MS^{2}L: Multi-Task Self-Supervised Learning for Skeleton Based Action Recognition

Lilang Lin
Wangxuan Institute of Computer Technology,
Peking University
linlilang@pku.edu.cn

Sijie Song
Wangxuan Institute of Computer Technology,
Peking University
ssj940920@pku.edu.cn

Wenhan Yang
Wangxuan Institute of Computer Technology,
Peking University
yangwenhan@pku.edu.cn

Jiaying Liu*
Wangxuan Institute of Computer Technology,
Peking University
liujiaying@pku.edu.cn

ABSTRACT

In this paper, we address self-supervised representation learning from human skeletons for action recognition. Previous methods, which usually learn feature presentations from a single reconstruction task, may come across the overfitting problem, and the features are not generalizable for action recognition. Instead, we propose to integrate multiple tasks to learn more general representations in a self-supervised manner. To realize this goal, we integrate motion prediction, jigsaw puzzle recognition, and contrastive learning to learn skeleton features from different aspects. Skeleton dynamics can be modeled through motion prediction by predicting the future sequence. And temporal patterns, which are critical for action recognition, are learned through solving jigsaw puzzles. We further regularize the feature space by contrastive learning. Besides, we explore different training strategies to utilize the knowledge from self-supervised tasks for action recognition. We evaluate our multi-task self-supervised learning approach with action classifiers trained under different configurations, including unsupervised, semi-supervised and fully-supervised settings. Our experiments on the NW-UCLA, NTU RGB+D, and PKUMMD datasets show remarkable performance for action recognition, demonstrating the superiority of our method in learning more discriminative and general features. Our project website is available at https://langlandslin.github.io/projects/MSL/.

CCS CONCEPTS

• **Computing methodologies → Computer vision**.

KEYWORDS

Action recognition, multi-task, self-supervised learning

ACM Reference Format:
Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. 2020. MS^{2}L: Multi-Task Self-Supervised Learning for Skeleton Based Action Recognition. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413548

1 INTRODUCTION

Action recognition is a fundamental yet challenging problem in computer vision research. Demands on action recognition are growing rapidly to facilitate the applications such as video surveillance, human-computer interaction, video understanding.

In the past few years, many works [10, 38, 42] have been developed on recognition based on RGB videos and achieved many significant results. However, processing RGB videos can be very time-consuming and require a large storage space. Another data modality, human skeletons, which represent a person by the 3D coordinate positions of skeletal joints, draw much attention due to the light-weight representations, the robustness to variations of viewpoints, appearances, and surrounding distractions. Furthermore, skeleton sequences can be regarded as a high-level representation for human behavior, which attracts many researchers to study skeleton-based action recognition [4, 29, 30, 43, 45, 46, 49]. Leveraging the merits of recurrent layers, many works [4, 30, 45, 46, 49] build their framework based on Recurrent Neural Networks (RNN) to model temporal evolution of different actions. Considering that skeletons are naturally with graph structures, graph convolution networks (GCN) are applied in skeleton-based action recognition and show outstanding performance [29, 43]. However, these models are trained in a fully-supervised manner and require massive labeled training examples. Besides, annotating training data can be tedious and expensive. How to effectively learn feature representations from skeleton data with less annotation efforts remains a concerned problem.

Recently, there are a few attempts [32, 48] exploring representation learning from unlabeled skeleton data. These models achieve feature learning by an encoder-decoder structure, the input of which is masked or original skeleton sequences, and the goal is to reconstruct skeleton sequences from the encoded features. We argue

[*] Corresponding author. This work was supported by the National Key R&D Program of China under Grand No.2018AAA0102700, National Natural Science Foundation of China under contract No.61772043, and Beijing Natural Science Foundation under contract No.4192025.

MM '20, October 12–16, 2020, Seattle, WA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413548

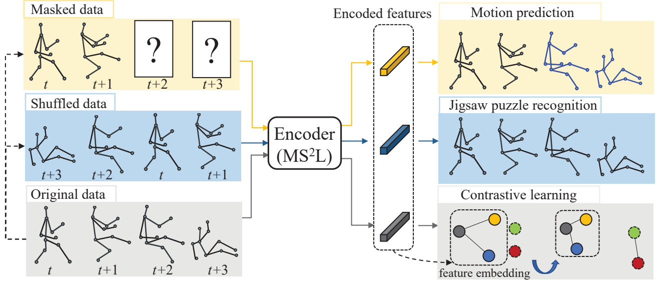


Figure 1: Our proposed approach (MS^{2}L) learns more general and discriminative features benefiting from multiple self-supervised tasks.

there are two potential issues in the previous work: (1) The skeleton reconstruction focuses more on detailed skeleton coordinates, ignoring the high-level spatio-temporal information which is critical for action recognition. (2) Learning from a single task could lead to overfitting for the specific task [26]. Therefore, the learned features from previous works may not be discriminative and general enough for recognizing skeleton sequences.

To address the aforementioned issues, we propose a novel self-supervised learning method by optimizing multiple task simultaneously. As shown in Figure 1, we focus on combining different tasks to make the representations more diverse and describe different aspects of information. In our paper, we design three tasks, *i.e.*, the generation task for motion prediction, the classification task for solving jigsaw puzzles, and contrastive learning based on skeleton transformations. We aim to learn skeleton dynamics from motion prediction, model temporal evolution through solving jigsaw puzzles, and further regularize the feature space by contrastive learning. To fully utilize the knowledge learned from self-supervised learning tasks and facilitate action recognition, we explore different training strategies to train the action classifier. We provide comprehensive evaluation and analysis in our experiments to demonstrate the superiority of our proposed self-supervised learning approach.

In summary, our contributions include the following aspects:

- We propose a multi-task self-supervised learning framework for skeleton based action recognition. We aim to learn comprehensive and general feature representations, benefiting from motion prediction, jigsaw puzzle recognition and contrastive learning.
- To transfer the knowledge learned from self-supervised learning, we explore different training strategies, *i.e.*, moving pre-training and jointly training, towards better action recognition performance.
- Exhaustive experiments on different datasets validate the capacity of our representations learned by self-supervised tasks, which show superiority in action recognition under different configurations, including the unsupervised, semi supervised and fully-supervised learning settings, as well as transfer learning.

