IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 July 2023, accepted 9 August 2023, date of publication 15 August 2023,
date of current version 24 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3305372

RESEARCH ARTICLE

Contrastive Learning for Action Assessment Using Graph Convolutional Networks With Augmented Virtual Joints

CHUNG-IN JOUNG, SEUNGHWAN BYUN, AND SEUNGJUN BAEK, (Member, IEEE)
Korea University, Seongbuk-gu, Seoul 02841, South Korea

Corresponding author: Seungjun Baek (sjbaek@korea.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant through the Ministry of Science and ICT (MSIT), Korea Government, under Grant 2022R1A5A1027646 and Grant 2021R1A2C1007215, and in part by the Korea Medical Device Development Fund grant funded by the Korea Government under Project 1711195279 and Project RS-2021-KD000009.

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT A fine-grained detection of posture problems for action assessment has a wide range of applications for health care, sports, and rehabilitation. However, there exist many design challenges, e.g., the difficulty of detecting subtle deviations in actions from standard ones, lack of annotated datasets, and even multiple posture problems that may be present in a single action. In this paper, we propose a contrastive learning framework leveraging graph convolutional networks to address these challenges. We introduce Augmented Virtual Joint which is a learned position in space where its associated graphs provide a holistic view of spatio-temporal dynamics of body joints, offering a flexible and generalized representation of actions. Next, we propose Degraded Negative Contrasting, which judiciously contrasts incorrect action samples for effective discrimination of incorrect actions from correct ones. We also propose Frame-Selective Pooling which provides a simple yet effective selection of important frames from action clips. Experiments show that, as compared with the state-of-the-art architectures, the proposed model consistently achieves the best performance under a lack of training data and in the presence of multiple posture problems, which demonstrates its efficacy for fine-grained evaluation of actions.

INDEX TERMS Graph convolutional networks, augmented virtual joints, contrastive learning, pooling, fine-grained action classification.

I. INTRODUCTION

The task of action assessment concerns evaluating *how well* an action is performed. Typically, action assessment focuses on automatically judging or scoring a given action [1], [2], [3], [4], [5], however, in daily actions/exercises, detecting problems would be more practical instead of assigning scores. In this work, we consider *classification*-type action assessment, and focus on physical exercise/workout, e.g., squat, push-up, so as to detect incorrect postures, e.g., ''inward knees'' or ''upwards head'' in a squat. The problem is also related to fine-grained action recognition [6], [7], [8], [9], [10]; however, our problem has even finer, almost down to joint-level, granularity. Fine-grained action recognition classifies *different* kinds of actions, whereas we detect posture problems in the *same* kind of action with lower inter-class variance.

However, the detection of fine-grained posture problems faces several challenges. It is hard to create a large-scale dataset due to the difficulty of annotation. To our knowledge, Squat [11] and AI-Hub Fitness [12] are the only publicly available datasets of exercise videos with detailed annotations. Moreover, *multiple* posture problems may exist in a single action. For example, a push-up action may exhibit *both* ''bad elbow'' and ''bad neck'' problems [12], i.e., incorrect angles of both elbow and neck, each of which are separate classes of posture problems. Multiple posture problems exacerbate the dataset scarcity because it is extremely hard to create annotated datasets covering all the combinations.

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano.

In this paper, we propose contrastive learning associated with a new type of skeleton-based GCN for fine-grained action classification. In order to capture subtle discrepancies in joint trajectories during action execution, we introduce the concept of Augmented Virtual Joints (AVJ). Virtual Joints (VJ) are learnable locations augmented in space, not restricted to the human body, which are virtually connected to body joints. We consider two kinds of graphs of VJs representing global and local connectivity, to which separate graph convolutional layers are applied.

A typical contrastive approach simply pulls and repels positive and negative samples. Instead, we propose Degraded Negative Contrasting which judiciously performs negative contrasting; a relatively strong contrasting is applied for correct and incorrect actions as compared to the contrasting among different types of incorrect actions, and contrastive network and classifier are *jointly* trained in order to boost classification performance. We show that a carefully designed contrastive network in combination with action classifiers is effective in distinguishing between correct and incorrect actions, which is the key task in our action assessment.

In addition, we propose Frame Selective Pooling (FSP) which is a temporal pooling module aimed at attending to important frames, so that only the frames containing significant changes in motions are reflected in the network.

Our contributions are summarized as follows: 1) We propose a GCN with Augmented Virtual Joints which provides a global and flexible view on human actions. 2) We propose Degraded Negative Contrasting for effective classification of posture problems relative to standard actions. 3) We propose Frame Selective Pooling for simple and effective temporal filtering of frames. 4) Experiments show that our model achieves the best performance on detecting single-posture and multiple-posture problems on Squat [11] and AI-Hub Fitness [12] datasets as compared to state-of-the-art architectures.

II. RELATED WORK

1) ACTION RECOGNITION AND ASSESSMENT

Recently, skeleton-based action recognition has been actively studied using GCNs [13]. ST-GCN [14], the first work to apply GCN to skeleton-based action recognition, proposed to extract the spatio-temporal features of action using graph structure induced from the human body. 2s-AGCN [15] proposed adaptive graph convolution to overcome the limitations of fixed graph structures and introduced an additional information stream called bone stream obtained from a differential of joint coordinates. AS-GCN [16] introduces actional and structural links which capture action-specific and skeletal dependencies respectively, and uses the stacked layers of actional-structural and temporal graph convolutions. Shift-GCN [17] proposed spatial and temporal shift graph convolution motivated by shift CNNs [18] so as to flexibly extend the range of receptive fields. Shift-GCN also has low computational complexity by utilizing non-local shift graph operations. Most of existing GCN models rely on body joints to learn motion dynamics. However, our method combines body joints with learnable augmented virtual joints and provides a flexible representation to capture motion dynamics from diverse perspectives.

