Structural Knowledge Distillation for Efficient Skeleton-Based Action Recognition

Cunling Bian, Wei Feng, *Member, IEEE*, Liang Wan, *Member, IEEE*, and Song Wang, *Senior Member, IEEE*

***Abstract*— Skeleton data have been extensively used for action recognition since they can robustly accommodate dynamic circumstances and complex backgrounds. To guarantee the action-recognition performance, we prefer to use advanced and time-consuming algorithms to get more accurate and complete skeletons from the scene. However, this may not be acceptable in time- and resource-stringent applications. In this paper, we explore the feasibility of using low-quality skeletons, which can be quickly and easily estimated from the scene, for action recognition. While the use of low-quality skeletons will surely lead to degraded action-recognition accuracy, in this paper we propose a structural knowledge distillation scheme to minimize this accuracy degradations and improve recognition model's robustness to uncontrollable skeleton corruptions. More specifically, a teacher which observes high-quality skeletons obtained from a scene is used to help train a student which only sees low-quality skeletons generated from the same scene. At inference time, only the student network is deployed for processing low-quality skeletons. In the proposed network, a graph matching loss is proposed to distill the graph structural knowledge at an intermediate representation level. We also propose a new gradient revision strategy to seek a balance between mimicking the teacher model and directly improving the student model's accuracy. Experiments are conducted on Kenetics400, NTU RGB+D and Penn action recognition datasets and the comparison results demonstrate the effectiveness of our scheme.**

***Index Terms*— Skeleton-based action recognition, structural knowledge distillation, graph matching loss, graph convolutional network, gradient revision.**

Manuscript received May 29, 2020; revised December 5, 2020 and January 7, 2021; accepted January 17, 2021. Date of publication February 9, 2021; date of current version February 17, 2021. This work was supported, in part, by the NSFC under Grants U1803264, 62072334, 61672376, 61671325. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Riccardo Leonardi. *(Corresponding authors: Wei Fang; Song Wang.)*

Cunling Bian and Wei Feng are with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics (SMARC), State Administration of Cultural Heritage, Tianjin 300350, China, and also with the Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin 300350, China (e-mail: clbian@tju.edu.cn; wfeng@ieee.org).

Liang Wan is with the School of Computer Software, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, and also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics (SMARC), State Administration of Cultural Heritage, Tianjin 300350, China.

Song Wang is with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics (SMARC), State Administration of Cultural Heritage, Tianjin 300350, China, also with the Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin 300350, China, and also with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: songwang@cec.sc.edu).

Digital Object Identifier 10.1109/TIP.2021.3056895

I. Introduction

With wide applications in video surveillance, human-machine interaction, and sport video analysis, human action recognition has recently attracted much attention in the computer-vision community [1], [2]. Multiple-modality information, such as appearance, depth, optical flows, and body skeletons [3], [4] have been used for human action recognition. Compared with the other modalities, skeletons capture the compact positions of the major body joints and are robust to inconsistent appearances, different environments, and varying viewpoints [5]. It has been proved empirically and theoretically that body joints provide highly effective information about human motion underlying different actions [6].

Obtaining the skeleton data is the prerequisite and most time/resource consuming step in the skeleton-based action recognition. By using sensor technology, special equipments like motion capture system and Kinect can directly collect skeleton data over time. However, the high cost and limited receptive field prevent them from being used in many applications. Given that most videos are taken by RGB cameras, it is higher desirable to use human pose estimation algorithms to get skeletons from RGB videos [7]–[9]. However, to get more accurate action recognition, we need more accurate skeletons, which require the use of more advanced and time-consuming human pose estimation methods. This prevents skeleton-based action recognition from being used in many time- and resource-stringent applications.

Certainly we can handle this problem by using low-quality skeletons (LQS), generated by human-pose estimation methods which sacrifice accuracy to reduce resource consumption, for action recognition. But this will lead to an action-recognition performance drop compared to the use of high-quality skeletons (HQS) estimated by costly methods. The focus of this paper is to develop a new approach that can minimize this action-recognition performance drop from the use of HQS to the use of LQS. Our basic idea is to train a deep neural network using HQS and then make it handle LQS for the deployment, with an expectation that the more knowledge underlying HQS can be embedded in the trained network for facilitating the LQS-based action recognition, i.e., HQS are only available for the training but not for testing.

To achieve this goal, following the basic principle of knowledge distillation [10], we propose a new structural knowledge distillation (SKD) scheme to distill information from HQS

and then convey it to LQS-based network training. More specifically, we start with training a deep network as a teacher that takes HQS as input. Then, we freeze the teacher and train a student network that takes LQS as input. The student is expected to not only classify the action correctly, but also "mimic" the teacher by producing the same representations at different levels. This is achieved by minimizing a graph matching loss [11], [12] between their intermediate representations and a divergence loss between the output distributions (class probabilities) predicted by the two networks. To seek a balance between mimicking teacher model and directly improving student model's recognition accuracy, we also propose a gradient revision strategy by considering their gradient angles. We conduct extensive evaluations on Kinetics400, NTU RGB+D and Penn action datasets by trying four scenarios of combined HQS and the corresponding LQS: i) manual annotation v.s. algorithm estimation, ii) high-cost v.s. low-cost human pose estimation, iii) regular v.s. reduced frames-per-second (fps) rate, iv) complete v.s. occluded skeleton. Experiments show that a network trained using the proposed new SKD scheme can significantly improve the LQS-based action-recognition performance.

To summarize, the main contributions of this paper are three-folds.

- We introduce a knowledge distillation scheme for skeleton-based action recognition. With this scheme, we only use LQS at the inference stage, but with less influence to the recognition accuracy.
- We develop a teacher-student graph convolutional network (GCN) and propose a graph matching loss to distill unary and pairwise structural knowledge at an intermediate representation level.
- We propose a gradient revision strategy to seek a balance between mimicking teacher model and directly improving student model's accuracy.

The remainder of the paper is organized as follows. Section II gives a brief review of the related work on skeleton-based action recognition, knowledge distillation and human pose estimation. In Section III, we describe our proposed structural knowledge distillation scheme. Section IV describes the benchmark datasets, pose estimation algorithms and experimental setting and reports the experiments, comparisons, and discussions, followed by a brief conclusion in Section V.

II. Related Work

A. Skeleton-Based Action Recognition

Conventional methods for skeleton-based action recognition use handcrafted features to represent the human body [13], [14]. With the development of deep neural network, data-driven methods have been shown to be superior over conventional ones. As far as we know, there are primarily three kinds of neural network structures for skeleton-based action recognition: RNN (Recurrent Neural Network), CNN and GCN. RNN-based methods usually model the skeleton data as a sequence of the coordinate vectors based on the established traversal strategy [7], [15]. CNN-based methods represent the skeleton data as a pseudo-image. Compared with RNNs, they have better parallelization and are easier to train [16]. In recent years, the properties of convolutional filters have been extended to graph data structure [17], [18]. They construct locally connected neighborhoods from the input graph, which is served as the receptive fields of a convolutional architecture. Therefore, the generalized convolution based propagation rules, being very similar to the inference of classical graph theoretical models [19]–[21], can be directly applied to graphs, which bring a significant boost for GCN. Instead of representing skeleton data as sequences or images, GCN-based methods model the data as a graph with joints as vertexes and bones as edges, which is a more intuitive way to represent the human body organization [18], [22], [23]. GCN eliminates the need for designing handcrafted part assignment or traversal rules, thus achieves better performance than many previous methods.

Most existing methods for skeleton-based action recognition deal with complete skeletons. But, it is inevitable that the captured skeletons are noisy and incomplete in reality. Such uncontrollable corruptions, e.g. joint missing or noise in detection [24], seriously affect the performance of existing methods. Therefore, it is very important to make classifiers more robust to these uncontrollable corruptions. However, there is very few work in this regard in skeleton-based action recognition.

B. Knowledge Distillation

Knowledge distillation is originally proposed as a model compression technique, which transfers knowledge from a complex model (teacher) to a simple model (student) by using class probabilities of the complex model as 'soft target' for the simple one [25]. A large quantity of approaches have been proposed to reinforce the efficiency of student models' learning capability. Romero *et al.* firstly put forward the concept of hint learning aiming at reducing the distance between intermediate representations of the student and teacher networks [26]. Agoruyko *et al.* considered this issue from the perspective of attention mechanism, attempting to force the student to mimic the attention maps of a powerful teacher network [27]. Recently, knowledge distillation has also been extended to instance relationships [28], [29].

A singular instance of generalized knowledge distillation is privileged information [30]. Learning under this paradigm provides a model trained with additional information available only in the training phase and not at testing time. Ge *et al.* proposed a learning approach to recognize low-resolution faces via distilling knowledge from high-resolution faces to extract discriminative features [31]. Karessli *et al.* introduced a weakly-supervised teacher-student training framework that leverages the power of statistical models combined with the rich visual information to learn visual cues [32]. Such learning paradigm has also been applied to action recognition [3], [10]. For instance, Crasto *et al.* introduced two learning approaches to training a standard 3D CNN, operating on RGB frames. That mimics the motion stream based on distillation framework, and as a result avoids flow computation at the testing time [33]. In order to reduce the number of FLOPs, Bhardwaj *et al.* used the idea of distillation

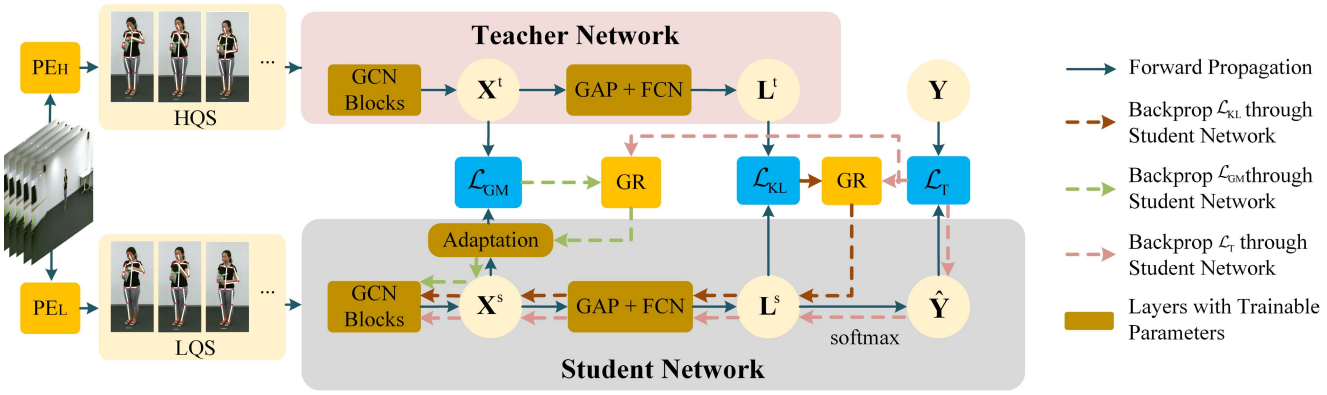


Fig. 1. The overall pipeline of the proposed SKD scheme for skeleton-based action classification.

albeit in a different setting, where a computationally-heavy teacher which looks at all the frames in the video is used to train a computationally-efficient student which looks at only a small fraction of frames in the video [3]. In our case, HQS is the privileged information available for training, along with LQS, but only LQS is available at the testing time.