The rest of the paper is organized as follows: Sec. 2 reviews previous works on self-supervised learning and skeleton-based action recognition. Sec. 3 introduces our proposed self-supervised learning approach and training strategies in detail. We present our experiment results and analysis in Sec. 4. Concluding remarks are given in Sec. 5.

2 RELATED WORK

In this section, we first introduce related work on self-supervised learning, and then give a brief review on skeleton-based action recognition.

2.1 Self-Supervised Learning

Self-supervised learning aims to learn feature representations from a huge amount of unlabelled data. It has been verified that self-supervised pre-training can help supervised learning [6] and it has a variety of applications in a broad range of computer vision topics [14, 25]. Self-supervised learning is usually achieved by pretext tasks, which utilize easy-to-obtain automatically generated supervision without human expensive annotation.

Many efforts have been devoted to designing pretext tasks to learn image representations from unlabelled image data [3, 8, 23, 24, 41, 47]. Doersh *et al.* [3] proposed to train a convolutional neural network to reorder perturbed image patches. Following the idea, the works in [23, 24, 41] predict a permutation of multiple shuffled image patches to better model spatial relationships, which are called jigsaw puzzles. There are also other pretext tasks, such as colorizing grayscale images [47] or predicting image rotation angles [8, 44]. More recently, Chen *et al.* [1] proposed a visual representation learning method with contrastive learning for contrastive prediction task, which forces on feature representation between positive pairs more similar than those between negative ones.

Recent studies also pay attention to representation learning for sequential data such as videos. A common way is to predict the video frame orders [7, 17, 21] to learn the temporal patterns. To further learn spatio-temporal representations, Vondrick *et al.* [36] proposed a spatio-temporal 3D convolution integrated with a generative adversarial network. Kim *et al.* [16] tackled the problem by solving space-time cubic puzzles inspired by jigsaw puzzles in the image domain. A more recent work [37] improves the spatio-temporal feature representations in a finer granularity by regressing motion and appearance statics along spatial and temporal dimensions. In our work, we use multiple tasks to learn spatial and temporal patterns, respectively. Besides, we apply the contrastive learning to constrain the feature space by sampling positive and negative pairs.

2.2 Skeleton-Based Action Recognition

Early skeleton-based action recognition methods are generally based on hand-crafted features by utilizing the geometry relationships of skeletal joints [9, 20, 34, 35, 38]. Recent methods for skeleton-based action recognition pay more attention to utilizing deep networks as their basic models. Benefiting from the merits of recurrent layers for sequential data, Du *et al.* [5] proposed a pioneer work based on a hierarchical recurrent neural network. Zhu *et al.* [49] explored the co-occurrence of joints by introducing a group sparsity constraint on the recurrent neural network. To more adaptively learn the co-occurrence patterns of skeletal

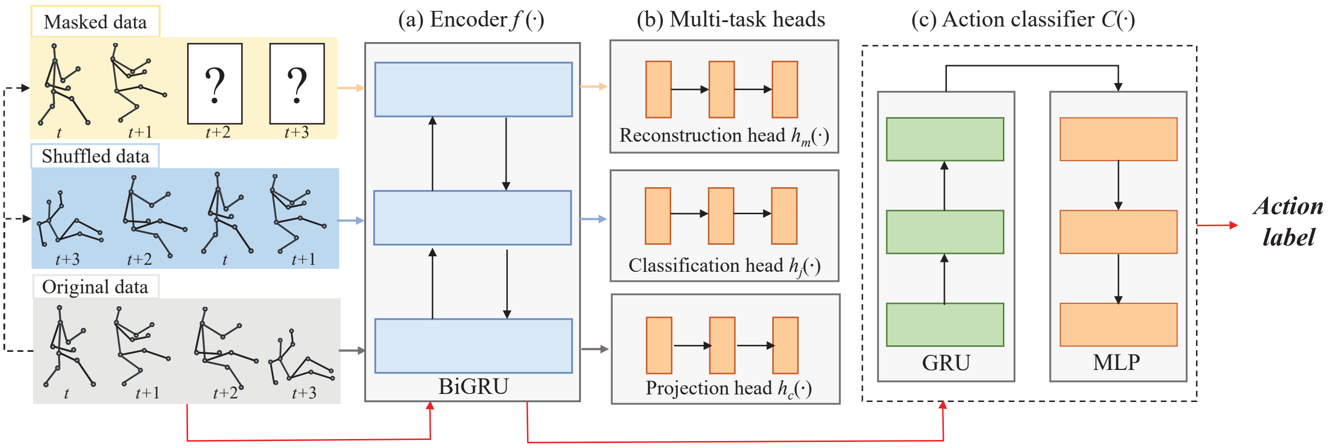


Figure 2: The structure of our network. (a) Encoder. (b) Multi-task heads. (c) Classifier for action recognition. We use yellow, blue and grey arrows to indicate the pipeline for motion prediction, jigsaw puzzle recognition and contrastive learning, respectively. Action recognition is achieved with the red pipeline.

joints, attention-based methods are proposed to automatically select important skeletal joints [30, 31, 46] and video frames [30, 31]. However, recurrent neural networks always suffer from gradient vanishing [12], which may cause optimization problems. Then, convolutional neural networks attract more attention for the skeleton-based action recognition. To adapt the input for convolutional neural network, a new representation for 3D skeleton data is proposed in [15], which converts the problem of action recognition to image classification. To better model the structural information of the skeleton data, Yan *et al.* [43] employed graph convolution networks to represent skeleton data on graphs. To make the graph representation more flexible, attention mechanisms are applied in [28, 29] to adaptively capture discriminative features based on spatial configurations and temporal dynamics.