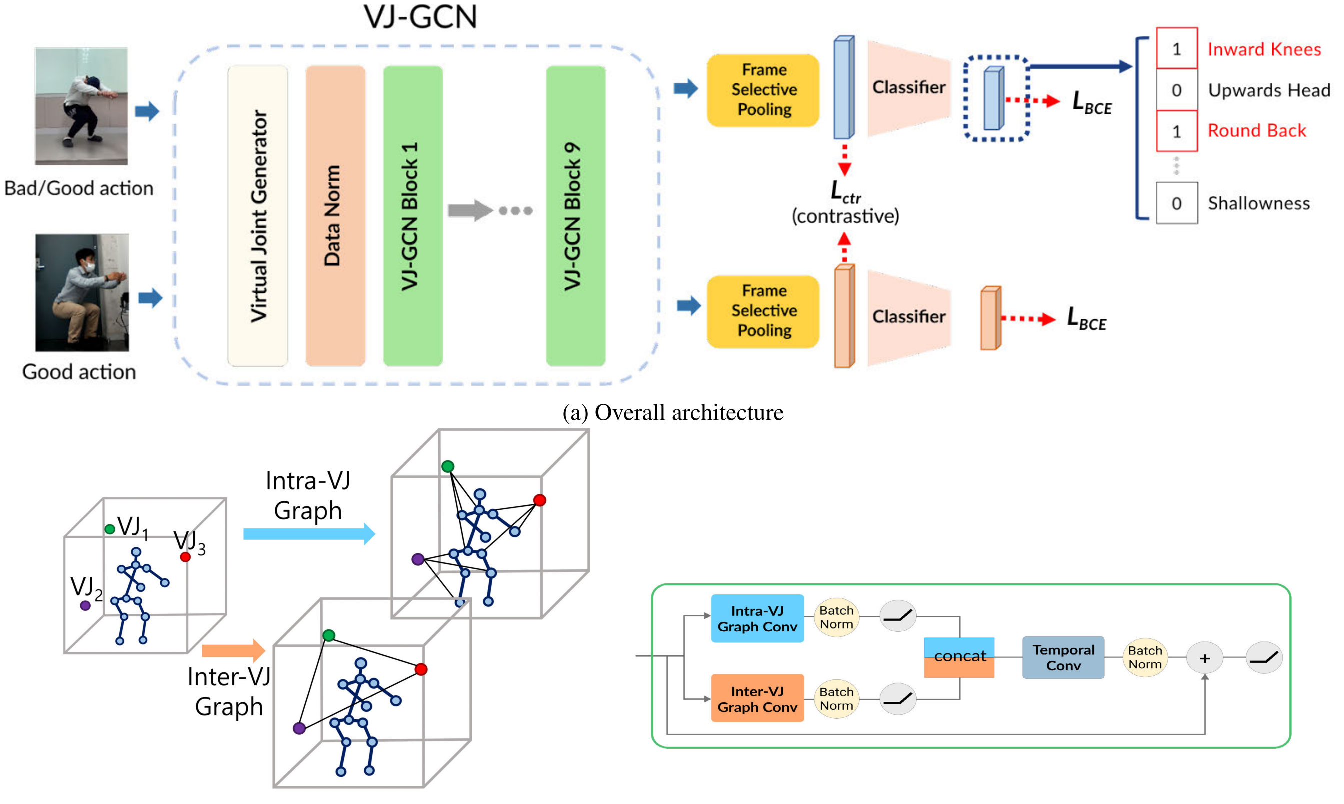
Another line of work explores innovations in the graph representation of human actions. DDGCN [19] proposes to extract spatial-temporal correlations between different parts of skeleton and uses directed graphs to capture hierarchical and sequential structures in human actions. CTR-GCN [20] proposed to learn and refines channel-wise topologies of the action graphs. CTR-GCN first learns the shared topology over channels, and refine it for each channel with channel-specific correlation. In this paper, we also seek for effective graph representation and propose GCN based on augmented virtual joints to provide a macroscopic as well as flexible view of the spatio-temporal action features. Some of the recent works which address skeleton-based action recognition with CNN approaches are PoseConv3D [21] and Ta-CNN [22]. PoseConv3D uses a 3D heatmap as input to the network, and Ta-CNN proposes to map joint coordinates to latent features.

Our work is also related to Action Quality Assessment (AQA) which evaluates the quality of execution of actions [1], [4], [5], [23]. AQA is applicable for automatic scoring of athletes' performance [1], [5] and also for improving surgical skills [24], [25], [26]. Existing works, however, mostly study estimating the AQA scores, whereas we focus on the assessment through fine-grained classification of postures.

2) CONTRASTIVE LEARNING

Contrastive learning aims at measuring the similarity among samples. The basic framework was established by Siamese networks [27], [28], [29] with shared weights and the loss function which "pulls" positive and "repels" negative sample pairs depending on whether the pair is from the same class. Recently, contrastive methods for self-supervised or semi-supervised learning have received much attention [30], [31], [32], [33]. SimCLR [30] leverages a large number of positive pairs for training using image augmentation. SupCon [34] uses not only augmented samples but also those in the same class as positive pairs.

Recent attempts on contrastive learning for video and action tasks include TCL [32] which is a semi-supervised contrastive learning and creates positive pairs through varying speeds of the same action video. CVRL [35] proposed self-supervised learning for videos by applying temporally consistent spatial augmentation to videos to produce positive pairs. Recent approaches to contrastive learning methods focus on handling noises in data for robust representations. For example, RINCE [36] and Sel-CL [37] propose learning methods under noisy views and labels, respectively. In our work, we judiciously apply contrastive learning depending on the correctness of actions so as to boost the discrimination between correct and incorrect postures with small inter-class variance.



(a) Overall architecture

(b) Intra-VJ and Inter-VJ Graphs

(c) VJ-GCN block

FIGURE 1. Illustration of the proposed architecture. (a) Overall architecture. Our model takes input of either (*Bad*, *Good*) or (*Good*, *Good*) action pair for contrastive learning. (*Bad*, *Bad*) action pairs are not used. (b) Examples of Intra-VJ and Inter-VJ graphs with 3 virtual joints. Only a subset of edges is shown for the Intra-VJ graph. (c) Structure of a VJ-GCN block. The input and output have the shape of (*channels*, *frames*, *joints*).

III. METHOD

The entire pipeline of the proposed network is illustrated in Fig. 1. An action clip is input to the network which has K binary outputs where there are K categories of incorrectly performed actions. Each output represents the existence of incorrectness in the given clip. For example, if the input action of squat contains both ‘‘upwards head’’ and ‘‘shallowness’’ problems, the outputs corresponding to classes of those incorrect actions are set to 1. An action is considered ‘‘correct’’ if none of K posture problems exist; otherwise, the action is considered ‘‘incorrect’’.

A. AUGMENTED VIRTUAL JOINTS

A challenging aspect of action assessment is that the model evaluates a single type of action, e.g., a squat, but should be able to detect subtle discrepancies in the action compared to correct ones. Thus, the model should fully capture complex interactions among the body joints. In skeleton-based action classification/assessment using graphs [14], [15], [17], [20], the graph is naturally induced from the local connectivity of human anatomy, i.e., vertices are joints and edges are skeletal connections. However, the question is, are there better graph representations for fine-grained classification of actions?

We propose a more general and flexible graph representation for skeleton-based action data. We introduce graphs with Augmented Virtual Joints (AVJ) as follows. A virtual joint (VJ) is a learnable location in space such that it is connected to *all* the body joints. Thus a VJ provides a holistic or global view of body joints from outside the body. The location of a VJ is not restricted to the human body, e.g., it may be even located far from and above the body, providing a ‘‘birds-eye’’ view of actions.

We augment multiple VJs, where each VJ provides a view from a different angle, and encodes the spatio-temporal dynamics of the associated body joints. The number of VJs is denoted by N which is a hyperparameter. We consider two graphs associated with VJs: Intra-VJ and Inter-VJ graphs. Intra-VJ graph has body joints and VJs as vertices, where there exist edges from each VJ to all the body joints. Inter-VJ graph is the complete graph of VJs only, e.g., see Fig. 1. A macroscopic view on actions is captured by Intra-VJ graphs, whereas Inter-VJ graphs capture the local connectivity among VJs. In addition, all the body joints are inter-connected through VJs within two hops, facilitating the exchange of node features across the GCN layers. Thus the combination of Intra-VJ and Inter-VJ graphs is able to capture local and global perspectives on body joints.

1) GRAPHS OF VIRTUAL JOINTS

Let us define the adjacency matrices for Intra-VJ and Inter-VJ graphs. Let J denote the number of body joints. There are $J+N$ nodes, and the node indices $1, \ldots, J$ and $J+1, \ldots, J+$

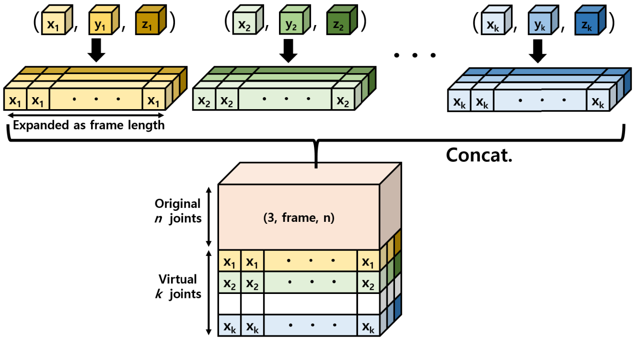


FIGURE 2. VJ-Generator. The module augments the input with virtual joints by adding k virtual joints to the original joints. The shape of the input changes from (channels, frames, n joints) to (channels, frames, $n+k$ joints).

