Actionlet-Dependent Contrastive Learning for Unsupervised Skeleton-Based Action Recognition

Lilang Lin, Jiahang Zhang, Jiaying Liu*
Wangxuan Institute of Computer Technology, Peking University, Beijing, China

Abstract

The self-supervised pretraining paradigm has achieved great success in skeleton-based action recognition. However, these methods treat the motion and static parts equally, and lack an adaptive design for different parts, which has a negative impact on the accuracy of action recognition. To realize the adaptive action modeling of both parts, we propose an ***Act****ionlet-Dependent* ***C****ontrastive* ***L****ea****r****ning method (ActCLR). The actionlet, defined as the discriminative subset of the human skeleton, effectively decomposes motion regions for better action modeling. In detail, by contrasting with the static anchor without motion, we extract the motion region of the skeleton data, which serves as the actionlet, in an unsupervised manner. Then, centering on actionlet, a motion-adaptive data transformation method is built. Different data transformations are applied to actionlet and non-actionlet regions to introduce more diversity while maintaining their own characteristics. Meanwhile, we propose a semantic-aware feature pooling method to build feature representations among motion and static regions in a distinguished manner. Extensive experiments on NTU RGB+D and PKUMMD show that the proposed method achieves remarkable action recognition performance. More visualization and quantitative experiments demonstrate the effectiveness of our method. Our project website is available at* `https://langlandslin.github.io/projects/ActCLR/`

1. Introduction

Skeletons represent human joints using 3D coordinate locations. Compared with RGB videos and depth data, skeletons are lightweight, privacy-preserving, and compact to represent human motion. On account of being easier and more discriminative for analysis, skeletons have been widely used in action recognition task [19,23,31,32,46,48].

Supervised skeleton-based action recognition methods [3,27,28] have achieved impressive performance. However, their success highly depends on a large amount of labeled training data, which is expensive to obtain. To get rid of the reliance on full supervision, self-supervised learning [16, 32, 34, 49] has been introduced into skeleton-based action recognition. It adopts a two-stage paradigm, *i.e.* first applying pretext tasks for unsupervised pretraining and then employing downstream tasks for finetuning.

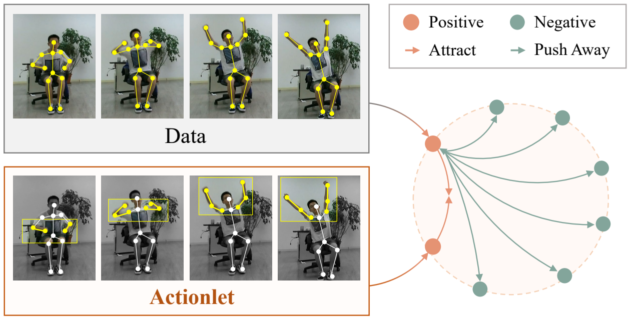

Figure 1. Our proposed approach (ActCLR) locates the motion regions as actionlet to guide contrastive learning.

According to learning paradigms, all methods can be classified into two categories: reconstruction-based [14, 32, 41] and contrastive learning-based. Reconstruction-based methods capture the spatial-temporal correlation by predicting masked skeleton data. Zheng *et al.* [49] first proposed reconstructing masked skeletons for long-term global motion dynamics. Besides, the contrastive learning-based methods have shown remarkable potential recently. These methods employ skeleton transformation to generate positive/negative samples. Rao *et al.* [24] applied Shear and Crop as data augmentation. Guo *et al.* [8] further proposed to use more augmentations, *i.e.* rotation, masking, and flipping, to improve the consistency of contrastive learning.

These contrastive learning works treat different regions of the skeleton sequences uniformly. However, the motion regions contain richer action information and contribute more to action modeling. Therefore, it is sub-optimal to directly apply data transformations to all regions in the previous works, which may degrade the motion-correlated infor-

*Corresponding author. This work is supported by the National Natural Science Foundation of China under contract No.62172020.

mation too much. For example, if the mask transformation is applied to the hand joints in the hand raising action, the motion information of the hand raising is totally impaired. It will give rise to the false positive problem, *i.e.*, the semantic inconsistency due to the information loss between positive pairs. Thus, it is necessary to adopt a distinguishable design for motion and static regions in the data sequences.

To tackle these problems, we propose a new actionlet-dependent contrastive learning method (ActCLR) by treating motion and static regions differently, as shown in Fig. 1. An *actionlet* [38] is defined as a conjunctive structure of skeleton joints. It is expected to be highly representative of one action and highly discriminative to distinguish the action from others. The actionlet in previous works is defined in a supervised way, which relies on action labels and has a gap with the self-supervised pretext tasks. To this end, in the unsupervised learning context, we propose to obtain actionlet by comparing the action sequence with the average motion to guide contrastive learning. In detail, the average motion is defined as the average of all the series in the dataset. Therefore, this average motion is employed as the static anchor without motion. We contrast the action sequence with the average motion to get the area with the largest difference. This region is considered to be the region where the motion takes place, *i.e.*, actionlet.

Based on this actionlet, we design a motion-adaptive transformation strategy. The actionlet region is transformed by performing the proposed semantically preserving data transformation. Specifically, we only apply stronger data transformations to non-actionlet regions. With less interference in the motion regions, this motion-adaptive transformation strategy makes the model learn better semantic consistency and obtain stronger generalization performance. Similarly, we utilize a semantic-aware feature pooling method. By extracting the features in the actionlet region, the features can be more representative of the motion without the interference of the semantics in static regions.

We provide thorough experiments and detailed analysis on NTU RGB+D [17, 26] and PKUMMD [18] datasets to prove the superiority of our method. Compared to the state-of-the-art methods, our model achieves remarkable results with self-supervised learning.