Though these models have achieved excellent performance on skeleton-based action recognition, they heavily rely on expensive annotation of action sequences. To get rid of action labeling, Zheng *et al.* [48] proposed an unsupervised framework using an encoder-decoder structure to re-generate the masked input sequence. To force the encoder to learn a more informative representation, a more recent work [32] is proposed to improve the encoder and weaken the decoder. However, the models in [32, 48] are learned from a single task, *i.e.*, skeleton sequence reconstruction, leading to limited feature representation ability. To learn more intrinsic features, our work introduces multiple self-supervised tasks to further enhance feature representation learning without the help of human annotations.

3 MULTIPLE SELF-SUPERVISED LEARNING (MS^{2}L)

In this section, we present our self-supervised learning techniques. We first provide a general description of our approach. Then, we introduce specific instantiations of our approach.

3.1 Preliminaries

We focus on the self-supervised feature learning for skeleton data. Then, we apply the learned features on skeleton-based action recognition. Basically, our overall framework consists of an encoder $f(\cdot)$ to extract features from skeleton data, and an action classifier $C(\cdot)$ to assign action labels to the input sequence. Supposing the i^{th} skeleton sequence is $\mathbf{X}^i = \{\mathbf{x}_1^i, \dots, \mathbf{x}_T^i\}$, where $\mathbf{x}_t^i$ represents the t^{th} frame. Then, the result for action recognition is $p^i = C\left(f\left(\mathbf{X}^i\right)\right)$, where p^i is the probability distribution over all the action categories. In our work, our goal is to learn powerful feature representations from the encoder $f(\cdot)$ with self-supervised learning. Besides, we explore different settings and strategies to train the action classifier $C(\cdot)$ with the learned features.

3.2 Multiple Self-Supervised Tasks

We now describe our self-supervised learning techniques. To learn generalizable and robust skeleton features, we consider multiple self-supervised tasks, *i.e.*, the generation task for motion prediction, the classification task for solving video jigsaw puzzles, and contrastive learning based on skeleton transformations. We aim to model skeleton dynamics through motion prediction and learn temporal patterns by solving jigsaw puzzles. Finally, we utilize contrastive learning to further regularize the feature space for more inherent representations. Figure 2 shows the pipeline of our model. The tasks share the encoder $f(\cdot)$ and adopt different heads for different objectives. Next, we present these self-supervised tasks in detail, respectively.

Motion Prediction. Given the past motion sequence, the motion prediction task focuses on forecasting the most likely future poses of a person by modeling skeleton dynamics. Inspired by *Seq2Seq* [33], we apply an encoder-decoder with recurrent layers to achieve the task. The encoder $f(\cdot)$ reads in parts of the input sequences and extracts representations from inputs. The decoder $h_m(\cdot)$, which is shown as a reconstruction head in Figure 2, receives the learned

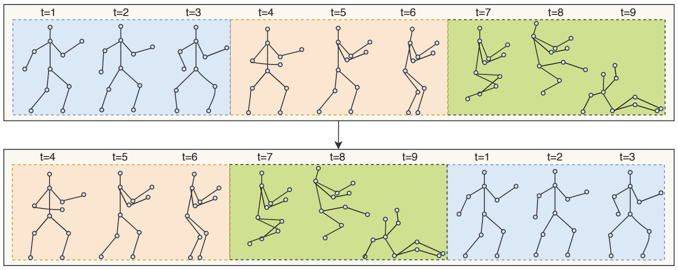


Figure 3: Our method of skeleton jigsaw puzzles. Different color means different segments and we shuffle these segments randomly to create various permutations.

representations and generates sequences to reconstruct the whole input sequences. To avoid overfitting, we augment the original data by injecting random noises into the input sequences. Those random noises are sampled from a Gaussian distribution to avoid the network remember the input sequences.

To formulate motion prediction, recall that the original skeleton sequence is $\mathbf{X}^i = \{\mathbf{x}^i_1, \ldots, \mathbf{x}^i_T\}$, and the masked sequence is $\mathbf{X}^i_m = \{\mathbf{x}^i_1, \ldots, \mathbf{x}^i_{T'} | T' < T\}$. Then, we inject random noise into the input sequence $\mathbf{X}^i_m$ to get the noisy input sequence $\tilde{\mathbf{X}}^i_m$. The future motion sequence is predicted by $\hat{\mathbf{X}}^i_m = h_m\left(f\left(\tilde{\mathbf{X}}^i_m\right)\right)$, where $\hat{\mathbf{X}}^i_m = \{\hat{\mathbf{x}}^i_{T'+1}, \ldots, \hat{\mathbf{x}}^i_T\}$, we use mean square error (MSE) to estimate the parameters of the network as follows:

$$\mathcal{L}_m = \sum_{i=1}^{N} \sum_{t=T'+1}^{T} \|\hat{\mathbf{x}}^i_t - \mathbf{x}^i_t\|_2^2, \quad (1)$$

where N is the batch size.

Jigsaw Puzzle. Solving the problem of jigsaw puzzles aims to predict the correct permutation from the shuffled sequences. In our work, we apply jigsaw puzzles for skeleton sequences in the temporal domain so the network is able to learn temporal patterns. To generate puzzles from the skeleton sequences, each sequence is divided into P segments equally and there are $\frac{T}{P}$ frames in a segment. We shuffle these segments randomly and there are P! ways to shuffle them. The network is trained to predict the correct order of the shuffled segments. An example of jigsaw puzzle can be viewed in Figure 3.

With the shared encoder $f(\cdot)$, we apply a classification head $h_j(\cdot)$ to obtain the classification results to recognize video jigsaw puzzles. Specifically, we use an MLP as our classification head. The task is trained with the loss $\mathcal{L}_j$, which is formulated as cross entropy loss for classification as follows:

$$\mathcal{L}_j = -\sum_{i=1}^{N} y^i \log h_j(f(\mathbf{X}^i_j)), \quad (2)$$

where $\mathbf{X}^i_j$ is the shuffled sequence of original data $\mathbf{X}^i$ and y^i is one-hot vector indicating the action label.