N denote the body joints and VJs respectively. The adjacency matrices for intra- and inter-VJ graphs are given as follows:

$$A_{\text{intra}}(i,j) = \begin{cases} 1 & (i,j) \text{ is a body joint–VJ pair} \\ 0 & \text{otherwise} \end{cases}$$

$$A_{\text{inter}}(i,j) = \begin{cases} 1, & (i,j) \text{ is a pair of VJs} \\ 0, & \text{otherwise} \end{cases}$$

In Intra-VJ graphs, a VJ is connected to all the body joints and should capture their spatio-temporal dynamics. Since the body joints change locations over time, their trajectories can be reflected in VJ locations by "calibrating" the VJ locations to the trajectory of body joints over time. Specifically, the location of i-th VJ at frame t $(t = 0, 1, 2, \dots)$, or $v_i(t) \in \mathbb{R}^3$ is defined as follows. Let $x_j(t)$ denote the location of j-th body joint at t-th frame.

$$v_i(t) := v_i^0 + \sum_{j=1}^{J} \beta_{i,j} \left[\sum_{\tau=0}^{t} \alpha^{t-\tau} x_j(\tau) \right] \tag{1}$$

$v_i^0 \in \mathbb{R}^3$ is called the "default location" of the i-th virtual joint, and is a parameter to be learned through backpropagation. The input to our network is a concatenated tensor representing the positions of body joints and VJs, where the trajectory of VJs is defined by (1) (VJ generator in Fig. 1c and Fig. 2). The second term of (1) represents the calibration of location of virtual joints over time. $\beta_{i,j}$ are adjustable parameters determining how much the change in body joints should be reflected. The body joint trajectories are exponentially averaged over time with a discount factor $\alpha \in [0, 1]$, as in the term in the brackets of (1), in order to take the history of joint trajectories into account.

For example, if we let $\beta_{i,j} = 1/J$, the VJ calibrates itself according to the center of mass of body joints averaged over time. If weights are set to small numbers, then VJ effectively has a fixed location, providing a fixed camera view of the body joints. Note $\beta_{i,j}$ can be either set as hyperparameters or as a learnable parameter. Below we summarize the benefits of VJs.

AVJs enable flexible design: The number of VJs, N, is a hyperparameter by which one can control the model complexity. The number of edges of a complete graph of body joints is always fixed to $O(J^2)$. However, the number of edges in Intra-VJ and Inter-VJ graphs combined is $O(NJ + N^2)$. We can adjust N, e.g., we can make N large for complex and refined models, or make N small if the number of training samples is severely limited and under the risk of overfitting.

AVJs allow more general representation: An arbitrary graph on body joints can be represented using a graph on VJs. For example, suppose we set the number of VJs equal to J, the number of body joints, and place each VJ at the same location as each body joint. If, $v_i^0 = (0, 0, 0)$ and $\beta_{i,i} = 1$ and $\beta_{i,j} = 0$ for $j \neq i$ and $\alpha = 0$, from (1), either Intra- or Inter-VJ graph represents a complete graph on body joints. Thus any graph of body joints can be represented as a graph on VJs by properly setting edge weights in adjacency matrices.

2) GCN OF VIRTUAL JOINTS

VJ-GCN has separate GCN layers for intra-VJ and inter-VJ graphs, see Fig. 1c. The outputs from the streams are later concatenated, and fed to the temporal convolution layer. We apply the standard form of the GCN layer as follows. Let $M_i \in \mathbb{R}^{(J+N)\times(J+N)}$ denote the learnable matrix of edge weights, W_i denotes the convolutional weight matrix. $\Lambda_i \in \mathbb{R}^{(J+N)\times(J+N)}$ is diagonal matrix for $i = \{\text{intra}, \text{inter}\}$, and $\Lambda_i(j,j) := \sum_n \{A_i(j,n) + I(j,n)\}$ as in [13] where $A(j,n)$ denotes (j,n) element of matrix A. Let f_{in} denote the input at the GCN layer. We have that

$$\hat{A}_i := \Lambda_i^{-\frac{1}{2}}(A_i + I)\Lambda_i^{-\frac{1}{2}}, \ i = \text{intra, inter} \tag{2}$$

$$f_{\text{out},\,i} := (\hat{A}_i \otimes M_i) f_{\text{in}} W_i, \ i = \text{intra, inter} \tag{3}$$

$$f_{\text{out}} := [\sigma(f_{\text{out,intra}}); \sigma(f_{\text{out,inter}})] \tag{4}$$

where $\sigma(\cdot)$ denotes the ReLU activation, and $[x; y]$ represents the concatenation of x and y along the channel axis. Our network uses two separate layers of GCN for Intra-VJ and Inter-VJ graphs in order to perform a separate encoding of global and local connectivity information.

The outputs from each GCN are concatenated. In order to learn temporal features, the concatenated output is input to the temporal convolutional layer as proposed in [14]. We use 2D convolution with kernel size $1 \times k$, where 1 and k are the kernel size along the joint and time axis, respectively.

B. CONTRASTIVE LEARNING

1) DEGRADED NEGATIVE CONTRASTING

In typical contrastive learning with supervision [34], the samples in the same classes are considered positive pairs, and those in different classes as negative pairs. Given a sample, the model is trained to "pull" positives and "repel" negatives. In problems detecting incorrect postures, the correct actions serve as a *reference*, i.e., the incorrectness is measured by how much the given action deviates from the reference. Thus, it is clear that correct and incorrect action samples should be strongly repelled.

But what about repelling two different incorrect actions? In fine-grained action classification, incorrect actions may share similar posture problems even though they are labeled differently. For example, in push-up, there are "bad elbows" and "bad hands" classes. "bad elbows" means the elbow angle is incorrect, while "bad hands" means the wrong positions of hands. However, "bad hands" typically accompanies problems in the elbow angle as well, because the hand positions significantly affect loads on the elbow joints in push-ups [38]. Such a common problem in the elbow may not be learned as an incorrect posture by the model, if the "bad elbows" and "bad hands" classes are strongly repelled and are moved far away from each other in the embedding space.

To address such problems, we propose Degraded Negative Contrasting (DNC). In DNC, positive pairs are mutually pulled as usual; however, negative pairs belonging to different classes of incorrect actions are set to repel weaker, so that the representations for incorrect actions sharing common problems are not excessively far apart. The idea is to use a contrastive loss function such that, the degree of repulsion is "degraded" for certain types of negative contrasting.

We will use the following loss function modified from [34]. Let $z_{(\cdot)}$ denote a normalized vector of embeddings. Let I denote the set of indices of the current minibatch.

$$\begin{aligned}L_{\text{ctr}} &= \sum_{i\in I} L_{\text{ctr},i}\\ &= \sum_{i\in I}\left[\frac{-1}{|P(i)|}\sum_{p\in P(i)}\frac{\exp(z_{\text{i}}^{T}z_{\text{p}}/\tau)}{\sum\limits_{a\in A(i)}\exp\left\{(z_i^T z_a - C\cdot g(i,a))/\tau\right\}}\right]\end{aligned} \tag{5}$$

where sample i is the anchor; $A(i) := I \setminus \{i\}$; $P(i)$ is the set of positives, i.e., the samples in $A(i)$ whose labels are the same as an anchor; function $g(i, a)$ is 1 if (i) both samples i and a are incorrect actions, (ii) i and a belongs to different classes; otherwise $g(i, a) = 0$. Thus, when contrasting anchor in an incorrect action class with negatives which are also in an incorrect action class, the strength of repulsion will be degraded due to $C > 0$.

C is a hyperparameter representing the "bias" subtracted from the normalized similarity $z_i^T z_a$. Since the similarity is guaranteed to be in $[-1, 1]$, C can be adjusted within a fixed range to a impose penalty on contrasting negative samples belonging to different classes of incorrect actions. Our experiments show that, in some cases, the optimal C can be effectively infinite, i.e., *not* repelling negatives of incorrect action classes at all yields the best performance.

