Hierarchical Graph Convolutional Networks for Action Quality Assessment

Kanglei Zhou, Yue Ma, Hubert P. H. Shum, *Senior Member, IEEE*, and Xiaohui Liang

***Abstract*—Action quality assessment (AQA) automatically evaluates how well humans perform actions in a given video, a technique widely used in fields such as rehabilitation medicine, athletic competitions, and specific skills assessment. However, existing works that uniformly divide the video sequence into small clips of equal length suffer from intra-clip confusion and inter-clip incoherence, hindering the further development of AQA. To address this issue, we propose a hierarchical graph convolutional network (GCN). First, semantic information confusion is corrected through clip refinement, generating the 'shot' as the basic action unit. We then construct a scene graph by combining several consecutive shots into meaningful scenes to capture local dynamics. These scenes can be viewed as different procedures of a given action, providing valuable assessment cues. The video-level representation is finally extracted via sequential action aggregation among scenes to regress the predicted score distribution, enhancing discriminative features and improving assessment performance. Experiments on the AQA-7, MTL-AQA, and JIGSAWS datasets demonstrate the superiority of the proposed hierarchical GCN over state-of-the-art methods.**

***Index Terms*—Action quality assessment, Graph convolutional neural networks, Human action understanding.**

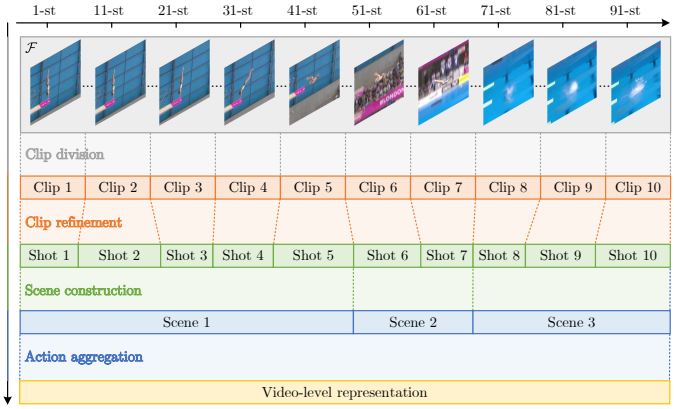

Fig. 1. The main idea of the proposed hierarchical method. The first goal is to correct the semantic information confusion of initial clips in order to obtain the shots. Secondly, meaningful scenes are obtained through the shot reduction process. The final enhanced video-level representation is extracted by using the action aggregation operation.

I. Introduction

As an important extension of human action recognition [1], [2], automated vision-based action quality assessment (AQA) from a given action instance can be used as an alternative to avoid personal judgment bias [3]. The goal of AQA is to quantify *how well* actions are performed [4] from the same class, as opposed to action recognition [5], [6] that is to identify *what* actions are performed in a given action from different classes. Since the difference among intra-class samples is always subtle, AQA is considered to be a challenging problem [7]. In recent years, AQA has gained increasingly widespread attention due to its wide range of real-world applications such as rehabilitation medicine [8], [9], [10], athletic competition [6], [11], [12], and specific skills assessment [13], [14].

Existing AQA approaches mainly focus on sports analysis, which can be divided into pose-based methods and vision-based methods according to the difference of the input modality. Early methods [4] mainly depend on pose-based features to regress quality scores. However, performing pose estimation in the sports domain is difficult [15]. The estimated poses are usually inaccurate with missing parts due to the crouching and occlusion of the athlete's body, as shown in previous works [4], [15]. Such errors at any moment may affect the final assessment performance. Most importantly, pose-only features do not take into account visual cues like splashes, which are crucial to judging. The state-of-the-art research focuses on vision-based methods [11], [12], [16], [17] as they can make full use of the visual features provided by image sequences and have achieved great success in recent years.

A general pipeline of vision-based AQA frameworks includes three phases: feature extraction, aggregation, and score regression. Existing AQA datasets [18], [19], [20] are mainly collected in sports competitions such as the Olympic games. They are usually quite small with a couple of hundred samples, *e.g.*, there are only 176 gymnastic vault samples in the AQA-7 dataset [18]. To avoid over-fitting due to training from small datasets, most existing methods adopt powerful backbone networks such as C3D [21] and I3D [6] as the feature extractor, which are usually pre-trained on large action recognition datasets and it is hypothesized that action recognition features can transfer well to the AQA task. These backbones with the structure of 3D convolutional neural networks (CNNs) are very memory and computationally intensive, which limits their usage to only small-sized clips. Additionally, the whole action sequence at every moment provides vital clues for AQA that the athlete can make or lose points [19]. Therefore, extracting

Manuscript created October, 2020; This work was supported by the National Natural Science Foundation of China (Project Number: 62272019). (Corresponding author: Xiaohui Liang)

Kanglei Zhou and Yue Ma are with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China (e-mail: {zhoukanglei, super_mayue}@buaa.edu.cn).