C. Human Pose Estimation

Algorithms for RGB image based human pose estimation can be grouped into two categories: top-down and bottom-up algorithms. The former runs a person detector first and estimates body joints within the detected bounding boxes. Examples of top-down methods include PoseNet [34], HourglassNet [35] and HRNet [36]. They benefit from the advances of person detectors and vast amounts of available labeled persons in the form of bounding boxes. But a coin has two sides – top-down methods are effective but struggle when person bounding boxes overlap and their runtime is proportional to the number of detected people. The bottom-up algorithms estimate each body joint first and all the detected joints are then grouped to form a unique pose. They are more robust when people in the image are in close proximity. They also decouple runtime complexity from the number of people in the image by using greedy decoders in combination with additional tools, such as Associative Embedding [37], Part Affinity Fields [8], and Part Intensity and Association Fields [38]. In this paper, we systematically study the skeleton-based action recognition using two representative bottom-up pose estimation algorithms – Openpose [8] and PifPaf [38].

III. Structural Knowledge Distillation

The focus of this work is to design a scheme which makes a network look at LQS at inference time while still allowing it to leverage the information from HQS at training time. To achieve this, SKD is proposed wherein the teacher is given access to HQS and the student only deals with LQS. This section describes our approach in terms of its architecture, loss functions used to train the teacher and student networks, the gradient revision strategy and the ways to reduce the cost of skeleton generation.

As shown in Figure 1, a high performance pose estimation method PE_H and a low performance pose estimation method PE_L are applied to the same scene to generate HQS and LQS, respectively. Then the two skeleton sequences are fed into the following networks. We first train the teacher network minimizing cross-entropy loss $\mathcal{L}_\text{T}$ with HQS data. Then, we focus on training the student network with the help of the teacher, i.e., the model just trained, with the whole scheme shown in this figure. As the teacher model has been pre-trained with HQS data before, the weights of the teacher are frozen, while training the student. The student, receiving the input LQS data, is trained with $\mathcal{L}_\text{T}$, while also receiving guidances $\mathcal{L}_\text{KL}$ and $\mathcal{L}_\text{GM}$ from the teacher using the Gradient Revision (GR) strategy. When testing, only LQS is used as the input to compute the action class score.

Skeleton sequence of an action can be naturally organized as an undirected spatial-temporal graph $(\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{\mathbf{v}_{ti} \mid t = 1, \ldots, T; i = 1, \ldots, N\}$ denotes a set of vertices, corresponding to T frames and N body joints per frame, and $\mathcal{E}$ is the set of edges, indicating the connections between nodes. The edge set consists of two parts: temporal connections that link each joint with its counterpart in the neighboring frames, and spatial connections that include both direct and indirect kinematic dependencies in each frame. $\mathbf{A} = [a_{pq}]_{M\times M}$ with $M = TN$ is the adjacency matrix, with $a_{pq} = 1$ if vertices p and q are connected, and otherwise $a_{pq} = 0$. The matrix $\mathbf{A}$ fully describes the skeleton sequence structure.

To deal with the skeleton sequence structure, each GCN block usually contains a spatial graph convolution followed by a temporal convolution, which alternately extracts spatial and temporal features, respectively. The spatial graph convolution operation introduces weighted average of neighboring features for each joint, where the neighbors are defined based on the nodes' distance to the root node, to capture more refined location information. Let $\mathbf{X}_\text{in} \in \mathbb{R}^{b\times T\times N\times d_\text{in}}$ be the input features of spatial-temporal graph, where d_in is the input feature dimension of vertices and b is the batchsize, and $\mathbf{X}_\text{out} \in \mathbb{R}^{b\times T\times N\times d_\text{out}}$ be the output features obtained from spatial graph convolution, where d_out is the dimension of

output features. Therefore, the operation of spatial-temporal GCN block can be expressed as

$$\mathbf{X}_{\text{out}} = \text{GCNBlock}(\mathbf{X}_{\text{in}}, \mathbf{A}). \tag{1}$$

Originated from ST-GCN [18], a series of GCN based skeleton-based action recognition methods are proposed, such as AGCN [23] and AS-GCN [22]. In this paper, we propose a general scheme not to rely on a specific network architecture but cover a family of architectures used for skeleton-based action recognition. In another words, we believe most GCN-based methods can take the place of GCN blocks in the proposed scheme.

A. Teacher Network

The teacher network looks at HQS and computes a graph encoding $\mathbf{X}^{\text{t}}$ of the skeleton sequence with several consecutive spatial-temporal GCN blocks. After that, a global average pooling (GAP) is performed on the $\mathbf{X}^{\text{t}}$ to get a **1D** feature vector for each sequence and the dimension is reduced to the number of the classes $\mathcal{Y}$ by a fully convolutional network (FCN). The entire architecture is shown in the upper part of Figure 1. The parameters of the teacher network are trained using a standard multi-label classification loss, which is a sum of the cross-entropy loss for each of the $\mathcal{Y}$ classes. This loss is referred as $\mathcal{L}_{\text{T}}$ where the equation stands for cross entropy between the true labels $\mathbf{Y}$ and the predictions $\hat{\mathbf{Y}}$,

$$\mathcal{L}_{\text{T}} = -\frac{1}{b}\sum_{i=1}^{b}\sum_{j=1}^{|\mathcal{Y}|}\mathbf{Y}_{ij}\log\hat{\mathbf{Y}}_{ij} + \left(1-\mathbf{Y}_{ij}\right)\log\left(1-\hat{\mathbf{Y}}_{ij}\right). \tag{2}$$

B. Student Network

The student network processes LQS and computes a graph encoding $\mathbf{X}^{\text{s}}$ from the spatial-temporal graph. The GCN blocks architecture of the student is not necessary to be the same as the above teacher network, but $\mathbf{X}^{\text{t}}$ and $\mathbf{X}^{\text{s}}$ need to have the same size here. As the scale of the intermediate representations is usually different between the teacher and student networks, we introduce an adaptation layer on the top of the student network's intermediate layer to match the scale of the teacher network. The adaptation layer is represented by $r(\cdot;\boldsymbol{\mu})$ with parameter $\boldsymbol{\mu}$ learned during the knowledge distillation. As the representation layers of the teacher and student networks are structurally identical, we add 1×1 convolution to minimize the number of parameters and keep the size constant.

The student is trained to minimize the loss:

$$\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{T}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{GM}}. \tag{3}$$

The loss consists of three parts: the first term is the task-specific loss $\mathcal{L}_{\text{T}}$ and the other two are the imitation losses which are Kullback Leibler (KL) divergence loss $\mathcal{L}_{\text{KL}}$ and graph matching loss $\mathcal{L}_{\text{GM}}$, respectively. In the following, we discuss these three losses in detail.

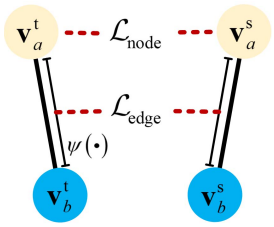


Fig. 2. Structure of graph matching loss $\mathcal{L}_{\text{GM}}$ (v_i^{t}): teacher's representation of i vertex; v_i^{s}: student's representation of i vertex; $\psi(\cdot)$: a relational potential function that measures a relation energy between two vertex. The graph matching Loss aims to distill not only unary but also pairwise structural knowledge between teacher and student at intermediate representation level.

1) Task-Specific Loss $\mathcal{L}_{\text{T}}$: For the action classification task, the objective of training is to minimize the cross-entropy error between the predicted values and the correct labels. Therefore, the task-specific loss $\mathcal{L}_{\text{T}}$ for the student network takes the same form as that in the teacher network, i.e., the one given in Eq. (2).

2) KL Divergence Loss $\mathcal{L}_{\text{KL}}$: Let the post-softmax predictions for the teacher and student networks be $\mathbf{P}^{\text{t}} = \text{softmax}(\mathbf{L}^{\text{t}})$ and $\mathbf{P}^{\text{s}} = \text{softmax}(\mathbf{L}^{\text{s}})$ respectively, where $\mathbf{L}^{\text{t}}$ and $\mathbf{L}^{\text{s}}$ are the pre-softmax predictions, also referred as logits. As the post-softmax have hosted some relative uncertainties that are more informative, a temperature parameter τ is used to soften predictions $\mathbf{P}^{\text{t}}$ and $\mathbf{P}^{\text{s}}$ to soft predictions $\widetilde{\mathbf{P}}^{\text{t}}$ and $\widetilde{\mathbf{P}}^{\text{s}}$ as

$$\widetilde{\mathbf{P}}^{\text{t}} = \text{softmax}(\frac{\mathbf{L}^{\text{t}}}{\tau}), \quad \widetilde{\mathbf{P}}^{\text{s}} = \text{softmax}(\frac{\mathbf{L}^{\text{s}}}{\tau}). \tag{4}$$

Then, considering $\widetilde{\mathbf{P}}^{\text{t}}$ as the target, we define the KL-divergence loss function

$$\mathcal{L}_{\text{KL}}(\widetilde{\mathbf{P}}^{\text{s}}||\widetilde{\mathbf{P}}^{\text{t}}) = \frac{\tau^2}{b}\sum_{i=1}^{b}\sum_{j=1}^{|\mathcal{Y}|}\widetilde{\mathbf{P}}_{ij}^{\text{s}}\log\frac{\widetilde{\mathbf{P}}_{ij}^{\text{s}}}{\widetilde{\mathbf{P}}_{ij}^{\text{t}}}. \tag{5}$$

In this way the student network learns to match the probability estimate of the teacher. $\mathcal{L}_{\text{KL}}$ is a frequently used loss function for knowledge distillation.