2) JOINT TRAINING WITH CONTRASTIVE AND CLASSIFICATION LOSSES

To promote contrasting correct and incorrect action classes, we propose to *jointly* train the contrastive network and the classifier. Typically these networks are trained separately, e.g., the contrastive network is first trained, and then the classifier is fine-tuned in a two-stage approach, or the back-propagation from classifiers is blocked at the contrastive networks [30], [34], [39]. By contrast, we train the networks simultaneously, so that the gradients from classifiers flow to contrastive networks. In this way, the embeddings for contrast are learned jointly with the classification logits, so that the classification performance can be boosted. In addition, since data augmentation is seldom used for skeleton-based datasets [14], [15], [17], the augmented positive pairs may not be available for training contrastive networks, which potentially makes the separate training ineffective. Thus the contrastive network and classifier can have synergistic effects in discriminating incorrect actions from standard actions by joint training. A similar attempt was made for face recognition using Siamese networks [40].

The network output consists of K binary classification outputs representing the existence of each of K incorrect actions for the input clip. Thus we consider binary cross-entropy (BCE) loss per action class. Let us denote the ground truth label and prediction for i-th clip for class k by $y_{i,k}$ and $\hat{y}_{i,k}$ respectively. The BCE loss is given by

$$L_{\text{bce},\,i} = -\frac{1}{K}\sum_{k=1}^{K}(y_{i,k}\log(\hat{y}_{i,k}) + (1-y_{i,k})\log(1-\hat{y}_{i,k}))$$

$$L_{\text{bce}} = \frac{1}{|I|}\sum_{i\in I} L_{\text{bce,i}}$$

Thus, the final loss function is the combination of L_{bce} and L_{ctr} with hyperparameter $\lambda > 0$:

$$L = L_{\text{ctr}} + \lambda \cdot L_{\text{bce}}. \tag{6}$$

C. FRAME SELECTIVE POOLING

Action video sequences can be quite long. However, not all the frames are equally important for action assessment. For example, there are frames in which the subject makes transitions to another posture, providing important hints on action assessment, while in some frames the subject may stay in a neutral position. One can utilize the attention modules [41], [42], [43], [44], [45] which learn to assign higher weights to important frames. However such modules require extra training, and the convergence can be an issue if the video sequence is very long.

We take a simpler approach and propose Frame-Selective Pooling (FSP) as follows. FSP performs the following temporal pooling: for each joint, the feature values are sorted over the temporal dimension, and m largest values are selected and averaged. That is, m most significant values are chosen from the feature map for each joint. FSP is aimed at focusing on the frames in which the target action is actively carried out. As shown in Fig 3, the network is in turn trained towards assigning high feature values to the important frame, as we observe from the patterns in the feature maps, i.e., the frames with pose transitions have higher values. FSP does not require additional training; the only hyperparameter we choose is m, the number of selected frames.

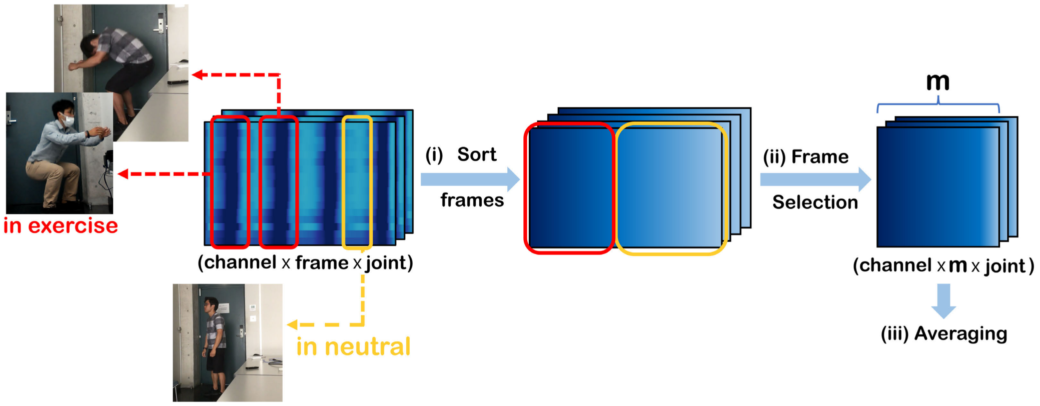


FIGURE 3. Frame Selective Pooling (FSP). First, the input tensors are sorted over the temporal dimension (horizontal direction). Second, *m* frames are selected and averaged. The red box represents the frames that contain dynamic movements of body joints whereas the yellow box represents the frames in which the subject is in a static pose.

TABLE 1. Dataset Configuration.

		Train	Test	Val
Squat	Single	813	407	174
	Multiple	-	107	-
Standing Side Crunch	Single	240	59	Cross Validation
	Multiple	-	1310	
Push up	Single	114	42	Cross Validation
	Multiple	-	676	
Knee Push up	Single	114	42	Cross Validation
	Multiple	-	674	

D. NETWORK ARCHITECTURE: SUMMARY

In summary, there are three key components in the proposed architecture: VJ-GCN provides multiple viewpoints of actions from various angles and locations to spot subtle problems in the action execution; DNC penalizes negative pairs of incorrect actions in the contrastive training to enhance contrasting between good and bad actions; FSP selects important action frames, which reduces feature dimension and noises. Although each component can be individually applied to action assessment tasks, they can be harmoniously combined for enhancing the fine-grained detection of erroneous actions, as we propose in this paper.

IV. EXPERIMENTS

A. DATASETS

Detailed configurations of datasets used in the experiments are provided in Table 1-5.

1) SQUAT

In Squat Dataset [11] released at CVPR 2019, we consider Single Individual Dataset containing 7 classes of incorrect squat postures. We used VideoPose3D [46] for pose estimation which outputs 17 joints in 3D coordinates. Two classes of *warped back* and *frontal knee* were not used in our experiments, because a precise pose estimation was difficult due to self-occlusion, and the difference from the correct postures was too small. The original dataset did not contain videos with multiple posture problems. Thus, we created a dataset of 107 squat video clips with 4 combinations of multiple posture problems: *round back+shallowness*, *inward knees+round back*, *inward knees+shallowness*, and *upwards head+shallowness* based on classes specified in [11]. We followed a similar protocol as [11] for the dataset construction, e.g., the subject looks at various directions, and each video has 300 frames.

2) AI-HUB FITNESS

AI-Hub [12] is an integrated AI platform maintained by the Korean Government (Ministry of Science and ICT, National Information Society Agency) which provides various AI infrastructures including datasets on vision, audio, healthcare, and autonomous driving, etc. The entire Fitness dataset contains 200K clips, each of which is 16–48 frames long, on personal exercise including videos of single posture problems as well as multiple posture problems (combination of single posture problems). In our experiments, we chose 3 types of exercises: standing side crunch, push up, and knee push up.

B. EXPERIMENTAL SETTINGS

We would like to address two important questions in our experiments: (1) how does a model perform under data scarcity? (2) does the model generalize well to the detection of multiple posture problems? The goal is to address data scarcity problems for fine-grained action evaluation, where the problem worsens for collecting and annotating action data with multiple posture problems.

The models are trained under varying sizes of training datasets: the training set sizes for squat data vary as 100, 200, 300 to 813, whereas 813 is the entire size of the dataset. For standing side crunch, the training dataset only contained 240 samples, thus we used the training sets of sizes 120

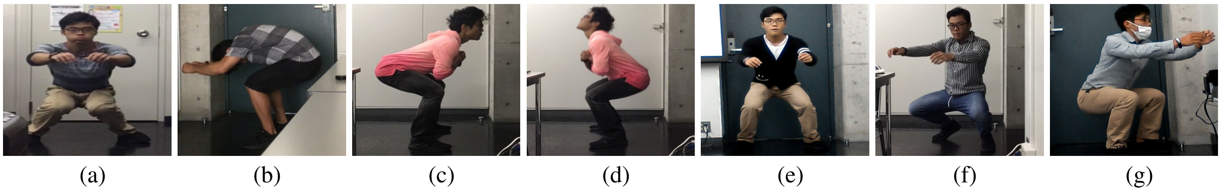


FIGURE 4. Squat Dataset [11]. (a) :Inward knees, (b):Round back, (c):Warped Back, (d):Upwards Head, (e):Shallowness, (f):Frontal knee, (g):Good.