In summary, our contributions are summarized as follows:

- We propose a novel unsupervised actionlet-based contrastive learning method. Unsupervised actionlets are mined as skeletal regions that are the most discriminative compared with the static anchor, *i.e.*, the average motion of all training data.
- A motion-adaptive transformation strategy is designed for contrastive learning. In the actionlet region, we employ semantics-preserving data transformations to learn semantic consistency. And in non-actionlet regions, we apply stronger data transformations to obtain stronger generalization performance.
- We utilize semantic-aware feature pooling to extract motion features of the actionlet regions. It makes features to be more focused on motion joints without being distracted by motionless joints.

2. Related Work

In this section, we first introduce the related work of skeleton-based action recognition, and then briefly review contrastive learning.

2.1. Skeleton-Based Action Recognition

Skeleton-based action recognition is a fundamental yet challenging field in computer vision research. Previous skeleton-based motion recognition methods are usually realized with the geometric relationship of skeleton joints [7, 36, 37]. The latest methods pay more attention to deep networks. Du *et al.* [6] applied a hierarchical RNN to process body keypoints. Attention-based methods are proposed to automatically select important skeleton joints [28–30, 47] and video frames [29, 30] to learn more adaptively about the simultaneous appearance of skeleton joints. However, recurrent neural networks often suffer from gradient vanishing [11], which may cause optimization problems. Recently, graph convolution networks attract more attention for skeleton-based action recognition. To extract both the spatial and temporal structural features from skeleton data, Yan *et al.* [40] proposed spatial-temporal graph convolution networks. To make the graphic representation more flexible, the attention mechanisms are applied in [3, 27, 28] to adaptively capture discriminative features based on spatial composition and temporal dynamics.

2.2. Contrastive Learning

Contrastive representation learning can date back to [9]. The following approaches [1, 13, 35, 39, 42] learn representations by contrasting positive pairs against negative pairs to make the representations between positive pairs more similar than those between negative pairs. Researchers mainly focus on how to construct pairs to learn robust representations. SimCLR proposed by Chen *et al.* [2] uses a series of data augmentation methods, such as random cropping, Gaussian blur and color distortion to generate positive samples. He *et al.* [10] applied a memory module that adopts a queue to store negative samples, and the queue is constantly updated with training. In self-supervised skeleton-based action recognition, contrastive learning has also attracted the attention of numerous researchers. Rao *et al.* [24] applied MoCo for contrastive learning with a single stream. To utilize cross-stream knowledge, Li *et al.* [15] proposed a multiview contrastive learning method and Thoker *et al.* [34] employed multiple models to learn from different skeleton representations. Guo *et al.* [8] proposed to use more extreme augmentations, which greatly improve the effect of contrastive learning. Su *et al.* [33] proposed novel representation learning by perceiving motion consistency and continuity. Following MoCo v2 [10], they exploit InfoNCE loss to optimize contrastive learning:

$$\mathcal{L}_{\text{CL}} = -\log \frac{\exp(\text{sim}(\mathbf{z}_q^i, \mathbf{z}_k^i)/\tau)}{\exp(\text{sim}(\mathbf{z}_q^i, \mathbf{z}_k^i)/\tau) + K}, \tag{1}$$

where $\mathbf{z}_q^i = g_q(f_q(\mathbf{X}_q^i))$ and $\mathbf{z}_k^i = g_k(f_k(\mathbf{X}_k^i))$. $K = \sum_{j=1}^{M} \exp(\text{sim}(\mathbf{z}_q^i, \mathbf{m}^j)/\tau)$ and τ is a temperature hyperparameter. $f_q(\cdot)$ is an online encoder and $f_k(\cdot)$ is an offline encoder. $g_q(\cdot)$ is an online projector and $g_k(\cdot)$ is an offline projector. The offline encoder $f_k(\cdot)$ is updated by the momentum of the online encoder $f_q(\cdot)$ by $f_k \leftarrow \alpha f_k + (1-\alpha) f_q$, where α is a momentum coefficient. $\mathbf{m}^j$ is the negative sample, stored in memory bank $\mathbf{M}$. $\text{sim}(\cdot, \cdot)$ is the cosine similarity.

3. Actionlet-Based Unsupervised Learning

In this section, we introduce unsupervised actionlet for contrastive representation learning, which is based on MoCo v2 described in Sec. 2.2. First, we describe the unsupervised actionlet extraction method. Then, the motion-adaptive data transformation and the semantic-aware feature pooling are introduced.

3.1. Unsupervised Actionlet Selection

Traditional actionlet mining methods rely on the action label to identify the motion region, which cannot be employed in the unsupervised learning context. Inspired by contrastive learning, we propose an unsupervised spatio-temporal actionlet selection method to mine the motion region as shown in Fig. 2. The actionlet is obtained by comparing the differences between an action sequence and the static sequence where we assume no motion takes place.

Specifically, we introduce the average motion as the static anchor, which is regarded as the sequence without motion. Resort to this, we contrast the action sequences between the static anchor to realize actionlet localization. The details of the proposed method are described below.

Average Motion as Static Anchor. In the process of obtaining the sequence without action occurrence, we observe that most of the action sequences have no action in most of the regions. The motion usually occurs in a small localized area, such as the hand or head. Therefore, as shown in Fig. 4, we can easily obtain the static anchor via average all the actions in the dataset, since most of the sequence has no motion in most of the regions and this average is a relatively static sequence. It is formalized as:

$$\bar{\mathbf{X}} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{X}^i), \tag{2}$$

where $\mathbf{X}^i$ is the i^{th} skeleton sequence and N is the size of the dataset.

