Modeling the Relative Visual Tempo for Self-supervised Skeleton-based Action Recognition

Yisheng Zhu[1], Hu Han[2,3], Zhengtao Yu[4], Guangcan Liu[5*]
[1]Nanjing University of Posts and Telecommunications
[2]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS)
[3]University of the Chinese Academy of Sciences
[4]Faculty of Information Engineering and Automation, Kunming University of Science and Technology
[5]School of Automation, Southeast University
yszhu1995@163.com, huhan@ict.ac.cn, ztyu@hotmail.com, guangcanliu@seu.edu.cn

Abstract

Visual tempo characterizes the dynamics and the temporal evolution, which helps describe actions. Recent approaches directly perform visual tempo prediction on skeleton sequences, which may suffer from insufficient feature representation issue. In this paper, we observe that relative visual tempo is more in line with human intuition, and thus providing more effective supervision signals. Based on this, we propose a novel Relative Visual Tempo Contrastive Learning framework for skeleton action Representation (RVTCLR). Specifically, we design a Relative Visual Tempo Learning (RVTL) task to explore the motion information in intra-video clips, and an Appearance-Consistency (AC) task to learn appearance information simultaneously, resulting in more representative spatiotemporal features. Furthermore, skeleton sequence data is much sparser than RGB data, making the network learn shortcuts, and overfit to low-level information such as skeleton scales. To learn high-order semantics, we further design a new Distribution-Consistency (DC) branch, containing three components: Skeleton-specific Data Augmentation (SDA), Fine-grained Skeleton Encoding Module (FSEM), and Distribution-aware Diversity (DD) Loss. We term our entire method (RVTCLR with DC) as RVTCLR+. Extensive experiments on NTU RGB+D 60 and NTU RGB+D 120 datasets demonstrate that our RVTCLR+ can achieve competitive results over the state-of-the-art methods. Code is available at https://github.com/Zhuysheng/RVTCLR.

*Corresponding author. This work was supported in part by the New Generation AI Major Project of Ministry of Science and Technology of China under Grant 2018AAA0102501 and in part by the National Natural Science Foundation of China (NSFC) under Grant U21B2027.

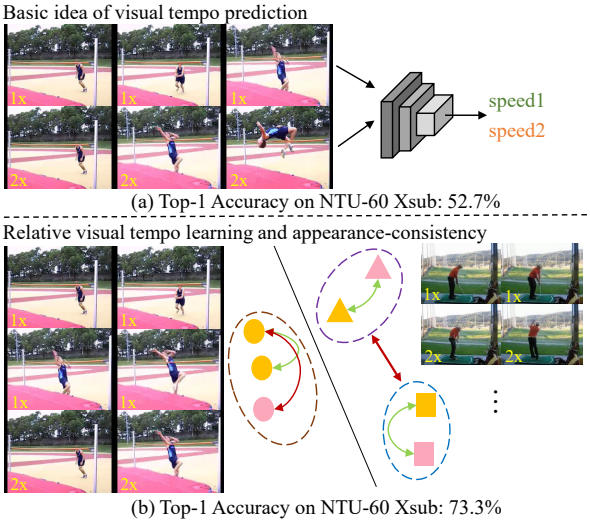


Figure 1. The basic visual tempo prediction is considered a classification task, where the learned model is used to assign tempo labels to individual video clips. We introduce relative visual tempo learning and appearance-consistency based on contrastive learning. It's more human-intuitive for modeling actions. The top-1 accuracy on NTU-60 Xsub benchmark supports our claim.

1. Introduction

As one of the most fundamental topics in video understanding, human action recognition has been widely explored in many real-world scenarios, such as human-computer interaction [22], autonomous driving [30], and so on [34, 17]. Skeleton data provides more abstract and well-structured information with less computation and storage than raw RGB video. It is less susceptible to camera viewpoint changes and background distractions. Thus, skeleton-based action recognition has attracted extensive at-

tention [39, 21, 28, 7, 35]. However, most of these methods rely heavily on full supervision. The collection of massive annotations is labor-intensive and time-consuming. Under this circumstance, learning action representations directly from the data in a self-supervised manner has attracted increasing attentions.

Several existing works [11, 38, 41] draw inspiration from video self-supervised learning and directly apply video pretext tasks on skeleton sequences, such as jigsaw puzzle recognition [1, 9] and temporal order prediction [15, 37]. For sequence data, a commonly used technique to model spatiotemporal information is visual tempo prediction [42, 29]. Action visual tempo describes the speed at which actions are performed, which is crucial for distinguishing actions that exhibit similar temporal evolutions (e.g., walking and jogging). The basic idea is shown in the upper part of Figure 1. For each video, clips are sampled at different sampling frequencies to mimic different visual tempos (e.g., $1\times$ and $2\times$), and predictions are made using the backbone network. However, there are limitations in directly applying it to skeleton sequences. People perform actions at their own tempo due to the influence of gender, age, etc. Even at the same tempo, the athlete's walking speed is visibly faster than that of a child in a third-person perspective. Therefore, it's obviously inappropriate to treat tempo prediction as a classification task. Furthermore, this basic strategy mainly focuses on motion information and cannot explicitly encourage the model to explore appearance information that is equally important for recognizing actions. Recently, contrastive learning has shown its great potential in extracting informative features [10, 6] in skeleton-based action recognition. Contrastive learning typically converts the naive classification task to a matching problem and learns an embedding space in which augmentations of same skeleton sequence are kept closer together, while different augmentations are far apart. In this way, the above issues can be alleviated in an elegant way. However, skeleton sequence data is much sparser than RGB data, and applying contrastive learning naively without explicit guidance may lead to model overfitting low-level information such as skeleton scales and angles, while failing to learn high-order semantics, resulting in insufficient feature representation capabilities.