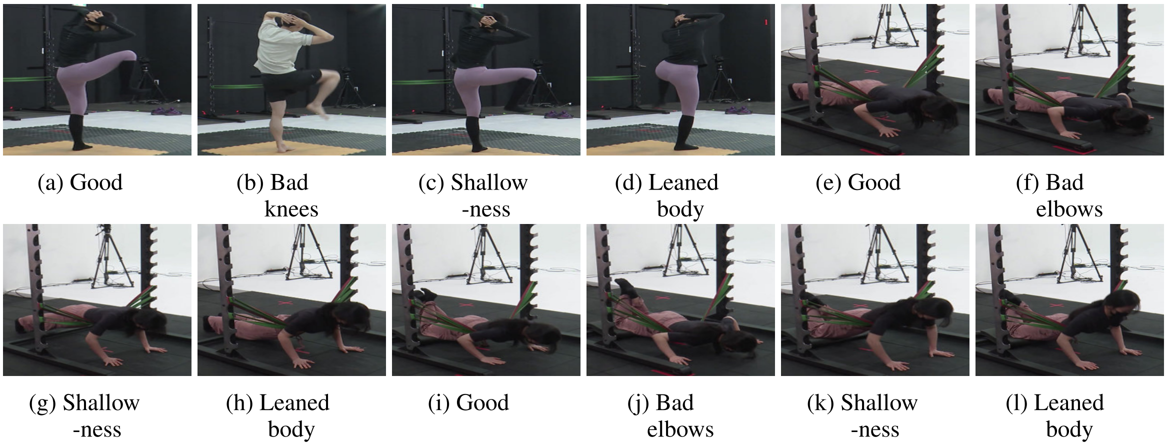


FIGURE 5. AI-Hub Fitness Dataset [12]. (a)–(d): Standing side crunch, (e)–(h): Push up, (i)–(l): Knee Push up.

TABLE 2. Configuration of Squat Dataset.

Squat	Train	Test
Inward knees	133	48
Round back	161	95
Upwards head	154	86
Shallowness	180	105
Good	185	73
Total (Single posture problem)	813	407
Round back + Shallowness	-	32
Inward knees + Round back	-	27
Inward knees + Shallowness	-	20
Upwards head + Shallowness	-	28
Total (Multiple posture problems)	-	107

and 240. For push-up and knee push-up, the total number of training samples was already limited to 114, thus we used the full training set. The training sets contain a roughly equal number of samples per class, e.g., the dataset of size 100 contains 5 classes with 20 video clips each. The models are trained only by the datasets with a single posture problem, and the datasets with multiple posture problems were used only for testing. To goal is to evaluate whether the model generalizes well for detecting unseen combinations of posture problems.

Our model has K binary outputs, each for detecting the presence of the corresponding incorrect actions. Recall that binary classification for each class is necessary for detecting multiple posture problems. For the evaluation metric, we use the F1-score of binary classifications averaged over

TABLE 3. Dataset configuration of Standing side crunch.

Standing side crunch	Train	Test
Leaned body	40	9
Not staring front	40	10
Shallowness	41	10
Bad knees	39	10
Bad hands	40	10
Good	40	10
Multiple posture problem	-	1310
Total	240	1369

TABLE 4. Dataset configuration of Push up.

Push up	Train	Test
Leaned body	19	7
Bad elbows	19	7
Shallowness	19	7
Bad hands	19	7
Bad head	19	7
Good	19	7
Multiple posture problem	-	676
Total	114	718

TABLE 5. **Dataset configuration of Knee push up.**

Knee push up	Train	Test
Leaned body	19	7
Bad elbows	19	7
Shallowness	19	7
Bad hands	19	7
Bad head	19	7
Good	19	7
Multiple posture problem	-	674
Total	114	716

the classes of incorrect actions. When evaluating F1-score, we assume that the "positive" outcome for a given class is the existence of the incorrect posture in the input action. Also, F1-score is suitable for dealing with the label imbalance. For example, in the Squat dataset containing 5 classes, only roughly 20% of the dataset contains positives for a given class which is well below 50%. The results for other metrics for classification such as accuracy, precision, and recall are presented in Supplementary Materials.

1) TRAINING SETTINGS

In both experiments with Squat and AI-Hub Fitness, we train networks for 150 epochs with batch size 64 using SGD optimizer with 0.9 momentum and 0.0001 of weight decay but 0.01 and 0.05 learning rate for each dataset respectively with cosine annealing scheduler. We used 5-fold cross-validation for AI-Hub Fitness.

We performed hyperparameter tuning for batch size, m, and the number of VJs. The batch sizes are set to either 8 or 16. The number of VJs is set to either 8 or 16 which is comparable to the number of body joints. For the squat dataset, the length of video clips is 300 frames, and that of the AI-Hub dataset is 16 frames. For m, we chose either 100 or 200 as m for the squat dataset, and 8 or 16 for AI-Hub dataset.

C. PERFORMANCE COMPARISON

1) BASELINES

We make comparisons with state-of-the-art architectures for action classification: ST-GCN [14], 2S-AGCN [15], Shift-GCN [17], and CTR-GCN [20]. For fair comparison using joint coordinates only, we used the "joint stream" mode of the models using multiple streams: 2S-AGCN, Shift-GCN, and CTR-GCN. We considered two versions of our network: VJ-GCN is the proposed network without contrastive training, i.e., classifier only. VJ-GCN + DNC is the model with contrastive training using Degraded Negative Contrasting. Through hyperparameter search, the number of VJs is set to 8 and 16 for Squat and AI-Hub Fitness datasets, respectively.

2) SINGLE- AND MULTIPLE-POSTURE PROBLEMS

As shown in Table 6, either VJ-GCN or VJ-GCN+DNC consistently achieves the best F1-score in all the configurations of training dataset sizes. For example, in the Squat dataset, when the training set has only 100 samples, the performance gains obtained by the proposed schemes are relatively higher than the other cases of dataset sizes, in both datasets for single- and multiple-posture problems. The results show that our scheme can be relatively robust under data scarcity.

We also observe that VJ-GCN performs well even without contrastive training. In Table 6, 5J-GCN shows performances comparable to other state-of-the-art architectures. Thus, the macroscopic view of human actions captured by the graphs of virtual joints seems to be important in fine-grained action classification.

We also observe that our scheme is effective in detecting multiple posture problems which are unseen combinations of single posture problems. The results show that, although trained only on datasets with single-posture problems, our model can generalize well to the detection of multiple posture problems, and ease the laborious task of annotating combined posture problems in action datasets.

3) CONTRASTIVE TRAINING

We show that simply adding contrastive network to classifiers does not necessarily result in a good performance. Table. 6 shows the results of applying SupCon [34] to ST-GCN [14] where both methods are widely used. For "ST-GCN, SupCon", we took the conventional two-stage approach for contrastive training: an instance of ST-GCN network is contrastively pre-trained and a classifier is appended for fine-tuning.

As shown in Table 6, the addition of contrastive training can even hamper performance: in all cases, "ST-GCN, SupCon" underperforms the original ST-GCN. In particular, data augmentation is mostly avoided for skeleton-based datasets [14], [15], [17], thus the contrastive learning relatively lacks positive pairs compared to image classification problems. An implication of the results is that, when the data availability is limited, the conventional approaches using contrastive learning can be harmful.

By contrast, simultaneous training for BCE and contrastive losses combined with DNC helps improving the performance of the original VJ-GCN. We found that the optimal penalty parameter C for DNC was mostly about 0.1–0.2; however, in 25% of the total cases, we have $C = 20$ which is a significantly high (practically infinite) penalty. Thus, one can be often advised *not* to perform negative contrasting *at all* for negative sample pairs from differing classes, which demonstrates the effectiveness of DNC.