Difference Activation Mapping for Actionlet Localization. To obtain the region where the motion takes place, we input the skeleton sequence $\mathbf{X}^i$ with the average motion $\bar{\mathbf{X}}$ into the offline encoder $f_k(\cdot)$ to obtain its corresponding dense features $\mathbf{h}_{ctv}^i = f_k(\mathbf{X}^i)$ and $\bar{\mathbf{h}}_{ctv} = f_k(\bar{\mathbf{X}})$, where c means channel dimension, t temporal dimension, and v joint dimension. After global average pooling (GAP), we then apply the offline projector $g_k(\cdot)$ to obtain global features $\mathbf{z}^i = g_k(\text{GAP}(\mathbf{h}_{ctv}^i))$ and $\bar{\mathbf{z}} = g_k(\text{GAP}(\bar{\mathbf{h}}_{ctv}))$. Then we calculate the cosine similarity of these two features. This can be formalized as:

$$\text{sim}(\mathbf{z}^i, \bar{\mathbf{z}}) = \frac{\langle \mathbf{z}^i, \bar{\mathbf{z}} \rangle}{\|\mathbf{z}^i\|_2 \|\bar{\mathbf{z}}\|_2}, \tag{3}$$

where $\langle \cdot, \cdot \rangle$ is the inner product.

To find the region where this similarity can be reduced, we back-propagate and reverse the gradient of this similarity to the dense feature $\mathbf{h}_{ctv}^i$. These gradients then are global average pooled over the temporal and joint dimensions to obtain the neuron importance weights α_c^i:

$$\begin{aligned} \Delta \mathbf{h}_{ctv}^i &= \frac{\partial(-\text{sim}(\mathbf{z}^i, \bar{\mathbf{z}}))}{\partial \mathbf{h}_{ctv}^i}, \\ \alpha_c^i &= \frac{1}{T \times V} \sum_{t=1}^{T} \sum_{v=1}^{V} \sigma(\Delta \mathbf{h}_{ctv}^i), \end{aligned} \tag{4}$$

where $\sigma(\cdot)$ is the activation function.

These importance weights capture the magnitude of the effect of each channel dimension on the final difference. Therefore, these weights α_c^i are considered difference activation mapping. We perform a weighted combination of the difference activation mapping and dense features as follows:

$$\mathbf{A}_{tv}^i = \sigma \left(\sum_{c=1}^{C} \alpha_c^i \mathbf{h}_{ctv}^i \right) \mathbf{G}_{vv}, \tag{5}$$

where $\sigma(\cdot)$ is the activation function and $\mathbf{G}_{vv}$ is the adjacency matrix of skeleton data for importance smoothing. The linear combination of maps selects features that have a negative influence on the similarity. The actionlet region is the area where the value of the generated actionlet $\mathbf{A}_{tv}^i$ exceeds a certain threshold, while the non-actionlet region is the remaining part.

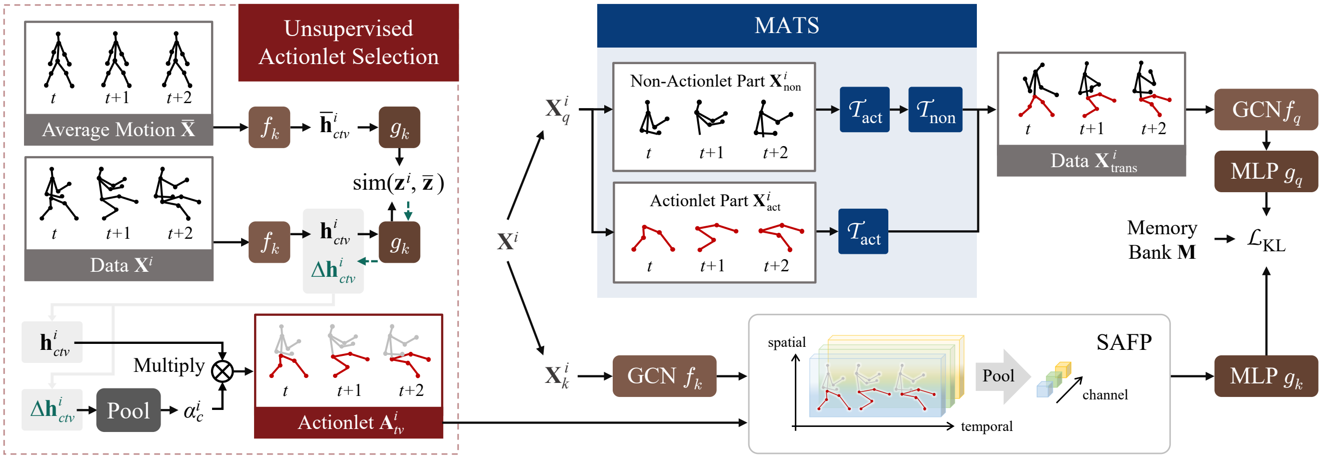
Unsupervised Actionlet Selection
t t+1 t+2
Average Motion $\bar{\mathbf{X}}$
f_k
$\bar{\mathbf{h}}^i_{ctv}$
g_k
sim($\mathbf{z}^i$, $\bar{\mathbf{z}}$)
Data $\mathbf{X}^i$
$\mathbf{h}^i_{ctv}$
$\Delta\mathbf{h}^i_{ctv}$
Multiply
Pool
α^i_c
Actionlet $\mathbf{A}^i_{tv}$
MATS
$\mathbf{X}^i_q$
Non-Actionlet Part $\mathbf{X}^i_{non}$
Actionlet Part $\mathbf{X}^i_{act}$
$\mathcal{T}_{act}$
$\mathcal{T}_{non}$
Data $\mathbf{X}^i_{trans}$
GCN f_q
MLP g_q
Memory Bank $\mathbf{M}$
$\mathcal{L}_{KL}$
$\mathbf{X}^i$
$\mathbf{X}^i_k$
GCN f_k
SAFP
spatial
temporal
channel
MLP g_k

Figure 2. The pipeline of actionlet-dependent contrastive learning. In unsupervised actionlet selection, we employ the difference from the average motion to obtain the region of motion. For contrastive learning, we employ two streams, *i.e.*, the online stream and the offline stream. The above stream is the online stream, which is updated by gradient. The below is the offline stream, which is updated by momentum. We get the augmented data $\mathbf{X}^i_{\text{trans}}$ by performing motion-adaptive data transformation (MATS) on the input data $\mathbf{X}^i_q$ with the obtained actionlet. In offline feature extraction, we employ semantic-aware feature pooling (SAFP) to obtain the accurate feature anchor. Finally, utilizing similarity mining, we increase the similarity between positives and decrease the similarity between negatives.

