Skeleton Cloud Colorization for Unsupervised 3D Action Representation Learning

Siyuan Yang1  Jun Liu2*  Shijian Lu1  Meng Hwa Er1  Alex C. Kot1
1Nanyang Technological University  2Singapore University of Technology and Design
siyuan005@e.ntu.edu.sg  jun_liu@sutd.edu.sg  {Shijian.Lu, emher, eackot}@ntu.edu.sg

Abstract

Skeleton-based human action recognition has attracted increasing attention in recent years. However, most of the existing works focus on supervised learning which requiring a large number of annotated action sequences that are often expensive to collect. We investigate unsupervised representation learning for skeleton action recognition, and design a novel skeleton cloud colorization technique that is capable of learning skeleton representations from unlabeled skeleton sequence data. Specifically, we represent a skeleton action sequence as a 3D skeleton cloud and colorize each point in the cloud according to its temporal and spatial orders in the original (unannotated) skeleton sequence data. Leveraging the colorized skeleton point cloud, we design an auto-encoder framework that can learn spatial-temporal features from the artificial color labels of skeleton joints effectively. We evaluate our skeleton cloud colorization approach with action classifiers trained under different configurations, including unsupervised, semi-supervised and fully-supervised settings. Extensive experiments on NTU RGB+D and NW-UCLA datasets show that the proposed method outperforms existing unsupervised and semi-supervised 3D action recognition methods by large margins, and it achieves competitive performance in supervised 3D action recognition as well.

1. Introduction

Human action recognition is a fast developing area due to its wide applications in human-computer interaction, video surveillance, game control, etc. According to the types of input data, human action recognition can be grouped into different categories such as RGB-based [2, 35, 38, 42, 47, 49, 55, 56], depth-based [26, 28, 45] and 3D skeleton-based [1, 9, 17, 18, 20, 21, 31], etc. Among these types of inputs, 3D skeleton data, which represents a human body by the locations of keypoints in the 3D space, has attracted increasing attention in recent years. Compared with RGB videos or depth data, 3D skeleton data encodes high-level representations of human behaviors and it is generally lightweight and robust to the variations in appearances, surrounding distractions, viewpoint changes, etc. Additionally, skeleton sequences can be easily captured by depth sensors and thus a large number of supervised methods have been designed to learn spatio-temporal representations for skeleton based action recognition.

Deep neural networks have been widely studied to model the spatio-temporal representation of skeleton sequences under supervised scenarios [9, 20, 21, 31]. For example, Recurrent Neural Network (RNN) has been explored for modelling skeleton actions since it can capture temporal relations well [6, 20, 21, 29, 52]. Convolutional Neural Networks (CNNs) have also been explored to build skeleton-based recognition frameworks by converting joint coordinates to 2D maps [5, 9, 13, 43]. In addition, Graph Convolution Networks (GCNs) have attracted increasing attention due to their outstanding performance [27, 31, 48]. However, all these methods are supervised which require a large number of labelled training samples that are often costly to collect. How to learn effective feature representations with minimal annotations thus becomes critically important. To the best of our knowledge, only a few works [10, 14, 19, 37, 54] explored representation learning from unlabelled skeleton data for the task of action recognition, where the major approach is to reconstruct skeleton sequences from the encoded features via certain encoder-decoder structures. Unsupervised skeleton-based action representation learning remains a great challenge.

In this work, we propose to represent skeleton sequences as a 3D skeleton cloud, and design an unsupervised representation learning scheme that learns features from spatial and temporal color labels. We treat a skeletal sequence as a spatial-temporal skeleton cloud by stacking the skeleton data of all frames together, and colorize each point in the cloud according to its temporal and spatial orders in the original skeleton sequence. Specifically, we learn spatial-temporal features from the corresponding joints’ colors by
leveraging a point-cloud based auto-encoder framework as shown in Fig. 1. By repainting the whole skeleton cloud, our network can achieve unsupervised skeleton representation learning successfully by learning both spatial and temporal information from unlabeled skeleton sequences.

The contributions of this paper are threefold. First, we formulate unsupervised action representation learning as a 3D skeleton cloud repainting problem, where each skeleton sequence is treated as a skeleton cloud and can be directly processed with a point cloud auto-encoder framework. Second, we propose a novel skeleton cloud colorization scheme that colorizes each point in the skeleton cloud based on its temporal and spatial orders in the skeleton sequence. The color labels ‘fabricate’ self-supervision signals which boost unsupervised skeleton action representation learning significantly. Third, extensive experiments show that our method outperforms state-of-the-art unsupervised and semi-supervised skeleton action recognition methods by large margins, and its performance is also on par with supervised skeleton-based action recognition methods.