V. ABLATION STUDY

In this section, we conduct an ablation study of the proposed method with respect to VJs, Frame Selective Pooling, and Degraded Negative Contrasting. Note that the ablation study for VJ-GCN + FSP without DNC in comparison to prior methods has been provided in Table. 6 (see the row "VJ-GCN"). In the ablation study, we use full-sized training dataset.

TABLE 6. Performances comparison on Squat [11] and AI-hub Fitness dataset [12].

Action type	Squat				Standing side crunch		Push up	Knee push up
Size of Training Dataset	100	200	300	813	120	240	114	114
Methods	F1 scores for a single posture problem							
ST-GCN [14]	43.68	52.38	60.10	69.79	61.02	72.32	26.19	30.95
Js-AGCN [15]	47.95	53.37	62.23	71.26	54.24	74.01	25.40	24.60
1s Shift-GCN [17]	42.04	51.89	53.04	67.32	61.02	77.40	14.29	20.63
CTR-GCN(joint) [20]	45.47	51.56	52.71	64.04	55.37	62.15	27.38	26.98
ST-GCN [14], SupCon [34]	34.48	35.47	43.68	61.25	30.79	37.29	9.52	11.90
VJ-GCN	**51.89**	58.62	63.28	72.09	62.71	73.45	23.02	30.95
VJ-GCN + DNC	51.40	**59.44**	**64.70**	**74.38**	**68.36**	**77.97**	**35.71**	**33.33**

Action type	Squat				Standing side crunch		Push up	Knee push up
Size of Training Dataset	100	200	300	813	120	240	114	114
Methods	F1 scores for multiple posture problems							
ST-GCN [14]	54.83	57.63	65.42	66.98	53.43	56.04	26.90	26.40
Js-AGCN [15]	50.78	61.06	65.42	69.47	52.24	53.62	33.88	35.19
1s Shift-GCN [17]	51.72	61.68	61.53	61.37	51.32	53.45	25.18	24.95
CTR-GCN(joint) [20]	53.99	48.29	49.07	62.93	53.97	55.93	39.83	35.16
ST-GCN [14], SupCon [34]	42.99	45.48	59.81	61.06	36.49	43.76	24.80	17.87
VJ-GCN	53.58	66.04	66.67	**69.78**	51.07	52.98	37.81	30.88
VJ-GCN + DNC	**67.29**	**66.05**	**67.60**	66.67	**54.65**	**57.29**	**43.21**	**36.04**

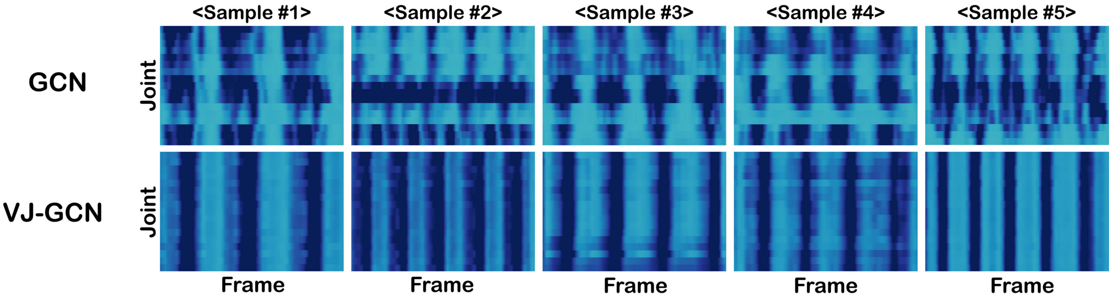


FIGURE 6. Sample feature maps in Squat actions from the experiments in Table. 7.

1) VIRTUAL JOINTS (VJs)

To evaluate the effectiveness of Virtual Joints, we compared VJ-GCN and a GCN with body joints only, which uses the adjacency graph based on body joints derived from human anatomy. As shown in Table 7, our methods outperformed GCN with body joints in most cases. We also present a qualitative evaluation of VJs. Fig. 6 shows examples of feature values at joints over the action frames. The darker color represents the features with higher activation values, and the brighter blue pattern represents the frames where the subject is in neutral postures. The figure shows how good the proposed VJ-GCN is at understanding body joint dynamics because feature outputs are highly activated across all the joints at important action frames. However, in case of GCN which uses only body joints, some dynamics do not stand out clearly whereas in case of 'VJ-GCN' all the body joints are activated well during exercise. VJ-GCN shows more regular patterns of emphasis across frames. Thus, VJ-GCN is consistently better at "attending" to important frames for discriminating good and bad actions, which leads to improved classification results.

2) FRAME SELECTIVE POOLING (FSP)

Table 8 shows the results for VJ-GCN with and without FSP. The ranges of hyperparameter m are 1–300 for Squat and 1–16 for the AI hub dataset. When m is equal to the maximum frame length, FSP is equivalent to average pooling (denoted as VJ-GCN without FSP in Table. 8). In most cases, FSP provides additional performance gains in spite of too short frame length, especially in the AI hub dataset.

3) DEGRADED NEGATIVE CONTRASTING (DNC)

For the ablation study associated with the proposed scheme which is DNC combined with 1-stage contrastive training, i.e., a joint training of contrastive networks and classifiers, we consider two cases: (1) VJ-GCN combined with DNC in 2-stage contrastive training, i.e., a separate training of contrastive networks and classifiers; (2) VJ-GCN and 1-stage contrastive training without DNC.

For Case (1), we first contrastively train the network, and later fine-tune the classifier. The results are shown in row "VJ-GCN, DNC" of Table 9. The performance is poor as expected, because the number of samples to pre-train

TABLE 7. Ablation study on VJs (F1-score).

Action type	Squat		Standing side crunch		Push up		Knee push up	
Methods	Single	Multiple	Single	Multiple	Single	Multiple	Single	Multiple
GCN with body joints only	68.31	60.43	62.71	51.71	21.43	32.88	26.19	**32.27**
VJ-GCN	**72.09**	**69.78**	**73.45**	**52.98**	**23.02**	**37.81**	**30.95**	30.88
GCN with body joints only+DNC	69.29	66.36	67.80	50.44	26.19	36.20	29.37	**37.89**
VJ-GCN + DNC	**74.38**	**66.67**	**77.97**	**57.29**	**35.71**	**43.21**	**33.33**	36.04

TABLE 8. Ablation study for Frame Selective Pooling. Hyperparameter *m* is specified for each experiment.

Action type	Squat	Standing side crunch	Push up	Knee push up
Methods	F1 scores for a single posture problem			
VJ-GCN without FSP	71.76	**73.45**	**23.02**	23.81
VJ-GCN with FSP	**72.09** (m=100)	**73.45** (m=16)	**23.02** (m=8)	**30.95** (m=4)
Methods	F1 scores for multiple posture problems			
VJ-GCN without FSP	65.73	**52.98**	36.88	**31.25**
VJ-GCN with FSP	**69.78** (m=100)	**52.98** (m=16)	**37.81** (m=8)	30.88 (m=4)

TABLE 9. Ablation study for Degraded Negative Contrasting.

Action type	Squat		Standing side crunch		Push up		Knee push up	
Methods	Single	Multiple	Single	Multiple	Single	Multiple	Single	Multiple
VJ-GCN	70.11	65.42	72.32	51.15	21.43	38.99	21.43	32.10
VJ-GCN, N-pair-mc	48.44	50.31	70.62	47.24	11.90	27.74	**33.33**	34.60
VJ-GCN, DNC	55.99	64.18	28.81	34.81	2.38	2.09	16.67	4.93
VJ-GCN + DNC	**74.38**	**66.67**	**77.97**	**57.29**	**35.71**	**43.21**	**33.33**	**36.04**

VJ-GCN is limited. We have presented a similar result in Table. 6, e.g., see row "STGCN, SupCon". In 2-stage training, we first train the contrastive network for 100 and 50 epochs with contrastive loss on Squat and AI-hub datasets respectively, and fine-tuned the classifier for 100 more epochs.