3.2. Actionlet-Guided Contrastive Learning

To take full advantage of the actionlet, we propose an actionlet-dependent contrastive learning method, shown in Fig. 2. We impose different data transformations for different regions by a motion-adaptive data transformation strategy module (MATS). Moreover, the semantic-aware feature pooling module (SAFP) is proposed to aggregate the features of actionlet region for better action modeling.

Motion-Adaptive Transformation Strategy (MATS). In contrastive learning, data transformation $\mathcal{T}$ is crucial for semantic information extraction and generalization capacity. How to design more diverse data transformations while maintaining relevant information for downstream tasks is still a challenge. Too simple data transformation is limited in numbers and modes and cannot obtain rich augmented patterns. However, data transformations that are too difficult may result in loss of motion information. To this end, we propose motion-adaptive data transformations for skeleton data based on actionlet. For different regions, we propose two transformations, actionlet transformation and non-actionlet transformation.

- **Actionlet Transformation $\mathcal{T}_{\text{act}}$**: Actionlet data transformations are performed within the actionlet regions. Inspired by the previous work [8], we adopt four spatial data transformations {Shear, Spatial Flip, Rotate, Axis Mask}; two temporal data transformations {Crop, Temporal Flip}; and two spatio-temporal data transformations {Gaussian Noise, Gaussian Blur}.

Besides, Skeleton AdaIN is proposed as a mixing method of global statistics. We randomly select two skeleton sequences and then swap the spatial mean and temporal variance of the two sequences. This transformation is widely used in style transfer [12]. Here, we are inspired by the idea of style and content decomposition in style transfer and regard the motion-independent information as style and the motion-related information as content. Therefore, we use Skeleton AdaIN to transfer this motion independent noise between different data. The noisy pattern of the data is thus augmented by this transfer method. This transformation can be formalized as:

$$\mathbf{X}^i_{\text{adain}} = \sigma(\mathbf{X}^j)\left(\frac{\mathbf{X}^i - \mu(\mathbf{X}^i)}{\sigma(\mathbf{X}^i)}\right) + \mu(\mathbf{X}^j), \qquad (6)$$

where $\sigma(\cdot)$ is the temporal variance, $\mu(\cdot)$ is the spatial mean, and $\mathbf{X}^j$ is a randomly selected sequence. All these data transformations maintain the action information.

- **Non-Actionlet Transformation $\mathcal{T}_{\text{non}}$**: To obtain stronger generalization, several extra data transformations are applied to the non-actionlet regions in addition to the above data transformation.

We apply an intra-instance data transformation {Random Noise} and an inter-instance data transformation {Skeleton Mix}. The random Noise has larger variance. Skeleton Mix is an element-wise data mixing method, including Mixup [44], CutMix [43], and ResizeMix [25]. Because these transformations are performed on non-actionlet regions, they do not change the action semantics. Therefore, the transformed data are used as positive samples with the original data.

- **Actionlet-Dependent Combination**: To merge the data transformations of the two regions, we utilize actionlets to combine. It is formalized as:

$$\mathbf{X}^i_{\text{trans}} = \mathbf{A}^i_{tv} \odot \mathbf{X}^i_{\text{act}} + (1 - \mathbf{A}^i_{tv}) \odot \mathbf{X}^i_{\text{non}}, \tag{7}$$

where $\mathbf{X}^i_{\text{trans}}$ is the final transformed data, $\mathbf{X}^i_{\text{act}}$ and $\mathbf{X}^i_{\text{non}}$ are transformed with actionlet transformations $\mathcal{T}_{\text{act}}$. Then, $\mathbf{X}^i_{\text{non}}$ is performed non-actionlet transformations $\mathcal{T}_{\text{non}}$. $\mathbf{A}^i_{tv}$ represents the actionlet.

Semantic-Aware Feature Pooling (SAFP). To extract the motion information more accurately, we propose a semantic-aware feature pooling method along the spatial-temporal dimension. This method focuses only on the feature representation of the actionlet region, thus reducing the interference of other static regions for motion feature extraction. It is formalized as:

$$\text{SAFP}(\mathbf{h}^i_{ctv}) = \sum_{t=1}^{T}\sum_{v=1}^{V} \mathbf{h}^i_{ctv} \left(\frac{\mathbf{A}^i_{tv}}{\sum_{t=1}^{T}\sum_{v=1}^{V} \mathbf{A}^i_{tv}} \right). \tag{8}$$

This semantic-aware feature aggregation approach effectively extracts motion information and makes the features more distinguishable. We utilize this semantic-aware feature pooling operation in the offline stream to provide accurate anchor features.

Training Overview. In this part, we conclude our framework of contrastive learning in detail:

1) Two encoders are pre-trained using MoCo v2 [10], an online encoder $f_q(\cdot)$ and an offline encoder $f_k(\cdot)$. The online encoder is updated via back-propagation gradients, while the offline encoder is a momentum-updated version of the online encoder as described in Sec. 2.2.

2) The offline network $f_k(\cdot)$ inputs the original data $\mathbf{X}^i$ and we employ the unsupervised actionlet selection module to generate actionlet regions $\mathbf{A}^i_{tv}$ in the offline stream in Sec. 3.1.