3) Graph Matching Loss $\mathcal{L}_{\text{GM}}$: In addition to the logits, it has been demonstrated that using the intermediate representation of the teacher as a hint can help the training process and improve the final performance of the student [26]. The main purpose of $\mathcal{L}_{\text{GM}}$ is to let $\mathbf{X}^{\text{s}}$ mimic $\mathbf{X}^{\text{t}}$. L_2 distance is usually used to measure the similarity between teacher and student representations which are the outputs of a convolutional layer. However, our work is mainly built on GCN and the structures of intermediate representations are spatial-temporal graph. In such a GCN, the features between vertexes are indispensable and are always neglected in traditional hint learning. In order to obtain more comprehensive supervision information at this level in the teacher network, here we propose a new graph matching loss $\mathcal{L}_{\text{GM}}$ by considering not only the representations associated to each vertex, but also the features computed between vertexes, which is shown in Figure 2. This way, we combine both the unary and pairwise structural knowledge for spatial-temporal graph representation.

We define the graph matching loss as

$$\mathcal{L}_{\text{GM}} = \lambda \mathcal{L}_{\text{node}} + (1-\lambda)\mathcal{L}_{\text{edge}}, \tag{6}$$

where λ is the parameter balancing the contributions of different loss terms during training. $\mathcal{L}_{\text{node}}$ and $\mathcal{L}_{\text{edge}}$ are unary and pairwise knowledge distillation losses respectively.

The objective for unary structural knowledge distillation $\mathcal{L}_{\text{node}}$ is to minimize the L_2 distance between teacher's and student's representations at each vertex. Combined with the adaption layer, it can be expressed as

$$\mathcal{L}_{\text{node}} = \frac{1}{bTN}\sum_{i=1}^{b}\sum_{j=1}^{T}\sum_{k=1}^{N}\|r(\mathbf{X}^{\text{s}}_{ijk};\boldsymbol{\mu}) - \mathbf{X}^{\text{t}}_{ijk}\|^2, \tag{7}$$

where T and N are the sizes of temporal and spatial dimensions of $\mathbf{X}^{\text{s}}$ and $\mathbf{X}^{\text{t}}$. It directly forces the student to match the teacher's output but largely ignore the structural information between vertices. $r(\mathbf{x};\boldsymbol{\mu}) = \boldsymbol{\mu}_{\text{weight}}\mathbf{x} + \boldsymbol{\mu}_{\text{bias}}$ represents an adaptation layer, where $\mathbf{x}$ is the input and $\boldsymbol{\mu}$ consists of $\boldsymbol{\mu}_{\text{weight}}$ and $\boldsymbol{\mu}_{\text{bias}}$. b is the batchsize. $\mathcal{L}_{\text{edge}}$ aims to transfer structural knowledge using mutual relations between vertices in the intermediate representation of GCN. A relation potential $\psi(\cdot)$ for each TN-tuple of vertices is calculated and then used to distill knowledge from teacher to student. The objective for structural knowledge distillation is expressed as

$$\mathcal{L}_{\text{edge}} = \frac{1}{b}\sum_{i=1}^{b} l\big(\psi(r(\mathbf{X}^{\text{s}}_{i11};\boldsymbol{\mu}),\ldots,r(\mathbf{X}^{\text{s}}_{iTN};\boldsymbol{\mu})), \psi(\mathbf{X}^{\text{t}}_{i11},\ldots,\mathbf{X}^{\text{t}}_{iTN})\big), \tag{8}$$

where $\psi(\cdot)$ is a relational potential function that measures a relation energy of the given TN-tuple, and $l(\cdot)$ is a loss that penalizes difference between the teacher and student. $\mathcal{L}_{\text{edge}}$ trains the student network to form the same relational structure as that of the teacher in terms of the used $\psi(\cdot)$. Therefore, relational potential function $\psi(\cdot)$ plays an essential role in this loss definition. In this work, for the sake of simplicity, we just take one simple yet effective potential function: distance-wise loss function to exploit pairwise relations between vertices. More specificially, *distance-wise* potential function $\psi_{\text{D}}(\cdot)$ measures the Euclidean distance between the two vertices in the intermediate representation as

$$\psi_{\text{D}}(\mathbf{v}_i,\mathbf{v}_j) = \frac{1}{\beta}\|\mathbf{v}_i - \mathbf{v}_j\|^2, \tag{9}$$

where $\beta = \frac{1}{|\mathcal{V}^2|}\sum_{(\mathbf{v}_i,\mathbf{v}_j)\in\mathcal{V}^2}\|\mathbf{v}_i - \mathbf{v}_j\|^2$. Here β is a normalization factor of the distance. To consider relative distances among other pairs, we set β to be the average distance between pairs from $\mathcal{V}^2$ in the mini-batch. Using the distance-wise potentials measured in both the teacher and student, Eq (8) is transformed to

$$\mathcal{L}_{\text{edge}} = \frac{1}{b}\sum_{k=1}^{b}\sum_{i,j=1\ldots T}\sum_{m,n=1\ldots N} l_\delta\big(\psi_{\text{D}}(r(\mathbf{X}^{\text{s}}_{kim};\boldsymbol{\mu}), r(\mathbf{X}^{\text{s}}_{kjn};\boldsymbol{\mu})), \psi_{\text{D}}(\mathbf{X}^{\text{t}}_{kim},\mathbf{X}^{\text{t}}_{kim})\big), \tag{10}$$

where $l_\delta(\cdot)$ is the Huber loss,

$$l_\delta(x,y) = \begin{cases} \frac{1}{2}(x-y)^2, & |x-y| \le 1, \\ |x-y| - \frac{1}{2}, & \text{otherwise.} \end{cases} \tag{11}$$

C. Gradient Revision

In this section, we propose Gradient Revision (GR), a strategy for knowledge distillation in the privilege information context. The main feature of GR is that it revises the gradients of knowledge distillation according to the task-specific gradient. When adopting knowledge distillation method to train a student model, blindly mimicking teacher model features is too blunt as it fails to account for shift distribution between teacher and student data. Occasionally, there are conflicts between mimicking teacher model and directly improving student model's accuracy in the training stage. Based on this consideration, we focus on minimizing negative backward transfer, which may increase the task-specific loss, to deal with these conflicts.

GR tries to ensure that minimizing knowledge distillation loss would not increase the task-specific loss at every training step using the relationships between their gradients. We define $f^{\text{s}}_{\boldsymbol{\theta}}$, $f^{\text{t}}_{\boldsymbol{\omega}}$ to be the student and the teacher models, parameterized by $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$, respectively. $\mathcal{D}^{\text{s}}$ and $\mathcal{D}^{\text{t}}$ are the datasets for them. Formally, the objective of GR is:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{KD}}\left(f^{\text{s}}_{\boldsymbol{\theta}}, f^{\text{t}}_{\boldsymbol{\omega}}, \mathcal{D}^{\text{s}}, \mathcal{D}^{\text{t}}\right) \\ &\text{s.t. } \mathcal{L}_{\text{T}}\left(f^{\text{s}}_{\boldsymbol{\theta}}, \mathcal{D}^{\text{s}}\right) \le \mathcal{L}_{\text{T}}\left(f^{\text{s,pre}}_{\boldsymbol{\theta}}, \mathcal{D}^{\text{s}}\right), \end{aligned} \tag{12}$$

where $f^{\text{s,pre}}_{\boldsymbol{\theta}}$ is the network trained before parameter update. Assuming that the function is locally linear (as is happens around small optimization steps), we can diagnose increases in the task-specific loss by computing the angle between the loss gradient vectors of knowledge distillation $\mathbf{g}_{\text{KD}}$, and the current classification task $\mathbf{g}_{\text{T}}$. If $\mathbf{g}^\top_{\text{KD}}\mathbf{g}_{\text{T}} \ge 0$, then the proposed parameter update $\mathbf{g}_{\text{KD}}$ is unlikely to increase the loss for the classification task. On the other hand, if the inequality constraint is violated, then the classification task would experience an increase in loss after the parameter update depending on $\mathbf{g}_{\text{KD}}$. Revising $\mathbf{g}_{\text{KD}}$ with a minimal change to satisfy the inequity constraint is an effective trade-off. It can avoid the increasing of the task-specific loss and in the mean time decreases the knowledge distillation loss as much as possible. Therefore, whenever the angle is great than 90°, GR projects $\mathbf{g}_{\text{T}}$ to the closest in the L_2 norm gradient $\tilde{\mathbf{g}}$ that keeps the angle within the bounds. Formally, the optimization problem GR is given by:

$$\text{argmin}_{\tilde{\mathbf{g}}}\frac{1}{2}\|\mathbf{g}_{\text{KD}} - \tilde{\mathbf{g}}\|_2^2 \quad \text{s.t.} \quad \tilde{\mathbf{g}}^\top\mathbf{g}_{\text{T}} \ge 0. \tag{13}$$

The gradient update rule can be obtained by solving this constrained optimization problem, which is expressed as:

$$\tilde{\mathbf{g}} = \mathbf{g}_{\text{KD}} - \frac{\mathbf{g}^\top_{\text{KD}}\mathbf{g}_{\text{T}}}{\mathbf{g}^\top_{\text{T}}\mathbf{g}_{\text{T}}}\mathbf{g}_{\text{T}}. \tag{14}$$

The formal proof of the update rule of Eq (14) is given in Appendix A. Algorithm 1 summarizes the training of GR with

Algorithm 1 Training With GR Over Privilege Information

Require: $\mathcal{D}^{\mathrm{s}}, \mathcal{D}^{\mathrm{t}}, f^{\mathrm{s}}_{\boldsymbol{\theta}}, f^{\mathrm{t}}_{\boldsymbol{\omega}}$.
Ensure: $(\mathbf{x}^{\mathrm{s}}, \mathbf{y}) \in \mathcal{D}^{\mathrm{s}}, (\mathbf{x}^{\mathrm{t}}, \mathbf{y}) \in \mathcal{D}^{\mathrm{t}}, \mathbf{x}^{\mathrm{s}}$ and $\mathbf{x}^{\mathrm{t}}$ belong to identical videos.
  for $(\mathbf{x}^{\mathrm{s}}, \mathbf{x}^{\mathrm{t}}, \mathbf{y})$ in $(\mathcal{D}^{\mathrm{s}}, \mathcal{D}^{\mathrm{t}})$ **do**
    $\mathbf{g}_{\mathrm{T}} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{T}} \left(f^{\mathrm{s}}_{\boldsymbol{\theta}}, \mathbf{x}^{\mathrm{s}}, \mathbf{y}\right)$;
    for $q = 1, ..., Q$ **do**
      $\mathbf{g}^{q}_{\mathrm{KD}} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}^{q}_{\mathrm{KD}} \left(f^{\mathrm{s}}_{\boldsymbol{\theta}}, f^{\mathrm{t}}_{\boldsymbol{\omega}}, \mathbf{x}^{\mathrm{s}}, \mathbf{x}^{\mathrm{t}}\right)$;
      $\tilde{\mathbf{g}}^{q} \leftarrow \mathrm{GR}\left(\mathbf{g}^{q}_{\mathrm{KD}}, \mathbf{g}_{\mathrm{T}}\right)$;
    end for
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha(\mathbf{g}_{\mathrm{T}} + \tilde{\mathbf{g}}_1 + ... + \tilde{\mathbf{g}}_Q)$;
  end for

