5,6]. Further, Convolutional Neural Networks (CNNs) based methods [7,8] propose to convert skeleton sequences to RGB images and then adopt CNNs to learn the discriminative spatio-temporal features. In recent years, due to the similarity between the human skeleton and graph structure, some methods [9–11] have also been proposed to capture the spatio-temporal relationship based on Graph Convolutional Networks (GCNs), which have achieved superior performances. Nevertheless, the methods mentioned above are all trained in a supervised manner, which requires numerous labeled data to learn the action representation. Actually, the cost of labeling large-scale datasets is particularly high. Therefore, self-supervised skeleton-based action recognition has recently received more and more attention due to its ability to better utilize unlabeled data to learn more general feature representations.

* Corresponding author.
E-mail addresses: levigty@stu.pku.edu.cn (T. Guo), liumengyuan@pku.edu.cn (M. Liu), hongliu@pku.edu.cn (H. Liu), guoquanwang@stu.pku.edu.cn (G. Wang), wenhaoli@pku.edu.cn (W. Li).

https://doi.org/10.1016/j.patcog.2024.110333
Received 14 July 2023; Received in revised form 24 December 2023; Accepted 8 February 2024
Available online 9 February 2024
0031-3203/© 2024 Elsevier Ltd. All rights reserved.

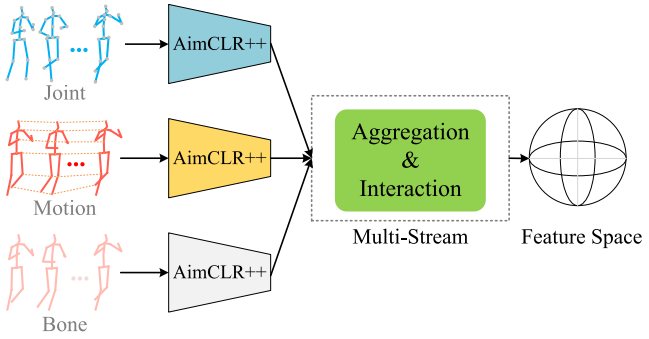


Fig. 1. Illustration of proposed 3s-AimCLR++ for self-supervised skeleton-based action recognition. 3s-AimCLR++ consists of single-stream contrastive learning and multi-stream information aggregation and interaction to build a better feature space.

For self-supervised skeleton-based action recognition, several works [12,13] focus on generative pretext tasks like reconstruction, motion prediction, and so on. Generative pretext tasks focus on the spatio-temporal details, forcing the model to learn better feature representations. With the development of contrastive self-supervised learning, many works [14–16] show their ability to make feature representations have better discrimination. Several works [17,18] rely on the contrastive learning framework, achieving better performance by pulling the positive pairs closer and pushing the negative pairs away. There are also some methods [19,20] that combine the generative task with contrastive learning to enable the model to extract discriminative representations in the multi-task learning manner.

Among them, compared with the generative methods, the contrastive learning methods pay more attention to the instance-level information rather than the detailed information and then construct a more discriminative feature space that is more suitable for downstream tasks, so it has received extensive attention. Further, we rethink existing contrastive self-supervised skeleton-based action recognition methods and find several drawbacks. We hope to come up with effective solutions to these problems and then improve the contrastive self-supervised skeleton-based action recognition methods.

Problem 1 (*How to Effectively Bring More Spatial–Temporal Movement Patterns?*)**.** Contrastive self-supervised methods use data augmentation to construct positive pairs and studies [21] have shown that data augmentation design is crucial. In SkeletonCLR [18], it just uses *Shear* and *Crop* to generate augmented sequences and we call it "normal augmentation" in this paper. As shown in Fig. 3, normal augmentation brings about limited spatio-temporal movement patterns. It is believed that seeing richer movement patterns in the pretext task clearly improves the feature extraction capability of the encoder. Thus, we hope to bring richer movement patterns through stronger data augmentations, thereby improving the generalizability of learned representations.

Problem 2 (*How to Better Use the Introduced Spatial–Temporal Information?*)**.** AS-CAL [17] proposes to use plenty of spatio-temporal augmentations, but the performance is not that satisfactory. We find that stronger data augmentations bring novel movement patterns while the augmented skeleton sequence may not keep the identity of the original sequence. Therefore, directly using extreme augmentation may not necessarily be able to boost the performance due to the drastic changes in original identity. Thus, additional efforts are needed to explore a more suitable way to use the introduced spatial–temporal information.

Problem 3 (*How to Better Choose the Positive and Negative Pairs?*)**.** In contrastive learning, two different augmented samples from the same sample are considered as positive pairs, while samples in the memory bank are all treated as negative samples. However, the samples in the memory bank are not necessarily all negative samples. It makes the learning process unreasonable to a certain extent.

Problem 4 (*How to Better Aggregate Multi-Stream Information?*)**.** For a given skeleton sequence, the joint information, motion information, and bone information are easy to obtain. However, existing methods [18,22] usually simply ensemble the results of different streams without considering the complementary information between different streams. Thus, there is an urgent need to find an efficient multi-stream information aggregation and interaction strategy to construct a more discriminative feature space.

To this end, a three-stream contrastive learning framework utilizing abundant information mining for self-supervised action representation (3s-AimCLR++) is proposed. Specifically, the framework of the proposed method is shown in Fig. 1. It includes AimCLR++ (Fig. 2) for a single stream and information aggregation and interaction strategy for multi-stream. Firstly, we hope to use extreme augmentation to bring more spatial–temporal movement patterns. Then the Distributional Divergence Minimization (DDM) loss is proposed to better use the introduced spatial–temporal information. Single-Stream Nearest Neighbors Mining is used to expand the positive set to make the learning process more reasonable. Furthermore, we propose a Multi-Stream Aggregation and Interaction (MSAI) strategy to better fuse information.

In summary, we have made the following contributions:

- We propose a new framework for self-supervised skeleton-based action recognition, called 3s-AimCLR++, which aims to learn more discriminative feature representations from extremely augmented sequences and conduct effective aggregation and interaction of multi-stream information.
- For single-stream, extreme augmentation is proposed to introduce more movement patterns. Then the DDM loss is proposed to gently learn from the introduced movement patterns. In order to make the positive set more reasonable, we further propose the Single-Stream Nearest Neighbors Mining.
- For multi-stream, Multi-Stream Aggregation and Interaction (MSAI) strategy is proposed to better fuse multi-stream information rather than simply ensemble the results at the decision level.
- State-of-the-art results are achieved under a variety of evaluation protocols on NTU-60, NTU-120, and PKU-MMD datasets, showing the superiority of our framework to learn better feature representations.

This paper is an extension of our conference paper [23]. The current paper extends AimCLR [23] in several ways. (1) We remove the EADM module to simplify the structure of AimCLR and explore the combinations of extreme augmentation to propose AimCLR++ for a single stream. (2) We propose an efficient strategy to aggregate and interact the multi-stream information. (3) Equipped with these new designs, our 3s-AimCLR proposed in [23] is upgraded to 3s-AimCLR++. Experimental results show that our 3s-AimCLR++ can achieve better performance and generalization ability. (4) We add a lot of experiments such as exploring the robustness of the method to different extreme augmentations, the number of most similar neighbors in SNNM, the performance of transfer learning, and the quantitative and qualitative experiments, to systematically show the effectiveness of the proposed method. We observe that the proposed 3s-AimCLR++ achieves consistent and substantial gains on three public datasets.

The remainder of this paper is organized as follows. We first discuss the related work in Section 2. Then we introduce our single-stream AimCLR++ in Section 3 and 3s-AimCLR++ in Section 4. After that, Section 5 provides the experimental results and Section 6 concludes this paper.

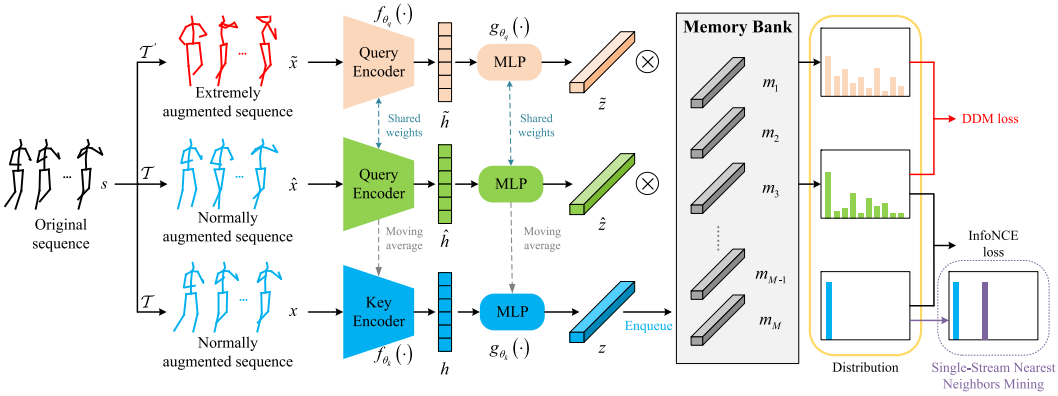


Fig. 2. The pipeline of the proposed single-stream AimCLR++. Through the normal augmentation $\mathcal{T}$ and extreme augmentation $\mathcal{T}'$, x, $\hat{x}$ and $\tilde{x}$ are obtained from the input sequence s. The query encoder and a query MLP are utilized to extract $\hat{z}$ and $\tilde{z}$. The key encoder and a key MLP are used to obtain z. z are stored in the memory bank in each training step, serving as negative samples for the next training steps. While using InfoNCE loss, we propose DDM loss to minimize the distribution divergence of $\hat{z}$, $\tilde{z}$. Furthermore, we propose single-stream nearest neighbors mining to expand the positive set.