Contrastive Learning. To further regularize the feature learning and encourage the network to learn inherent representations, we adopt contrastive learning by mapping the transformed data into a common feature space. Inspired by *SimCLR* [1], our network learns representations by maximizing cosine similarity between transformed modalities of the same original data. For each original sample, we consider multiple transformations. Specifically, we randomly sample N examples and apply $(M-1)$ kinds of transformation operators to obtain NM samples. Then for each original sample, we can construct (M-1) positive pairs with its transformed samples, and construct negative pairs with other samples.

A projection head $h_c(\cdot)$ is designed to map the encoded sequences into the feature space. Let $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{NM}$ be the feature extracted from the output of encoder $f(\cdot)$, for any integer k from 1 to N, $\mathbf{z}_{(k-1)M+1}$ is the original data and the sequences from $\mathbf{z}_{(k-1)M+2}$ to $\mathbf{z}_{kM}$ are the transformed samples from the original sequence $\mathbf{z}_{(k-1)M+1}$. $\bar{\mathbf{z}}_i = \frac{1}{M}\sum_{j=(i-1)M+1}^{iM} \mathbf{z}_j$ denotes the mean features of original and transformed data for $\mathbf{z}_{(i-1)M+1}$. Similar to recent works [1], we use $sim(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}/\|\mathbf{x}\|_2\|\mathbf{y}\|_2$ define the cosine similarity between $\mathbf{x}$ and $\mathbf{y}$. We define the loss function as follows:

$$\mathcal{L}_c = -\sum_{i=1}^{MN} log \frac{exp(sim(\mathbf{z}_i, \bar{\mathbf{z}}_k))}{\sum_{j=1}^{N} exp(sim(\mathbf{z}_i, \bar{\mathbf{z}}_j))}, \quad (3)$$

where k is $\lceil \frac{i}{M} \rceil$. This extension can adapt to an arbitrary number of transformation operators and gain a better constraint in the feature space. In practice, we adopt two transformation operators in our work. One is temporal masking, and the other is temporal jigsaw, which are actually the input of motion prediction and jigsaw puzzle recognition, respectively.

3.3 Training for Action Recognition

With the feature representations from the encoder $f(\cdot)$, we build an action classifier $C(\cdot)$ on top of the encoder to achieve action recognition. We consider different settings to train the classifiers, including the unsupervised setting, semi-supervised setting and fully-supervised setting. In the unsupervised setting, the encoder is trained only with self-supervised tasks introduced above, and then we train the action classifier independently by optimizing the cross-entropy loss with the encoder fixed. In the semi-supervised and fully-supervised settings, we are allowed to train the encoder and classifier jointly. Here, we explore two different training strategies for the semi-supervised and fully-supervised settings towards better performance for action recognition.

Moving Pretraining Strategy. The previous pretraining method [48] initializes the encoder with learned weights and finetunes the whole network. However, that may cause severe destruction of the extracted features learned from self-supervised tasks. To address the issue, we adopt a novel pretraining scheme, using a linear regularization mechanism to adjust the weights between self-supervised tasks and the action recognition task. This can help stabilize the training process when switching between different tasks.

Specifically, let $\mathcal{L}_{cls}$ be a standard cross-entropy loss for action recognition, $\mathcal{L}_{self} = \mathcal{L}_m + \mathcal{L}_j + \mathcal{L}_c$ is the sum of self-supervised learning losses from motion prediction, jigsaw puzzle recognition and contrastive learning. When we initialize the encoder with the weights trained by self-supervised tasks, our network would take several epochs to perform moving pretrained supervised learning, during which we train the network with self-supervised tasks and supervised learning tasks jointly with a changeable parameter to adjust the proportion of two tasks with a loss as follows:

$$\mathcal{L}_{moving} = \theta \mathcal{L}_{cls} + (1-\theta)\mathcal{L}_{self}, \tag{4}$$

where θ increases from 0 to 1 linearly and is fixed at 1 finally.

Jointly Training Strategy. Another alternative to train for action recognition is to optimize the networks jointly from scratch. Training data are fed into the encoder to extract features and then into their corresponding heads and the action classifier. The loss function can be defined as follows:

$$\mathcal{L}_{joint} = \mathcal{L}_{cls} + \omega \mathcal{L}_{self}, \tag{5}$$

where ω is a non-negative scalar weight to balance the two terms. In practice, ω is set as 1. We will show the action recognition performance with different training strategies and give an analysis on that in the experiments.

4 EXPERIMENT RESULTS

For evaluation, we conduct our experiments on the following three datasets: the North-Western UCLA dataset [39], the NTU RGB+D dataset [27], and the PKUMMD dataset [18]. Our goal is to evaluate whether our feature encoder $f(\cdot)$ trained with the proposed self-supervised learning approach can generate good feature representations for action recognition. Thus, we consider action classifiers trained under different settings (*i.e.*, unsupervised, self-supervised, and fully supervised). We also apply our approach to transfer learning. Finally, we give an ablation study to illustrate the effectiveness of each component in our work.

4.1 Dataset and Settings

North-Western UCLA (NW-UCLA) [39] This dataset is captured by Kinect v1 and contains 1494 videos in 10 action categories performed by 10 subjects. Each body has 20 skeleton joints. There are three views of each action and we use the first two views for training and the third view for testing, which contains $1,018$ videos and 462 videos, respectively.

NTU RGB+D Dataset (NTU) [27] This is a large scale dataset including $56,578$ videos with 60 action labels and 25 joints for each body, including interactions with pairs and individual activities. We test our method under the cross-subject protocol, that the training and testing are split by different subjects, leading to $40,091$ videos for training and $16,487$ videos for testing.