To this end, we propose RVTCLR: a Relative Visual Tempo Contrastive Learning framework for skeleton action Representation. The basic idea is shown at the bottom of Figure 1. First, we observe that it's intuitively plausible to compare relative visual tempos within videos rather than to predict a specific visual tempo for each video. Specifically, for each video, we sample 3 clips with different visual tempos (e.g., $1\times$, $1\times$, and $2\times$) to construct the contrastive pairs and train a network to pull the anchor-positive pairs closer while repelling the anchor-negative pairs. Furthermore, to make the representations explicitly focus on appearance information, we design another Appearance-Consistency (AC) task. In this task, pairs from the same video are attracted no matter their visual tempos, while pairs from different videos are pushed away. In this way, the learned representations are expected to focus on both skeleton motion and appearance information simultaneously.

In addition, in order to encourage the models to learn high-order semantics, we introduce a new Distribution-Consistency (DC) branch based on RVTCLR, which contains three components: Skeleton-specific Data Augmentation (SDA), Fine-grained Skeleton Encoding Module (FSEM), and Distribution-aware Diversity (DD) Loss. We refer to this as RVTCLR+. First, we leverage SDA to generate more difficult contrastive pairs by applying more skeleton-specific transformations (e.g., gaussian noise) at the input level, since augmentation plays a key role in learning better representations, as demonstrated by SimCLR [3] and MoCo [8]. However, too much strong augmentations may blur the joint connections compared to the normal augmented sequence (i.e., crop and shear), resulting in performance degradation. Benefiting from recent attention mechanisms [5, 13, 44], we try to emphasize these connections by designing a novel module, FSEM, which contains an Intra-Inter-Part block (IIPB) for local spatial modeling and a Non-local block (NLB) [32] for global spatiotemporal modeling. Finally, inspired by [33, 6], we introduce a DD loss to minimize the distributional divergence between the normal augmented view and our DC branch. By combining these components, we hope that the DC branch can better learn local and global spatiotemporal features, which help to extract discriminative high-order semantics.

To sum up, the contributions of this work include: (1) We propose a novel contrastive representation learning framework named RVTCLR. The proposed RVTCLR leverages Relative Visual Tempo Learning (RVTL) task to learn better skeleton motion information. By combining another Appearance-Consistency (AC) task, our model explicitly learns to concentrate on skeleton appearance information simultaneously. (2) To encourage models to focus on high-order semantics, we propose RVTCLR+ by introducing a new Distribution-Consistency (DC) branch. This DC branch contains three components: Skeleton-specific Data Augmentation (SDA), Fine-grained Skeleton Encoding Module (FSEM), and Distribution-aware Diversity (DD) Loss. (3) These contrastive tasks are jointly trained using a two-branch structure such that the models can learn both spatiotemporal and high-order semantics simultaneously.

2. Related Work

Self-supervised skeleton-based action recognition. To meet the requirements of real-time recognition, it's necessary for the models to be able to directly extract dis-

criminative action representations from the label-free online videos. Recent progress in self-supervised skeleton-based action recognition can be summarized into two categories: pretext-task based learning [11, 38, 41, 43, 23] and contrastive learning [18, 10, 6, 25, 31, 24]. For example, Zheng et al. [43] design a skeleton inpainting architecture to learn the long-term dynamics. Lin et al. [11] integrate multiple tasks such as jigsaw puzzle recognition to learn more general skeleton features. Xu et al. [38] propose reverse sequential predictions based on encoder-decoder structure to extract motion pattern. However, the representations learned by these methods may not be good enough, in the sense that they could be exclusively particular to the pre-designed tasks. Inspired by the success of contrastive learning in image classification (e.g., instance discrimination [36], SimCLR [3], and MoCo [8]), Rao et al. [18] first propose to perform contrastive learning among different augmentations of unlabeled skeleton data, to learn inherent action patterns. Thoker et al. [25] propose to generate contrastive pairs based on different input-representations of the skeleton sequences, i.e., graph, sequence, and image representation. By leveraging multiple views of the skeleton data, i.e., joint, bone, and motion, Li et al. [10] introduce SkeletonCLR and CrosSCLR to perform the single-view and cross-view contrastive learning. Guo et al. [6] further propose AimCLR to learn from extremely augmented skeleton sequences.

Nevertheless, none of the aforementioned methods concentrate on visual tempo, which is crucial for characterizing human action dynamics. To our best knowledge, Su et al. [24] propose motion consistency and continuity learning, which has overlap with our framework. However, the differences are obvious, which mainly lie in three aspects: (1) contrastive pairs are generated differently. In [24], speed-changed clips are considered as positive pairs, while we focus on relative visual tempo and regard these clips as negative pairs. (2) appearance information is modeled differently. In [24], appearance information is implicitly modeled in the contrastive learning process, while we design another Appearance-Consistency (AC) task specifically for spatial modeling. (3) learned features are enhanced differently. In [24], learned features are enhanced by designing a self-reconstruction based motion continuity module, while we introduce a novel Distribution-Consistency (DC) branch to guide the models focus on high-order semantics.

Visual tempo modeling. Visual tempo describes the speed of human movements, which has already been applied into various action recognition methods [4, 40, 14]. For example, Feichtenhofer et al. [4] first propose SlowFast network to explore the potential of different visual tempos. It consists of two pathways, operating at different frame rates, to capture both spatial semantics and motion dynamics. When it comes to video self-supervised learning [42, 29, 2], the potential of visual tempo in motion modeling is further verified. Yao et al. [42] utilize video playback rates as self-supervision signals and propose playback rate perception to learn spatiotemporal features in a collaborative discrimination-generation manner. The above methods usually require assigning visual tempos to each video clip according to different sampling rates, and then elaborate learning paradigms through reconstruction or prediction, but this is suboptimal in learning discriminative representations. Chen et al. [2] argue that relative speed is more in line with motion pattern. They propose a new video self-supervised learning framework (called RSPNet) to leverage the relative speed between two video clips to supervise the representation learning. Inspired by RSPNet, the proposed RVTCLR focuses on relative visual tempo learning with skeleton-specific modifications.