2. Related work

Supervised Skeleton-based Action Recognition. Traditional skeleton-based action recognition methods [3,4] mainly focus on designing hand-crafted features to represent the skeleton sequences. With the rapid development of deep learning in recent years, some methods [5,6] use RNNs to model the temporal feature of the skeleton sequence. Since RNNs suffer from gradient vanishing, some methods [7, 8] convert the 3D skeleton sequence into a pseudo image and leverage CNNs to achieve competitive results. However, both RNNs and CNNs fail to fully represent the structure of the skeleton data because the skeleton data are naturally embedded in the form of graphs rather than a vector sequence or 2D matrices. Recently, with the introduction of ST-GCN [9], GCNs have been widely used in the field of skeleton-based action recognition, and a variety of GCN-based methods [24] have emerged on the basis of ST-GCN [9] to better model the spatio-temporal relationship. In this paper, to verify the superiority of our proposed framework, we adopt the widely-used ST-GCN as the encoder to extract features.

Self-supervised Learning. In recent years, due to the huge annotations costs, self-supervised methods have received a lot of attention. Early pretext tasks such as predicting rotation, solving jigsaw puzzle, and temporal order prediction is designed to learn good features. Contrastive learning is one such pretext task that relies on instance discrimination. The goal is to classify a set of positives (augmented version of the same instance) against a set of unrelated negatives which helps the model learn good features. Recent works have demonstrated exceptional performance using these techniques in a variety of domains [25–28], including images, videos, graphs, text, etc. SimCLR [29], MoCo [14,15], and VICReg [30] frameworks have been very popular due to their ease of use and general applicability. Recently, several non-contrastive learning approaches like BYOL [31], SimSiam [32], MAE [33] have shown promising performance but contrastive learning still offers complementary benefits. In this work, we focus on contrastive learning which have been shown to be superior to non-contrastive ones for self-supervised skeleton-based action recognition. And we design SNNM and MSAI to further improve the performance of contrastive learning in this field.

Self-supervised Skeleton-based Action Recognition. Self-supervised learning is a specific type of unsupervised learning. It defines specific pretext tasks as the learning targets to learn powerful representation from unlabeled data. For the skeleton sequence data, there are generative pretext tasks, contrastive pretext tasks, and multiple pretext tasks.

For generative pretext tasks, LongTGAN [12] proposes to use the encoder–decoder to reconstruct the input sequence to obtain useful feature representation. P&C [13] is also based on the reconstruction pretext task, it proposes a training strategy to weaken the decoder, forcing the encoder to learn more discriminative features. Cheng et al. [34] propose to predict the motion of the 3D skeleton as a pretext task and use the Hierarchical Transformer to encode the skeleton sequence. Yang et al. [22] design a novel skeleton cloud colorization technique to learn skeleton representations. Xu et al. [35] propose the Motion Capsule Autoencoder (MCAE) to address the "transformation invariance" challenge in the motion representations. These methods based on frame-level generation are considered not inherently appropriate for representation learning for skeleton-based action recognition [36]. Indeed, methods based on frame-level generation, like those mentioned previously, may not be inherently suitable for representation learning in the context of skeleton-based action recognition [36]. There might be limitations in effectively capturing the underlying spatiotemporal relationships and structures present in the skeleton data using such frame-level generation approaches.

For contrastive pretext tasks, AS-CAL [17] proposes to use plenty of spatio-temporal augmentations and use momentum LSTM and memory bank. SkeletonCLR [18] is built based on MoCo v2 [14,15] and uses *Shear* and *Crop* as the augmentations, which achieves competitive performance. Further, CrosSCLR [18] proposes a cross-view consistent

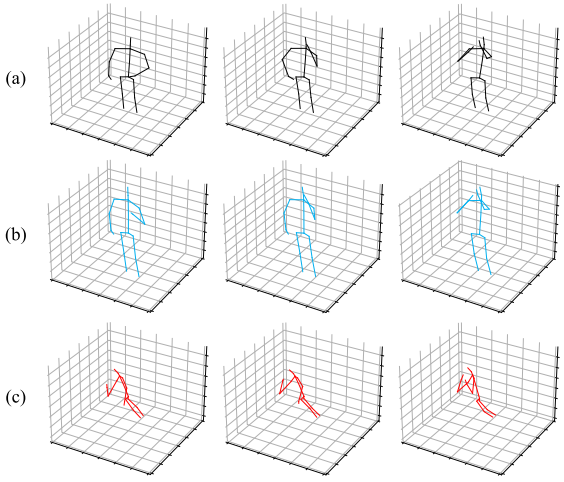


Fig. 3. The visualization of (a) the original sequence (black), (b) normally augmented sequence [18] (blue), and (c) the strongly augmented sequence (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

knowledge mining strategy to improve the performance of SkeletonCLR. ST-CL [36] acquires action-specific features by regarding the spatio-temporal continuity of motion tendency as the supervisory signal. Thoker et al. [37] propose inter-skeleton contrastive learning, which learns from multiple different input skeleton representations in a cross-contrastive manner.

The generative pretext tasks pay more attention to the detailed information of the skeleton sequence, and the contrastive pretext task pays more attention to the discriminative information of the skeleton sequence at the instance level. In order to learn more general features, there are also several methods to do multiple pretext tasks. MS^{2}L [19] proposes to use multiple pretext tasks (i.e., motion prediction, jigsaw puzzle recognition, and contrastive learning) and combine them to encourage the Bi-GRU encoder to capture more suitable features. CP-STN [20] combines contrastive learning paradigms and generative pretext tasks in one framework by using asymmetric spatial and temporal augmentations to enable network extracting discriminative representations. Tanfous et al. [38] propose to reconcile AS-CAL and P&C by developing a taxonomy of self-supervised learning for action recognition. PCRP [39] proposes a novel framework by prototypical contrast and reverse prediction to fully learn inherent semantic similarity.

Our method uses contrastive pretext tasks to improve self-supervised action recognition from both single-stream contrastive learning and multi-stream information aggregation. Better performance is achieved even without the use of multiple pretext tasks due to better introduction and attention to spatio-temporal information.

3. AimCLR++

For a given skeleton sequence, existing methods usually utilize the information of different streams (i.e., joint, motion and bone). In this section, we propose a contrastive learning framework named AimCLR++ which is shown in Fig. 2 and could be used for single-stream data. The proposed AimCLR++ consists of the following key components: (1) Spatio-Temporal Data Augmentation, (2) Distributional Divergence Minimization Loss, (3) Single-Stream Nearest Neighbors Mining. In order to easily understand the main idea of our method, we first give an overview of AimCLR++, then introduce each component in detail.

3.1. Overview of AimCLR++

As shown in Fig. 2, for the input sequence s, we apply normal augmentation $\mathcal{T}$ to obtain the normally augmented sequence x and $\hat{x}$. And the extreme augmentation $\mathcal{T}'$ is applied to obtain the extremely augmented sequence $\tilde{x}$. Then the query encoder f_{θ_q} is utilized to extract features: $\hat{h} = f_{\theta_q}(\hat{x})$ and $\tilde{h} = f_{\theta_q}(\tilde{x})$. A query multilayer perceptron (MLP) g_{θ_q} is used to project the features to a new space: $\hat{z} = g_{\theta_q}(\hat{h})$, $\tilde{z} = g_{\theta_q}(\tilde{h})$. The key encoder f_{θ_k} and the key MLP g_{θ_k} are the momentum updated version of f_{θ_q} and g_{θ_q}.

The memory bank $\mathbf{M} = \{m_i\}_{i=1}^{M}$ containing M negative samples is a first-in-first-out queue updated by z. After each iteration, z will enqueue while the earliest embedding in $\mathbf{M}$ will dequeue. $\mathbf{M}$ provides numerous negative embeddings while the new calculated z is the positive embedding. The pipeline of our baseline method SkeletonCLR [18] is shown in the bottom part of Fig. 2. It use the InfoNCE loss [40]:

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

where τ is the temperature hyper-parameter. Here, z, $\hat{z}$ are normalized and dot product $\hat{z} \cdot z$ is to compute their similarity.

For the proposed extremely augmented view, it introduces rich spatio-temporal information while may not keep the identity of the original sequence. For this reason, InfoNCE loss is not suitable for this view, which has also been verified in the experiment. Therefore, we propose the Distributional Divergence Minimization (DDM) loss (Section 3.3). And the Single-Stream Nearest Neighbors Mining (Section 3.4) is proposed to expand the positive set to make the learning process more reasonable.