To the best of our knowledge, this is the first work that converts the unsupervised skeleton representation learning problem into a novel skeleton cloud repainting task.

2. Related work

Skeleton-based action recognition. Skeleton-based action recognition has attracted increasing interest recently. Unlike traditional methods that design hand-craft features [8, 39, 40, 46], deep-learning based methods employ Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Graph Convolution Networks (GCNs) to learn skeleton-sequence representation directly. Specifically, RNNs have been widely used to model temporal dependencies for skeleton-based action recognition. For example, [6] uses a hierarchical RNN model to represent human body structures and temporal dynamics of the body joints. [20, 21] proposes a 2D Spatio-Temporal LSTM framework to employ the hidden sources of action related information over both spatial and temporal domains concurrently. [52] adds a view-adaptation scheme to the LSTM to regulate the observation viewpoints.

CNN-based methods [5, 9, 13, 43] have also been proposed for skeleton action recognition. They usually transform the skeleton sequences to skeleton maps of same target size and then use CNNs to learn the spatial and temporal dynamics. For example, [5, 13] transform a skeleton sequence to an image by treating the joint coordinate (x,y,z) as the R, G, and B channels of a pixel. [9] transforms the 3D skeleton data to three skeleton clips for robust action feature learning. [43] presents a “scene flow to action map” representation for action recognition with CNNs.

Inspired by the observation that human 3D-skeleton is naturally a topological graph, Graph Convolutional...
Networks (GCN) have attracted increasing attention in skeleton-based action recognition. For example, [48] presents a spatial-temporal GCN to learn both spatial and temporal patterns from skeleton data. [31] uses a non-local method with the spatial-temporal GCN to boost performance. [30] uses bone features for skeleton-based action recognition. [27] recognizes actions by searching for different graphs at different layers via neural architecture search.

Though the aforementioned methods achieve very impressive performance, they are all supervised requiring large amount of labelled data which is prohibitively time-consuming to collect. In this work, we study unsupervised representation learning in skeleton-based action recognition which mitigates the data labelling constraint greatly.

**Unsupervised representation learning for action recognition.** Unsupervised action recognition aims to learn effective feature representations by predicting future frames of input sequences or by re-generating the sequences. Most existing methods focus on RGB videos or RGB-D videos. For example, [36] uses a LSTM based Encoder-Decoder architecture to learn video representations [23] uses a RNN based encoder-decoder framework to predict the sequences of flows computed with RGB-D modalities. [15] uses unlabeled video to learn view-invariant video representations.

Unsupervised skeleton-based action recognition was largely neglected though a few works attempt to address this challenging task very recently. For example, [54] presents a GAN encoder-decoder to re-generate masked input sequences. [10] adopts a hierarchical fusion approach to improve human motion generation. [37] presents a decoder-weakening strategy to drive the encoder to learn discriminative action features.

The aforementioned methods all process skeleton sequences frame by frame and extract temporal features from ordered sequences. We instead treat a skeleton sequence as a novel colored skeleton cloud by stacking human joints of each frame together. We design a novel skeleton colorization scheme and leverage the color information for unsupervised spatial-temporal representation learning.

### 3. Method

In this section, we present our skeleton cloud colorization representation learning method that converts the skeleton sequence to a skeleton cloud, and colorizes each point in the cloud by its spatial-temporal properties. In particular, we present how to construct the skeleton cloud in Section 3.1 and describe the colorization step in Section 3.2. Repainting pipeline and training details are described in Section 3.3 and Section 3.4, respectively.

#### 3.1. Data Processing

Given a skeleton sequence $S$ under the global coordinate system, the $j^{th}$ skeleton joint in the $t^{th}$ frame is denoted as $v_{t,j} = [x_{t,j}, y_{t,j}, z_{t,j}], t \in (1, \cdots, T), j \in (1, \cdots, J)$, where $T$ and $J$ denote the number of frames and body joints, respectively. Generally, skeleton data is defined as a sequence and the set of joints in the $t^{th}$ frame are denoted as $V_t = \{v_{t,j}|j = 1, \ldots, J\}$. We propose to treat all the joints in a skeleton sequence as a whole by stacking all frames' data together, and Fig. 1 illustrates the stacking framework. We name the stacked data as skeleton cloud and denote it by $P_r = \{v_t,j = [x_{t,j}, y_{t,j}, z_{t,j}]|t = 1, \ldots, T; j = 1, \ldots, J\}$. The obtained 3D skeleton cloud therefore consists of $N = T \times J$ 3D points in total. We use $P_r$ to denote raw skeleton cloud so as to differentiate it from the colorized clouds to be described later.