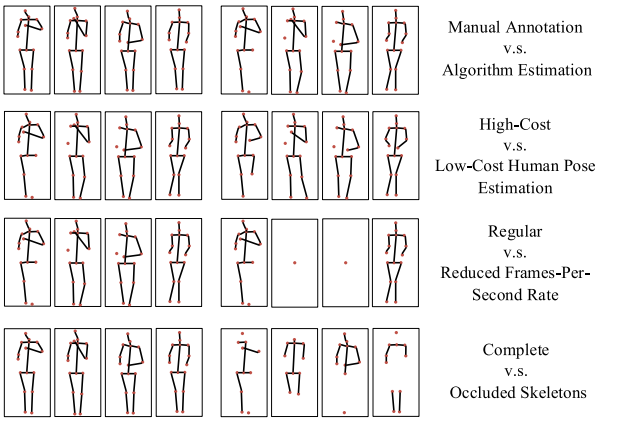


Fig. 3. The demonstration of the four scenarios forming HQS and LQS which aim to evaluate the proposed SKD scheme.

more than one knowledge distillation losses, where α is learning rate and Q is the number of knowledge distillation losses. In our proposed method, we consider $Q = 3$ knowledge distillation losses $\mathcal{L}^{1}_{\mathrm{KD}} = \mathcal{L}_{\mathrm{KL}}$, $\mathcal{L}^{2}_{\mathrm{KD}} = \mathcal{L}_{\mathrm{node}}$, and $\mathcal{L}^{3}_{\mathrm{KD}} = \mathcal{L}_{\mathrm{edge}}$. In the ablation studies of the later experiments, we will compare with the cases using fewer knowledge distillation losses. Each example in any of these dataset $\mathcal{D}$ consists of a tuple defined by an input $\mathbf{x}$ and a target vector $\mathbf{y}$. In this study, $\mathbf{x}$ and $\mathbf{y}$ are a skeleton sequence and its action label respectively.

D. Different-Quality Skeletons

There are many factors in pose estimation can affect the quality of skeletons in different degrees. In this paper, we perform an extensive evaluation of the proposed scheme with different ways to construct LQS: Three of them construct LQS by reducing the cost of skeleton generation, and one has LQS with missing joints caused by occlusions or broken sensors, as shown in Figure 3.

1) Manual Annotation v.s. Algorithm Estimation: Despite the great development of human pose estimation with deep learning, there is still an inescapable gap between machine and human. Some challenges are remained to be unsolved, such as the influence of body part occlusion and rare poses. In this context, manually annotated skeletons are HQS while the outputs of a pose-estimation algorithms are LQS.

2) High-Cost v.s. Low-Cost Human Pose Estimation: Using a smaller network is one of the most direct and effective manners to reduce the cost of pose estimation. Meanwhile, it is undeniable that taking a weaker network will decrease the performance for a specific algorithm. In this paper, a recently proposed pose estimation algorithm PifPaf is used to generate skeletons on every frame of the video given that its backbone network can be easily changed. The model with ShuffleNet v2 1× backbone gets 50.2 AP on COCO 2017 val set while with ResNet152 can reach 67.8 AP [38]. ShuffleNet is much smaller but faster than ResNet152. In this context, ShuffleNet is selected to generate LQS and ResNet152 generates HQS.

3) Regular v.s. Reduced Frames-Per-Second Rate: Another scenario forming LQS is that skeletons are generated on temporally sampled frames, which clearly save the execution times of pose estimation. It can reduce the computational cost but is bound to lose some fine movement information. This setup will verify whether the proposed scheme can handle the quality degradation in temporal dimension. To realize it, we use a pose estimation algorithm to generate skeletons on each frame of a video as HQS. While there are many strategies to reduce the skeleton FPS rate [3], in this paper LQS is obtained by uniformly sampling k frames from the video for pose estimation.

4) Complete v.s. Occluded Skeleton: It is a very common scenario of pose estimation methods that the obtained skeletons are incomplete due to broken sensors or occlusions. By using more reliable equipments, such as a motion capture system, more complete skeletons could be generated and used as HQS to guide the incomplete-skeleton based action recognition. Following the work of [24], we construct a synthetic occlusion dataset based on the NTU 3D Skeleton dataset, where every joint is set to be occluded (set to zero) with a certain probability. In this context, the original complete skeletons from NTU 3D Skeleton dataset are HQS while the occluded skeletons from the synthetic NTU 3D Skeleton dataset are LQS.

IV. Experimental Results

A. Setup

1) Dataset: We evaluate the proposed scheme on three public available action recognition benchmark datasets: Kinetics400 dataset [39], NTU RGB+D dataset [7] and Penn dataset [40]. Deepmind **Kinetics400** human action dataset contains video clips retrieved from YouTube. Because Kinetics is a video-based dataset for action recognition which only provides raw video clips without skeleton data, we have to extract skeletons from each frame in Kinetics using pose estimation algorithms. Following [18], 260,000 clips with reasonable pose estimations are selected for our experiment and they are divided into a training set of 240,000 clips and a testing set of 20,000 clips. **NTU RGB+D** is the largest public dataset for multi-modal video action recognition. It consists of 56,880 videos with more than four million frames, available in four modalities: RGB videos, depth sequences, infrared frames, and 3D skeleton data. Each action is captured by three cameras at the same height but from different horizontal

angles. Therefore, the dataset is very challenging due to the large intra-class and viewpoint variations. The original paper of the NTU RGB+D dataset recommends two benchmarks [7]: 1) Cross-subject (CS): the dataset is divided into a training set and a testing set, where the actors in the two subsets are different. 2) Cross-view (CV): the training set contains videos that are captured by cameras 2 and 3, and the testing set contains videos that are captured by camera 1. We follow this convention and report performance on both benchmarks. **Penn** is an in-the-wild human action dataset containing 2,326 challenging consumer videos. The main difference between Penn and other action recognition datasets is that each video in Penn is manually annotated with 13 human body joints in 2D frame-by-frame. In evaluation, following their original paper we adopt an even split of the videos for the training and testing sets [40].

2) Pose Estimation Algorithm: In this work, we take the joint locations in the pixel coordinate system as input and discard the raw RGB frames. Two representative pose estimation algorithms, Openpose [8] and PifPaf [38], are used to obtain the skeletons from RGB frames. **Openpose** is the first public available real-time toolbox for multi-person 2D pose detection, including body, foot, hand, and facial keypoints. Since this toolbox can only handle a fixed input images' resolution, we need to first resize all the video frames' resolution to 340×256. Openpose is then applied to process each frame independently and produces 2D coordinates $(X; Y)$ in the pixel coordinates system with confidence score C for 18 joints. Thus, each joint is represented with a tuple of $(X; Y; C)$. We select 2 people with the highest average joint confidence in each video for the multi-person cases and pad every skeleton sequence by replaying it from the start to have the predetermined uniform length $T = 300$. Therefore, the shape of tensor for each sample is $(18, 3, T, 2)$. **PifPaf** is one of the state-of-the-art bottom-up methods on the pose estimation task [38]. In addition to excellent performance, the main reason for choosing PifPaf is its release of the pre-trained models with different network backbones. Therefore, we can analyze how effectively the proposed scheme works in handling skeleton quality degradation for action recognition. The output data format of PifPaf is broadly similar with Openpose, except for the number of the estimated joints. PifPaf gives coordinates for 17 joints in COCO format. Therefore, the shape of tensor for each sample is $(17, 3, T, 2)$.

3) Evaluation: We train the models on the training set and report Top-1 and Top-5 accuracies on the testing set. Since reducing the cost of skeleton generation is one of the advantages for SKD, we also provide params, FLOPs and running time of inference model as indicators. To facilitate the understanding, We also follow a naming rule to represent different variants of our scheme. We first denote three variable components of our scheme, i.e. pose estimation network, action recognition network, frame sampling rate, as PENet, ARNet, and FSR, respectively. In this work, PENet$\in$ {Openpose, Pifpaf, Human}, where 'Human' indicates manually annotated poses. ARNet$\in${ST-GCN, AS-GCN}, and FSR takes different values of k. For simplicity, we name our scheme as SKD(PENet, ARNet, FSR), where PENet, ARNet and FSR are the ones used for the deployed student network. Meanwhile, ResNet152 and ShuffleNet v2 $1\times$ are abbreviated as ResNet and ShflNet for short. Two versions of PifPaf with the above network backbones are named as PifPaf-ResNet and PifPaf-ShflNet, respectively.

TABLE I
QUANTITATIVE RESULTS ON PENN ACTION DATASET

Model	Accuracy	
	Top-1	Top-5
HQS-T(Human, ST-GCN, k=1)	83.43	98.33
LQS-S(PifPaf-ResNet, ST-GCN, k=1)	63.74	92.99
LQS-SWA-S(PifPaf-ResNet, ST-GCN, k=1)	68.08	95.33
SKD(PifPaf-ResNet, ST-GCN, k=1)	**81.87**	**95.44**

We compare SKD(PENet, ARNet, FSR) with the following networks: (a) HQS-T(PENet, ARNet, FSR): The original teacher network which processes HQS. This, in some sense, provides the upper bound of the action-recognition performance. (b) LQS-S(PENet, ARNet, FSR): We consider a network which is trained from scratch using LQS as baseline. But unlike SKD, it does not get any guidance from a teacher. (c) LQS-SWA-S(PENet, ARNet, FSR): We also consider a network which is trained from scratch using smoothed LQS as baseline. The transformation is simple, which uses sliding window average to fill missing values and denoise along the time dimension, and aims to improve skeleton quality.

4) Hyperparameters: For all experiments, we use Stochastic Gradient Descent optimizer with nesterov to train models. The initial learning rate is set to 0.1, momentum equals to 0.9, batch size is 16 and weight decay is 10^{-4}. We set the epoch number to 60 and decay the learning rate by 0.1 after 15, 30 and 45 epochs. When training student network using the proposed scheme, we directly set the temperature $\tau = 1$, and $\lambda = 0.5$. The sliding window size is set to 3 for all experiments. All experiments are performed on Pytorch deep learning framework with one GeForce RTX 2080Ti GPU.