PKU Multi-Modality Dataset (PKUMMD) [18] PKU-MMD is a new large scale benchmark for continuous multi-modality 3D human action understanding and covers a wide range of complex human activities with well annotated information. It contains almost $20,000$ action instances and 5.4 million frames in 52 action categories. Each sample consists of 25 body joints. PKUMMD consists of two subsets, *i.e.*, part I and part II. Part I is an easier version for action recognition, while part II is more challenging with more skeleton noise caused by the large view variation. We conduct experiments under the cross subject protocol on the two subsets, respectively.

To train the network, all the skeleton sequences are temporally down-sampled to 200 frames. For the motion prediction, we add random noise to the former 50 frames and mask the latter 150 frames. For the skeleton jigsaw task, we divide the sequence into 3 segments so there are 6 ways to shuffle the sub-sequences.

The architecture is set as four parts. First, the shared encoder $f(\cdot)$ is a 1-layer bidirectional GRU with 30 units in each layer. The reconstruction head $h_m(\cdot)$ for motion prediction, the classification head $h_j(\cdot)$ for solving jigsaw puzzles, and the projection head $h_c(\cdot)$ for contrastive learning are 1 FC layer ($dim = 60$). The classifier $C(\cdot)$ includes a 1-layer unidirectional GRU with 60 units to be compatible with the dimensions of the output of the encoder $f(\cdot)$ and an MLP for recognition. All networks are initialized with a random uniform distribution.

To optimize our network, Adam optimizer [22] is used and the learning rate declines from 0.01 to 0.0001 with 0.1 decay rate for every 100 iterations. We train the network on one NVIDIA Titan X GPU with a batch size of 32 for NW-UCLA and 128 for NTU RGB+D, PKUMMD datasets, respectively.

4.2 Evaluation and Comparison

In this section, we explore whether the representations learned by our multi-task self-supervised model (MS^{2}L) are meaningful for action recognition. To give a comprehensive and thorough evaluation, we conduct experiments under different settings, including unsupervised, semi-supervised and fully supervised approaches. We also show the comparison results with other state-of-the-art methods, respectively.

Unsupervised Approaches. In the unsupervised setting, the feature extractor, *i.e.*, the encoder $f(\cdot)$, is independently trained with some pretext tasks. Then, the feature representation is evaluated by classifiers. In our experiments, we evaluate feature representations with a linear classifier, which is trained on top of the frozen encoder $f(\cdot)$, and action recognition accuracy is used as a measurement for representation quality. We test the following configurations:

- **MS^{2}L Rand-Unsupervised (MS^{2}L Rand-U)**: We only train the linear classifier and freeze the encoder $f(\cdot)$ which is randomly initialized. We regard this configuration as our baseline.
- **LongT GAN [48]**: This work designs a conditional skeleton inpainting architecture for learning a fix-dimensional representation with additional adversarial training strategies. Specifically, this model uses the feature of the original data and randomly masked skeleton data to recover the original data. And the trained weights of the encoder $f(\cdot)$ can be used for recognition. We construct the network according to the paper.
- **MS^{2}L**: It is our full system, where the encoder $f(\cdot)$ is trained by MS^{2}L independently, then we train the linear classifier with encoder $f(\cdot)$ fixed.

In Table 1, we show the results of the baseline (*MS^{2}L* Rand-U), the prior work *LongT GAN*, and the proposed *MS^{2}L*. As we can see, our approach achieves better performance over random baseline and *LongT GAN*. This improvement verifies that our methods can force the network to extract more effective features. It is worth noting that the feature dimension from our encoder $f(\cdot)$ is 60 while that from *LongT GAN* is 800. Therefore, we achieve better performance with much more compact feature representations compared to *LongT GAN*.

Semi-Supervised Approaches. In semi-supervised learning, the training process utilizes both labeled data and unlabeled data. Generally, the encoder $f(\cdot)$ is pretrained with some pretext tasks with unlabeled data, then jointly trained with the classifier $C(\cdot)$ with

Table 1: Comparison of action recognition results with unsupervised learning approaches.

Models	NW-UCLA	PKUMMD part I	PKUMMD part II	NTU
MS^{2}L Rand-U	60.61	62.80	20.70	41.10
LongT GAN	74.30	67.70	25.95	52.14
MS^{2}L (**Our**)	76.81	64.86	27.63	52.55

Table 2: Comparison of action recognition results with semi-supervised learning approaches.

Models	NW-UCLA	PKUMMD part I	PKUMMD part II	NTU
1% *labeled data*:				
MS^{2}L Rand-SS	17.09	34.46	11.79	32.18
LongT GAN	18.26	35.78	12.37	35.22
MS^{2}L (**Our**)	21.28	36.42	13.03	33.10
10% *labeled data*:				
MS^{2}L Rand-SS	58.63	67.95	22.81	62.49
LongT GAN	59.94	69.51	25.71	62.03
MS^{2}L (**Our**)	60.45	70.30	26.10	65.17

labeled data. In our experiments, we respectively sample 1% and 10% data randomly from the training set as *labeled data* and regard the rest as *unlabeled data*.

- **MS^{2}L Rand-Semi supervised (MS^{2}L Rand-SS)**: The encoder $f(\cdot)$ is initialized with random weights. Then we finetune all the weights of the network with *labeled data*.
- **LongT GAN [48]**: To apply this work in semi-supervised learning, we train the weights of GAN with *unlabeled data* and then finetune all the weights with *labeled data*.
- **MS^{2}L**: We train our full system in the semi-supervised setting. We use both *labeled data* and *unlabeled data* to independently train the encoder $f(\cdot)$ with MS^{2}L. Then we train the classifier $C(\cdot)$ and finetune the encoder $f(\cdot)$ with *labeled data* jointly.

From Table 2, we can notice that only with a small subsets of the datasets, our method can always improve the baseline considerably and performs better than *LongT GAN*.