#### 3.2. Skeleton Cloud Colorization

Points within our skeleton cloud are positioned with 3D spatial coordinates ($x, y, z$) which is similar to normal point cloud that consists of unordered points. The spatial relation and temporal dependency of skeleton cloud points are crucial in skeleton-based action recognition, but they are largely neglected in the aforementioned raw skeleton cloud data. We propose an innovative skeleton cloud colorization method to exploit the spatial relation and temporal dependency of skeleton cloud points for unsupervised skeleton-based action representation learning.

**Temporal Colorization:** Temporal information is critical in action recognition. To assign each point in the skeleton cloud a temporal feature, we colorize the skeleton cloud points according to their relative time order (from 1 to $T$) in the original skeleton sequence. Different colorization schemes have been reported and here we adopt the colorization scheme that uses 3 RGB channels [4] as illustrated in Fig. 2. The generated color is actually relatively linear under this color scheme. Hence, points from adjacent frames are assigned with similar color under this distribution.
A colored skeleton cloud can be denoted by original 3D coordinate information, the temporally colored with red, green and blue channels (RGB channels) a point will have a 3-channels features that can be visualized with this temporal-index based colorization scheme, each point will have a 3-channels features that can be visualized with red, green and blue channels (RGB channels) to represent its temporal information. Together with the original 3D coordinate information, the temporally colorized skeleton cloud can be denoted by \( P_t = \{ v_{t,j}^p, [x_{t,j}, y_{t,j}, z_{t,j}, r_{t,j}^r, g_{t,j}^g, b_{t,j}^b] | t = 1, ..., T; j = 1, ..., J \}. \)

**Spatial Colorization:** Besides temporal information, spatial information is also very important for action recognition. We employ similar colorization scheme to colorize spatial information as in NTU RGB+D [29] which is important to the skeleton action recognition. We therefore propose a person-level colorization scheme for action recognition.

We focus on the scenarios that human interactions involve two persons only, and apply different colors to the points of different persons. Specifically, we encode the first person’s joints with red color and the second person’s joints with blue color as illustrated in Fig. 4. The person-level colored clouds can thus be denoted by \( P_p = \{ v_{t,j,n}^p, [x_{t,j,n}, y_{t,j,n}, z_{t,j,n}, 1, 0, 0] | t = 1, ..., T; j = 1, ..., J; n = 1 \} \cup \{ v_{t,j,n}^p, [x_{t,j,n}, y_{t,j,n}, z_{t,j,n}, 0, 0, 1] | t = 1, ..., T; j = 1, ..., J; n = 2 \}. \)

**Person-level Colorization:** Human actions contain rich person interaction information as in NTU RGB+D [29] which is important to the skeleton action recognition. We therefore propose a person-level colorization scheme for action recognition.

We focus on the scenarios that human interactions involve two persons only, and apply different colors to the points of different persons. Specifically, we encode the first person’s joints with red color and the second person’s joints with blue color as illustrated in Fig. 4. The person-level colored clouds can thus be denoted by \( P_p = \{ v_{t,j,n}^p, [x_{t,j,n}, y_{t,j,n}, z_{t,j,n}, 1, 0, 0] | t = 1, ..., T; j = 1, ..., J; n = 1 \} \cup \{ v_{t,j,n}^p, [x_{t,j,n}, y_{t,j,n}, z_{t,j,n}, 0, 0, 1] | t = 1, ..., T; j = 1, ..., J; n = 2 \}. \)

**Spatial Colorization:** Besides temporal information, spatial information is also very important for action recognition. We employ similar colorization scheme to colorize spatial information as in NTU RGB+D [29] which is important to the skeleton action recognition. We therefore propose a person-level colorization scheme for action recognition.

**Spatial Colorization:** Besides temporal information, spatial information is also very important for action recognition. We employ similar colorization scheme to colorize spatial information as in NTU RGB+D [29] which is important to the skeleton action recognition. We therefore propose a person-level colorization scheme for action recognition.