B. Results and Discussion

Since we have three benchmark action recognition datasets, four scenarios forming HQS and LQS, five different combinations of loss functions and three kinds of SKD network architectures, as detailed in later ablation studies, the total number of experiments that we can run is very large. To avoid redundancy, we only try the low-cost pose estimation model for constructing LQS on NTU RGB+D dataset. Apart from the ablation studies, SKD in all the experiments is trained using a combination of the three loss functions ($\mathcal{L}_{\mathrm{T}}$, $\mathcal{L}_{\mathrm{KL}}$, $\mathcal{L}_{\mathrm{GM}}$) with the GR and adaption layer.

1) Manual Annotation v.s. Algorithm Estimation: In Table I, it can be found that there is a huge gap in using manually annotated and automatically estimated skeletons for action recognition. Top-1 classification accuracy for HQS-T(Human, ST-GCN, $k = 1$) on Penn action dataset is 83.43% while it drops to 63.74% for LQS-S(PifPaf-ResNet, ST-GCN, $k = 1$). This result clearly shows that the quality of skeletons significantly influence the recognition performance. The proposed method, SKD(PifPaf-ResNet, ST-GCN, $k = 1$), using the model trained on manually annotated skeletons as a

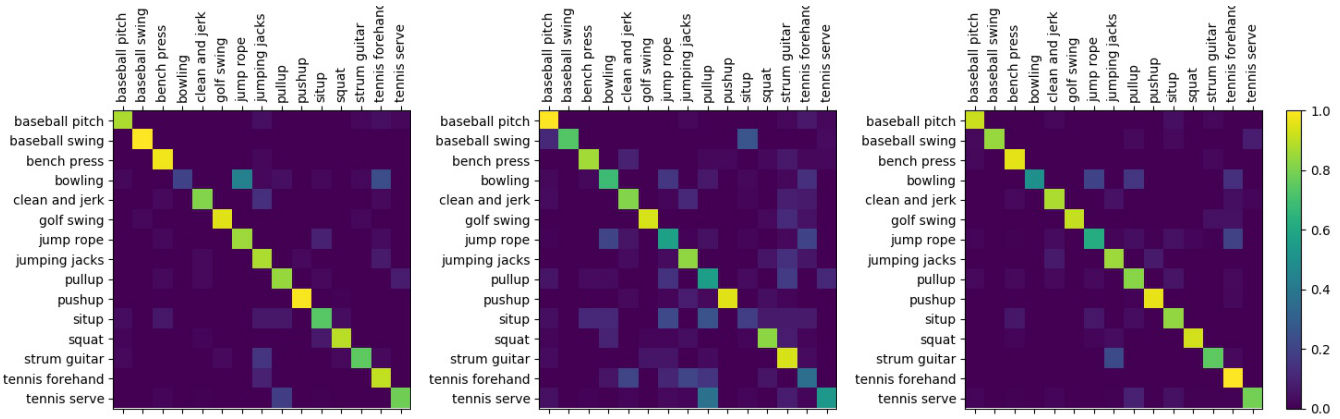


Fig. 4. The confusion matrices on Penn action dataset obtained by HQS-T(Human, ST-GCN, $k = 1$)(left), LQS-S(PifPaf-ResNet, ST-GCN, $k = 1$)(middle) and SKD(PifPaf-Res5Net, ST-GCN, $k = 1$)(right) respectively.

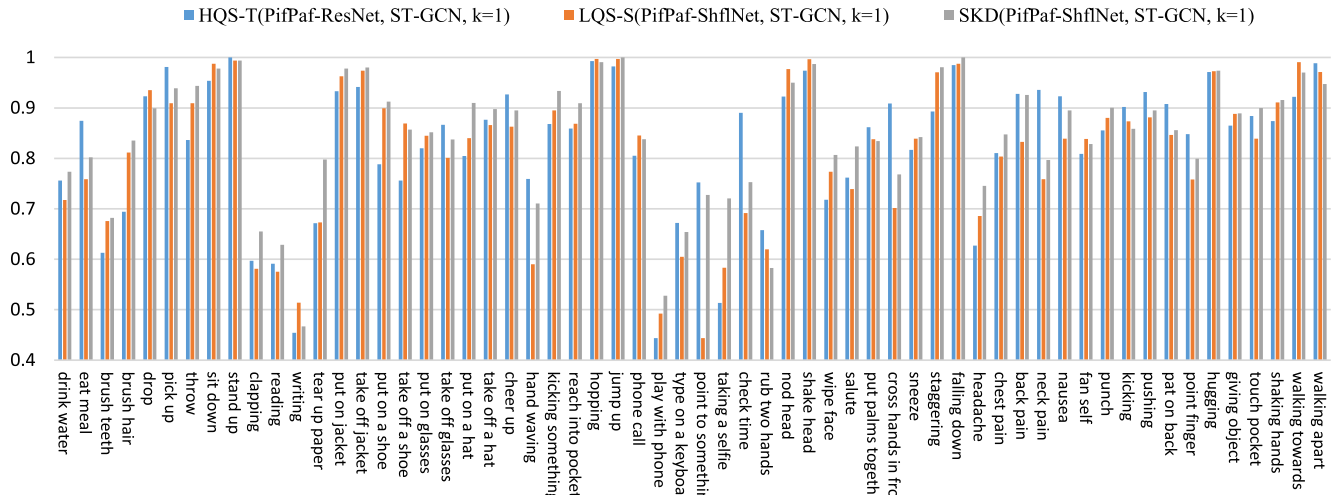


Fig. 5. Classification accuracy for each action on NTU RGB+D action dataset. There are 60 action classes including both individual and mutual actions.

teacher to help LQS-S(PifPaf-ResNet, ST-GCN, $k = 1$), can achieve a much high recognition accuracy of 81.87%. While LQS-SWA-S plays a positive role, its effect is not nearly as good as SKD. The confusion matrices for these methods are shown in Figure 4. The proposed scheme performs very well on most of the actions. The recognition performance of some actions, such as "sit up" and "tennis forehand" is significantly improved after applying SKD. We can also discover from the figure that the performance of SKD(PifPaf-ResNet, ST-GCN, $k = 1$) is closer to HQS-T(Human, ST-GCN, $k = 1$) than to LQS-S(PifPaf-ResNet, ST-GCN, $k = 1$). It means that, using only the automatically estimated LQS, SKD model's performance is still comparable to the model that takes the manually annotated HQS as input.

2) High-Cost v.s. Low-Cost Human Pose Estimation: In Table II, we summarize the results obtained using different PENet, ARNet, and FSR on NTU RGB+D dataset. These experiments have illustrated that the vast majority operations serve pose estimation process, so declining cost of it is the most efficient way for time- and resource-stringent applications. It can be drawn from the results of HQS-T(PifPaf-ResNet, ST-GCN, $k = 1$) and LQS-S(PifPaf-ShflNet, ST-GCN, $k = 1$) that switching PifPaf network backbone from ResNet to ShflNet can reduce the space complexity by 96% and accelerate the recognition speed by 54%, but also leading to a significant drop in recognition performance. In SKD(PifPaf-ShflNet, ST-GCN, $k = 1$), we use ShflNet for pose estimation in the student network and ResNet for the teacher network. It can be observed that the Top-1 and Top-5 accuracy in both CS and CV settings are improved as compared to LQS-S(PifPaf-ShflNet, ST-GCN, $k = 1$). Although LQS-SWA-S sometimes also achieves better results than LQS-S, its performance is not as stable as SKD. One possible reason is that the main purpose of SKD is to learn action recognition effective features from teacher while the improvement process of LQS have no strong correlation with action recognition. To investigate the difference of recognition rates of different actions, we further depict and compare the accuracies of the actions. Figure 5 shows the results of cross-view evaluation on the NTU RGB+D dataset. For half of the actions, SKD(PifPaf-ShflNet, ST-GCN, $k = 1$) gets the highest accuracies. But for ten actions, i.e., "drop", "sit down", "take of a shoe" and "phone call", LQS-S(PifPaf-ShflNet, ST-GCN, $k = 1$) gets the best results. For the rest of actions, i.e., "eat meal", "pick up", "stand up" and "take of glasses", HQS-T(PifPaf-ResNet, ST-GCN, $k = 1$) achieves the highest performance.

TABLE II

QUANTITATIVE RESULTS ON NTU RGB+D ACTION DATASET. FOR THE METRICS OF PARAMS, FLOPS AND TIME, REPRESENTATION 'A+B' INDICATES THE METRIC VALUE A OF POSE ESTIMATION MODEL AND B OF ACTION RECOGNITION MODEL

Model	Params (M)	FLOPs (G)	Time (S)	Accuracy			
				CS		CV	
				Top-1	Top-5	Top-1	Top-5
HQS-T(PifPaf-ResNet, ST-GCN, k=1)	60.19+2.80	4010.84+9.38	36.60+0.0065	82.03	98.20	88.99	99.46
LQS-S(PifPaf-ShflNet, ST-GCN, k=1)	2.28+2.80	103.64+9.38	16.80+0.0065	76.58	96.70	81.12	98.19
LQS-SWA-S(PifPaf-ShflNet, ST-GCN, k=1)	2.28+2.80	103.64+9.38	16.80+0.0065	78.35	96.69	80.29	97.94
SKD(PifPaf-ShflNet, ST-GCN, k=1)	2.28+2.80	103.64+9.38	16.80+0.0065	**80.80**	**97.56**	**84.42**	**98.80**
LQS-S(PifPaf-ResNet, ST-GCN, k=$\frac{1}{10}$)	60.19+2.80	401.08+9.38	3.66+0.0065	74.54	96.49	79.42	98.18
LQS-SWA-S(PifPaf-ResNet, ST-GCN, k=$\frac{1}{10}$)	60.19+2.80	401.08+9.38	3.66+0.0065	74.42	96.37	83.91	99.00
SKD(PifPaf-ResNet, ST-GCN, k=$\frac{1}{10}$)	60.19+2.80	401.08+9.38	3.66+0.0065	**78.19**	**97.54**	**84.96**	**99.12**
LQS-S(PifPaf-ShflNet, ST-sGCN, k=1)	2.28+1.89	103.64+5.77	16.80+0.0047	75.25	96.19	78.51	97.69
LQS-SWA-S(PifPaf-ShflNet, ST-sGCN, k=1)	2.28+1.89	103.64+5.77	16.80+0.0047	77.12	96.37	79.02	97.58
SKD(PifPaf-ShflNet, ST-sGCN, k=1)	2.28+1.89	103.64+5.77	16.80+0.0047	**79.79**	**97.40**	**83.39**	**98.62**
LQS-S(PifPaf-ShflNet, ST-2sGCN, k=1)	2.28+0.98	103.64+2.16	16.80+0.0028	73.32	95.71	72.95	96.04
LQS-SWA-S(PifPaf-ShflNet, ST-2sGCN, k=1)	2.28+0.98	103.64+2.16	16.80+0.0028	73.84	95.14	76.93	97.07
SKD(PifPaf-ShflNet, ST-2sGCN, k=1)	2.28+0.98	103.64+2.16	16.80+0.0028	**76.36**	**96.96**	**79.69**	**98.21**

These indicate that although HQS-T is superior to SKD in the overall accuracies, it is not as good as SKD according to the number of actions which get highest accuracies. To sum up, there is an obvious performance boost for most of the actions after applying SKD and the proposed scheme can effectively handle the skeleton quality degradation caused by a lost-cost pose estimation model for action recognition. The proposed SKD scheme can not only handle the skeleton quality degradation but also compress recognition model at the same time. To prove it, we test SKD with two compact ST-GCNs, ST-sGCN and ST-2sGCN on NTU RGB+D. ST-sGCN is constructed by abandoning the 2nd, 6th and 9th st-gcn blocks in ST-GCN while ST-2sGCN abandon the 2nd, 3rd, 6th, 7th, 9th and 10th st-gcn blocks. As shown in Table II, SKD gets better results on ST-sGCN and ST-2sGCN than LQS-S, which clearly indicates that SKD can also transfer knowledge to a compact student network simultaneously.