Supervised Approaches. In the supervised setting, the encoder $f(\cdot)$ is pretrained by pretext tasks, and then the encoder $f(\cdot)$ and classifier $C(\cdot)$ are jointly trained with the full training data. We first evaluate our proposed approach in the supervised setting with different training strategies introduced in Sec. 3.2. The configurations are as follows:

- **MS^{2}L Rand-Supervised (MS^{2}L Rand-S)**: Our baseline structure initializes the weights of the encoder $f(\cdot)$ randomly and learns them with action labels jointly with the classifier $C(\cdot)$.
- **MS^{2}L Pretrain**: We initialize the encoder $f(\cdot)$ with the learned weights from self-supervised tasks and then learn the classifier $C(\cdot)$ for action recognition with encoder $f(\cdot)$ fixed.
- **MS^{2}L Moving**: We train the network with moving strategy, introduced in Sec.3.2. The encoder $f(\cdot)$ is pretrained with pretext tasks. Then we train the encoder $f(\cdot)$ and classifier $C(\cdot)$ jointly by switching the pretext tasks and classification task gradually.
- **MS^{2}L Jointly**: This strategy requires to train the model with self-supervised tasks and supervised task at the same time. We train the network from scratch with both self-supervised tasks and supervised task with fixed ω in Eq. 5.

Table 3: Comparison of action recognition results with supervised learning approaches on the NW-UCLA dataset.

Models	NW-UCLA
HBRNN-L [5]	78.50
SK-CNN [19]	86.10
VA-LSTM [45]	70.71
Denoised-LSTM [2]	80.30
MS^{2}L Rand-S	83.86
MS^{2}L Pretrain (**Our**)	85.26
MS^{2}L Moving (**Our**)	85.32
MS^{2}L Jointly (**Our**)	86.75

The results on the NW-UCLA, NTU and PKUMMD datasets are shown in Table 3, 4, 5, respectively. We improve the performance from 83.86% to 85.32% and 86.75% by moving pretraining and jointly training on NW-UCLA. And on larger datasets, the performance improves from 83.49% to 84.43% and 85.17% on PKUMMD part I and from 40.97% to 42.57% and 45.70% on PKUMMD part II by moving pretraining and jointly training, respectively. The best performance also improves from 78.44% to 78.56% on NTU dataset.

Compared to the baseline, our self-supervised learning method achieves significant improvement. Using the moving pretrained strategy helps to make the network change from self-supervised tasks to supervised task gradually and remember the prior knowledge learned by self-supervised tasks. It is also observed that jointly training achieves more gain than moving pretraining strategy. We explain it as that jointly training can force the network to extract features for different tasks, so the features can be relatively more general and contain richer information.

Table 4: Comparison of action recognition results with supervised learning approaches on the PKUMMD dataset.

Models	Part I	Part II
ST-GCN [43]	84.07	48.20
VA-LSTM	84.10	50.00
MS^{2}L Rand-S	83.49	40.97
MS^{2}L Pretrain (**Our**)	83.46	42.45
MS^{2}L Moving (**Our**)	84.43	42.57
MS^{2}L Jointly (**Our**)	85.17	45.70

Table 5: Comparison of action recognition results with supervised learning approaches on the NTU dataset.

Models	NTU
LSTM [13]	71.90
BLSTM	71.40
STA-LSTM [31]	73.40
TPN [40]	75.30
ST-GCN [43]	81.50
MS^{2}L Rand-S	78.44
MS^{2}L Pretrain (**Our**)	78.33
MS^{2}L Moving (**Our**)	78.46
MS^{2}L Jointly (**Our**)	78.56

Compared with the state-of-the-art, our model performs better on NW-UCLA and PKUMMD part I datasets and can be competitive to the previous methods on PKUMMD part II and NTU datasets.

Transfer Learning Performance. To further evaluate whether the proposed MS^{2}L is able to gain knowledge to related tasks, we investigate the transfer learning performance of our model.

Generally, the representations learned from large scale data more generalizable, as illustrated in [11]. Therefore, in our experiments, we regard the NTU and PKUMMD part I as source datasets and PKUMMD part II as the target dataset. We pretrain our model with the source datasets respectively and then fine-tune the whole network on the target dataset. We evaluate transfer learning performance using the accuracy of action recognition on PKUMMD part II and compare the results with those trained with full supervision from PKUMMD part II and *LongT GAN*. The configurations are as follows:

- **MS^{2}L Rand-Transfer (MS^{2}L Rand-T)**: We initialize the network randomly and then finetune the whole weights from scratch on PKUMMD part II.
- **LongT GAN [48]**: To perform transfer learning with *LongT GAN*, we pretrain the generator on the source dataset and then train the entire network on the target dataset.
- **MS^{2}L**: The encoder $f(\cdot)$ is trained by self-supervised tasks independently on source datasets, then we train the full system on the target dataset.

Table 6 shows the transfer learning results. Our self-supervised model outperforms the supervision baseline, improving the result from 40.97% to 44.14% when pretrained on PKUMMD part I and 45.81% when pretrained on NTU, respectively.

Compared with *LongT GAN*, our model can also show superiority. *LongT GAN* employs adversarial training strategies to reconstruct the whole skeleton data. Therefore, the network is trained to focus more on the details of skeletal joints. The domain gap in detailed skeleton settings of different datasets makes it hard to transfer the knowledge from the source dataset to the target dataset. Our proposed method, however, maps the skeleton data from different datasets to a common feature space with contrastive learning, and then achieve high-level domain knowledge transfer. The results in Table 6 illustrate the superiority of our proposed approach.

Table 6: Comparison of the transfer learning performance.

Models	Accuracy on PKUMMD part II
MS^{2}L Rand-T	40.97
PKUMMD part I:	
LongT GAN	43.61
MS^{2}L (**Our**)	44.14
NTU:	
LongT GAN	44.83
MS^{2}L (**Our**)	45.81

Table 7: Comparison of combinations of self-supervised tasks.

Method	NW-UCLA
MS^{2}L Rand-S	83.86
Prediction	84.88
Jigsaw	84.11
Contrastive	85.47
Prediction & Jigsaw	84.90
Contrastive & Jigsaw	84.82
Prediction & Contrastive	85.71
Prediction & Jigsaw & Contrastive	86.75

4.3 Ablation Study

Next, we conduct ablation experiments to give more analysis of our proposed approach. All the ablation studies are performed on the NW-UCLA dataset.