3) We perform data transformation $\mathcal{T}$ to obtain two different views $\mathbf{X}^i_q$ and $\mathbf{X}^i_k$. And we apply motion-adaptive transformation strategy (MATS) to enhance the diversity of $\mathbf{X}^i_q$ in Sec. 3.2.

4) For feature extraction, in online stream, $\mathbf{z}^i_q = (g_q \circ \text{GAP} \circ f_q \circ \text{MATS})(\mathbf{X}^i_q)$, where $g_q(\cdot)$ is an online projector and GAP is the global average pooling. To provide a stable and accurate anchor feature, we utilize the semantic-aware feature pooling (SAFP) method in Sec. 3.2 to generate offline features $\mathbf{z}^i_k = (g_k \circ \text{SAFP} \circ f_k)(\mathbf{X}^i_k)$, where $g_k(\cdot)$ is an offline projector.

5) A memory bank $\mathbf{M} = \{\mathbf{m}^i\}_{i=1}^{M}$ is utilized to store offline features. The offline features extracted from the offline data in each batch are stored in the memory bank, and the bank is continuously updated using a first-in first-out strategy.

6) Following recent works [20, 45], we exploit similarity mining to optimize:

$$\begin{aligned} \mathcal{L}_{\text{KL}}(\mathbf{p}^i_q, \mathbf{p}^i_k) &= -\mathbf{p}^i_k \log \mathbf{p}^i_q, \\ \mathbf{p}^i_q &= \text{SoftMax}(\text{sim}(\mathbf{z}^i_q, \mathbf{M})/\tau_q), \\ \mathbf{p}^i_k &= \text{SoftMax}(\text{sim}(\mathbf{z}^i_k, \mathbf{M})/\tau_k), \end{aligned} \tag{9}$$

where $\text{sim}(\mathbf{z}^i_q, \mathbf{M}) = [\text{sim}(\mathbf{z}^i_q, \mathbf{m}^j)]_{j=1}^{M}$, which indicates the similarity distribution between feature $\mathbf{z}^i_q$ and other samples in $\mathbf{M}$. For the elements $\mathbf{p}^{ij}_k$ of $\mathbf{p}^i_k$ greater than the elements $\mathbf{p}^{ij}_q$ of $\mathbf{p}^i_q$, these corresponding features $\mathbf{m}^j$ in the memory bank are positive samples. This is because the network increases the similarity of the output with these features.

4. Experiment Results

For evaluation, we conduct our experiments on the following two datasets: the NTU RGB+D dataset [17, 26] and the PKUMMD dataset [18].

4.1. Datasets and Settings

- **NTU RGB+D Dataset 60 (NTU 60)** [26] is a large-scale dataset which contains 56,578 videos with 60 action labels and 25 joints for each body, including interactions with pairs and individual activities.

- **NTU RGB+D Dataset 120 (NTU 120)** [17] is an extension to NTU 60 and the largest dataset for action recognition, which contains 114,480 videos with 120 action labels. Actions are captured with 106 subjects with multiple settings using 32 different setups.

- **PKU Multi-Modality Dataset (PKUMMD)** [18] covers a multi-modality 3D understanding of human actions. The actions are organized into 52 categories and include almost 20,000 instances. There are 25 joints in each sample. The PKUMMD is divided into part I and part II. Part II provides more challenging data, because the large view variation causes more skeleton noise.

To train the network, all the skeleton sequences are temporally down-sampled to 50 frames. The encoder $f(\cdot)$ is based on ST-GCN [40] with hidden channels of size 16, which is a quarter the size of the original model. The projection heads for contrastive learning and auxiliary tasks are all multilayer perceptrons, projecting features from 256 dimensions to 128 dimensions. τ_q is 0.1 and τ_k is 0.04. We employ a fully connected layer $\phi(\cdot)$ for evaluation.

To optimize our network, Adam optimizer [21] is applied, and we train the network on one NVIDIA TitanX GPU with a batch size of 128 for 300 epochs.

Table 1. Comparison of action recognition results with unsupervised learning approaches on NTU dataset.

Models	Stream	NTU 60 xview	NTU 60 xsub	NTU 120 xset	NTU 120 xsub
AimCLR [8]	joint	79.7	74.3	63.4	63.4
ActCLR	joint	**86.7**	**80.9**	**70.5**	**69.0**
AimCLR [8]	motion	70.6	66.8	54.4	57.3
ActCLR	motion	**84.4**	**78.6**	**67.8**	**68.3**
AimCLR [8]	bone	77.0	73.2	63.4	62.9
ActCLR	bone	**85.0**	**80.1**	**68.2**	**67.8**
3s-AimCLR [8]	joint+motion+bone	83.8	78.9	68.8	68.2
3s-ActCLR	joint+motion+bone	**88.8**	**84.3**	**75.7**	**74.3**

Table 2. Comparison of action recognition results with unsupervised learning approaches on NTU 60 dataset. † indicates that results reproduced on our settings of feature dimension size.

Models	Architecture	xview	xsub
Single-stream:			
LongT GAN [49]	GRU	48.1	39.1
MS^{2}L [16]	GRU	-	52.5
AS-CAL [24]	LSTM	64.8	58.5
P&C [32]	GRU	59.3	56.1
SeBiReNet [22]	SeBiReNet	79.7	-
ISC [34]	GCN & GRU	78.6	76.3
AimCLR [8]	GCN	79.7	74.3
CMD† [20]	GRU	81.3	76.8
GL-Transformer [14]	Transformer	83.8	76.3
CPM [45]	GCN	84.9	78.7
ActCLR	GCN	**86.7**	**80.9**
Three-stream:			
3s-Colorization [41]	DGCNN	83.1	75.2
3s-CrosSCLR [15]	GCN	83.4	77.8
3s-AimCLR [8]	GCN	83.8	78.9
3s-CMD† [20]	GRU	85.0	79.9
3s-SkeleMixCLR [5]	GCN	87.1	82.7
3s-CPM [45]	GCN	87.0	83.2
3s-ActCLR	GCN	**88.8**	**84.3**

4.2. Evaluation and Comparison

To make a comprehensive evaluation, we compare our method with other methods under variable settings.