3) Regular v.s. Reduced Frames-Per-Second Rate: Table II and Table III show the action recognition performance by varying FSR for constructing HQS and LQS. $k = \frac{1}{10}$ means that pose is estimated only on one out of every ten frames, which can save 90% computational cost for pose estimation. We can see that the decrease in skeleton frame rate leads to a significant performance drop in both NTU RGB+D and Kinetics400. Based on the proposed scheme, Top-1 accuracy in the CS and CV settings on NTU RGB+D can be improved by 3.65% and 5.54% respectively, by comparing SKD(PifPaf-ResNet, ST-GCN, $k = \frac{1}{10}$) with LQS-S(PifPaf-ResNet, ST-GCN, $k = \frac{1}{10}$). Although Openpose is efficient to realize real-time pose estimation, reducing skeleton frame rate is still meaningful for low computing power devices and heavy computing tasks. From the results of LQS-T(OpenPose, ST-GCN, $k = \frac{1}{10}$) and SKD(OpenPose, ST-GCN, $k = \frac{1}{10}$) on Kinetics400 in Table III, we can observe that the scheme's benefits are very clear: it can achieves 21.88% and 12.17% increase for Top-1 and Top-5 accuracies respectively, when compared to the baseline LQS-based method.

4) Complete v.s. Occluded Skeleton: During the process of skeleton generation, broken sensors or part occlusions may occur, leading to the total missing of some joints in detection. The purpose of random occlusion experiments is to imitate this situation. The occlusion probability for each joint is set to 0.3 and 0.6, respectively. An overall summary of the results is shown in V, from which we can see that there is a rapid performance degradation when the occluded probability increases due to increased action information loss. Although SKD obviously alleviates the performance deterioration caused by joint occlusion, we found that a simple data completion method, such as sliding window average, can work even better by making up for the loss of action information. Note that the moving window approach is not corruption agnostic, i.e., it cannot work for any arbitrary low quality skeleton, but only specific cases like when the occlusions are at random. Whereas the proposed SKD approach is in some sense agnostic to the kind of skeleton corruptions since it makes no assumptions on how the skeleton quality maybe degraded. Meanwhile, there is no conflict between SKD and such simple data-completion methods. As shown at the bottom of V, we combine both SDK and sliding window average methods, and get further improved performance. This shows that skeleton quality enhancement methods and the proposed SKD can complement each other for action recognition.

The proposed SKD scheme can also be applied to other GCN based recognition models. We evaluate the effect of SKD with several state-of-the-art GCN-based approaches on the synthetic NTU 3D Skeleton dataset. The results are also given in V. We can see that SKD successfully improves all of them in handling incomplete skeletons and this verifies the generality of the proposed SKD to different GCN-based action recognition models. In summary, SKD has strong robustness to uncontrollable corruptions in the form of noise or errors in pose estimation.

This requirement might not always be satisfied that generating HQS for one specific action recognition dataset. In this situation, SKD can be adpoted to trained an effective feature extraction module on one datasset which has HQS as privilege information. The pretrained module can be adopted as a feature extraction module into LQS-based action recognition model on another dataset, where only LQS are available. In order to prove the feasibility of this method, we first

TABLE III

QUANTITATIVE RESULTS ON KINETICS400 ACTION DATASET.FOR THE METRICS OF PARAMS, FLOPS AND TIME, REPRESENTATION 'A+B' INDICATES THE METRIC VALUE A OF POSE ESTIMATION MODEL AND B OF ACTION RECOGNITION MODEL

Model	Params (M)	FLOPs (G)	Time (S)	Accuracy	
				Top-1	Top-5
HQS-T(OpenPose, ST-GCN, k=1)	52.31+2.80	71997.00+9.95	22.20+0.0066	30.77	53.53
Feature Enc. [14]	52.31+ -	71997.00+ -	22.20+ -	14.90	25.80
Deep LSTM [7]	52.31+0.17	71997.00+0.10	22.20+0.0072	16.40	35.30
Temporal Conv. [41]	52.31+0.93	71997.00+1.12	22.20+0.0065	20.30	40.00
LQS-S(OpenPose, ST-GCN, k=$\frac{1}{10}$)	52.31+2.80	7199.70+9.95	2.22+0.0066	23.54	43.73
LQS-SWA-S(OpenPose, ST-GCN, k=$\frac{1}{10}$)	52.31+2.80	7199.70+9.95	2.22+0.0066	21.44	41.21
SKD(OpenPose, ST-GCN, k=$\frac{1}{10}$)	52.31+2.80	7199.70+9.95	2.22+0.0066	**28.68**	**49.05**

TABLE IV

TOP1 ACCURACY ON THE CS BENCHMARK OF MININTU120 OCCLUSION DATASATE USING PRETRAINED ST-GCN MODEL FROM SYNTHETIC NTU 3D SKELETON DATASET

Pretrained Model	Occluded Probability	
	0.3	0.6
ST-GCN [18]	79.13	78.39
SKD(ST-GCN)	**81.77**	**79.95**

pretrained two action recognition models on the synthetic NTU 3D skeleton dataset with and without SKD, respectively. And then, the models are fineturned on another occlusion dataset. The dataset is consist of ten action categories sampled from NTU RGB+D 120 dataset [42], where subjects and categories are completely different from the synthetic NTU 3D skeleton datasate. Follow the above mentioned occlusion strategy, we build this dataset, which is named as miniNTU120 occlusion datasate. The results are shown in IV. Here, we can see that model's Top-1 accuracies using pretrained SKD(ST-GCN) model are 81.77% and 79.95%, while without SKD are 79.13% and 78.39%. Therefore, though obtaining an effect effective feature extraction module to handle LQS, SKD palys a positive role in cross-dataset training and testing.

5) Visualization of Model Representations: Apart from quantitatively evaluating the models in term of accuracy, we can also examine whether the representations learned by the teacher and student are indeed similar. To do this, we choose the first six classes (class1: drink water, class2: eat meal, class3: brush teeth, class4: brush hair, class5: drop, class6: pick up) in the NTU RGB+D dataset and visualize the TSNE-embeddings of the representations computed by SKD(PifPaf-ShflNet, ST-GCN, $k = 1$), LQS-S(PifPaf-ShflNet, ST-GCN, $k = 1$) and HQS-T(PifPaf-ResNet, ST-GCN, $k = 1$) for identical input samples. Specifically, we take the logits output as the representation. In the left of Figure 6, we use the darker shade of a color to represent SKD(PifPaf-ShflNet, ST-GCN, $k = 1$) TSNE-embeddings and a lighter shade of the same color to represent LQS-S(PifPaf-ShflNet, ST-GCN, $k = 1$) TSNE-embeddings. We observe that the distribution of the dark shades and lighter shades are obviously different, indicating different representations between SKD(PifPaf-ShflNet, ST-GCN, $k = 1$) and LQS-S(PifPaf-ShflNet, ST-GCN, $k = 1$). Meanwhile, in the right of Figure 6, we use the darker shade of a color to represent SKD(PifPaf-ShflNet, ST-GCN, $k = 1$) embeddings and a lighter shade of the same color to represent HQS-T(PifPaf-ResNet, ST-GCN, $k = 1$) embeddings. It can be seen that the darker shades and the lighter shades of the same color show very good spatial overlap, indicating that SKD(PifPaf-ShflNet, ST-GCN, $k = 1$) and HQS-T(PifPaf-ResNet, ST-GCN, $k = 1$) produce more similar representations than the first one. From these results, we can draw a conclusion that SKD brings the student network to learn better representations by mimicking teacher network's representation.

TABLE V

TOP1 ACCURACY OF APPLYING SKD AND SWA WITH STATE-OF-THE-ART GCN-BASED APPROACHES WITH RANDOM OCCLUSION ON THE CS BENCHMARK OF SYNTHETIC NTU 3D SKELETON DATASET

Model	Year	Occluded Probability	
		0.3	0.6
ST-GCN [18]	2018	76.58	75.00
SWA-ST-GCN	-	80.71	75.90
SKD(ST-GCN)	-	79.03	76.88
SKD*(ST-GCN)	-	**82.43**	**78.60**
AS-GCN [23]	2019	80.09	78.01
SWA-AS-GCN	-	83.69	81.08
SKD(AS-GCN)	-	81.97	79.66
SKD*(AS-GCN)	-	**85.30**	**82.91**
Shift-GCN [43]	2020	81.90	80.25
SWA-Shift-GCN	-	83.43	80.86
SKD(Shift-GCN)	-	83.62	81.52
SKD*(Shift-GCN)	-	**84.63**	**82.55**
MS-G3D [44]	2020	81.94	76.58
SWA-MS-G3D	-	84.35	82.23
SKD(MS-G3D)	-	82.83	78.52
SKD*(MS-G3D)	-	**85.54**	**83.26**

*: These methods use sliding window average to fill missing values.

Figure 7 shows the neural response magnitude of each node in the last layer of ST-GCN, from which we can observe that HQS-T and LQS-S pay too much attention to minority joints. It means that part of joints dominate the recognition in these methods. On the contrary, for SDK, almost all of the joints are activated, and the neural response magnitude of each node is similar to each other. It indicates that SKD explores discriminative features from all the joints. Meanwhile, we also zoom in the high response areas of HQS and the related areas of LQS. It can be observed that the differences in neural response magnitude between HQS and LQS are smaller by using SKD, verifying that our method successfully makes the model to mimic HQS-based features when processing LQS.