After computing the total loss, the query encoder and query MLP are updated via gradients while the key encoder and key MLP are updated as a moving-average of the query encoder and query MLP. We denote the parameters of the query encoder and query MLP as θ_q and those of the key encoder and key MLP as θ_k. Then the key encoder and key MLP are updated as:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q, \tag{2}$$

where $m \in [0, 1)$ is a momentum coefficient. The key encoder is updated slowly based on the query encoder change, which ensures stable key representations.

Based on the above, the query encoder is unsupervisedly trained. At last, we use the query encoder to extract feature representations for downstream tasks.

3.2. Spatio-temporal data augmentations

As shown in Fig. 3, the movement patterns introduced by the normal augmentations used in SkeletonCLR [18] are relatively limited. Those carefully designed augmentations limit the encoder to further explore the novel patterns exposed by other augmentations. What is more, it is believed that seeing richer movement patterns in the pretext task clearly improves the feature extraction capability of the encoder. Therefore, extreme augmentation is introduced to bring more novel movement patterns to learn more general feature representation. According to the characteristics of the skeleton sequence and several previous works [17,18], we adopt the following data augmentations.

Definition 1 (*Normal Augmentations* $\mathcal{T}$)**.** In the baseline method SkeletonCLR [18], *Shear* and *Crop* are used as the augmentations. We define "*Shear* + *Crop*" as normal augmentation in subsequent descriptions.

Shear: The shear augmentation is a linear transformation [17,18,37]. The shape of 3D coordinates of body joints is inclined at a random angle. The transformation matrix is defined as:

$$A = \begin{bmatrix} 1 & a_{12} & a_{13} \\ a_{21} & 1 & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix}, \tag{3}$$

where $a_{12}, a_{13}, a_{21}, a_{23}, a_{31}, a_{32}$ are shear factors sampled randomly from $[-\beta, \beta]$. β is the shear amplitude. Here we set $\beta = 0.5$. Then the sequence is multiplied by the transformation matrix A on the channel dimension.

Crop: For image classification tasks, *Crop* is a very commonly used data augmentation, because it can increase the diversity while maintaining the distinction of original samples. For the temporal skeleton sequence, specifically, we symmetrically pad some frames to the sequence and then randomly crops it to the original length [18,37]. The padding length is defined as T/γ, γ is the padding ratio and here we set $\gamma = 6$.

Definition 2 (*Extreme Augmentation* $\mathcal{T}'$)**.** We hope to explore a more general framework in which extreme augmentation can introduce more novel movement patterns than normal augmentation. Thus, we define the extreme augmentation as adding several augmentations on the basis of normal augmentation to introduce more movement patterns. And we introduce some augmentations as options. Spatial augmentations like *Spatial Flip*, *Rotate*, *Axis Mask*, temporal augmentations like *Temporal Flip* and spatio-temporal augmentations like *Gaussian Noise* and *Gaussian Blur*. We use the combination of these augmentations and normal augmentation as "Extreme Augmentation" to finally get one extremely augmented sequence. According to the experimental results, the "Extreme Augmentation" we finally use is "Normal Augmentation + *Spatial Flip* + *Rotate* + *Axis Mask* + *Gaussian Noise* + *Gaussian Blur*".

Spatial Flip: The skeletal structure of the human body is symmetrical, and we are able to distinguish the actions after the skeletal symmetry exchange. For example, kicking with the left leg and kicking with the right leg should be considered the same class of "kick". Thus, the skeleton sequence of each frame is symmetrically augmented based on the probability of $p = 0.5$. Specifically, the position of the torso in the center of the skeleton remains unchanged, while the positions of the left and right sub-skeletons are exchanged.

Rotate: Considering the change of camera position, we introduce random "rotate" to the skeleton sequence [17,18,37]. For all skeleton joints, we randomly select a main rotation axis $a \in \{X, Y, Z\}$ and choose a random rotation angle $[0, \pi/6]$, and the remaining two rotation axis randomly chooses the angle $[0, \pi/180]$. This is also in line with our common sense of recognizing actions: the change in the observation view does not affect the action itself.

Gaussian Noise: To simulate the noisy positions caused by estimation or annotation [17], we add Gaussian noise $\mathcal{N}(0, 0.01)$ over joint coordinates of the original sequence with the probability of $p = 0.5$.

Gaussian Blur: We randomly sample $\sigma \in [0.1, 2.0]$ for the Gaussian kernel [17], which is a sliding window with length of 15. Joint coordinates of the original sequence are blurred with the probability of $p = 0.5$ by the kernel $G(\cdot)$ below:

$$G(t) = \exp(-\frac{t^2}{2\sigma^2}) \quad t \in \{-7, -6, \dots, 6, 7\}, \tag{4}$$

where t denotes the relative position from the center skeleton, and the length of the kernel is set to 15 corresponding to the total span of t.

Axis Mask: For the 3D skeleton sequence, we hope that the sequence projected to 2D space can be used as its augmented sequence. Specifically, we randomly select an axis $a \in \{X, Y, Z\}$ and apply the zero-mask with the probability of $p = 0.5$.

Temporal Flip: In PCRP [39], reverse prediction is proposed to learn more high-level information (e.g. movement order) that are meaningful to human perception. Hence, temporal flip is used as one of the data augmentation options. Specifically, the skeleton sequence $S = s_1, \dots, s_T$ is reversed to $S' = s_T, \dots, s_1$ with the probability of $p = 0.5$.

3.3. Distributional divergence minimization loss

From Fig. 2, for a query $\hat{z}$, the new calculated z is the positive embedding and numerous negative embeddings are stored in the memory bank $\mathbf{M} = \{m_i\}_{i=1}^{M}$. Thus, we can obtain a conditional distribution which encodes the likelihood of the query $\hat{z}$ being assigned to the embedding m_i in the memory bank $\mathbf{M}$:

$$p(m_i \,|\, \hat{z}) = \frac{\exp(\hat{z} \cdot m_i/\tau)}{\exp(\hat{z} \cdot z/\tau) + \sum_{i=1}^{M} \exp(\hat{z} \cdot m_i/\tau)}. \tag{5}$$

Similarly, we can also have the likelihood of positive pairs for the query $\hat{z}$ being assigned to its positive embedding z:

$$p(z \,|\, \hat{z}) = \frac{\exp(\hat{z} \cdot z/\tau)}{\exp(\hat{z} \cdot z/\tau) + \sum_{i=1}^{M} \exp(\hat{z} \cdot m_i/\tau)}. \tag{6}$$

Then the InfoNCE loss in Eq. (1) can be rewritten in another form:

$$\begin{aligned} \mathcal{L}_{Info} = &- q(z \,|\, \hat{z}) \log p(z \,|\, \hat{z}) \\ &- \sum_{i=1}^{M} q(m_i \,|\, \hat{z}) \log p(m_i \,|\, \hat{z}), \end{aligned} \tag{7}$$

where $q(z \,|\, \hat{z})$ and $q(m_i \,|\, \hat{z})$ are the ideal distributions, $p(z \,|\, \hat{z})$ and $p(m_i \,|\, \hat{z})$ are the distributions learned by the network. To avoid the unknown ideal distribution exploration, InfoNCE loss [40] regards $\hat{z}$ as a one-hot distribution, where positive pairs have $q(z \,|\, \hat{z}) = 1$ and negative pairs satisfy $q(m_i \,|\, \hat{z}) = 0 (i \in [1, M])$. It means that InfoNCE loss maximizes the agreement of positive samples while minimizing the agreement of negative samples.

To explore the novel movement patterns from the extreme augmentations, a straightforward approach is directly using the extremely augmented sequence as a query and the normally augmented sequence as the key in InfoNCE loss. However, compared to the normally augmented sequence, the extremely augmented sequence may not keep the identity of the original sequence due to the dramatic changes in movement patterns, leading to performance degradation.

On the other hand, it is almost impossible to obtain the ideal likelihood distribution. Fortunately, CLSA [41] found that the normally augmented query and the extremely augmented query share similar distribution for a randomly initialized network. It inspires us that the distribution of normally augmented query over the memory bank can be used to supervise that of the extremely augmented query. It avoids directly using one-hot distribution for extremely augmented sequences and is able to explore the novel patterns exposed by the extreme augmentations.

Similar to Eqs. (5) and (6), we obtain conditional distributions for $\tilde{z}$ based on its positive samples and negative samples $p(z \,|\, \tilde{z})$ and $p(m_i \,|\, \tilde{z})$. Then, we propose to minimize the following distributional divergence between the normally augmented query $\hat{z}$ and the extremely augmented query $\tilde{z}$ such that:

$$\begin{aligned} \mathcal{L}_{ddm} = &- p(z \,|\, \hat{z}) \log p(z \,|\, \tilde{z}) \\ &- \sum_{i=1}^{M} p(m_i \,|\, \hat{z}) \log p(m_i \,|\, \tilde{z}). \end{aligned} \tag{8}$$

The distribution $p(z \,|\, \hat{z})$ is a much better supervision of $p(z \,|\, \tilde{z})$ instead of the one-hot distribution, which is also verified in the experimental part. This will avoid the problem of performance degradation caused by directly using InfoNCE loss for extremely augmented query $\tilde{z}$, and improve the generalization of the learned representation.