3. Methods

3.1. Preliminaries

SkeletonCLR. SkeletonCLR [10] utilizes ST-GCN [39] as its backbone and follows the practice in MoCo [8] to learn skeleton action representations. Suppose that a skeleton sequence $s = (s^1, s^2, ...s^T)$ contains T consecutive skeleton frames, where $s^i \in \mathbb{R}^{3\times V\times M}$ means 3D coordinates of V joints for M actors. A data augmentation module $\mathcal{T}$ is first utilized to randomly transform the given s into different augmentations x_q and x_k. Then, a query encoder $f(\cdot;\theta_q)$ and a momentum updated key encoder $f(\cdot;\theta_k)$ are leveraged to encode x_q and x_k into hidden space h_q and h_k. These embeddings are further passed through a projector $g(\cdot;\theta_q)$ and its momentum updated version $g(\cdot;\theta_k)$ to get the final query features q and key features k. In each training step, samples in the queue $\mathbf{Q} = \{m_o\}_{o=1}^K$ are progressively replaced by the key features k following a first-in-first-out scheme. Following MoCo, q and k serve as positive pairs while q and embeddings in $\mathbf{Q}$ serve as negative pairs. The InfoNCE loss [36] is used to guide the network learning, which can be formulated as:

$$\mathcal{L}_{Info} = -\log \frac{\exp(q\cdot k/\tau)}{\exp(q\cdot k/\tau) + \sum_{o=1}^{K}\exp\left(q\cdot m_o/\tau\right)} \quad (1)$$

where τ is the temperature hyper-parameter, and $\cdot$ represents the similarity measured by dot product.

After computing the InfoNCE loss, the parameters of θ_q are updated by gradient back-propagation while the parameters of θ_k are updated as their moving-average:

$$\theta_k \leftarrow \alpha\theta_k + (1-\alpha)\theta_q \quad (2)$$

where α is a momentum coefficient (set to 0.999 by default).

Nearest Neighbors Mining. Traditional InfoNCE regards all samples in the memory bank as negative. This overly hard approach may lead to non-generic feature embeddings. To address the above issue, [10, 6] propose

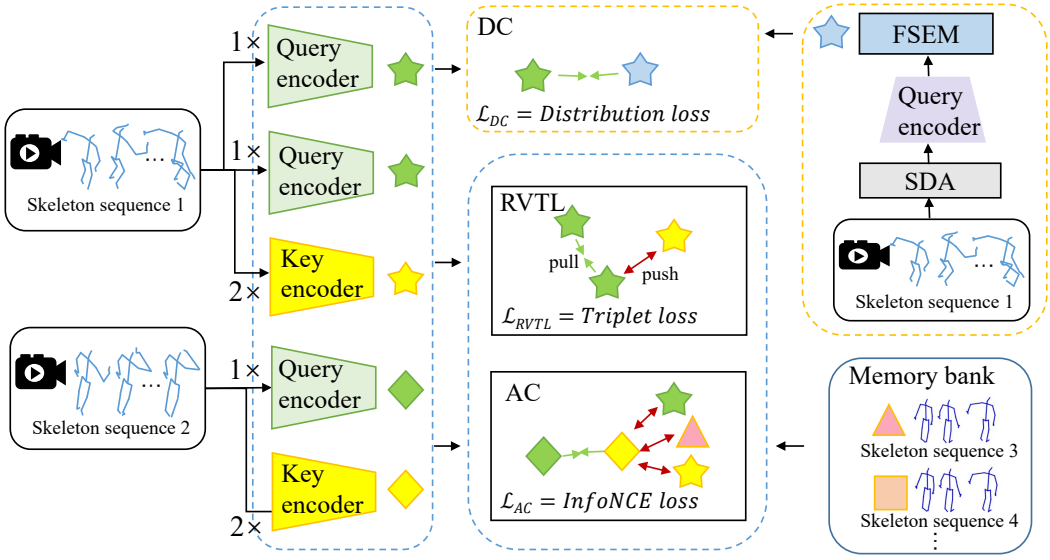


Figure 2. The pipeline of our method. For each skeleton sequence, we sample clips with varying visual tempos. Then, we use a two-branch structure to extract discriminative feature from each sequence. Specifically, in RVTCLR, the Relative Visual Tempo Learning (RVTL) is responsible for modeling motion dynamics. The Appearance-Consistency (AC) is responsible for modeling appearance characteristics. These two tasks are combined into one branch to achieve a more comprehensive spatial-temporal representation. In another Distribution-Consistency (DC) branch, we introduce Skeleton-specific Data Augmentation (SDA), Fine-grained Skeleton Encoding Module (FSEM), and Distribution-aware Diversity (DD) Loss to guide the network to concentrate on extracting high-order semantics.

to leverage Nearest Neighbors Mining (NNM) to generate more positive pairs:

$$\mathcal{L}_{NNM} = -\log \frac{\exp(q \cdot k/\tau) + \sum_{o \in N_+} \exp(q \cdot m_o/\tau)}{\exp(q \cdot k/\tau) + \sum_{o=1}^{K} \exp(q \cdot m_o/\tau)} \tag{3}$$

where N_+ represents the top-N nearest neighbors that are most similar to the query features q in the memory queue.

Since the model learned in the early training stages may not be strong enough to provide confident nearest neighbors, [10, 6] propose to perform a two-stage training, i.e., first train the model using Equation 1, and then train the model to mine the nearest neighbors using Equation 3.

In this paper, we mainly implement our method based on SkeletonCLR and two-stage training strategy.