6) Ablation Studies: We intend to identify the effect of different loss functions and architectures on SKD in these

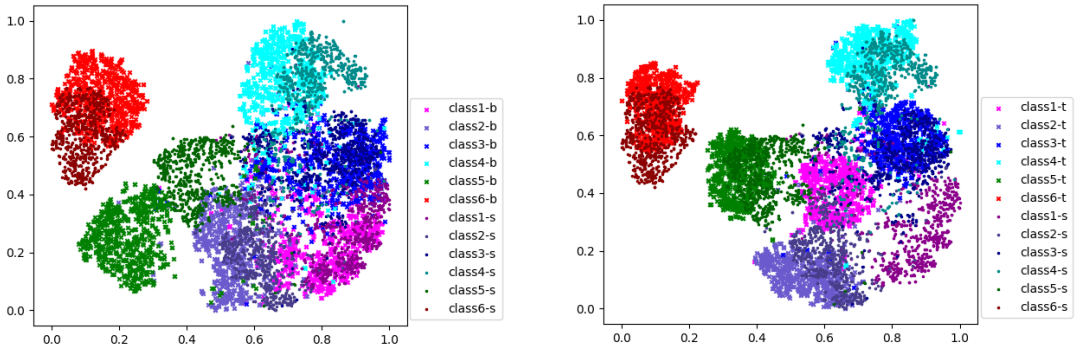


Fig. 6. (Left) TSNE-Embedding of SKD(PifPaf-ShflNet, ST-GCN, $k = 1$) and LQS-S(PifPaf-ShflNet, ST-GCN, $k = 1$) representations. (Right) TSNE-Embedding of SKD(PifPaf-ShflNet, ST-GCN, $k = 1$) and HQS-T(PifPaf-ResNet, ST-GCN, $k = 1$) representations. Post-fixes $-$t, $-$b and $-$s denote HQS-T(PifPaf-ResNet, ST-GCN, $k = 1$), LQS-S(PifPaf-ShflNet, ST-GCN, $k = 1$) and SKD(PifPaf-ShflNet, ST-GCN, $k = 1$) embedding, respectively.

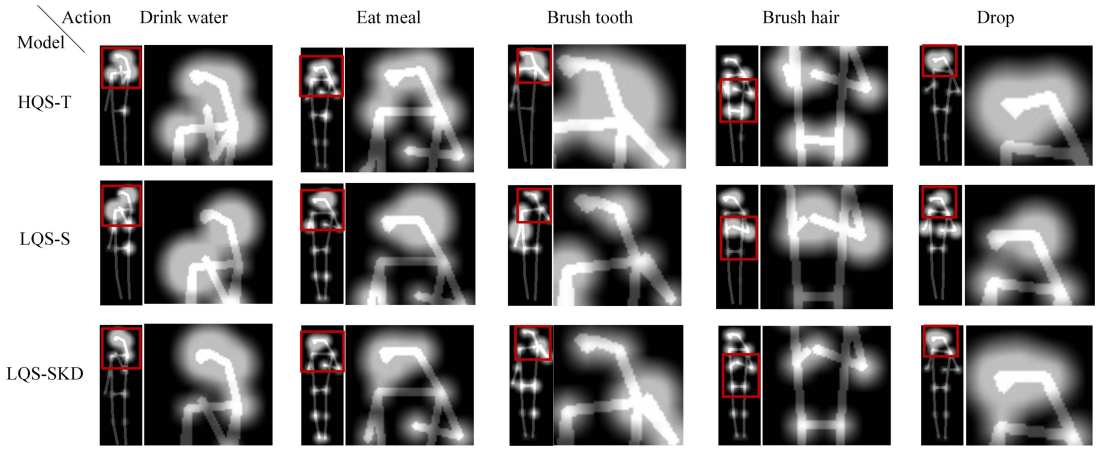


Fig. 7. Examples of the neural response magnitude of each joint in the last layer of ST-GCN for HQS-T(PifPaf-ResNet, ST-GCN, $k = 1$)(top), LQS-S(PifPaf-ShflNet, ST-GCN, $k = 1$)(middle) and SKD(PifPaf-ShflNet, ST-GCN, $k = 1$)(bottom). Meanwhile, the high response areas of HQS and the related areas of LQS are zoomed in. The larger the white area at a joint, the higher its response magnitude. From left to right are skeletons from five different actions and each action provides skeletons of different quality.

TABLE VI

QUANTITATIVE RESULTS USING DIFFERENT COMBINATIONS OF KNOWLEDGE-DISTILLATION LOSS FUNCTIONS ON NTU RGB+D ACTION DATASET

Model	Loss Function	Accuracy			
		CS		CV	
		Top-1	Top-5	Top-1	Top-5
HQS-T(PifPaf-ResNet, ST-GCN, k=1)	$\mathcal{L}_{\mathrm{T}}$	82.03	98.20	88.99	99.46
LQS-S(PifPaf-ShflNet, ST-GCN, k=1)	$\mathcal{L}_{\mathrm{T}}$	76.58	96.70	81.12	98.19
SKD(PifPaf-ShflNet, ST-GCN, k=1)	$\mathcal{L}_{\mathrm{T}} + \mathcal{L}_{\mathrm{KL}}$	77.61	96.88	81.52	98.36
SKD(PifPaf-ShflNet, ST-GCN, k=1)	$\mathcal{L}_{\mathrm{T}} + \mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{node}}$	78.56	97.12	83.82	98.74
SKD(PifPaf-ShflNet, ST-GCN, k=1)	$\mathcal{L}_{\mathrm{T}} + \mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{edge}}$	78.32	97.04	83.44	98.56
SKD(PifPaf-ShflNet, ST-GCN, k=1)	$\mathcal{L}_{\mathrm{T}} + \mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{GM}}$	**80.80**	**97.56**	**84.42**	**98.80**

ablation studies. The experiments are carried on the NTU RGB+D dataset in the context of using low-cost pose estimation model.

Firstly, we focus on exploiting how loss functions can affect the performance of SKD. In this paper, we consider a task-specific loss $\mathcal{L}_{\mathrm{T}}$ and three knowledge-distillation losses $\mathcal{L}_{\mathrm{KL}}, \mathcal{L}_{\mathrm{node}}, \mathcal{L}_{\mathrm{edge}}$. We train SKD(PifPaf-ShflNet, ST-GCN, $k = 1$) with different combinations of these loss functions. Specifically, each combination contains $\mathcal{L}_{\mathrm{T}}$ and one, two, or all three knowledge distillation loss functions. And a total of 4 groups are constructed. The results are summarized in Table VI, from which we can see that regardless of the loss functions' form, knowledge distillation plays a positive role in all the experiments. This is accordance with the original purpose of distillation loss by giving the student more comprehensive supervision. Meanwhile, it can also be observed that using the two intermediate-representation distillation losses, $\mathcal{L}_{\mathrm{node}}$ and $\mathcal{L}_{\mathrm{edge}}$, together can achieve better performance than using them separately, which proves that unary and pairwise structural knowledge would better be considered at the same time in GCN-based distillation.

TABLE VII
QUANTITATIVE RESULTS WITH DIFFERENT NETWORK ARCHITECTURES ON NTU RGB+D ACTION DATASET

Model	Architecture	Accuracy			
		CS		CV	
		Top-1	Top-5	Top-1	Top-5
HQS-T(PifPaf-ResNet, ST-GCN, k=1)	-	82.03	98.20	88.99	99.46
LQS-S(PifPaf-ShflNet, ST-GCN, k=1)	-	76.58	96.70	81.12	98.19
SKD(PifPaf-ShflNet, ST-GCN, k=1)	w/o Adaption + w/o GR	77.73	96.91	83.06	98.61
SKD(PifPaf-ShflNet, ST-GCN, k=1)	w Adaption + w/o GR	80.07	97.36	83.36	98.66
SKD(PifPaf-ShflNet, ST-GCN, k=1)	w Adaption + w GR	**80.80**	**97.56**	**84.42**	**98.80**

TABLE VIII
$\mathcal{L}_{\text{node}}$ AND $\mathcal{L}_{\text{edge}}$ OF SKD(PIFPAF-SHFLNET, ST-GCN, $k = 1$) WITH DIFFERENT NETWORK ARCHITECTURES ON NTU RGB+D ACTION DATASET

	CS		CV	
	w/o Adaption + w/o GR	w Adaption + w/o GR	w/o Adaption + w/o GR	w Adaption + w/o GR
$\mathcal{L}_{\text{node}}$	1.0223	0.9133	1.1077	1.1068
$\mathcal{L}_{\text{edge}}$	0.0395	0.0334	0.0610	0.0600

We also conduct an ablation study to show whether the adaption layer in student network is necessary to match the scale of the teacher. We show the effect of adaption layer in Table VII, where the student's performance with and without adaption layer are compared. Comparing the results of the first two SKD methods, we can see that the Top-1 accuracy with the adaption layer is 2.34% CS and 0.3% CV higher than that without the adaption layer. As shown in VIII, the adaptation layer can reduce $\mathcal{L}_{\text{node}}$ and $\mathcal{L}_{\text{edge}}$ by 10.66% and 15.44% respectively for CS settings while only by 0.08% and 1.64% for CV settings. It is consistent with the performance in VII, where the adaptation layer brings obvious gain for CS but marginal gain for CV. We can also find from Table VIII that $\mathcal{L}_{\text{node}}$ and $\mathcal{L}_{\text{edge}}$ for CV are larger than those for CS under the same conditions. It means that the intermediate representation scale gap in CV is larger than that in CS and the former brings greater challenge to the adaptation layer, which is one possible reason that results in varying improvements for these two benchmarks. The experiment indicates that adaption layer between the teacher network and the student network in SKD is indeed necessary but still with limited ability.

The last ablation study is on the usefulness of the GR. The experiment results are also shown in Table VII. Here, we can see that the model's Top-1 accuracy with GR is 80.80% CS and 84.42% CV while the one without GR is 80.07% CS and 83.36% CV. Therefore, the GR indeed further improves the performance of the proposed SDK scheme.

V. CONCLUSION

In this paper, we introduced a structural knowledge-distillation scheme for efficient skeleton-based action recognition. In this scheme, we exploited the potential of high-quality skeletons in the training stage and made the trained model to better handle low-quality skeletons. Specifically, we start with the training of a deep network as a teacher network that takes high-quality skeletons as input. Then, we freeze the teacher network and train a student network that takes the low-quality skeletons as input. The student network is expected to mimic the probability distribution and intermediate representations of the teacher network, along with minimizing a cross entropy loss for action recognition. In consideration of the graph structured intermediate representation in GCN, a graph matching loss is proposed to distill both the unary and the pairwise structural knowledge. Meanwhile, a gradient revision strategy is presented to deal with the conflict between mimicking teacher model and directly improving the student's accuracy. In experiments, an extensive evaluation was conducted under four scenarios of paired high-quality and low-quality skeletons. On the Kinetics400, NTU RGB+D and Penn datasets, our experiments showed that the proposed method clearly enhances model's robustness to uncontrollable skeleton corruptions and boosts the action-recognition performance on low-cost and low-quality skeletons.