3.4. Single-stream nearest neighbors mining

In MoCo [14], two different augmented samples from the same sample are considered as positive pairs, while samples in the memory bank are all treated as negative samples. However, the samples in the memory bank are not necessarily all negative samples [16]. Therefore, we hope that the nearest neighbors of query $\hat{z}$ and $\tilde{z}$ over the memory bank $\mathbf{M}$ should be considered as positive samples to expand the positive set (see Fig. 4).

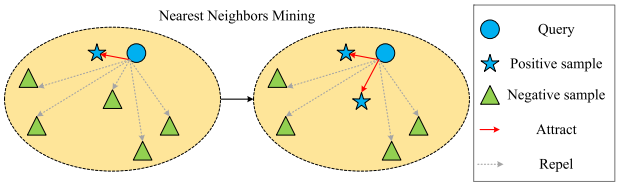


Fig. 4. Illustration of the proposed single-stream nearest neighbors mining. In the traditional contrastive learning framework, the goal is to pull the positive pairs closer and push the negative samples away. The idea of the proposed single-stream nearest neighbors mining is to make several negative samples that are similar to query be positive samples.

Specifically, N_+^n is the index set of the nearest top-k_1 neighbors that are most similar to the normally augmented query $\hat{z}$ in the memory bank $\mathbf{M}$. Similarly, we could also have the N_+^e to represent the index set of the nearest top-k_1 neighbors of the extremely augmented query $\tilde{z}$. Thus, we can set the nearest top-k_1 neighbors as positive samples to make the learning process more reasonable:

$$\begin{aligned} \mathcal{L}_{snnm} = &- \log \frac{\exp(\hat{z} \cdot z/\tau)}{\exp(\hat{z} \cdot z/\tau) + \sum_{i=1}^{M} \exp(\hat{z} \cdot m_i/\tau)} \\ &- \sum_{j \in N_+} \log \frac{\exp(\hat{z} \cdot m_j/\tau)}{\exp(\hat{z} \cdot z/\tau) + \sum_{i=1}^{M} \exp(\hat{z} \cdot m_i/\tau)}. \end{aligned} \tag{9}$$

In this equation, $N_+ = N_+^n \cup N_+^e$. Compared to Eq. (1), Eq. (9) will lead to a more regular space by pulling close more high-confidence positive samples.

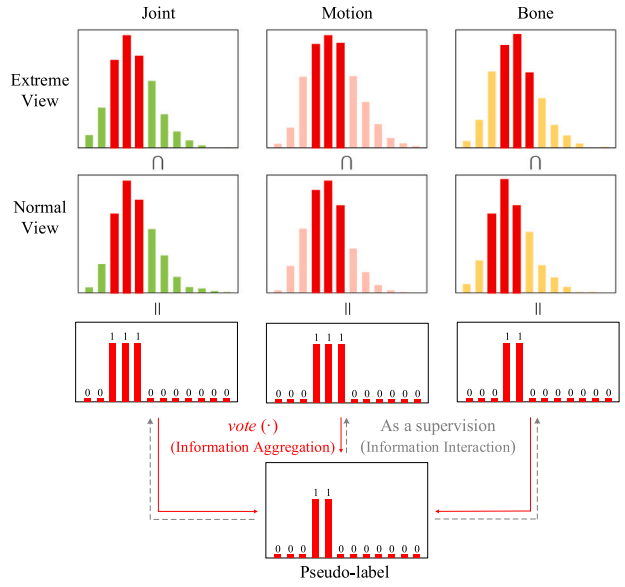


Fig. 5. Illustration of the proposed multi-stream aggregation and interaction. The strategy is proposed to fuse the information of joint, motion, and bone to obtain a pseudo-label representing positive or negative. Then the pseudo-label is used as a supervision to supervise each single-stream feature distribution.

4. 3s-AimCLR++

Existing methods [18,22] usually directly ensemble the results of the three streams (i.e., joint, motion and bone) at the decision-level to obtain the final result. We hope to make better use of the complementarity of the three-stream information, thus, we propose Multi-Stream Aggregation and Interaction (MSAI) strategy to achieve 3s-AimCLR++. As shown in Fig. 1, we first hold AimCLR++ for each single-stream which is described in Section 3. Then we use the Multi-Stream Aggregation and Interaction (MSAI) strategy to fuse multi-stream information to build a better feature space.

4.1. Multi-stream aggregation and interaction

As shown in Fig. 5, on the basis of Single-Stream Nearest Neighbors Mining, we could calculate N_+^n and N_+^e with nearest top-k_2 neighbors for each single-stream. Then we calculate $N_+ = N_+^n \cap N_+^e$ for single-stream to preserve high-confidence results to obtain N_+^j, N_+^m and N_+^b for joint, motion and bone stream, respectively. For multi-stream, $N_+^{3s} = vote(N_+^j, N_+^m, N_+^b)$. Here, $vote(\cdot)$ means when there are more than two streams that think the element should be 1, it will be 1, otherwise, it will be 0. Thus, the pseudo-label is obtained.

Then the pseudo-label is utilized to set the nearest neighbors as positive samples. Like Eq. (9), we calculate $\mathcal{L}_{msai}$ for each single-stream. It can better fuse the multi-stream aggregation and interaction to make the learning process more reasonable.

4.2. Training loss function

In the early training stage, the model is not stable and strong enough to provide reliable nearest neighbors and do information interaction. Thus, we use different loss functions in different training stages for 3s-AimCLR++:

$$\begin{cases} \mathcal{L}_1 = \sum (\alpha \mathcal{L}_{Info} + \beta \mathcal{L}_{ddm}) \\ \mathcal{L}_2 = \sum (\alpha \mathcal{L}_{snnm} + \beta \mathcal{L}_{ddm}) \\ \mathcal{L}_3 = \sum (\alpha \mathcal{L}_{msai} + \beta \mathcal{L}_{ddm}) \end{cases}, \tag{10}$$

where $\sum$ means adding up the losses of the three streams. In the first stage, the model of the three-stream is trained with the loss function $\mathcal{L}_1$. Then in the second training stage, the loss function is $\mathcal{L}_2$ to start mining the single-stream nearest neighbors. Then in the third training stage, the loss function is $\mathcal{L}_3$ to do the multi-stream information aggregation and interaction. Here, α and β are the coefficients to balance the loss. Though other values may achieve better results, we use $\alpha = \beta = 1$ to make 3s-AimCLR++ more general.

5. Experiments and discussions

5.1. Dataset

NTU RGB+D 60 Dataset [42]: It contains 56,578 action sequences and 60 action classes. There are two evaluation protocols: cross-subject (xsub) and cross-view (xview). In xsub, half of the subjects are used as training sets, and the rest are used as testing sets. In xview, the sequences of camera 2 and 3 are used for training while the sequences of camera 1 are used for testing.

NTU RGB+D 120 Dataset [43]: It is NTU RGB+D 60 based extension, whose scale is up to 120 action classes and 113,945 sequences. There are two evaluation protocols: cross-subject (xsub) and cross-setup (xset). In xsub, actions performed by 53 subjects are for training and the others are for testing. In xset, all 32 setups are separated as half for training and the other half for testing.

PKU-MMD Dataset [44]: It contains almost 20,000 action sequences covering 51 action classes. It consists of two subsets. Part I is an easier version for action recognition, while part II is more challenging with more noise caused by view variation. We conduct experiments under the cross-subject protocol on the two subsets.

5.2. Experimental settings

All the experiments are conducted on the PyTorch framework. For data pre-processing, we follow SkeletonCLR and CrosSCLR [18] for fair comparisons.

Self-supervised Pretext training. For pretext parameter settings, we follow that in SkeletonCLR and CrosSCLR for fair comparisons. Specifically, ST-GCN [9] is adopted as the encoder, the batch size is 128, the feature dimension is 128, the size of the memory bank is 32 768, the momentum coefficient m is set to 0.999, and the temperature hyper-parameter τ is set to 0.07. For optimization, we use SGD with momentum (0.9) and weight decay (0.0001). The model is trained for 400 epochs with a learning rate of 0.1 (decreases to 0.01 at epoch 350). We set $k_1 = 4$ in the SNNM and $k_2 = 20$ in the MSAI. For the three-stage training strategy mentioned in Section 4.2, the encoder is trained with $\mathcal{L}_1$ in the first 150 epochs, and is trained with $\mathcal{L}_2$ in the 150–300 epochs while trained with $\mathcal{L}_3$ in the remaining epochs.

KNN Evaluation Protocol. We fix the trained encoder and the K-Nearest Neighbor (KNN) classifier is used on the features of the trained encoder. For the reported KNN results, $K = 20$.