3.2. RVTCLR

Relative Visual Tempo Learning (RVTL). Observing that each person performs actions at his/her own visual tempo, we introduce a RVTL task, which aims to learn better skeleton motion information. Specifically, given a skeleton sequence s with T frames, we first sample 3 clips c_i, c_j and c_k with visual tempos v_i, v_j and v_k, respectively, where $v_i = v_j \neq v_k$. Note that visual tempos can be set arbitrarily, i.e., $\{1\times, 2\times, 3\times, ..\}$. However, to avoid the temporal ambiguity in skeleton sequences, we only consider $1\times$ and $2\times$, which represent that the sampling interval is set to 1 and 2 frames. We sample from the first frame and keep a duration of $T/2$ for each clip. Then, we apply normal skeleton augmentations $\mathcal{T}$ on these three clips respectively to construct a triplet, i.e., anchor $a = \mathcal{T}(c_i)$, positive $p = \mathcal{T}(c_j)$ and negative $n = \mathcal{T}(c_k)$. This triplet is further fed into SkeletonCLR with a projection head $g_r(\cdot;\theta_{q_r})$ and its momentum updated $g_r(\cdot;\theta_{k_r})$ to generate the encoded features (q_a, k_p, k_n). Finally, these features are normalized and combined as anchor-positive pairs (q_a, k_p) and anchor-negative pairs (q_a, k_n), and our goal is to pull (q_a, k_p) closer while pushing (q_a, k_n) away. The assumption here is that the network can only succeed in such a RVTL task if it understands the intrinsic visual tempo of each clip and learns discriminative motion information. This task is realized by a triplet loss [19]:

$$\mathcal{L}_{RVTL} = \max\{-(q_a \cdot k_p - q_a \cdot k_n) + \text{margin}, 0\} \tag{4}$$

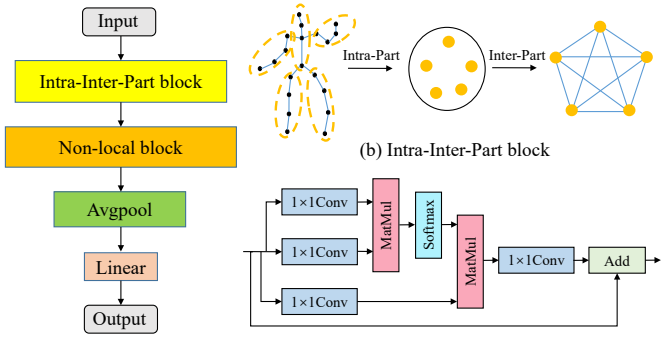
where the margin (set to 1 by default) is a hyper-parameter to control the distance between two components.

Appearance-Consistency (AC). Appearance information is also essential for recognizing actions. To this end, we introduce another AC task to explicitly learn such information. Specifically, given a skeleton sequence, we first sample two clips c_i, c_j with visual tempos randomly selected in $\{1\times, 2\times\}$. Then, they are normally augmented and sent to SkeletonCLR with another projection head $g_a(\cdot;\theta_{q_a})$ and its momentum updated $g_a(\cdot;\theta_{k_a})$ to generate the encoded features (q_i, k_j). Note that the query and key encoders in AC share weights with the encoders in RVTL while the

projection heads $g_a(\cdot;\theta_{q_a})$ and $g_a(\cdot;\theta_{k_a})$ have weights independent of $g_r(\cdot;\theta_{q_r})$ and $g_r(\cdot;\theta_{k_r})$. Here, the goal is to pull q_i and k_j closer while pushing away q_i and negative samples in queue $\mathbf{Q}$. Although from the same sample, clips with various visual tempos show different temporal motion dynamics, by pulling such a pair closer while pushing away different samples, we want the model to explicitly focus on appearance information. This task is realized by the InfoNCE loss:

$$\mathcal{L}_{AC} = -\log \frac{\exp(q_i \cdot k_j/\tau)}{\exp(q_i \cdot k_j/\tau) + \sum_{o=1}^{K} \exp(q_i \cdot m_o/\tau)} \tag{5}$$

Based on SkeletonCLR, we jointly train these tasks. In the first training stage, we train the model with the loss function: $\mathcal{L}_1 = \lambda_1 \mathcal{L}_{RVTL} + \lambda_2 \mathcal{L}_{AC}$, and then in the second training stage, we try to obtain more positive samples by using $\mathcal{L}_2 = \lambda_1 \mathcal{L}_{RVTL} + \lambda_2 \mathcal{L}_{AC'}$, where λ (set to 1 by default) is the coefficient to balance the loss and $\mathcal{L}_{AC'}$ is defined in Equation 3.



(a) FSEM module (b) Intra-Inter-Part block (c) Non-local block

Figure 3. Illustration of FSEM. (a) The workflow of FSEM. (b) IIPB is responsible for local spatial modeling, containing intra-part and inter-part relational reasoning submodules. (c) NLB is responsible for global spatial-temporal modeling.

3.3. RVTCLR+

The success of contrastive learning depends on the construction of contrastive pairs and the difficulty of learning paradigm. Given that the skeleton sequences are much sparser, naive contrastive learning may risk overfitting low-level information such as skeleton scales. It would be even better if we could design a new branch to explicitly learn skeleton high-order semantics. We thus propose RVTCLR+ by designing our **Distribution-Consistency (DC)**.

SDA. We leverage SDA to generate more difficult contrastive pairs. The SDA contains normal augmentations $\mathcal{T}$: crop and shear, and additional strong augmentations $\mathcal{T}'$: gaussian noise, gaussian blur, and channel mask. Given a skeleton sequence s, we apply $\mathcal{T}$ and $\mathcal{T}'$ to get the transformed version: $e = \mathcal{T}'(\mathcal{T}(s))$. Afterwards, we send it to the weight-sharing query encoder to encode e into hidden space h_e, which is further passed through a projector $g_e(\cdot;\theta_{q_e})$ to obtain the final embeddings q_e.

FSEM. Such a direct modeling method may be suboptimal, since the additional augmentations may cause blurred joint connections. Considering that skeleton sequences are spatiotemporal cubes, we design FSEM to highlight these connections. As shown in Figure 3, our FSEM contains an Intra-Inter-Part block (IIPB), which is responsible for local spatial modeling, and a Non-local block (NLB), which is responsible for global spatial-temporal modeling. Given the encoded $h_e \in \mathbb{R}^{C\times T\times V}$, in IIPB, we first divide each human skeleton into 5 parts according to the physical topology of the body. Then, average pooling is applied on each part to generate the intra-part representations $h_{e'} \in \mathbb{R}^{C\times T\times 5}$. Two 1×1 convolution layers are further utilized to establish the inter-part relations:

$$h_{\hat{e}} = Conv2(ReLU(Conv1(h_{e'}, W_1)), W_2) \tag{6}$$

where $W_1 \in \mathbb{R}^{\frac{C}{2}\times C}$ and $W_2 \in \mathbb{R}^{C\times \frac{C}{2}}$ are learnable parameters. $ReLU$ is the activation function. The intuition here is that connections between body parts can sometimes better describe actions (e.g., the coordination of arm and leg in 'running'). By using IIPB, we establish the part-based connections that seek to mitigate the effects of blurred joints.