1) Linear Evaluation. In the linear evaluation mechanism, a linear classifier $\phi(\cdot)$ is applied to the fixed encoder $f(\cdot)$ to classify the extracted features. We adopt action recognition accuracy as a measurement. Note that this encoder $f(\cdot)$ is fixed in the linear evaluation protocol.

Compared with other methods in Tables 1, 2 and 3, our model shows superiority on these datasets. We find that the transformation that 3s-CrosSCLR [15] and 3s-AimCLR [8] design in the contrastive learning task is unified for different regions, which makes the data transformation interfere with the motion information. On the contrary, our method adopts MATS for semantic-aware motion-adaptive data transformation. Thus, the features extracted by our method maintain better action information which is more suitable for downstream tasks.

Table 3. Comparison of action recognition results with unsupervised learning approaches on NTU 120 dataset. † indicates that results reproduced on our settings of feature dimension size.

Models	Architecture	xset	xsub
Single-stream:			
AS-CAL [24]	LSTM	49.2	48.6
AimCLR [8]	GCN	63.4	63.4
CMD† [20]	GRU	66.0	65.4
GL-Transformer [14]	Transformer	68.7	66.0
CPM [45]	GCN	69.6	68.7
ActCLR	GCN	**70.5**	**69.0**
Three-stream:			
3s-CrosSCLR [15]	GCN	66.7	67.9
3s-AimCLR [8]	GCN	68.8	68.2
3s-CMD† [20]	GRU	69.6	69.1
3s-SkeleMixCLR [5]	GCN	70.7	70.5
3s-CPM [45]	GCN	74.0	73.0
3s-ActCLR	GCN	**75.7**	**74.3**

2) Supervised Finetuning. We first pretrain the encoder $f(\cdot)$ in the self-supervised learning setting, and then finetune the entire network. We train the encoder $f(\cdot)$ and classifier $\phi(\cdot)$ using complete training data.

Table 4 displays the action recognition accuracy on the NTU datasets. This result confirms that our method extracts the information demanded by downstream tasks and can better benefit action recognition. In comparison with state-of-the-art supervised learning methods, our model achieves better performance.

3) Transfer Learning. To explore the generalization ability, we evaluate the performance of transfer learning. In transfer learning, we exploit self-supervised task pretrain-

Table 4. Comparison of action recognition results with supervised learning approaches on NTU dataset.

Models	Params	NTU 60 xview	NTU 60 xsub	NTU 120 xset	NTU 120 xsub
Single-stream:					
ST-GCN [40]	0.83M	88.3	81.5	73.2	70.7
SkeletonCLR [15]	0.85M	88.9	82.2	75.3	73.6
AimCLR [8]	0.85M	89.2	83.0	76.1	77.2
CPM [45]	0.84M	91.1	84.8	78.9	78.4
ActCLR	0.84M	**91.2**	**85.8**	**80.9**	**79.4**
Three-stream:					
3s-ST-GCN [40]	2.49M	91.4	85.2	77.1	77.2
3s-CrosSCLR [15]	2.55M	92.5	86.2	80.4	80.5
3s-AimCLR [8]	2.55M	92.8	86.9	80.9	80.1
3s-SkeleMixCLR [5]	2.55M	93.9	87.8	81.2	81.6
3s-ActCLR	2.52M	**93.9**	**88.2**	**84.6**	**82.1**

Table 5. Comparison of the transfer learning performance on PKUMMD dataset with linear evaluation pretrained on NTU 60.

Models	PKU I xview	PKU II xview
3s-AimCLR [8]	85.3	42.4
3s-ActCLR	**91.6**	**44.5**

Models	PKU I xsub	PKU II xsub
LongT GAN [49]	-	44.8
MS^{2}L [16]	-	45.8
ISC [34]	-	51.1
Hi-TRS [4]	-	55.0
3s-CrosSCLR [15]	-	51.3
3s-AimCLR [8]	85.6	51.6
3s-ActCLR	**90.0**	**55.9**

ing on the source data. Then we utilize the linear evaluation mechanism to evaluate on the target dataset. In linear evaluation, the encoder $f(\cdot)$ has fixed parameters without fine-tuning.

As shown in Table 5, our method achieves significant performance. Our method employs MATS to remove irrelevant information, and SAFP to retain information related to downstream tasks. This allows our encoder $f(\cdot)$ to obtain stronger generalization performance.

4) Unsupervised Action Segmentation. To explore the extraction of local features by our method, we used unsupervised action segmentation as an evaluation metric. We pretrain the encoder $f(\cdot)$ on the NTU 60 dataset. Then we utilize the linear evaluation mechanism to evaluate the results on the PKUMMD dataset. In linear evaluation, the encoder $f(\cdot)$ has fixed parameters without fine-tuning.

As shown in Table 6, our method achieves significant performance. Because our method focuses on the main occurrence region of the action, it is possible to locate the actions out of the long sequence.

Table 6. Comparison of the action segmentation performance on PKUMMD II xview dataset with linear evaluation pretrained on NTU 60 xview dataset.

Models	Stream	PKUMMD II xview			
		ACC	MACC	FWIoU	mIoU
AimCLR [8]	joint	39.77	28.68	26.79	15.67
ActCLR	joint	**51.29**	**31.97**	**35.24**	**21.38**
AimCLR [8]	motion	42.32	26.65	29.92	15.92
ActCLR	motion	**56.69**	**39.45**	**41.34**	**27.73**
AimCLR [8]	bone	54.22	39.52	39.41	27.36
ActCLR	bone	**59.09**	**41.14**	**41.54**	**28.89**

4.3. Ablation Study

Next, we conduct ablation experiments to give a more detailed analysis of our proposed approach.