Given a raw skeleton cloud, the three colorization schemes thus construct three colorized skeleton clouds \( P_t, P_s, P_p \).
3.3. Repainting Pipeline

Inspired by the success of self-supervised learning, our goal is to extract the temporal, spatial, and interactive information by learning to repaint the raw skeleton cloud \( P_r \) in self-supervised manner. As illustrated in Fig. 1, we use colorized skeleton clouds (temporal-level \( P_{r_{\tau}} \), spatial-level \( P_s \), and person-level \( P_p \)) as three kinds of self-supervision signals, respectively. The framework consists of an encoder \( E(.) \) and a decoder \( D(.) \). Since we have three colorization schemes, we have three pairs of encoders \( (E_{\tau}(.), E_s(.), \text{and } E_p(.) ) \) and decoders \( (D_{\tau}(.), D_s(.), \text{and } D_p(.)) \). Below we use the temporal colorization stream as the example to explain the model architecture and the training process.

Model Architecture: As mentioned in Section 3.2, the obtained skeleton cloud format is similar to that of the normal point cloud. We therefore adopt DGCNN [44] (designed for point cloud classification and segmentation) as the backbone of our framework and use the modules before the fully-connected (FC) layers to build our encoder 1.

In addition, we adopt the decoder 1 of FoldingNet [50] as the decoder of our network architecture. Since the input and output of FoldingNet are all \( N \times 3 \) matrices with 3D positions \((x, y, z)\) only, we enlarge the feature dimension to 6 to repaint both the position and color information. Assuming that the input is the raw point set \( P_r \) and the obtained repainted point set is \( \widehat{P}_r = D_{\tau}(E_{\tau}(P_r)) \), the repainting error between the ground truth temporal colorization \( P_{r_{\tau}} \) and the repainted \( \widehat{P}_r \) is computed by using the Chamfer distance:

\[
d_{CH}(P_{r_{\tau}}, \widehat{P}_r) = \max \{ A, B \} \quad \text{where} \quad \begin{align*}
A &= \frac{1}{|P_{r_{\tau}}|} \sum_{v_{r_{\tau}} \in P_{r_{\tau}}} \min_{v_{\tau} \in \widehat{P}_r} \| v_{r_{\tau}} - \widehat{v}_{\tau} \|_2 \quad (7) \\
B &= \frac{1}{|\widehat{P}_r|} \sum_{\widehat{v}_{\tau} \in \widehat{P}_r} \min_{v_{r_{\tau}} \in P_{r_{\tau}}} \| \widehat{v}_{\tau} - v_{r_{\tau}} \|_2 \quad (8)
\end{align*}
\]

where the \( \min_{v_{\tau} \in \widehat{P}_r} \| v_{r_{\tau}} - \widehat{v}_{\tau} \|_2 \) enforces that any 3D point \( v_{r_{\tau}} \) in temporally colorized skeleton cloud \( P_{r_{\tau}} \) has a matched 3D point \( \widehat{v}_{\tau} \) in the repainted point cloud \( \widehat{P}_r \). The term \( \min_{v_{r_{\tau}} \in P_{r_{\tau}}} \| \widehat{v}_{\tau} - v_{r_{\tau}} \|_2 \) enforces the matching vice versa.

The max operation enforces that the distance from \( P_{r_{\tau}} \) to \( \widehat{P}_r \) and vice versa need to be small concurrently.

By using the Chamfer distance, the encoder \( E_{\tau}(.) \) and decoder \( D_{\tau}(.) \) are forced to learn temporal dependency via the proposed temporal repainting scheme that reconstructs temporal-order colors. Similarly, the encoder \( E_s(.) \) and decoder \( D_s(.) \) will learn useful spatial relations during spatial repainting, and the encoder \( E_p(.) \) and decoder \( D_p(.) \) are pushed to distinguish the person index and learn interactive information. It is non-trivial to repaint \( P_r \) to colorized skeleton clouds. To balance the color repainting and unsupervised feature learning, we uniformly sample half of points in \( P_{r_{\tau}} \) for colorization. Specifically, in the temporal colorization stream, points corresponding to odd-order frames are colored, while the rest is not. In the spatial colorization, points with odd-index joints are colored based on joints orders, and even-indexed joints are not colored.