IIPB can capture the local dependencies of intra-frame parts. However, global inter-frame connections are ignored, which are also essential for describing actions. We introduce NLB to associate all possible frames of a skeleton sequence in a self-attention mechanism [27]. The non-local operation computes frame dependencies by enumerating all possible frames:

$$y^t = \frac{1}{\mathcal{C}(h_{\hat{e}})} \sum_{\forall t'} f\left(h_{\hat{e}}^t, h_{\hat{e}}^{t'}\right) g\left(h_{\hat{e}}^{t'}\right) \tag{7}$$

where $h_{\hat{e}}^t$ and $h_{\hat{e}}^{t'}$ denote the input at frame t and t', respectively. $g(\cdot)$ is a linear function. $\mathcal{C}(\cdot)$ is a normalization term. An embedded gaussian function $f(\cdot,\cdot)$ [32] computes the relationships between t and all t'. We wrap the non-local operation in Equation 7 into NLB with a residual layer as:

$$h_{\tilde{e}} = W_3 y + h_{\hat{e}} \tag{8}$$

where W_3 is a learnable embedding for y. Finally, we send it to the projector to generate the new embeddings $q_{\tilde{e}}$. Note that FSEM can be plugged into different GCN layers, we only insert the FSEM before the projector for the reduction of experiment cost.

DD Loss. The distributional divergence between weakly augmented counterparts and strongly augmented views enables the framework to explore novel patterns, which may help extract representative high-level features. Formally,

given the encoded query features q_a, the encoded key features k_p, and the memory queue $\mathbf{Q} = \{m_o\}_{o=1}^{K}$, we can obtain a conditional distribution:

$$p(k_p|q_a) = \frac{\exp(q_a \cdot k_p/\tau)}{\exp(q_a \cdot k_p/\tau) + \sum_{o=1}^{K} \exp(q_a \cdot m_o/\tau)} \quad (9)$$

which encodes the likelihood of the query q_a being assigned to the key k_p. By replacing k_p with m_o, we can also obtain the likelihood that the query q_a is assigned to the negative embeddings in $\mathbf{Q}$. Then, the InfoNCE loss in Equation 1 can be rewritten as:

$$\mathcal{L}_{Info} = -q(k_p|q_a) \log p(k_p|q_a) - \sum_{o=1}^{K} q(m_o|q_a) \log p(m_o|q_a) \quad (10)$$

where $q(\cdot|\cdot)$ is the ideal distribution of the likelihood, $p(k_p|q_a)$ is the distribution learned by network. In InfoNCE, $q(k_p|q_a)$ and $q(m_o|q_a)$ are simply set to 1 and 0 based on one-hot distribution, respectively. When applied to $q_{\widetilde{e}}$, a straightforward approach is to directly replace q_a in Equation 10 with $q_{\widetilde{e}}$. However, as demonstrated in [33, 6], the one-hot distribution cannot mimic the ideal distribution and thus cannot help representations learning any more. Conversely, the similarity distribution of weakly-augmented queries for the same instance in a queue can provide useful clues for strong-augmentation based learning. This inspires us to leverage $p(k_p|q_a)$ and $p(m_o|q_a)$ as the ideal distribution to supervise $q_{\widetilde{e}}$:

$$\begin{aligned} \mathcal{L}_{DC} = &-p(k_p|q_a) \log p(k_p|q_{\widetilde{e}}) \\ &- \sum_{o=1}^{K} p(m_o|q_a) \log p(m_o|q_{\widetilde{e}}) \end{aligned} \quad (11)$$

By combining DC, the final loss our RVTCLR+ can be formulated as $\mathcal{L}_1 = \lambda_1 \mathcal{L}_{RVTL} + \lambda_2 \mathcal{L}_{AC} + \lambda_3 \mathcal{L}_{DC}$ and $\mathcal{L}_2 = \lambda_1 \mathcal{L}_{RVTL} + \lambda_2 \mathcal{L}_{AC'} + \lambda_3 \mathcal{L}_{DC}$.

4. Experiments

4.1. Datasets

NTU RGB+D 60 Dataset [20]: It contains 56880 skeleton video clips over 60 action classes capture from 40 subjects and 3 different camera view angles. Each clip provides 25 body joints with 3D coordinates for at most 2 subjects. There are two evaluation benchmarks: cross-subject (Xsub) and cross-view (Xview). In Xsub, clips of 20 subjects are used for training, and the rest are used for testing. In Xview, clips of camera 2 and 3 are used for training, and clips of camera 1 are used for testing.

NTU RGB+D 120 Dataset [12]: It is an extension of NTU RGB+D 60 with 113945 samples over 120 classes capture from 106 subjects and 32 different camera setups. Two evaluation benchmarks are recommended: cross-subject (Xsub) and cross-setup (Xset). In Xsub, clips of 53 subjects are used for training, and the rest are used for testing. In Xset, clips of even camera IDs are used for training, and clips of odd IDs are used for testing.

4.2. Implementation Details

For data pre-processing, we follow CrosSCLR [10] and AimCLR [6] except for that we resize the length of skeleton sequences to 100 frames, rather than 50 frames (Actually, by $2\times$ temporal sampling, we also maintain 50 frames for each clip). The mini-batch size is set to 128.

Data Augmentation. For skeleton sequences, shear and crop are used as the normal augmentations $\mathcal{T}$. For strong augmentations $\mathcal{T}'$, we mainly use gaussian noise, gaussian blur, and channel mask. Since [18, 6] use more augmentations, we test if these augmentations (spatial flip, temporal flip, rotate) work in our approach.