1) Analysis of Motion-Adaptive Data Transformation. Data transformation is very important for consistency learning. To explore the influence of motion-adaptive data transformations, we test the action recognition accuracy under different data transformations. As shown in Table 7, the motion-adaptive transformation can obtain better performance than full region (the whole skeleton data) in different noise settings. It is also observed that when the noise strength increases, our performance degradation is much smaller than that of full region. This indicates that the design is more robust to data transformation.

To explore the influence of different data transformations on the contrastive learning effect, we test the action recognition accuracy under different data transformation combinations. As shown in Table 8, the consistency of the feature space is further enhanced with more data transformations. Thus, the performance of the downstream task is improved.

Table 7. Analysis of motion-adaptive data transformation on NTU 60 xview dataset with the joint stream.

Transformation	Region	KNN	Linear
Noise 0.01	Non-Actionlet	77.63	86.46
	Full Area	76.51	85.91
Noise 0.05	Non-Actionlet	**78.04**	**86.79**
	Full Area	75.28	84.20
Noise 0.1	Non-Actionlet	77.31	86.12
	Full Area	74.19	83.69
Skeleton Mix	Non-Actionlet	**78.04**	**86.79**
	Full Area	73.24	83.05

Table 8. Analysis of data transformation combinations on NTU 60 xview dataset with the joint stream. $\mathcal{T}$ is all the transformations. $\mathcal{T}_{\text{act}}$ is actionlet transformations. $\mathcal{T}_{\text{non}}$ is non-actionlet transformations. AdaIN refers to Skeleton AdaIN.

Modules	KNN	Linear
w/o $\mathcal{T}$	67.50	79.98
w/o (AdaIN + $\mathcal{T}_{\text{non}}$)	69.75	81.80
w/o $\mathcal{T}_{\text{non}}$	73.63	83.27
Full Version	**78.04**	**86.79**

Table 9. Analysis of semantic-aware feature pooling on NTU 60 xview dataset with the joint stream.

Modules	KNN	Linear
w/o SAFP	76.38	85.69
offline w/ SAFP	**78.04**	**86.79**
online w/ SAFP	76.02	85.25
online + offline w/ SAFP	76.71	85.92

2) Analysis of Semantic-Aware Feature Pooling. To explore the semantic-aware feature pooling, we perform this pooling on different streams. Table 9 shows the results of accuracy of action recognition under different settings. We note that better performance is obtained with offline, as it makes offline to generate better positive sample features for contrastive learning. Using this module in online reduces the benefits exposed by the non-actionlet transformation.

3) Analysis of Actionlet and Non-Actionlet Semantic Decoupling. In Fig. 3, we show the performance of extracting only actionlet region information and non-actionlet region information for action recognition. The accuracy of the actionlet region for action recognition is comparable to the accuracy of the whole skeleton data. In contrast, the performance of the features of non-actionlet regions for action recognition is much lower. This shows that the actionlet area does contain the main motion information.

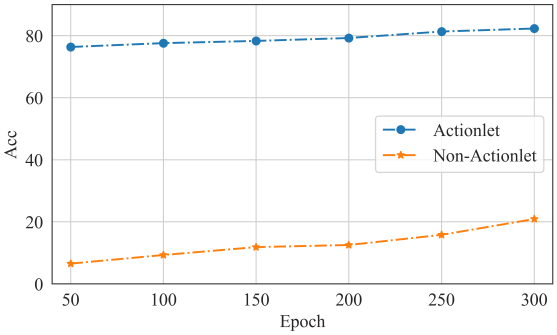

Figure 3. Action recognition accuracy of actionlet regions and non-actionlet regions.

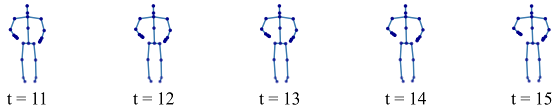

Figure 4. Visualization of the average motion. No obvious action takes place in the average motion sequence and can therefore be considered as a static anchor.

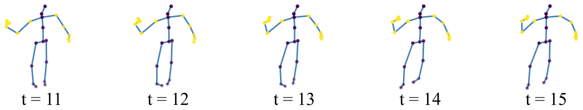

Figure 5. Visualization of the actionlet for a "throw" sequence. The yellow joints are the actionlet. Note that hand movements are mainly selected, indicating that the actionlet is reasonable.

4) Visualization of Average Motion and Actionlet. Fig. 4 shows a visualization of the average motion and actionlet respectively. The average motion has no significant motion information and serves as a background. The actionlet, shown in Fig. 5, selects the joints where the motion mainly occurs. Our actionlet is spatio-temporal, because the joints with motion may change when the action is performed.

5. Conclusions

In this work, we propose a novel actionlet-dependent contrastive learning method. Using actionlets, we design motion-adaptive data transformation and semantic-aware feature pooling to decouple action and non-action regions. These modules make the motion information of the sequence to be attended to while reducing the interference of static regions in feature extraction. In addition, the similarity mining loss further regularizes the feature space. Experimental results show that our method can achieve remarkable performance and verify the effectiveness of our designs.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019. 2

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[3] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *IEEE ICCV*, 2021. 1, 2

[4] Yuxiao Chen, Long Zhao, Jianbo Yuan, Yu Tian, Zhaoyang Xia, Shijie Geng, Ligong Han, and Dimitris N Metaxas. Hierarchically self-supervised transformer for human skeleton representation learning. In *ECCV*, 2022. 7