3.4. Training for Skeleton Action Recognition

After the self-supervised repainting, we obtain three encoders (i.e., \( E_{\tau}(.) \), \( E_s(.) \), and \( E_p(.) \)) that capture meaningful temporal, spatial, interaction features, respectively. With the feature representations from the three encoders, we include a simple linear classifier \( f(.) \) on top of the encoder to perform action recognition as in [10, 19, 25, 37, 54]. We adopt different settings to train the classifier including unsupervised, semi-supervised, and supervised settings. In the unsupervised setting, the encoder is only trained by the skeleton cloud repainting method, and then we train the linear classifier with the encoder fixed by following previous unsupervised skeleton representation learning works [10, 19, 25, 37, 54]. In the semi-supervised and supervised settings, the encoder is first trained with unsupervised rep-

---

1 Detail structure of encoder and decoder can be found in supplementary.
representation learning and then fine-tuned with the linear classifier as in [19]. We use the standard cross-entropy loss as the classification loss $L_{cls}$.

### 4. Experiments

We conducted extensive experiments over two publicly accessible datasets including NTU RGB+D [29], and Northwestern-UCLA [41]. The experiments aim to evaluate whether our skeleton cloud colorization scheme can learn effective unsupervised feature representations for the task of skeleton action recognition. We therefore evaluate different experimental settings including unsupervised and semi-supervised as well as supervised.

#### 4.1. Datasets

**NTU RGB+D [29]:** NTU RGB+D consists of 56880 skeleton action sequences which is the most widely used dataset for skeleton-based action recognition research. In this dataset, action samples are performed by 40 volunteers and categorized into 60 classes. Each sample contains an action and is guaranteed to have at least two subjects as captured by three Microsoft Kinect v2 cameras from different views. The authors of this dataset recommend two benchmarks: (1) cross-subject (CS) benchmark where training data comes from 20 subjects and testing data comes from the other 20 objects; (2) cross-view (CV) benchmark where training data comes from camera views 2 and 3, and testing data comes from camera view 1.

**Northwestern-UCLA (NW-UCLA) [41]:** This dataset is captured by Kinect v1 cameras and it contains 1494 samples performed by 10 volunteers. It contains 10 action classes, and each body has 20 skeleton joints. Following the evaluation protocol in [41], the training set consists of samples from camera views 1 and 2, and the rest samples from camera view 3 form the testing set.

#### 4.2. Implementation Details

For NTU RGB+D, we adopt the pre-processing in [31] and uniformly sampled $T = 40$ frames from each skeleton sequence. The sampled skeleton sequences are constructed to a 2000-points skeleton cloud. For NW-UCLA, we adopt the pre-processing in [37] and uniformly sampled $T = 50$ frames from the skeleton sequences. The skeleton cloud has 1000 points.

In the unsupervised feature learning phase, we use Adam optimizer and set the initial learning rate at $1e^{-5}$ and reduce it to $1e^{-7}$ with cosine annealing. The dimension of the encoder output is 1024, and the batch size is 24 for both datasets. The training lasts for 150 epochs. In the classifier training phase, we use SGD optimizer with Nesterov momentum (0.9). We set the initial learning rate at 0.001 and reduce it to $1e^{-5}$ with cosine annealing. The batch size for NTU RGB+D and NW-UCLA is 32 and 64 and the training lasts for 100 epochs. We implement our method using PyTorch and all experiments were conducted on Tesla P100 GPUs with CUDA 10.1.
4.3. Comparison with the state-of-the-art methods

We conduct extensive experiments under three settings including unsupervised learning, semi-supervised learning and supervised learning. We also study three skeleton cloud colorization configurations: 1) ‘T-Stream (TS)’ that uses temporally colorized skeleton cloud as self-supervision; 2) ‘S-Stream (SS)’ that uses spatially colorized skeleton cloud as self-supervision; and 3) ‘P-Stream (PS)’ that uses person-level colorized cloud as self-supervision.

Unsupervised Learning. In the unsupervised setting, the feature extractor (i.e. the encoder $E(.)$) is trained with our proposed skeleton cloud colorization unsupervised representation learning approach. Then the feature representation is evaluated by the simple linear classifier $f(.)$, which is trained on the top of the frozen encoders $E(.)$. Such experimental setting for unsupervised learning has been widely adopted and practised in prior studies [10, 19, 25, 37, 54]. Here for fair comparisons, we use the same setting as these prior works.

We compare our skeleton cloud colorization method with prior unsupervised methods on NTU RGB+D and NW-UCLA datasets as shown in Tables 1 and 2. It can be seen that our proposed temporal colorization encoding (i.e. ‘TS’ colorization) outperforms prior unsupervised methods on NTU RGB+D dataset, especially under the cross-subject evaluation protocol. Additionally, the proposed ‘TS + SS’ colorization and ‘TS + SS + PS’ colorization outperform the state-of-the-art obviously on both cross subject and cross view protocols. For NW-UCLA, our method outperforms the state-of-the-art consistently for both configurations ‘TS’ and ‘TS + SS’ as shown in Table 2.