Self-supervised Pre-training. The baseline is SkeletonCLR which follows the MoCo [8] framework. The queue size $\mathbf{Q}$ and temperature τ are set to 32768 and 0.07, respectively. For the backbone, we adopt ST-GCN [39], but the number of channels in each layer is reduced to 1/4 of the original settings. For the optimizer, we use SGD with momentum 0.9 and weight decay 0.0001. The pre-training runs 300 epochs with the initial learning rate 0.1 and multiplied by 0.1 at 250 epochs. For the two-stage training strategy, we train 150 epochs with the loss function $\mathcal{L}_1$, and another 150 epochs with $\mathcal{L}_2$. In the second training stage, nearest neighbors N is set to 1.

Linear Evaluation Protocol. We add a linear classifier after the pre-trained encoder. During the linear evaluation, we freeze the parameters in encoder and only train the parameters in linear classifier. We use SGD with momentum 0.9. The model runs 100 epochs with the initial learning rate 3.0 and decayed by 10 at 80 epochs.

Finetune Protocol. We add a linear classifier after the pre-trained encoder. During the finetune, we train the whole model with SGD optimizer, the momentum is set to 0.9 and the weight decay is set to 0.0001. The model runs 100 epochs with the initial learning rate 0.01 and decayed by 10 at 80 epochs.

Semi-supervised Evaluation Protocol. We add a linear classifier after the pre-trained encoder. During the semi-supervised evaluation, we train the whole model with only 1% or 10% randomly selected labeled data. We choose SGD optimizer with momentum 0.9 and weight decay 0.0001. The model runs 100 epochs with the initial learning rate 0.01 and decayed by 10 at 80 epochs.

4.3. Ablation Study

Tempo Prediction or RVTCLR. We first compare RVTCLR with the tempo prediction. Table 1 shows the comparison results of the joint stream on the NTU RGB+D 60

dataset using the linear evaluation protocol. It is obviously found that tempo prediction cannot bring good performance compared to RVTL, which verifies our claim that predicting each video's specific visual tempo results in unreasonable action features. The outcome of AC demonstrates that explicit consideration of apparent information also plays an important role in modeling actions. AC+RVTL's performance proves that relative visual tempo and appearance modeling can work collaboratively to better concentrate on both skeleton motion and appearance clues. By further combining NNM, the performance of RVTCLR boosts to 74.4% (79.4%) on Xsub (Xview) benchmark, surpassing the tempo prediction by a large margin.

Table 1. Comparisons between Tempo Prediction and RVTCLR. We report linear evaluation results of the joint stream on the NTU-60 dataset.

Methods	NTU-60	
	Xsub	Xview
Tempo prediction	52.7	58.0
RVTL	63.1	73.4
AC	65.1	68.4
AC+RVTL	73.3	77.8
AC+RVTL+NNM	74.4	79.4

Effectiveness of RVTCLR+. We then verify the effectiveness of RVTCLR+. Table 2 shows the comparison results of three streams. From the table, we have several observations: (1) RVTCLR performs much better than SkeletonCLR in most cases. For example, we can obtain +4.2% (+9.2%) performance gain on Xsub (Xview) in the motion stream. By further combining DC branch, our RVTCLR+ can still significantly improve the accuracy by +13.3% and +12%, respectively, which shows the efficacy of our method. (2) DC plays a more important role in the motion and bone streams. The primary reason we believe is due to the inherent presence of higher-order semantics in these two streams. The sparsity of skeleton data, however, may hinder the ability of naive contrastive learning to fully release their potential. As a result, recognition success may be limited to relying solely on low-level information. Our DC, however, force the model to focus on high-level semantics, thereby enabling the learning of more discriminative representations. (3) our 2-stream and 3-stream fusion results are always better than the comparative counterparts.

We also conduct experiments in the motion stream to validate the effect of DC. As shown in Table 3 (a), we first determine whether all augmentations mentioned in [18, 6] applicable to SDA. Compared to the normal augmentation, we can find that gaussian noise, gaussian blur, and channel mask work well on both benchmarks. Although the others may perform better on Xsub, their results on Xview are relatively poor. Through empirical experimentation, we determine the former three augmentations as our best choice, yielding +12.7% and +11.4% accuracy increases over using the normal augmentation.

Table 2. Linear evaluation compared with SkeletonCLR. '2s' and '3s' means two-stream and three-stream fusion, respectively.

Methods	DC	Stream	NTU-60	
			Xsub	Xview
SkeletonCLR		joint	68.3	76.4
RVTCLR		joint	74.4	79.4
RVTCLR+	✓	joint	74.7	79.1
SkeletonCLR		motion	53.3	50.8
RVTCLR		motion	57.5	60.0
RVTCLR+	✓	motion	70.8	72.0
SkeletonCLR		bone	69.4	67.4
RVTCLR		bone	68.1	71.7
RVTCLR+	✓	bone	72.2	78.4
2s-SkeletonCLR		joint+motion	70.5	77.9
2s-RVTCLR		joint+motion	75.7	81.6
2s-RVTCLR+	✓	joint+motion	77.8	82.2
3s-SkeletonCLR		joint+motion+bone	75.0	79.8
3s-RVTCLR		joint+motion+bone	77.2	82.0
3s-RVTCLR+	✓	joint+motion+bone	79.7	84.6

FSEM is designed to emphasize the blurred joint connections. As shown in Table 3 (b), IIPB brings +3.3% performance improvement on Xview, proving its ability to adeptly capture the local spatial features. The accuracies of NLB are 70.0% and 70.7%, respectively, which demonstrate that constructing global dependencies among frames can achieve a more representative feature extraction. By combining these two blocks, the performance of our final FSEM further boots to 70.8% and 72.0%.

DD loss is introduced to help learn more representative high-level features. From the Table 3 (b), we can see a large performance drop when we replace the DD loss with the original one. This suggests that compared to the one-hot distribution, the weakly augmented view provides more suitable supervision signals.

Table 3. Comparisons of DC's each component. We report linear evaluation results of the motion stream on the NTU-60 dataset.