For Case (2), we consider jointly training contrastive network and classifier, however, do *not* use DNC. Note if, in the DNC loss function, we can control penalty parameter $C > 0$. By making C sufficiently small, the loss function reduces to that of SupCon [34]. Thus, the loss function in SupCon is a special case of DNC. For that reason, we have chosen another type of contrastive loss function instead of SupCon-type losses. We consider *multi-class N-pair loss (N-pair-mc)* [47]. The results in Table 9 show that VJ-GCN+DNC still outperforms VJ-GCN, *N-pair-mc*. This shows that applying contrastive loss judiciously based on correct/incorrect action classes indeed helps the fine-grained classification for action evaluation.

VI. VISUALIZATION OF VIRTUAL JOINTS AND SKELETON

In this section, we provide a qualitative analysis of the effectiveness of virtual joints. We visualized some examples of virtual joints and skeletons in Fig 7, 8. The body joints are represented by blue dots, whereas virtual joints are marked by red dots. Only a subset of virtual joints are shown for each case. We observe that virtual joints are scattered around the body, providing diverse perspectives on the action from

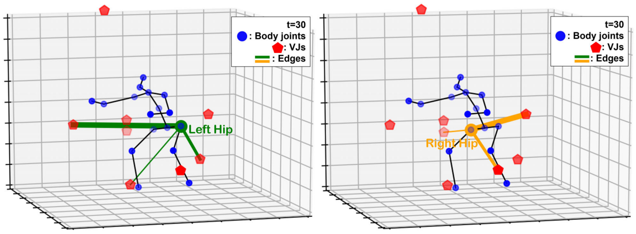


FIGURE 7. Visualization examples of virtual joints and skeletons of a *good* squat motion at frame $t = 30$. Some of the edges in Intra-VJ graphs are shown, where the thickness of edges represents the relative edge weights, e.g., thicker edges have larger weights. On the left, the edges with the three largest weights incident on the Left Hip joint are shown. On the right, the edges associated with the three largest weights incident on the Right Hip joint are shown. Different sets of VJs were connected to Left and Right Hip joints through those edges. Edge weights encode the relative importance of neighbors in determining the feature of a node; thus, diverse groups of VJs contributed to learning body joint features from various angles and distances.

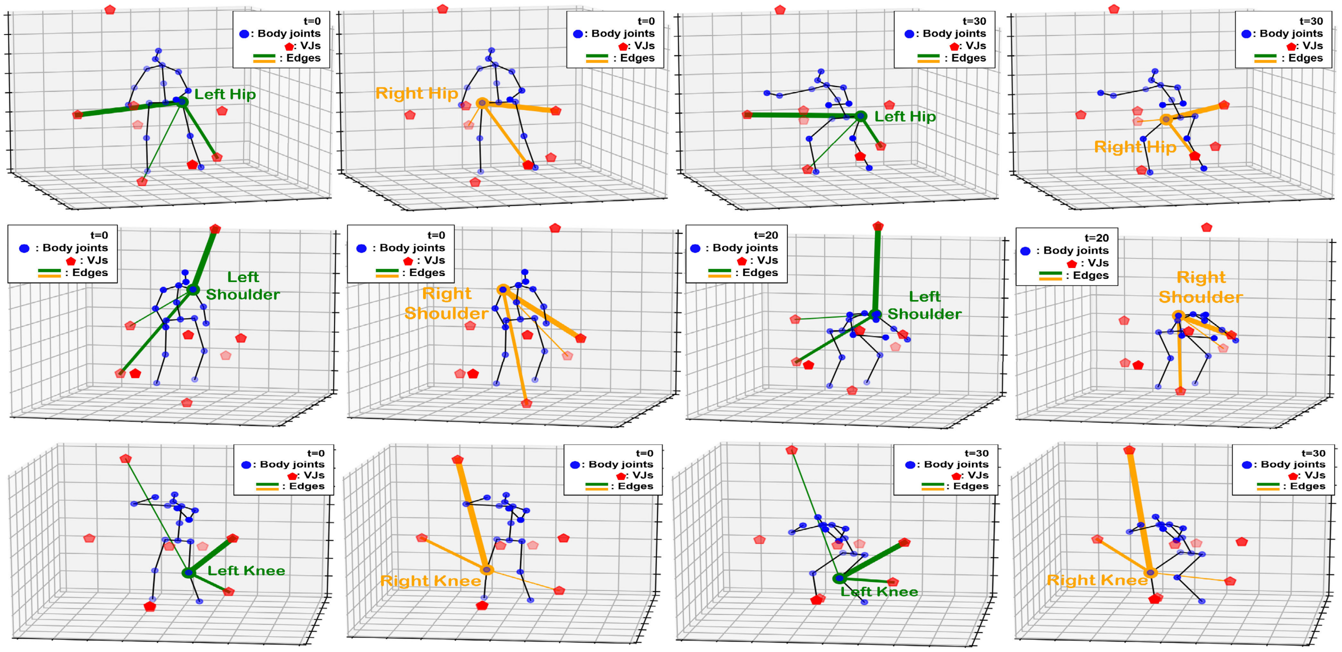


FIGURE 8. Visualization examples of virtual joints and skeletons where row 1–3 are *Good*, *Round back*, and *Inward knees* from the squat dataset respectively. Orange and green line segments represent edges with the three largest weights in the Intra-VJ graphs.

various angles. The learned locations of virtual joints are more or less evenly distributed over space.

We also observe that the locations of virtual joints change over time, which results from the "calibration" of the locations of virtual joints over time according to Eq. (1) of the main paper. The virtual joints appear to move at different velocities, because the learnable calibration parameter, $\beta_{i,j}$ in Eq. (1) of the main paper, depends on the locations of both virtual joint i and body joint j. Overall, the results show that virtual joints are able to provide macroscopic viewpoints on the spatio-temporal dynamics of human actions.

Fig. 7 depicts examples of VJs and the skeleton of body joints and the associated intra-VJ graphs. The figure shows the intra-VJ graph of a squat motion at frame $t = 30$, and the left and right figures depict some of the edges connecting VJs and body joints from the identical frame. The thickness of edges represents the relative size of edge weights in matrix M_i in Eq. (3) of the main paper. As in [14], [15], and [16], M_i represents the learned importance of edges, e.g., the strength of the connection between VJs and body joints where thicker edges represent larger weights. On the left (resp. right) of Fig. 7, the edges with the three largest weights connected to Left Hip (resp. Right Hip) joint are shown.

As shown in Fig. 7, different groups of VJs are deemed to be "important" for determining features of Left and Right Hip joints. This implies that diverse groups of VJs contribute to the body joint features from various locations and angles. Such diverse viewpoints provided by VJs appear to be the main reason behind the effectiveness of virtual joints in the fine-grained classification of actions through VJ-GCN. Similar observations can be made for different actions and body parts. Fig. 8 shows similar figures for good and bad squat actions where the edges for various joints are shown. We observe that VJs at diverse locations and angles participate in the encoding of joint features depending on the action types, body parts, and frame number.

VII. CONCLUSION AND LIMITATIONS

We proposed a contrastive learning framework with graph convolutions based on Augmented Virtual Joints. A new graph representation based on virtual joints which can provide a macroscopic view of actions was developed. Considering that incorrect actions have common posture problems, we proposed Degraded Negative Contrasting for judicious repelling of negative sample pairs in the contrastive training. Experiments have shown that our model achieved good performance in detecting both single and multiple posture problems even with insufficient training samples.

A practical application of the proposed algorithm will be a self-directed rehabilitation or an automated personal training system that can give detailed advice on proper poses for exercise. In that application, users can take videos of their rehabilitation or routine exercises and upload the video to a server that can automatically provide feedback on the proper poses. In the future, we plan to design a lightweight algorithm so that the automated assessment can be done in an on-device manner in real-time.