Linear Evaluation Protocol. We append a classification head (a fully connected layer together with a softmax layer) after the encoder and train the network with the encoder fixed. An SGD with an initial learning rate of 3.0 is used for 100 epochs. The learning rate decreases to 0.3 at epoch 80. The batch size is set to 128.

Semi-Supervised Evaluation Protocol. The encoder is trained with unlabeled data to get the initialized weights, then finetune the whole network with only 1% or 10% labeled data. An SGD with an initial learning rate of 0.1 (decreases by 10 at epoch 80) is used. The whole network is also trained for 100 epochs with batch size set to 128, except for PKU-MMD part II under 1% semi-supervised evaluation protocol (set to 52 due to the limited data).

Finetune Evaluation Protocol. The encoder is trained with unlabeled data to get the initialized weights, then finetune the whole network with all labeled data. An SGD with an initial learning rate of 0.1 (decreases to 0.01 at epoch 80) is used to train the whole network for 100 epochs. The batch size is set to 128.

Table 1
Linear evaluation results on three datasets.

Method	Encoder	Classifier	NTU-60		NTU-120		PKU-MMD		Publication & Year
			xsub	xview	xsub	xset	PartI	Part II	
LongTGAN [12]	GRU	FC	39.1	48.1	–	–	67.7[a]	26.0[a]	AAAI'2018
MS^{2}L [19]	GRU	GRU	52.6	–	–	–	64.9	27.6	ACMMM'2020
P&C [13]	GRU	FC	50.7	61.1	41.7	42.7	–	–	CVPR'2020
PCRP [39]	GRU	FC	53.9	63.5	41.7	45.1	–	–	TMM'2021
AS-CAL [17]	LSTM	FC	58.5	64.8	48.6	49.2	–	–	Infomation Science'2021
Tanfous et al. [38]	GRU	FC	67.1	**76.3**	59.1	61.5	–	–	WACV'2022
3s-AimCLR++ (Ours)	GRU	FC	**71.1**	73.8	**61.5**	**66.4**	**85.9**	**14.1**	–
4s-MG-AL [45]	ST-GCN	FC	64.7	68.0	46.2	49.5	–	–	TCSVT'2022
Cheng et al. [34]	Transformer	FC	69.3	72.8	–	–	–	–	ICME'2021
MCAE [35]	MCAE	FC	65.6	74.7	52.8	54.7	–	–	NIPS'2021
CP-STN [20]	ST-GCN	FC	69.4	76.6	55.7	54.7	–	–	ACML'2021
ST-CL [36]	GCN	FC	68.1	69.4	54.2	55.6	–	–	TMM'2022
3s-CrosSCLR [18]	ST-GCN	FC	77.8	83.4	67.9	66.7	84.9[a]	21.2[a]	CVPR'2021
Yang et al. [22]	DGCNN	FC	75.2	83.1	–	–	–	–	ICCV'2021
ASAR [46]	ST-GCN	FC	75.6	80.7	62.5	62.8	83.5	38.8	ICIP'2023
Thoker et al. [37]	GRU+CNN	FC	76.3	85.2	67.1	67.9	80.9	36.0	ACMMM'2020
3s-AimCLR [23]	ST-GCN	FC	78.9	83.8	68.2	68.8	87.8	38.5	AAAI'2022
3s-HiCLR [47]	ST-GCN	FC	80.4	**85.5**	70.0	70.4	–	–	AAAI'2023
3s-AimCLR++ (Ours)	ST-GCN	FC	**80.9**	**85.4**	**70.1**	**71.2**	**90.4**	**41.2**	–

[a] Denotes the result is reproduced according to the provided code.

Table 2
Linear evaluation results compared with SkeletonCLR on NTU-60 and NTU-120 dataset. "Δ" represents the gain compared to SkeletonCLR using the same stream data. "3s" means three-stream.

Method	Stream	NTU-60 (%)				NTU-120 (%)			
		xsub		xview		xsub		xset	
		acc.	Δ	acc.	Δ	acc.	Δ	acc.	Δ
SkeletonCLR [18]	Joint	68.3		76.4		56.8		55.9	
AimCLR (Ours) [23]	Joint	74.3	+6.0	79.7	+3.3	63.4	+6.6	63.4	+7.5
AimCLR++ (Ours)	Joint	**77.2**	+8.9	**81.5**	+5.1	**65.5**	+8.7	**67.8**	+11.9
SkeletonCLR [18]	Motion	53.3		50.8		39.6		40.2	
AimCLR (Ours) [23]	Motion	66.8	+13.5	70.6	+19.8	57.3	+17.7	54.4	+14.2
AimCLR++ (Ours)	Motion	**69.2**	+15.9	**74.5**	+23.7	**58.1**	+18.5	**56.1**	+15.9
SkeletonCLR [18]	Bone	69.4		67.4		48.4		52.0	
AimCLR (Ours) [23]	Bone	73.2	+3.8	77.0	+9.6	62.9	+14.5	63.4	+11.4
AimCLR++ (Ours)	Bone	**76.6**	+7.2	**80.7**	+13.3	**65.3**	+16.9	**66.6**	+14.6
3s-SkeletonCLR [18]	Joint+motion+bone	75.0		79.8		60.7		62.6	
3s-CrosSCLR [18]	Joint+motion+bone	77.8	+2.8	83.4	+3.6	67.9	+7.2	66.7	+4.1
3s-AimCLR (Ours) [23]	Joint+motion+bone	78.9	+3.9	83.8	+4.0	68.2	+7.5	68.8	+6.2
3s-AimCLR++ (Ours)	Joint+motion+bone	**80.9**	+5.9	**85.4**	+5.6	**70.1**	+9.4	**71.2**	+8.6

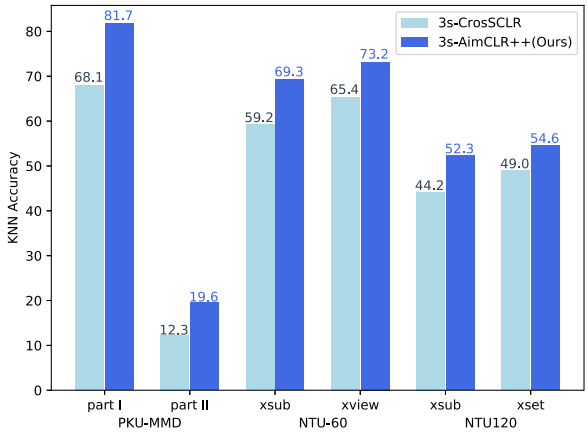


Fig. 6. Comparison of KNN accuracy of 3s-CrosSCLR [18] and our 3s-AimCLR++.

5.3. Comparison with state-of-the-art

We compare the proposed method with prior related methods under a variety of evaluation protocols.

KNN Evaluation Results. From Fig. 6, we can see that our 3s-AimCLR++ is better than 3s-CrosSCLR on the three datasets under the KNN evaluation protocol. The obvious gains of using a simple KNN classifier also show that the features learned by our 3s-AimCLR++ are more discriminative.

Linear Evaluation Results. As shown in Table 1, we compare the proposed 3s-AimCLR++ with other recent methods. For using GRU as the encoder, our 3s-AimCLR++ outperforms other methods on the three datasets under the linear evaluation protocol except on NTU-60 xview. For using stronger encoders, our 3s-AimCLR++ leads 3s-AimCLR 2.0% and 1.6% under the NTU-60 xsub and xview protocols, respectively. For the performance on NTU-120, our 3s-AimCLR++ also outperforms other methods by a large margin. This shows that our 3s-AimCLR++ is very competitive on multi-class large-scale datasets. For PKU-MMD dataset, Part II is more challenging with more skeleton noise caused by the view variation. Notably, 3s-CrosSCLR suffers on part II while our 3s-AimCLR and 3s-AimCLR++ perform well. It also proves that our methods have a strong ability to cope with movement patterns caused

Table 3
Semi-supervised evaluation results on PKU-MMD dataset and NTU-60 dataset.

Method	PKU-MMD (%)		NTU-60 (%)	
	Part I	Part II	xsub	xview
1% labeled data:				
LongTGAN (2018) [12]	35.8	12.4	35.2	–
MS^{2}L (2020) [19]	36.4	13.0	33.1	–
Thoker et al. (2021) [37]	37.7	–	35.7	38.1
AS-CAL (2021) [17]	–	–	47.2	53.5
3s-CrosSCLR (2021) [18]	49.7	10.2	51.1	50.0
Yang et al. (2021) [22]	–	–	48.3	52.5
3s-AimCLR (2022) [23]	57.5	15.1	54.8	54.3
3s-AimCLR++ (Ours)	**68.9**	**16.5**	**60.2**	**61.5**
10% labeled data:				
LongTGAN (2018) [12]	69.5	25.7	62.0	–
MS^{2}L (2020) [19]	70.3	26.1	65.2	–
AS-CAL (2021) [17]	–	–	52.2	57.3
Thoker et al. (2021) [37]	72.1	–	65.9	72.5
3s-CrosSCLR (2021) [18]	82.9	28.6	74.4	77.8
Yang et al. (2021) [22]	–	–	71.7	78.9
3s-AimCLR(Ours) (2022) [23]	86.1	33.4	**78.2**	81.6
3s-AimCLR++ (Ours)	**87.8**	**36.3**	77.4	**81.7**

Table 4
Finetuned results on NTU-60 and NTU-120 dataset.