SDA	NTU-60	
	Xsub	Xview
Normal	55.2	58.2
Gaussian noise	66.8	72.6
Gaussian blur	56.2	59.4
Channel mask	57.3	58.6
Rotate	55.5	57.1
Spatial flip	57.6	56.1
Temporal flip	59.4	55.9
Best Combination	67.9	69.6

(a) SDA

FSEM	NTU-60	
	Xsub	Xview
-	67.9	69.6
IIPB	68.0	72.9
NLB	70.0	70.7
IIPB+NLB	70.8	72.0

DD loss	NTU-60	
	Xsub	Xview
✓	70.8	72.0
-	64.6	63.6

(b) FSEM and DD loss

4.4. Performance Comparison

Linear Evaluation. Table 4 and Table 5 show the comparisons on NTU-60 and NTU-120 under linear evaluation protocol. From the Table 4, we observe that our joint-stream RVTCLR+ achieves competitive results compared to the

most recent SkeletonCLR [10] and AimCLR [6]. For example, 74.7% vs 74.3% on Xsub when compared to AimCLR. When multiple streams are fused, the performance of our method can be further improved. From the Table 5, we find that the performance of our 3s-RVTCLR+ remains at the forefront.

Finetune Evaluation. Table 6 shows the comparisons on NTU-60 and NTU-120 under fineture evaluation protocol. The table reveals that our single-stream RVTCLR+ achieves 0.8%~2.1% improvement over AimCLR on both datasets. Notably, 3s-ST-GCN [39] is pretrained under full supervision, and its performance is the ensemble of multi-streams, compared to it, our single-stream RVTCLR+ can still get comparable performance. When combining different streams, the accuracy of our 3s-RVTCLR+ can be further increased by 2.6%~5%, greatly outperforming AimCLR.

Table 4. Linear evaluation results on NTU-60 dataset.

Methods	NTU-60	
	Xsub	Xview
Single-stream		
LongT GAN [43]	39.1	48.1
P&C [23]	50.7	76.3
MS^{2}L [11]	52.6	-
PCRP [38]	54.9	63.4
AS-CAL [18]	58.5	64.8
CRRL [31]	67.6	73.8
SeBiReNet [16]	-	**79.7**
SkeletonCLR [10]	68.3	76.4
AimCLR [6]	74.3	**79.7**
RVTCLR+ (ours)	**74.7**	79.1
Three-stream		
3s-SkeletonCLR [10]	75.0	79.8
3s-Colorization [41]	75.2	83.1
3s-CrosSCLR [10]	77.8	83.4
3s-AimCLR [6]	78.9	83.8
3s-RVTCLR+ (Ours)	**79.7**	**84.6**

Table 5. Linear evaluation results on NTU-120 dataset.

Methods	NTU-120	
	Xsub	Xset
P&C [23]	42.7	41.7
PCRP [38]	43.0	44.6
AS-CAL [18]	48.6	49.2
CRRL [31]	56.2	57.0
ISC [25]	67.9	67.1
3s-CrosSCLR [10]	67.9	66.7
3s-AimCLR [6]	**68.2**	68.8
3s-RVTCLR+ (Ours)	68.0	**68.9**

Semi-supervised Evaluation. As shown in Table 7, with only 1% and 10% labeled data, the results of our 3s-RVTCLR+ far exceed MCC [24], 3s-CrosSCLR [10] , and 3s-AimCLR. For example, 3s-RVTCLR+ increases the accuracies by 2.5% and 4.6% compared with 3s-AimCLR on NTU-120 dataset. These results suggest that the representational capacity of our learned features is considerable.

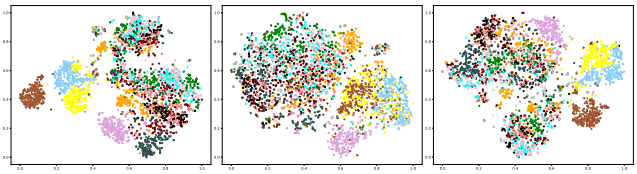
Qualitative Results. We apply t-SNE [26] with fixed settings to show the embedding distribution of SkeletonCLR and our RVTCLR+ on NTU-60 Xsub benchmark. For a fair comparison, we select the same 11 classes for visualization, where each class is represented by a dot of the same color. From the Figure 4, we can obviously see that the embeddings extracted from our method have better inter-class separability and intra-class compactness, indicating that our learned features are more discriminative.

Table 6. Finetune evaluation results on NTU-60 and NTU-120 dastasets. †means using the same bone stream. ‡means the model is pretrained under full supervision.

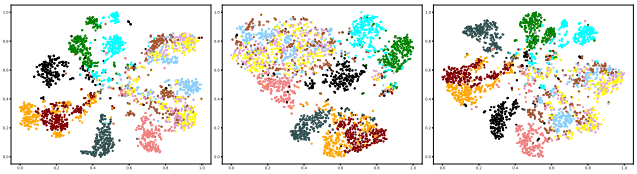
Methods	NTU-60		NTU-120	
	Xsub	Xview	Xsub	Xset
SkeletonCLR† [10]	82.2	88.9	73.6	75.3
MCC [24]	83.0	89.7	77.0	77.8
AimCLR† [6]	83.0	89.2	76.4	76.7
RVTCLR+†(Ours)	**84.4**	**91.3**	**77.2**	**78.4**
3s-ST-GCN‡ [39]	85.2	91.4	77.2	77.1
3s-CrosSCLR [10]	86.2	92.5	80.5	80.4
3s-AimCLR [6]	86.9	92.9	80.1	80.9
3s-RVTCLR+ (Ours)	**87.5**	**93.9**	**82.0**	**83.4**

Table 7. Seim-supervised evaluation on NTU-60 and NTU-120 dastasets. * means the results are obtained by their released codes.