Analysis of Self-Supervised Tasks. In this part, we explore the role that each self-supervised task plays in the learning process. The baseline is training the classifier $C(\cdot)$ independently and the encoder $f(\cdot)$ is with random weights. For evaluating the self-supervised tasks, we pretrain the encoder $f(\cdot)$ and then finetune the overall network.

- **Motion Prediction:** There are two ways to conduct motion prediction, *i.e.*, temporal motion prediction and spatial motion prediction. For the temporal motion prediction, our model generates future skeleton data conditioned on the past skeleton data, while the spatial motion prediction reads in the corrupted input sequence, of which a number parts of human key-points are masked and set to be zero, and we predict the masked regions. We pretrain the encoder $f(\cdot)$ with motion prediction and then finetune the whole network jointly. Table 8 shows the results of temporal and spatial motion prediction, respectively. We observe that the spatial masked data reconstruction task does not improve and even hurts their performance. We analyze it may be caused by two reasons. On the one hand, a spatially masked skeleton sequence may lose critical information to infer the unknown skeleton regions. Therefore, it is more difficult to predict the spatial coordinates. On the other hand, we apply GRU units as our backbone, which explicitly learn temporal patterns but ignore spatial modeling. With a more powerful backbone, such as a graphic model which explicitly models spatial relations, we may boost the performance from spatial motion prediction. We leave it as our future work. In our model, we use temporal motion prediction as one of the self-supervised tasks.

• **Jigsaw Puzzles:** We explore two different methods to shuffle the skeleton data to perform jigsaw puzzle recognition, *i.e.*, temporal jigsaw puzzle recognition, and spatial jigsaw puzzle recognition. Temporal jigsaw puzzle shuffles the original data temporally and our goal is to predict the correct order of shuffled sequences , while spatial jigsaw puzzle shuffles the order of key-points which are divided into five parts, namely four limbs and a trunk, and our goal is to recognize each part. Also, the tasks are trained jointly and evaluated in classification accuracy. We show the results in Table 7, utilizing the spatial information harms the performance of the recognition task. It is mainly because the spatial jigsaw is much more difficult to predict since there are more permutations than a temporal jigsaw. In our model, we choose temporal jigsaw puzzle recognition as one of the self-supervised tasks.

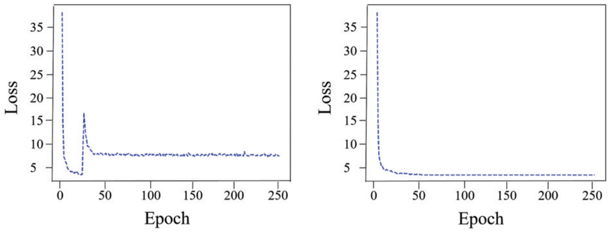
• **Contrastive Learning:** We evaluate different combinations of transformation operators for contrastive learning. In Table 8, we can observe that when we use spatial transformation operators always hurt recognition performance, whereas temporal transformation operators improve performance better. It is illustrated that the features from spatially masked skeletons and spatially shuffled skeletal joints lose much information to establish mapping with the features from original skeletons.

Table 7 shows the results from a single self-supervised task as well as their combinations. When applying a single self-supervised task, we can observe that the contrastive learning task always outperforms over the other two tasks. Contrastive learning aims to learn a common space between the original and transformed skeleton data. Therefore, the network is encouraged to learn more inherent feature representations. When two different self-supervised tasks are jointly optimized, they can enhance the performance because of the stronger restriction to the representation space. And using all the three tasks achieves the best performance. We explain it as the features extracted by the tasks jointly keep more aspects of information from the original sequences. That means the encoder $f(\cdot)$ can extract more general features.

Table 8: Analysis of self-supervised tasks. (P means Prediction; J means Jigsaw)

Method	NW-UCLA
MS^{2}L Rand-S	83.86
Temporal motion prediction	84.88
Spatial motion prediction	81.11
Temporal jigsaw puzzle recognition	84.11
Spatial jigsaw puzzle recognition	81.62
Contrastive(Spatial P + Temporal J)	81.35
Contrastive(Temporal P + Spatial J)	82.84
Contrastive(Spatial P + Spatial J)	82.45
Contrastive(Temporal P + Temporal J)	85.47

Training Strategy. Now we provide some insights into our moving pretraining strategy. Figure 4 shows the sum of all the losses of self-supervised tasks with different training strategies. Figure 4(a) shows the losses when we pre-train the self-supervised tasks and fine-tune the overall network for action recognition. We can observe that when we begin to finetune the overall network by jointly training the encoder $f(\cdot)$ and the classifier $C(\cdot)$, the losses of self-supervised learning increase sharply, which may destroy the feature representations learned from the self-supervised tasks. Figure 4(b) shows the losses of self-supervised tasks applied by our moving method. The smooth training loss curve illustrates the network is able to learn weights for action recognition while keeping the feature representations learned from the self-supervised tasks. The better results in Tables 3, 4, 5 confirm the effectiveness of moving pretraining.



(a) MS^{2}L Pretrain

(b) MS^{2}L Moving

Figure 4: Loss curves of MS^{2}L Pretrain and MS^{2}L Moving, respectively.

5 CONCLUSION

In this work, we propose a self-supervised learning approach for skeleton-based action recognition. To deal with the overfitting problem of learning skeleton representations from a single reconstruction task, we integrate multiple tasks to learn more general features. We apply motion prediction to model skeleton dynamics and jigsaw puzzle recognition to model temporal patterns, respectively. Besides, contrastive learning is adopted to further regularize the feature space and help learn intrinsic features. With comprehensive and thorough experiments on three datasets, we can show our model is a powerful feature extractor which outperforms the baseline significantly.