A limitation of our study is the lack of datasets: to our knowledge, Squat [11] and AI-Hub Fitness [12], are the only publicly available datasets that fit our needs. We used contrastive learning to alleviate the problem. By contrasting $O(N^2)$ sample pairs, the discriminative power of the classifier could be improved as compared to simple supervision using N samples. However, we believe our ideas are applicable to other problems of fine-grained action evaluation. Moreover, considering the importance of the topic, more datasets are expected to be available in the near future.

REFERENCES

[1] H. Pirsiavash, C. Vondrick, and A. Torralba, "Assessing the quality of actions," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 556–571.

[2] P. Parmar and B. Morris, "Action quality assessment across multiple actions," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1468–1476.

[3] P. Parmar and B. T. Morris, "What and how well you performed? A multitask learning approach to action quality assessment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 304–313.

[4] Y. Tang, Z. Ni, J. Zhou, D. Zhang, J. Lu, Y. Wu, and J. Zhou, "Uncertainty-aware score distribution learning for action quality assessment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9836–9845.

[5] P. Parmar and B. T. Morris, "Learning to score Olympic events," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 76–84.

[6] D. Shao, Y. Zhao, B. Dai, and D. Lin, "Intra- and inter-action understanding via temporal action parsing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 727–736.

[7] A. Piergiovanni and M. S. Ryoo, "Fine-grained activity recognition in baseball videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1740–1748.

[8] K. Hara, Y. Ishikawa, and H. Kataoka, "Rethinking training data for mitigating representation biases in action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3344–3348.

[9] J. Munro and D. Damen, "Multi-modal domain adaptation for fine-grained action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3723–3726.

[10] D. Shao, Y. Zhao, B. Dai, and D. Lin, "FineGym: A hierarchical video dataset for fine-grained action understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2613–2622.

[11] R. Ogata, E. Simo-Serra, S. Iizuka, and H. Ishikawa, "Temporal distance matrices for squat classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 2533–2542.

[12] Ministry of Science and ICT, National Information Society Agency. (2021). *AI Hub: Fitness Posture Dataset to Understand Fine-Grained Actions*. [Online]. Available: https://aihub.or.kr/aidata/8051

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017, pp. 1–14. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[14] M. Jiang, J. Dong, D. Ma, J. Sun, J. He, and L. Lang, "Inception spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. Int. Symp. Control Eng. Robot. (ISCER)*, Feb. 2022, pp. 208–213.

[15] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12018–12027.

[16] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3590–3598.

[17] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 180–189.

[18] B. Wu, A. Wan, X. Yue, P. Jin, S. Zhao, N. Golmant, A. Gholaminejad, J. Gonzalez, and K. Keutzer, "Shift: A zero FLOP, zero parameter alternative to spatial convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9127–9135.

[19] M. Korban and X. Li, "DDGCN: A dynamic directed graph convolutional network for action recognition," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 12365, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds. Glasgow, U.K.: Springer, 2020, pp. 761–776, doi: 10.1007/978-3-030-58565-5_45.

[20] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13339–13348.

[21] H. Duan, Y. Zhao, K. Chen, D. Lin, and B. Dai, "Revisiting skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2959–2968.

[22] K. Xu, F. Ye, Q. Zhong, and D. Xie, "Topology-aware convolutional neural network for efficient skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 36, 2022, pp. 2866–2874.

[23] T. S. Kim and A. Reiter, "Interpretable 3D human action analysis with temporal convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1623–1631.

[24] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, T. Ploetz, M. A. Clements, and I. Essa, "Automated video-based assessment of surgical skills for training and evaluation in medical schools," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 11, no. 9, pp. 1623–1636, Sep. 2016.

[25] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Evaluating surgical skills from kinematic data using convolutional neural networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018, pp. 214–221.

[26] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, and I. Essa, "Video and accelerometer-based motion analysis for automated surgical skills assessment," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 13, no. 3, pp. 443–455, Mar. 2018.

[27] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, no. 4, pp. 669–688, 1993.

[28] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2005, pp. 539–546.

[29] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML*, vol. 2, 2015, pp. 1–30.

[30] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[31] M. Zheng, F. Wang, S. You, C. Qian, C. Zhang, X. Wang, and C. Xu, "Weakly supervised contrastive learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10022–10031.

[32] A. Singh, O. Chakraborty, A. Varshney, R. Panda, R. Feris, K. Saenko, and A. Das, "Semi-supervised action recognition with temporal contrastive learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10384–10394.

[33] Y. Zhang, B. Hooi, D. Hu, J. Liang, and J. Feng, "Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning," 2021, *arXiv:2102.06605*.

[34] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 18661–18673.

[35] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, "Spatiotemporal contrastive video representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6960–6970.

[36] C.-Y. Chuang, R. D. Hjelm, X. Wang, V. Vineet, N. Joshi, A. Torralba, S. Jegelka, and Y. Song, "Robust contrastive learning against noisy views," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16649–16660.

[37] S. Li, X. Xia, S. Ge, and T. Liu, "Selective-supervised contrastive learning with noisy labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 316–325.

[38] M. J. Donkers, K.-N. An, E. Y. S. Chao, and B. F. Morrey, "Hand position affects elbow joint load during push-up exercise," *J. Biomechanics*, vol. 26, no. 6, pp. 625–632, Jun. 1993.

[39] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, M. Meila and T. Zhang, Eds. 2021, pp. 12310–12320. [Online]. Available: http://proceedings.mlr.press/v139/zbontar21a.html

[40] H. Hanselmann and H. Ney, "Learning local convolutional features for face recognition with 2D-warping," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 747–758.

[41] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proc. 21st AAAI Conf. Artif. Intell.*, S. P. Singh and S. Markovitch, Eds. San Francisco, CA, USA, 2017, pp. 4263–4270. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14437

[42] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-temporal attention-based LSTM networks for 3D action recognition and detection," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3459–3471, Jul. 2018, doi: 10.1109/TIP.2018.2818328.

[43] E. Kim, K. On, J. Kim, Y. Heo, S. Choi, H. Lee, and B. Zhang, "Temporal attention mechanism with conditional inference for large-scale multi-label video classification," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 11132, L. Leal-Taixe and S. Roth, Eds. Munich, Germany: Springer, 2018, pp. 306–316, doi: 10.1007/978-3-030-11018-5_28.

[44] Z. Wu, C. Xiong, C.-Y. Ma, R. Socher, and L. S. Davis, "AdaFrame: Adaptive frame selection for fast video recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1278–1287.

[45] S. N. Gowda, M. Rohrbach, and L. Sevilla-Lara, "SMART frame selection for action recognition," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI), 33rd Conf. Innov. Appl. Artif. Intell. (IAAI), 11th Symp. Educ. Adv. Artif. Intell. (EAAI)*, 2021, pp. 1451–1459. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16235

[46] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3D human pose estimation in video with temporal convolutions and semi-supervised training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7745–7754.

[47] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1857–1865.

CHUNG-IN JOUNG received the B.S. degree in computer engineering from Hongik University, in 2019, and the M.S. degree in computer engineering from Korea University, in 2022. His research interests include deep learning, computer vision, and action/video classifications.

SEUNGHWAN BYUN received the B.S. degree from Hongik University, in 2020, and the M.S. degree from Korea University, in 2023. His research interests include computer vision, AI in rehabilitation, or any other interesting topics dealing with vision and artificial intelligence.

SEUNGJUN BAEK (Member, IEEE) received the B.S. degree from Seoul National University, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Texas at Austin, in 2002 and 2007, respectively. From 2007 to 2009, he was a member of the Technical Staff with the DSP Systems Research and Development Center, Texas Instruments. In 2009, he joined the College of Informatics, Korea University, South Korea, where he is currently a Professor. His research interests include deep learning, data-driven optimization, mobile computing, and medical AI.