Methods	NTU-60		NTU-120	
	Xsub	Xview	Xsub	Xset
1% labeled data				
ISC [25]	35.7	38.1	-	-
3s-Colorization [41]	48.3	52.5	-	-
3s-CrosSCLR [10]	51.1	50.0	28.6*	28.0*
3s-AimCLR [6]	54.8	**54.3**	**34.8***	32.6*
3s-RVTCLR+ (Ours)	**54.9**	53.6	33.3	**32.8**
10% labeled data				
ISC [25]	65.9	72.5	-	-
3s-Colorization [41]	71.7	78.9	-	-
MCC-ST-GCN [24]	55.6	59.9	40.7	43.4
MCC-2s-AGCN [24]	60.8	65.8	47.0	51.8
MCC-AS-GCN [24]	59.2	63.1	44.9	47.8
3s-CrosSCLR [10]	74.4	77.8	61.3*	61.1*
3s-AimCLR [6]	78.2	81.6	64.8*	63.7*
3s-RVTCLR+ (Ours)	**79.5**	**83.7**	**67.3**	**68.3**



(a) SkeletonCLR, j (b) SkeletonCLR, m (c) SkeletonCLR, b



(d) RVTCLR+, j (e) RVTCLR+, m (f) RVTCLR+, b

Figure 4. The t-SNE visualization of embeddings on NTU-60 X-sub. These models are trained for 300 epochs. j, b, and m represent joint-stream, bone-stream, and motion-stream, respectively.

5. Conclusion

In this paper, we propose a Relative Visual Tempo Contrastive Learning framework for skeleton action representation (RVTCLR). The proposed RVTCLR combines Relative Visual Tempo Learning (RVTL) and Appearance-Consistency (AC) into a single-branch to obtain a more comprehensive spatial-temporal representation. Given the inherent sparsity of skeleton sequence data compared to RGB data, we design a new Distribution-Consistency (D-C) branch aimed at emphasizing high-order semantics and preventing the network from learning shortcuts. The DC branch consists of Skeleton-specific Data Augmentation (S-DA), Fine-grained Skeleton Encoding Module (FSEM), and Distribution-aware Diversity (DD) Loss. We refer to this new two-branch structure as RVTCLR+. Experimental results and visualization analysis verify that RVTCLR+ can obtain a more discriminative skeleton action representation.

References

[1] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 179–189, 2019.

[2] Peihao Chen, Deng Huang, Dongliang He, Xiang Long, Runhao Zeng, Shilei Wen, Mingkui Tan, and Chuang Gan. Rspnet: Relative speed perception for unsupervised video representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1045–1053, 2021.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020.

[4] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[5] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, pages 1–38, 2022.

[6] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 762–770, 2022.

[7] Xiaoke Hao, Jie Li, Yingchun Guo, Tao Jiang, and Ming Yu. Hypergraph neural network for skeleton-based action recognition. *IEEE Transactions on Image Processing*, 30:2263–2275, 2021.

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[9] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8545–8552, 2019.

[10] Linguo Li, Minsi Wang, Bingbing Ni, Hang Wang, Jiancheng Yang, and Wenjun Zhang. 3d human action representation learning via cross-view consistency pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4741–4750, 2021.

[11] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2490–2498, 2020.

[12] 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 Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2684–2701, 2019.

[13] Yang Liu, Keze Wang, Lingbo Liu, Haoyuan Lan, and Liang Lin. Tcgl: Temporal contrastive graph for self-supervised video representation learning. *IEEE Transactions on Image Processing*, 31:1978–1993, 2022.

[14] Yuanzhong Liu, Junsong Yuan, and Zhigang Tu. Motion-driven visual tempo learning for video-based action recognition. *IEEE Transactions on Image Processing*, 2022.

[15] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544, 2016.

[16] Qiang Nie, Ziwei Liu, and Yunhui Liu. Unsupervised 3d human pose representation with viewpoint and pose disentanglement. In *European Conference on Computer Vision*, pages 102–118, 2020.

[17] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[18] 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*, 569:90–109, 2021.

[19] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[20] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016.

[21] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019.

[22] Xiangbo Shu, Jinhui Tang, Guo-Jun Qi, Wei Liu, and Jian Yang. Hierarchical long short-term concurrent memory for

human interaction recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):1110–1118, 2019.

[23] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9631–9640, 2020.

[24] Yukun Su, Guosheng Lin, and Qingyao Wu. Self-supervised 3d skeleton action representation learning with motion consistency and continuity. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13328–13338, 2021.

[25] Fida Mohammad Thoker, Hazel Doughty, and Cees GM Snoek. Skeleton-contrastive 3d action representation learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1655–1663, 2021.

[26] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[28] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29:9532–9545, 2020.

[29] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *European Conference on Computer Vision*, pages 504–521, 2020.

[30] Jiadai Wang, Jiajia Liu, and Nei Kato. Networking and communications in autonomous driving: A survey. *IEEE Communications Surveys & Tutorials*, 21(2):1243–1274, 2018.

[31] Peng Wang, Jun Wen, Chenyang Si, Yuntao Qian, and Liang Wang. Contrast-reconstruction representation learning for self-supervised skeleton-based action recognition. *arXiv preprint arXiv:2111.11051*, 2021.

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[33] Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *arXiv preprint arXiv:2104.07713*, 2021.

[34] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[35] Yu-Hui Wen, Lin Gao, Hongbo Fu, Fang-Lue Zhang, Shihong Xia, and Yong-Jin Liu. Motif-gcns with local and non-local temporal blocks for skeleton-based action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[36] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[37] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019.

[38] Shihao Xu, Haocong Rao, Xiping Hu, Jun Cheng, and Bin Hu. Prototypical contrast and reverse prediction: Unsupervised skeleton based action recognition. *IEEE Transactions on Multimedia*, 2021.

[39] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI Conference on Artificial Intelligence*, pages 7444–7452, 2018.

[40] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020.

[41] Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C Kot. Skeleton cloud colorization for unsupervised 3d action representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13423–13433, 2021.

[42] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6548–6557, 2020.

[43] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[44] Yisheng Zhu and Guangcan Liu. Fine-grained action recognition using multi-view attentions. *The Visual Computer*, 36(9):1771–1781, 2020.