Method	Params(M)	FLOPs(G)	NTU-60 (%)		NTU-120 (%)	
			xsub	xview	xsub	xset
ST-GCN (2018) [9]	0.84	0.57	82.8	90.8	77.3	78.3
AimCLR++ (Ours)	0.84	0.57	**84.5**	**91.0**	**78.1**	**79.1**
3s-ST-GCN (2018) [9]	2.55	1.72	85.2	91.4	77.2	77.1
3s-CrosSCLR (2021) [18]	2.55	1.72	86.2	92.5	80.5	80.4
3s-AimCLR (2022) [23]	2.55	1.72	86.9	92.8	80.1	80.9
3s-AimCLR++ (Ours)	2.55	1.72	**87.1**	**93.0**	**82.5**	**83.2**

Table 5
Exploration of different pretext datasets for fine-tuning on PKU-MMD dataset.

Pretext task dataset	PKU-MMD (%)	
	Part I	Part II
w/o pretext task	92.7	54.9
PKU-MMD	94.6 (+1.9)	58.7 (+3.8)
NTU-60 xsub	94.4 (+1.7)	63.4 (+8.5)
NTU-60 xview	94.5 (+1.8)	63.9 (+9.0)
NTU-120 xsub	94.8 (+2.1)	64.4 (+9.5)
NTU-120 xset	94.9 (+2.2)	64.1 (+9.2)

by skeleton noise. Based on the above, our 3s-AimCLR++ performs well on both large-scale and small-scale datasets, which demonstrates the effectiveness and generalization of our method.

Comparisons with SkeletonCLR and CrosSCLR [18]**.** We conduct experiments on NTU-60 and NTU-120 datasets to compare the performance of our method with baseline SkeletonCLR and CrosSCLR in detail. As can be seen from Table 2, for the three different streams of the two datasets, our AimCLR++ substantially improves SkeletonCLR, especially in motion and bone stream. We believe this is due to our introduction and better use of novel movement patterns, which have greatly improved the performance of these two streams. For the results of three streams, our 3s-AimCLR++ far exceeds 3s-SkeletonCLR. Compared to 3s-CrosSCLR, which uses a cross-view consistent knowledge mining strategy, we also achieve leading positions on both datasets.

Semi-supervised Evaluation Results. In some cases, there is very little labeled data that can be used, which makes it difficult to train the data-driven encoders. As shown in Table 3, with a small amount of data (i.e., 1%, 10%), our 3s-AimCLR++ achieves competitive results for all configurations. The results of 3s-AimCLR and 3s-AimCLR++ of using 1% and 10% labeled data far exceed other methods. It also proves that the novel movement patterns brought by extreme augmentations have a huge impact when there is only a small amount of labeled data.

Table 6
Exploration of different pretext datasets for fine-tuning on NTU-60 dataset.

Pretext task dataset	NTU-60 (%)	
	xsub	xview
w/o pretext task	85.2	91.4
NTU-60	87.1 (+1.9)	93.0 (+1.6)
NTU-120 xsub	87.8 (+2.6)	93.3 (+1.9)
NTU-120 xset	87.3 (+2.1)	93.3 (+1.9)

Table 7
Ablation study on NTU-60 dataset. EA: Extremely Augmentation. DDM: Distributional Divergence Minimization. SNNM: Single-Stream Nearest Neighbors Mining. MSAI: Multi-Stream Aggregation and Interaction.

Method	NTU-60 (%)	
	xsub	xview
3s-SkeletonCLR	75.0	79.8
+EA (InfoNCE)	70.8	78.8
+EA (DDM)	76.3	81.5
+EA (DDM) + SNNM	79.6	84.5
+EA (DDM) + MSAI	77.4	83.1
+EA (DDM) + SNNM + MSAI	**80.9**	**85.4**

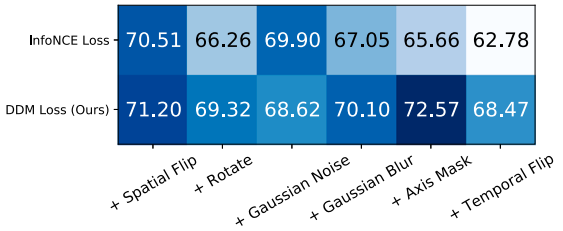


Fig. 7. Accuracy of adding a single augmentation for single-stream AimCLR++ on NTU-60 xsub.

Finetuned Evaluation Results. For fair comparisons, the ST-GCN used in the methods of Table 4 all have the same structure and parameters. For single-stream, the finetuned results of AimCLR++ outperform the supervised ST-GCN on these two datasets. For multi-stream, the finetuned 3s-AimCLR++ also outperforms the 3s-CrosSCLR and the supervised 3s-ST-GCN, indicating the effectiveness of our method.

Transfer Learning Results. As is common practice in the image fields, they perform self-supervised pre-training on the large scale ImageNet [48], then initialize the network with the learned weights and finally train on the small dataset. We also conduct experiments to do pretext tasks on a larger dataset without labels, then finetune on the small dataset. As shown in Table 5, when training from scratch, the accuracy of part I and part II is 92.7% and 54.9%. When we do the pretext task on the PKU-MMD dataset itself, we can gain the improvement of 1.9% and 3.8%. When applying the NTU-60 and NTU-120 datasets for self-supervised pre-training, we can significantly improve the accuracy by a large margin in part II, which illustrates the benefits of the transferability of learned representation. A similar trend can be seen for the transfer learning results on the NTU-60 dataset presented in Table 6. It illustrates our approach can significantly improve the accuracy of end-to-end training.

5.4. Ablation study

We conduct ablation studies on NTU-60 dataset to verify the effectiveness of different components of our method.

Effectiveness of Each Component. As shown in Table 7, 3s-SkeletonCLR [18] uses the normal augmentation and achieves the

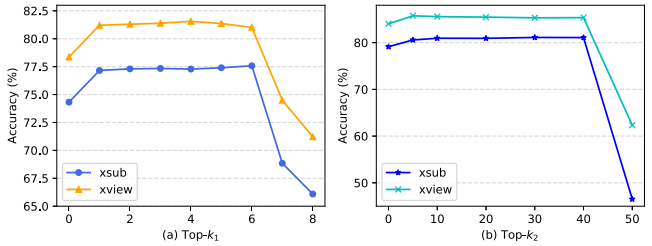


Fig. 8. Accuracy of different value of top-k_1 for single-stream AimCLR++ and different value of top-k_2 for 3s-AimCLR++.

Table 8
Accuracy of different combination of augmentations for single-stream AimCLR++ on NTU-60 xsub. NA: Normal Augmentation. top-x: it depends on the result of the second row of Fig. 7. i: Spatial Flip, ii: Rotate, iii: Gaussian Noise, iv: Gaussian Blur, v: Axis Mask, vi: Temporal Flip.

Extreme augmentation	DDM loss	InfoNCE loss
NA	70.91	70.66
NA + top-1(v)	72.57	65.66
NA + top-2(v + i)	72.95	68.86
NA + top-3(v + i + iv)	73.63	65.08
NA + top-4(v + i + iv + ii)	73.93	70.87
NA + top-5(v + i + iv + ii + iii)	**73.95**	4.81
NA + top-6(v + i + iv + ii + iii + vi)	73.40	4.88

Table 9
Results of AimCLR++ on NTU-60 with 2D/3D skeleton.

Evaluation protocol	NTU-60 (%)-joint			
	2D skeleton		3D skeleton	
	xsub	xview	xsub	xview
Linear evaluation	74.2	77.2	77.2	81.5
Semi-supervised evaluation (1%)	47.9	48.2	50.3	54.3
Semi-supervised evaluation (10%)	70.2	74.2	74.5	79.2
Supervised setting	80.6	87.8	82.8	90.8
Finetuned evaluation	82.8	89.2	84.5	91.0

Table 10
Results of AimCLR++ on NTU-60 joint stream with different backbone.

Backbone	Setting	NTU-60 (%)	
		xsub	xview
ST-GCN	Supervised setting	82.8	90.8
	AimCLR++ (Finetuned)	**84.5**	**91.0**
CTR-GCN	Supervised setting	83.9	90.9
	AimCLR++ (Finetuned)	**85.8**	**91.2**
UNIK	Supervised setting	82.3	90.1
	AimCLR++ (Finetuned)	**83.4**	**91.9**
DSTA-Net	Supervised setting	75.5	86.7
	AimCLR++ (Finetuned)	**80.3**	**87.3**

accuracy of 75.0% and 79.8% on xsub and xview respectively. While directly using InfoNCE loss on the extremely augmented query, the accuracies drop on both xsub and xview. It also illustrates that directly using InfoNCE loss for the extremely augmented query may not necessarily be able to boost the performance due to the drastic changes in original identity. However, while using DDM loss, the accuracy is improved by 1.3% and 1.7%. Further, when SNNM is introduced, the accuracy on xsub and xview improves by 3.3% and 3.0%, respectively. Notably, MSAI is also effective to bring about 1.1% and 1.6% gain. Our 3s-AimCLR++ achieves the highest accuracy when using all components. It also shows that the proposed components can make the encoder learn more suitable features.