APPENDIX A

We now show the derivation details from Eq. (13) to Eq. (14). First, Eq. (13) can be simplified to

$$\text{argmin}_{\tilde{\mathbf{g}}} \frac{1}{2}\tilde{\mathbf{g}}^\top\tilde{\mathbf{g}} - \mathbf{g}_{\text{KD}}^\top\tilde{\mathbf{g}} \quad \text{s.t.} \quad \tilde{\mathbf{g}}^\top\mathbf{g}_{\text{T}} \geq 0.$$

Then, the Lagrangian of this constrained optimization problem becomes:

$$\mathcal{L}(\tilde{\mathbf{g}}, \alpha) = \frac{1}{2}\tilde{\mathbf{g}}^\top\tilde{\mathbf{g}} - \mathbf{g}_{\text{KD}}^\top\tilde{\mathbf{g}} - \alpha\tilde{\mathbf{g}}^\top\mathbf{g}_{\text{T}}\,.$$

We can find the optimum $\tilde{\mathbf{g}}^*$ that minimizes $\mathcal{L}(\tilde{\mathbf{g}}, \alpha)$ by setting $\nabla_{\tilde{\mathbf{g}}}\mathcal{L}(\tilde{\mathbf{g}}, \alpha) = 0$. So, we have

$$\tilde{\mathbf{g}}^* = \mathbf{g}_{\text{KD}} + \alpha\mathbf{g}_{\text{T}}\,.$$

The dual of this optimization is $\mathcal{F}(\alpha) = \min_{\tilde{\mathbf{g}}} \mathcal{L}(\tilde{\mathbf{g}}, \alpha)$. By taking $\tilde{\mathbf{g}}^*$ into $\mathcal{L}(\tilde{\mathbf{g}}, \alpha)$, we have

$$\begin{aligned}\mathcal{F}(\alpha) &= \frac{1}{2}(\mathbf{g}_{\mathrm{KD}}^\top \mathbf{g}_{\mathrm{KD}} + 2\alpha \mathbf{g}_{\mathrm{KD}}^\top \mathbf{g}_{\mathrm{T}} + \alpha^2 \mathbf{g}_{\mathrm{T}}^\top \mathbf{g}_{\mathrm{T}}) - \mathbf{g}_{\mathrm{KD}}^\top \mathbf{g}_{\mathrm{KD}} \\ &\quad - 2\alpha \mathbf{g}_{\mathrm{KD}}^\top \mathbf{g}_{\mathrm{T}} - \alpha^2 \mathbf{g}_{\mathrm{T}}^\top \mathbf{g}_{\mathrm{T}} \\ &= -\frac{1}{2}\mathbf{g}_{\mathrm{KD}}^\top \mathbf{g}_{\mathrm{KD}} - \alpha \mathbf{g}_{\mathrm{KD}}^\top \mathbf{g}_{\mathrm{T}} - \frac{1}{2}\alpha^2 \mathbf{g}_{\mathrm{T}}^\top \mathbf{g}_{\mathrm{T}}\end{aligned}$$

By setting $\nabla_\alpha \mathcal{F}(\alpha) = 0$, we have the solution α^* to the dual:

$$\alpha^* = -\frac{\mathbf{g}_{\mathrm{KD}}^\top \mathbf{g}_{\mathrm{T}}}{\mathbf{g}_{\mathrm{T}}^\top \mathbf{g}_{\mathrm{T}}}.$$

Based on this, the gradient update rule is:

$$\tilde{\mathbf{g}}^* = \mathbf{g}_{\mathrm{KD}} - \frac{\mathbf{g}_{\mathrm{KD}}^\top \mathbf{g}_{\mathrm{T}}}{\mathbf{g}_{\mathrm{T}}^\top \mathbf{g}_{\mathrm{T}}} \mathbf{g}_{\mathrm{T}}\,.$$

REFERENCES

[1] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image Vis. Comput.*, vol. 60, pp. 4–21, Apr. 2017.

[2] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "RGB-D-based human motion recognition with deep learning: A survey," *Comput. Vis. Image Understand.*, vol. 171, pp. 118–139, Jun. 2018.

[3] S. Bhardwaj, M. Srinivasan, and M. M. Khapra, "Efficient video classification using fewer frames," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 354–363.

[4] J.-F. Hu, W.-S. Zheng, J. Pan, J. Lai, and J. Zhang, "Deep bilinear learning for RGB-D action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 335–351.

[5] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 1227–1236.

[6] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception Psychophys.*, vol. 14, no. 2, pp. 201–211, Jun. 1973.

[7] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1010–1019.

[8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7291–7299.

[9] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3D human pose and shape from a single color image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 459–468.

[10] N. C. Garcia, P. Morerio, and V. Murino, "Modality distillation with multiple stream networks for action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 103–118.

[11] W. Feng, Z.-Q. Liu, L. Wan, C.-M. Pun, and J. Jiang, "A spectral-multiplicity-tolerant approach to robust graph matching," *Pattern Recognit.*, vol. 46, no. 10, pp. 2819–2829, Oct. 2013.

[12] W. Feng and Z.-Q. Liu, "Region-level image authentication using Bayesian structural content abstraction," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2413–2424, Dec. 2008.

[13] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 588–595.

[14] B. Fernando, E. Gavves, M. J. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5378–5387.

[15] W. Zhu *et al.*, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3697–3704.

[16] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3288–3297.

[17] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2014–2023.

[18] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 7444–7452.

[19] W. Feng, J. Jia, and Z.-Q. Liu, "Self-validated labeling of Markov random fields for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1871–1887, Oct. 2010.

[20] C. Rother, P. Kohli, W. Feng, and J. Jia, "Minimizing sparse higher order energy functions of discrete variables," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1382–1389.

[21] W. Feng, J. Jia, and Z.-Q. Liu, "ESSP: An efficient approach to minimizing dense and nonsubmodular energy functions," 2014, *arXiv:1405.4583*. [Online]. Available: http://arxiv.org/abs/1405.4583

[22] 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 Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3595–3603.

[23] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 12026–12035.

[24] Y. Song, Z. Zhang, S. Caifeng, and W. Liang, "Richly activated graph convolutional network for robust skeleton-based action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Aug. 7, 2020, doi: 10.1109/TCSVT.2020.3015051.

[25] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531

[26] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*. [Online]. Available: http://arxiv.org/abs/1412.6550

[27] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," 2016, *arXiv:1612.03928*. [Online]. Available: http://arxiv.org/abs/1612.03928

[28] Y. Liu *et al.*, "Knowledge distillation via instance relationship graph," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 7096–7104.

[29] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," 2019, *arXiv:1907.09682*. [Online]. Available: http://arxiv.org/abs/1907.09682

[30] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik, "Unifying distillation and privileged information," 2015, *arXiv:1511.03643*. [Online]. Available: http://arxiv.org/abs/1511.03643

[31] S. Ge, S. Zhao, C. Li, and J. Li, "Low-resolution face recognition in the wild via selective knowledge distillation," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 2051–2062, Apr. 2019.

[32] N. Karessli, R. Guigourès, and R. Shirvany, "SizeNet: Weakly supervised learning of visual size and fit in fashion images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 335–343.

[33] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, "MARS: Motion-augmented RGB stream for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 7882–7891.

[34] G. Papandreou *et al.*, "Towards accurate multi-person pose estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4903–4911.

[35] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 483–499.

[36] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," 2019, *arXiv:1902.09212*. [Online]. Available: http://arxiv.org/abs/1902.09212

[37] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2277–2287.

[38] S. Kreiss, L. Bertoni, and A. Alahi, "PifPaf: Composite fields for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 11977–11986.

[39] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6299–6308.

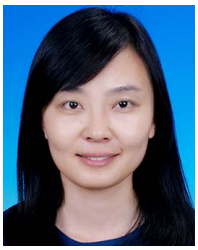
[40] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2248–2255.

[41] 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.

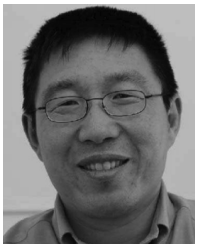
[42] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, Oct. 2020.
[43] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 183–192.
[44] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 143–152.

Cunling Bian received the B.S. degree in educational technology and the M.S. degree in educational information technology from the Ocean University of China, China, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree with the College of Intelligence and Computing, Tianjin University, China. His major research interests include visual intelligence, specifically including skeleton-based action recognition and multi-camera action analysis. He is also interested in solving preventive conservation problems of cultural heritages via artificial intelligence.

Wei Feng (Member, IEEE) received the Ph.D. degree in computer science from the City University of Hong Kong, in 2008. From 2008 to 2010, he was a Research Fellow with the Chinese University of Hong Kong and the City University of Hong Kong. He is currently a Professor with the School of Computer Science and Technology, College of Computing and Intelligence, Tianjin University, China. His major research interests include active robotic vision and visual intelligence, specifically including active camera relocalization and lighting recurrence, general Markov Random Fields modeling, energy minimization, active 3D scene perception, SLAM, and generic pattern recognition. Recently, he focuses on solving preventive conservation problems of cultural heritages via computer vision and machine learning. He is the Associate Editor of *Neurocomputing* and *Journal of Ambient Intelligence and Humanized Computing*.



Liang Wan (Member, IEEE) received the B.Eng. and M.Eng. degrees in computer science and engineering from Northwestern Polytechnical University, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, in 2007. She is currently a Full Professor with the College of Intelligence Computing, Tianjin University, China. Her main research interests include image processing, computer vision, and graphics, including image segmentation, gesture recognition, and medical image analysis.



Song Wang (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at UrbanaChampaign (UIUC), Champaign, IL, USA, in 2002. He was a Research Assistant with the Image Formation and Processing Group, Beckman Institute, UIUC, from 1998 to 2002. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, where he is currently a Professor. His current research interests include computer vision, image processing, and machine learning. He is currently serving as the Publicity/Web Portal Chair for the Technical Committee of Pattern Analysis and Machine Intelligence of the IEEE Computer Society, an Associate Editor for IEEE TRANSACTION ON PATTERN ANALYSIS and *Machine Intelligence*, *Pattern Recognition Letters*, and *Electronics Letters*. He is a member of the IEEE Computer Society.