[5] Zhan Chen, Hong Liu, Tianyu Guo, Zhengyan Chen, Pinhao Song, and Hao Tang. Contrastive learning from spatio-temporal mixed skeleton sequences for self-supervised skeleton-based action recognition. *arXiv:2207.03065*, 2022. 6, 7

[6] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE CVPR*, 2015. 2

[7] Yusuke Goutsu, Wataru Takano, and Yoshihiko Nakamura. Motion recognition employing multiple kernel learning of fisher vectors using local skeleton features. In *IEEE ICCV Workshops*, 2015. 2

[8] Tianyu Guo, Hong Liu, Zhan Chen, Mengyuan Liu, Tao Wang, and Runwei Ding. Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. *AAAI*, 2022. 1, 3, 4, 6, 7

[9] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE CVPR*, 2006. 2

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE CVPR*, 2020. 2, 3, 5

[11] Hochreiter, Sepp and Bengio, Yoshua and Frasconi, Paolo and Schmidhuber, Jürgen and others. Gradient flow in recurrent nets: The difficulty of learning longterm dependencies. In *A Field Guide to Dynamical Recurrent Networks*, 2001. 2

[12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE ICCV*, 2017. 4

[13] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H Adelson. Learning visual groups from co-occurrences in space and time. *arXiv:1511.06811*, 2015. 2

[14] Boeun Kim, Hyung Jin Chang, Jungho Kim, and Jin Young Choi. Global-local motion transformer for unsupervised skeleton-based action learning. *ECCV*, 2022. 1, 6

[15] Linguo Li, Minsi Wang, Bingbing Ni, Hang Wang, Jiancheng Yang, and Wenjun Zhang. 3D human action representation learning via cross-view consistency pursuit. In *IEEE CVPR*, 2021. 2, 6, 7

[16] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. MS2L: Multi-task self-supervised learning for skeleton based action recognition. In *ACM MM*, 2020. 1, 6, 7

[17] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. NTU RGB+D 120: A large-scale benchmark for 3d human activity understanding. *IEEE TPAMI*, 2019. 2, 5

[18] Jiaying Liu, Sijie Song, Chunhui Liu, Yanghao Li, and Yueyu Hu. A benchmark dataset and comparison study for multi-modal human action analytics. *ACM TOMM*, 2020. 2, 5

[19] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *IEEE CVPR*, 2020. 1

[20] Yunyao Mao, Wengang Zhou, Zhenbo Lu, Jiajun Deng, and Houqiang Li. CMD: Self-supervised 3d action representation learning with cross-modal mutual distillation. *ECCV*, 2022. 5, 6

[21] Whitney K Newey. Adaptive estimation of regression models via moment restrictions. *Journal of Econometrics*, 1988. 5

[22] Qiang Nie, Ziwei Liu, and Yunhui Liu. Unsupervised 3D human pose representation with viewpoint and pose disentanglement. In *ECCV*, 2020. 6

[23] Wei Peng, Xiaopeng Hong, Haoyu Chen, and Guoying Zhao. Learning graph convolutional network for skeleton-based human action recognition by neural searching. In *AAAI*, 2020. 1

[24] Haocong Rao, Shihao Xu, Xiping Hu, Jun Cheng, and Bin Hu. Augmented skeleton based contrastive action learning with momentum LSTM for unsupervised action recognition. *Information Sciences*, 2021. 1, 2, 6

[25] Sucheng Ren, Huiyu Wang, Zhengqi Gao, Shengfeng He, Alan Yuille, Yuyin Zhou, and Cihang Xie. A simple data mixing prior for improving self-supervised learning. In *IEEE CVPR*, 2022. 4

[26] Amir Shahroudy, Jun Liu, Tian Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3d human activity analysis. In *IEEE CVPR*, 2016. 2, 5

[27] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE CVPR*, 2019. 1, 2

[28] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *IEEE CVPR*, 2019. 1, 2

[29] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, 2017. 2

[30] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. Spatio-temporal attention-based lstm networks for 3d action recognition and detection. *IEEE TIP*, 2018. 2

[31] Yifan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *ACM MM*, 2020. 1

[32] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *IEEE CVPR*, 2020. 1, 6

[33] Yukun Su, Guosheng Lin, and Qingyao Wu. Self-supervised 3D skeleton action representation learning with motion consistency and continuity. In *IEEE ICCV*, 2021. 3

[34] Fida Mohammad Thoker, Hazel Doughty, and Cees GM Snoek. Skeleton-contrastive 3D action representation learning. In *ACM MM*, 2021. 1, 3, 6, 7

[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv:1906.05849*, 2019. 2

[36] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *IEEE CVPR*, 2014. 2

[37] Raviteja Vemulapalli and Rama Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *IEEE CVPR*, 2016. 2

[38] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE CVPR*, 2012. 2

[39] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE CVPR*, 2018. 2

[40] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 2, 5, 7

[41] Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C Kot. Skeleton cloud colorization for unsupervised 3D action representation learning. In *IEEE ICCV*, 2021. 1, 6

[42] Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *IEEE CVPR*, 2019. 2

[43] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE ICCV*, 2019. 4

[44] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv:1710.09412*, 2017. 4

[45] Haoyuan Zhang, Yonghong Hou, Wenjing Zhang, and Wanqing Li. Contrastive positive mining for unsupervised 3d action representation learning. *ECCV*, 2022. 5, 6, 7

[46] 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 *IEEE CVPR*, 2020. 1

[47] Pengfei Zhang, Jianru Xue, Cuiling Lan, Wenjun Zeng, Zhanning Gao, and Nanning Zheng. Adding attentiveness to the neurons in recurrent neural networks. In *ECCV*, 2018. 2

[48] Xikun Zhang, Chang Xu, and Dacheng Tao. Context aware graph convolution for skeleton-based action recognition. In *IEEE CVPR*, 2020. 1

[49] 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. 1, 6, 7