Effectiveness of the Combinations of Extreme Augmentations. In Section 3.2, we define that extreme augmentation can introduce more novel movement patterns than normal augmentation. Therefore, we first explore the performance of the AimCLR++ when the extreme augmentation is normal augmentation plus a single data augmentation. Fig. 7 shows the accuracy of adding a single augmentation for single-stream AimCLR++ on NTU-60 xsub. When the extreme augmentation is normal augmentation plus a single data augmentation, "*Axis Mask*" and "*Spatial Flip*" are relatively more effective. We can see that using DDM loss generally outperforms direct using InfoNCE loss, which demonstrates the effectiveness of the proposed DDM loss.

On the basis of adding a single augmentation, we also explored the combination of different augmentations. Based on the results in Fig. 7, we continue to superimpose augmentations to form extreme augmentation in Table 8. It can be seen that our DDM loss always performs well, while direct using InfoNCE loss generally causes performance degradation. It is worth mentioning that when the augmentations are too extreme, it is even difficult to converge for InfoNCE loss. It illustrates the robustness of our method to the combinations of different augmentations. It is a general framework in which extreme augmentation can introduce more novel movement patterns than normal augmentation.

Robustness to 2D and 3D Skeletons. In order to verify whether our method is effective on the 2D skeletons, we changed the 3D skeleton of the NTU-60 dataset to 2D, and the experimental results are shown in Table 9. It can be seen that using 2D data does not perform as well as using 3D data, because 3D data contains depth information compared to 2D data. Nonetheless, our method also achieves competitive results under multiple evaluation protocols when taking 2D skeletons as input.

Robustness to Different Backbones. ST-GCN [9] is widely used as a backbone in the field of skeleton-based action recognition. To verify the robustness of our method to different backbones, we explore the performance of our method when using a more advanced backbone. We tried CTR-GCN [49] (GCN-based), UNIK [50] (CNN-based) and DSTA-Net [51] (Transformer-based). As shown in Table 10, under the same experimental setting, our method can improve the performance of end-to-end training.

Influence of Top-k_1. From Fig. 8(a), $k_1 = 0$ means that do not use the Single-Stream Nearest Neighbors Mining (SNNM). It is worth mentioning that SNNM brings clear gains when $1 \leq k_1 \leq 6$. This is because SNNM expands positive samples to make the learning process more reasonable. When k_1 continues to increase, it has a bad impact on performance due to the introduction of too many low-confidence positive samples. Finally, we take k_1 to 4 according to performance.

Influence of Top-k_2. From Fig. 8(b), $k_2 = 0$ means that do not use the Multi-Stream Aggregation and Interaction (MSAI) strategy. It is worth mentioning that MSAI also brings clear gains when $0 < k_2 \leq 40$. This is because MSAI retains high confidence samples through "$\cap$" to learn better feature representations. Finally, we take k_2 to 20 according to performance.

Influence of the Size of Memory Bank. We also conducted experiments to verify the performance of different sizes of memory bank. As shown in Table 11(a), increasing the size of the memory bank will bring about an increase in performance, but at the same time it will bring about an increase in the amount of computation. To be consistent with the SkeletonCLR and CrosSCLR, the size of the memory bank is set to 32 768.

Influence of the Batch Size. Due to the limitation of computing power, we conducted experiments with batch settings of 32, 64, and 128. In SkeletonCLR and CrosSCLR, the batch size is set to 128. As shown in Table 11(b), the best results are also achieved when the batch size is set to 128.

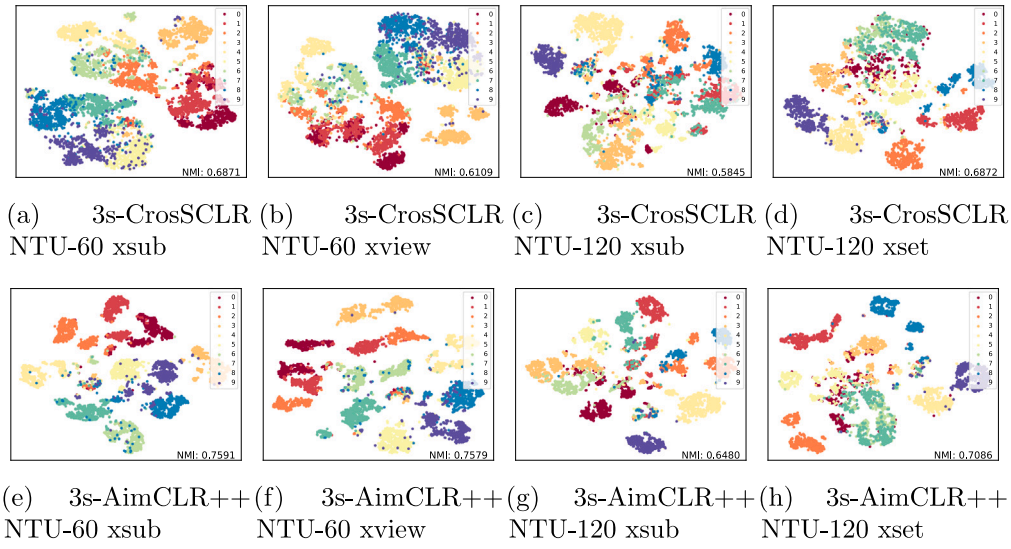


Fig. 9. The t-SNE visualization on NTU-60 and NTU-120 datasets. The reported t-SNE results are fair comparisons with the same randomly selected 10 class samples. NMI stands for Normalized Mutual Information, which is usually used as a quantitative indicator to measure the clustering effect.

Table 11
Performance of different size of memory bank and mini-batch.

(a)			(b)		
Memory bank	NTU-60 (%)		Batch size	NTU-60 (%)	
	xsub	xview		xsub	xview
16 384	80.3	84.9	32	79.7	84.7
32 768	**80.9**	85.3	64	80.7	85.3
65 536	**80.9**	**85.4**	**128**	**80.9**	**85.4**

Qualitative and Quantitative Results. T-SNE is applied with fix settings to show the embedding distribution in Fig. 9. The reported t-SNE results are fair comparisons with the same randomly selected 10 class samples and we also calculated the NMI (Normalized Mutual Information). From the results, we can draw a conclusion that 3s-AimCLR++ could cluster the embeddings of the same class closer than 3s-CrosSCLR. For the NMI, 3s-AimCLR++ leads 3s-CrosSCLR by a large margin on both NTU-60 and NTU-120 datasets. The huge gains under multiple evaluation protocols also corroborate the authenticity of the t-SNE results.

6. Conclusion

In this paper, 3s-AimCLR++ is proposed to explore the novel movement patterns brought by extreme augmentation for single-stream contrastive learning and effectively interact information for multi-stream. Specifically, for single-stream, extreme augmentation is proposed to bring novel movement patterns to improve the universality of the learned representations. The DDM loss is proposed to minimize the distributional divergence to use extreme augmentation in a more gentle way. In order to alleviate the irrationality of the positive set, the single-stream nearest neighbors mining strategy is further proposed to make the learning process more reasonable. For multi-stream, multi-stream aggregation and interaction strategy is proposed to better fuse the information. Experiments show that 3s-AimCLR++ significantly performs favorably against state-of-the-art methods under a variety of evaluation protocols. However, the limitation of our method is that using three streams will significantly increase computational complexity. In the future, we will explore how to reduce computational complexity and explore the application of the Siamese network without negative samples in this field.

CRediT authorship contribution statement

Tianyu Guo: Software, Writing – original draft. **Mengyuan Liu:** Conceptualization, Supervision, Writing – review & editing. **Hong Liu:** Project administration, Writing – review & editing. **Guoquan Wang:** Writing – review & editing. **Wenhao Li:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62203476), Natural Science Foundation of Shenzhen (No. JCYJ20230807120801002).

References

[1] L.L. Presti, M. La Cascia, 3D skeleton-based human action classification: A survey, Pattern Recognit. 53 (2016) 130–147.
[2] S. Du, Z. Yuan, P. Lai, T. Ikenaga, JoyPose: Jointly learning evolutionary data augmentation and anatomy-aware global–local representation for 3D human pose estimation, Pattern Recognit. 147 (2023) 110116.
[3] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: CVPR, 2012, pp. 1290–1297.
[4] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3D skeletons as points in a lie group, in: CVPR, 2014, pp. 588–595.
[5] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: CVPR, 2015, pp. 1110–1118.
[6] S. Song, C. Lan, J. Xing, W. Zeng, J. Liu, Spatio-temporal attention-based LSTM networks for 3D action recognition and detection, TIP 27 (7) (2018) 3459–3471.
[7] Y. Du, Y. Fu, L. Wang, Skeleton based action recognition with convolutional neural network, in: ACPR, 2015, pp. 579–583.