Semi-supervised Learning. We evaluate semi-supervised learning with the same protocol as in [19, 34] for fair comparisons. Under the semi-supervised setting, the encoder $E(.)$ is first pre-trained with colorized skeleton clouds and then jointly trained with the linear classifier $f(.)$ with a small ratio of action annotations. Following [19, 34], we derive labelled data by uniformly sampling 1%, 5%, 10%, 20%, 40% data from the training set of NTU RGB+D dataset, and 1%, 5%, 10%, 15%, 30%, 40% data from the training set of NW-UCLA dataset, respectively.

Tables 3 and 4 show experimental results on NTU RGB+D and NW-UCLA datasets, respectively. As Table 3 shows, our method performs better than the state-of-the-art consistently for all three configurations (‘TS’, ‘TS + SS’, and ‘TS + SS + PS’) on NTU RGB+D. For NW-UCLA dataset, our proposed temporal colorization encoding (‘TS’) performs better than the state-of-the-art as shown in Table 4. Additionally, our ‘TS + SS’ colorization scheme outperforms the state-of-the-art by large margins.

Supervised Learning. Following the supervised evaluate-
ation protocol in [19], we pre-train the encoder with our unsupervised skeleton colorization method and fine-tune the encoder and classifier by using labeled training data. Tables 5 and 6 show experimental results. We can observe that our method achieves superior performance on NW-UCLA, and it performs much better than the previous “Unsupervised Pre-trained” methods [15, 19] (first train under the unsupervised feature learning and then fine-tune the framework with the labeled data). On the large-scale NTU RGB+D, our method outperforms the “Unsupervised Pre-trained” works [15, 19] by large margins. Though our framework is not designed for supervised setting, its performance is even comparable to state-of-the-art supervised methods.

4.4. Ablation Study

Effectiveness of Our Skeleton Colorization: We verify the effectiveness of our skeleton cloud colorization on all three learning settings including unsupervised learning, semi-supervised learning, and fully-supervised learning. We compare our method with three baselines: 1) Baseline-U: it only trains the linear classifier and freezes the encoder which is randomly initialized; 2) Baseline-Semi: the encoder is initialized with random weight instead of pre-training by our unsupervised representation learning; 3) Baseline-S: the same as Baseline-Semi. We train the encoder and linear classifier jointly with action labels. The input for these three baselines is the raw skeleton cloud without color label information.

Tables 7, 8, and 9 show experimental results. It can be seen that all the three colorization strategies (i.e. temporal-level, spatial-level, and person-level) achieve significant performance improvement as compared with the baseline, demonstrating the effectiveness of our proposed colorization technique. Though the person-level colorization stream does not perform as well as the other two streams on the NTU RGB+D, it improves the overall performance while collaborating with the other two.

Effectiveness of Colorization Ratio: As mentioned in Section 3.3, we need to balance between the repainting and the spatial-temporal ordered feature learning. We observe that the unsupervised performance improves from 65.7% to 71.6% on the temporal stream on the NTU RGB+D (Cross-subject setting) when 50% points with the color information is provided, demonstrating the effectiveness of our proposed colorization scheme. Detailed comparisons can be found in the supplementary materials.

5. Conclusion

In this paper, we address unsupervised representation learning in skeleton action recognition, and design a novel skeleton cloud colorization method that is capable of learning skeleton representations from unlabeled data. We obtain colored skeleton cloud representations by stacking skeleton sequences to 3D skeleton cloud and coloring each point according to its temporal and spatial orders in the skeleton sequences. Additionally, spatio-temporal features are learned effectively from the corresponding joints’ colors from unlabeled data. The experiments demonstrate that our proposed method achieves superior unsupervised action recognition performance.

Acknowledgement. This work was done at Rapid-Rich Object Search (ROSE) Lab, Nanyang Technological University. This research is supported in part by the NTU-PKU Joint Research Institute (a collaboration between the Nanyang Technological University and Peking University that is sponsored by a donation from the Ng Teng Fong Charitable Foundation), the Science and Technology Foundation of Guangzhou Huangpu Development District under Grant 2019GH16, and China-Singapore International Joint Research Institute under Grant 206-A018001. This research work is also partially supported by the SUTD project PIE-SGP.AI-2020-02.
References


Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C Kot. Collaborative learning of gesture recognition and 3d hand pose estimation with multi-order feature analysis. In