REFERENCES

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

[2] Girum G Demisse, Konstantinos Papadopoulos, Djamila Aouada, and Bjorn Ottersten. 2018. Pose Encoding for Robust Skeleton-Based Action Recognition. (2018), 188–194.

[3] Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2015. Unsupervised visual representation learning by context prediction. In *Proc. IEEE Int'l Conference on Computer Vision*. 1422–1430.

[4] Yong Du, Yun Fu, and Liang Wang. 2016. Representation learning of temporal dynamics for skeleton-based action recognition. *IEEE Transactions on Image Processing* 25, 7 (2016), 3010–3022.

[5] Yong Du, Wei Wang, and Liang Wang. 2015. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 1110–1118.

[6] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11, Feb (2010), 625–660.

[7] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. 2017. Self-supervised video representation learning with odd-one-out networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 3636–3645.

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. In *Proc. Int'l Conference on Learning Representations*.

[9] Yusuke Goutsu, Wataru Takano, and Yoshihiko Nakamura. 2015. Motion Recognition Employing Multiple Kernel Learning of Fisher Vectors Using Local Skeleton Features. In *Proc. Int'l Conference for Machine Learning Workshops*. 79–86.

[10] R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 1735–1742.

[11] Kaiming He, Ross Girshick, and Piotr Dollár. 2019. Rethinking imagenet pre-training. In *Proc. IEEE Int'l Conference on Computer Vision*. 4918–4927.

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

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (1997), 1735–1780.

[14] Eric Jang, Coline Devin, Vincent Vanhoucke, and Sergey Levine. 2018. Grasp2vec: Learning object representations from self-supervised grasping. *arXiv preprint arXiv:1811.06964* (2018).

[15] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. 2017. A New Representation of Skeleton Sequences for 3D Action Recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 4570–4579.

[16] Dahun Kim, Donghyeon Cho, and In So Kweon. 2019. Self-supervised video representation learning with space-time cubic puzzles. In *Proc. AAAI Conference on Artificial Intelligence*, Vol. 33. 8545–8552.

[17] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2017. Unsupervised representation learning by sorting sequences. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 667–676.

[18] Jiaying Liu, Sijie Song, Chunhui Liu, Yanghao Li, and Yueyu Hu. 2020. A Benchmark Dataset and Comparison Study for Multi-Modal Human Action Analytics. *ACM Trans. Multimedia Comput. Commun. Appl.* 16, 2, Article 41 (2020), 24 pages.

[19] Mengyuan Liu, Hong Liu, and Chen Chen. 2017. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition* 68, 68 (2017), 346–362.

[20] Fengjun Lv and Ramakant Nevatia. 2006. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *Proc. European Conference on Computer Vision*. 359–372.

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

[22] Whitney K Newey. 1988. Adaptive estimation of regression models via moment restrictions. *Journal of Econometrics* 38, 3 (1988), 301–339.

[23] Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. European Conference on Computer Vision*. 69–84.

[24] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. 2018. Boosting self-supervised learning via knowledge transfer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 9359–9367.

[25] Andrew Owens and Alexei A Efros. 2018. Audio-visual scene analysis with self-supervised multisensory features. In *Proc. European Conference on Computer Vision*. 631–648.

[26] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[27] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. 2016. NTU RGB+ D: A large scale dataset for 3D human activity analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 1010–1019.

[28] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. 2019. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 12026–12035.

[29] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. 2019. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 1227–1236.

[30] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2017. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. In *Proc. AAAI Conference on Artificial Intelligence*. 4263–4270.

[31] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2018. Spatio-Temporal Attention-Based LSTM Networks for 3D Action Recognition and Detection. *IEEE Transactions on Image Processing* 27, 7 (2018), 3459–3471.

[32] Kun Su, Xiulong Liu, and Eli Shlizerman. 2019. PREDICT CLUSTER: Unsupervised Skeleton Based Action Recognition. *ArXiv* abs/1911.12409 (2019).

[33] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. (2014), 3104–3112.

[34] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014. Human action recognition by representing 3d skeletons as points in a lie group. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 588–595.

[35] Raviteja Vemulapalli and Rama Chellapa. 2016. Rolling rotations for recognizing human actions from 3D skeletal data. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 4471–4479.

[36] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. In *Proc. Advances in Neural Information Processing Systems*. 613–621.

[37] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. 2019. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 4006–4015.

[38] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. 2012. Mining actionlet ensemble for action recognition with depth cameras. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 1290–1297.

[39] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Songchun Zhu. 2014. Cross-View Action Modeling, Learning, and Recognition. (2014), 2649–2656.

[40] Peng Wang, Yuanzhouhan Cao, Chunhua Shen, Lingqiao Liu, and Heng Tao Shen. 2017. Temporal Pyramid Pooling-Based Convolutional Neural Network for Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 12 (2017), 2613–2622.

[41] Chen Wei, Lingxi Xie, Xutong Ren, Yingda Xia, Chi Su, Jiaying Liu, Qi Tian, and Alan L Yuille. 2019. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 1910–1919.

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

[43] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *Proc. AAAI Conference on Artificial Intelligence*. 7444–7452.

[44] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. 2019. S4l: Self-supervised semi-supervised learning. In *Proc. IEEE Int'l Conference on Computer Vision*. 1476–1485.

[45] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. 2017. View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition from Skeleton Data. In *Proc. IEEE Int'l Conference on Computer Vision*. 2117–2126.

[46] Pengfei Zhang, Jianru Xue, Cuiling Lan, Wenjun Zeng, Zhanning Gao, and Nanning Zheng. 2018. Adding attentiveness to the neurons in recurrent neural networks. In *Proc. European Conference on Computer Vision*. 135–151.

[47] Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In *Proc. European Conference on Computer Vision*. 649–666.

[48] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. 2018. Unsupervised Representation Learning With Long-Term Dynamics for Skeleton Based Action Recognition. In *Proc. AAAI Conference on Artificial Intelligence*. 2644–2651.

[49] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. 2016. Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks. In *Proc. AAAI Conference on Artificial Intelligence*. 3697–3703.