[8] M. Liu, H. Liu, C. Chen, Enhanced skeleton visualization for view invariant human action recognition, Pattern Recognit. 68 (2017) 346–362.
[9] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: AAAI, Vol. 32, 2018, (1).
[10] M. Dai, Z. Sun, T. Wang, J. Feng, K. Jia, Global spatio-temporal synergistic topology learning for skeleton-based action recognition, Pattern Recognit. 140 (2023) 109540.
[11] W. Peng, X. Hong, G. Zhao, Tripool: Graph triplet pooling for 3D skeleton-based action recognition, Pattern Recognit. 115 (2021) 107921.
[12] N. Zheng, J. Wen, R. Liu, L. Long, J. Dai, Z. Gong, Unsupervised representation learning with long-term dynamics for skeleton based action recognition, in: AAAI, Vol. 32, 2018, (1).
[13] K. Su, X. Liu, E. Shlizerman, Predict & cluster: Unsupervised skeleton based action recognition, in: CVPR, 2020, pp. 9631–9640.
[14] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: CVPR, 2020, pp. 9729–9738.
[15] X. Chen, H. Fan, R. Girshick, K. He, Improved baselines with momentum contrastive learning, 2020, arXiv preprint arXiv:2003.04297.
[16] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, A. Zisserman, With a little help from my friends: Nearest-neighbor contrastive learning of visual representations, in: ICCV, 2021, pp. 9588–9597.
[17] H. Rao, S. Xu, X. Hu, J. Cheng, B. Hu, Augmented skeleton based contrastive action learning with momentum LSTM for unsupervised action recognition, Inform. Sci. 569 (2021) 90–109.
[18] L. Li, M. Wang, B. Ni, H. Wang, J. Yang, W. Zhang, 3D human action representation learning via cross-view consistency pursuit, in: CVPR, 2021, pp. 4741–4750.
[19] L. Lin, S. Song, W. Yang, J. Liu, MS^{2}L: Multi-task self-supervised learning for skeleton based action recognition, in: ACMMM, 2020, pp. 2490–2498.
[20] Y. Zhan, Y. Chen, P. Ren, H. Sun, J. Wang, Q. Qi, J. Liao, Spatial temporal enhanced contrastive and pretext learning for skeleton-based action representation, in: ACML, 2021, pp. 534–547.
[21] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, P. Isola, What makes for good views for contrastive learning? NeurIPS 33 (2020) 6827–6839.
[22] S. Yang, J. Liu, S. Lu, M.H. Er, A.C. Kot, Skeleton cloud colorization for unsupervised 3D action representation learning, in: ICCV, 2021, pp. 13423–13433.
[23] T. Guo, H. Liu, Z. Chen, M. Liu, T. Wang, R. Ding, Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition, in: AAAI, 2022.
[24] L. Shi, Y. Zhang, J. Cheng, H. Lu, Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in: CVPR, 2019, pp. 12026–12035.
[25] C. Feichtenhofer, H. Fan, B. Xiong, R. Girshick, K. He, A large-scale study on unsupervised spatiotemporal representation learning, in: CVPR, 2021, pp. 3299–3309.
[26] S. Jenni, H. Jin, Time-equivariant contrastive video representation learning, in: ICCV, 2021, pp. 9970–9980.
[27] I.R. Dave, M.N. Rizve, C. Chen, M. Shah, TimeBalance: Temporally-invariant and temporally-distinctive video representations for semi-supervised action recognition, in: CVPR, 2023, pp. 2341–2352.
[28] S. Jenni, G. Meishvili, P. Favaro, Video representation learning by recognizing temporal transformations, in: ECCV, 2020, pp. 425–442.
[29] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: ICML, 2020, pp. 1597–1607.
[30] A. Bardes, J. Ponce, Y. LeCun, VICReg: Variance-invariance-covariance regularization for self-supervised learning, ICLR (2022).
[31] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al., Bootstrap your own latent-a new approach to self-supervised learning, NeurIPS 33 (2020) 21271–21284.
[32] X. Chen, K. He, Exploring simple siamese representation learning, in: CVPR, 2021, pp. 15750–15758.
[33] C. Feichtenhofer, Y. Li, K. He, et al., Masked autoencoders as spatiotemporal learners, NeurIPS 35 (2022) 35946–35958.
[34] Y.-B. Cheng, X. Chen, J. Chen, P. Wei, D. Zhang, L. Lin, Hierarchical transformer: Unsupervised representation learning for skeleton-based human action recognition, in: ICME, 2021, pp. 1–6.
[35] Z. Xu, X. Shen, Y. Wong, M.S. Kankanhalli, Unsupervised motion representation learning with capsule autoencoders, NeurIPS 34 (2021).
[36] X. Gao, Y. Yang, Y. Zhang, M. Li, J.-G. Yu, S. Du, Efficient spatio-temporal contrastive learning for skeleton-based 3D action recognition, TMM (2021).
[37] F.M. Thoker, H. Doughty, C.G. Snoek, Skeleton-contrastive 3D action representation learning, in: ACMMM, 2021, pp. 1655–1663.
[38] A.B. Tanfous, A. Zerroug, D. Linsley, T. Serre, How and what to learn: Taxonomizing self-supervised learning for 3D action recognition, in: WACV, 2022, pp. 2888–2897.
[39] S. Xu, H. Rao, X. Hu, J. Cheng, B. Hu, Prototypical contrast and reverse prediction: Unsupervised skeleton based action recognition, TMM (2021).
[40] A.v.d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2018, arXiv preprint arXiv:1807.03748.
[41] X. Wang, G.-J. Qi, Contrastive learning with stronger augmentations, 2021, arXiv preprint arXiv:2104.07713.
[42] A. Shahroudy, J. Liu, T.-T. Ng, G. Wang, NTU RGB + D: A large scale dataset for 3D human activity analysis, in: CVPR, 2016, pp. 1010–1019.
[43] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, A.C. Kot, NTU RGB + D 120: A large-scale benchmark for 3D human activity understanding, TPAMI 42 (10) (2019) 2684–2701.
[44] J. Liu, S. Song, C. Liu, Y. Li, Y. Hu, A benchmark dataset and comparison study for multi-modal human action analytics, TOMM 16 (2) (2020) 1–24.
[45] Y. Yang, G. Liu, X. Gao, Motion guided attention learning for self-supervised 3D human action recognition, TCSVT (2022).
[46] G. Wang, H. Liu, T. Guo, J. Guo, T. Wang, Y. Li, Self-supervised 3D skeleton representation learning with active sampling and adaptive relabeling for action recognition, in: ICIP, 2023, pp. 56–60.
[47] J. Zhang, L. Lin, J. Liu, Hierarchical consistent contrastive learning for skeleton-based action recognition with growing augmentations, in: AAAI, Vol. 37, 2023, pp. 3427–3435, (3).
[48] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, IJCV 115 (3) (2015) 211–252.
[49] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, W. Hu, Channel-wise topology refinement graph convolution for skeleton-based action recognition, in: ICCV, 2021, pp. 13359–13368.
[50] D. Yang, Y. Wang, A. Dantcheva, L. Garattoni, G. Francesca, F. Bremond, UNIK: A unified framework for real-world skeleton-based action recognition, 2021, arXiv preprint arXiv:2107.08580.
[51] L. Shi, Y. Zhang, J. Cheng, H. Lu, Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition, in: ACCV, 2020.



Tianyu Guo received the B.S. degree in electronic and information engineering in 2020. He is currently a graduate student in the School of Electronic and Computer Engineering, Peking University (PKU), China, under the supervision of Prof. H. Liu. His research interests include computer vision, machine learning, and action recognition.



Mengyuan Liu is an Assistant Professor at Peking University. He received a Ph.D. degree in 2017 under the supervision of Prof. Hong Liu from the School of EE&CS, Peking University (PKU), China. His research interests include human action recognition, human motion prediction, and human motion generation using RGB, depth, and skeleton data. Related methods have been published in T-IP, T-CSVT, T-MM, PR, CVPR, ECCV, ACM MM, AAAI, and IJCAI. He has been invited to be a Technical Program Committee (TPC) member for ACM MM and AAAI.



Hong Liu received the Ph.D. degree in Mechanical Electronics and Automation in 1996, and serves as a Full Professor in the School of EE&CS, Peking University (PKU), China. Prof. Liu has been selected as Chinese Innovation Leading Talent supported by "National High-level Talents Special Support Plan" since 2013. His research interests include computer vision, pattern recognition and robot motion planning. He is also reviewers for many international journals such as Pattern Recognition, IEEE Trans. on Signal Processing, and IEEE Trans. on PAMI.



Guoquan Wang received the B.S. degree from Hefei University of Technology (HFUT), China. He is a research graduate student studying at Peking University (PKU), China. His research interest lies in human action recognition and contrastive learning.



Wenhao Li received the B.E. degree in automation in 2019. He is currently working toward the Ph.D. degree in the School of Electronics Engineering and Computer Science, Peking University (PKU), China, under the supervision of Prof. Hong Liu. His research interests lie in deep learning, machine learning, and their applications to computer vision.