Hubert P. H. Shum is with the Department of Computer Science, Durham University, Durham DH1 3LE, UK (e-mail: hubert.shum@durham.ac.uk).

Xiaohui Liang is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China, and also with Zhongguancun Laboratory, Beijing, China (e-mail: liang_xiaohui@buaa.edu.cn).

keyframes is an effective approach for action recognition, but it is not feasible for AQA.

To trade off sequence length and visual cues, most works [11], [12], [7] uniformly divide the entire video sequence into small clips of equal length (usually 16 frames long), compute spatial-temporal features independently for each clip, and aggregate these clip-level features to generate the video-level representation on average. However, such clip division and aggregation strategies suffer from the intra-clip confusion and inter-clip incoherence problems, which greatly limits their performance. On the one hand, the uniform division causes incompleteness or redundancy in each clip of motion-related semantic information. That is, a clip may lack information about motion information or may also contain information belonging to other groups. We dub this situation as intra-clip confusion. Exact distribution of motion information in different clips is closely related to scoring for local action details. Hence, eliminating the information confusion problem is key for accurate assessment. On the other hand, a meaningful scoring procedure may span several motion units and a single unit is insufficient to observe the global dynamics of actions. We dub this case as inter-clip incoherence. Inaccurate organization of the motion information contained in the scene will lead to score errors in an action procedure and affect the final scoring. Therefore, it is crucial to explore the hierarchical relationship between the action procedure and the motion unit.

At the same time, existing aggregation methods such as average pooling [22] and long short-term memory (LSTM) [19] are impossible to explicitly explore the local and global dynamics of actions. To achieve a reasonable clip division, a straightforward solution [23], [7] is to perform the boundary detection [24] *w.r.t.* the corresponding basic motion units before training. However, excessive or incorrect divisions may lead to the loss of important temporal information that restricts AQA performance. In addition, manual labels of boundaries are expensive to obtain and are not included in the majority of existing datasets. Therefore, mining the potential relationships between different clips is an important compromise. Inspired by the structured video analysis [25], we define a basic motion unit as a shot, containing several frames. Several consecutive shots constitute a meaningful scene and a kind of action consists of different scenes. Based on such a hierarchical structure, we propose an end-to-end hierarchical graph convolutional network (GCN) to address the intra-clip confusion and inter-clip incoherence problems.

As both uniform clip division and dynamic boundary detection would lead to information confusion, we adopt an end-to-end manner to explore hierarchical relationships for accessing the quality of actions. As shown in Figure 1, we first propose the clip refinement module to correct the semantic information confusion of each clip. To solve the inter-clip incoherence problem, a scene graph is then constructed by combining several consecutive clips into a meaningful scene. Next, the video-level representation is extracted by action aggregation. Finally, considering that the action quality score given by multiple judges is often uncertain and the label distribution learning can fit AQA datasets better than the score regression [11], [16], we treat AQA as the score distribution regression and directly map the video-level representation to the Gaussian score distribution. The final score is sampled from such a score distribution. We have conducted extensive experiments on three AQA datasets, including AQA-7 [18], MTL-AQA [19], and JIGSAWS [20]. The experimental results demonstrate that the proposed method outperforms the state-of-the-art.

The source code of this research is available at https://github.com/ZhouKanglei/HGCN_AQA. The main contributions of our work are summarized as follows:

1) Addressing the intra-clip confusion problem, we propose a simple yet effective clip refinement module to correct the semantic information confusion of each clip.
2) Addressing the inter-clip incoherence problem, we construct a scene graph to capture the local dynamics of actions by combining several consecutive shots into a scene and propose an action aggregation operation to obtain the video-level representation.
3) The proposed hierarchical GCN can be used as a plug-and-play module for other methods. Extensive experiments on three AQA datasets demonstrate our method outperforms the state-of-the-art.

The rest of the whole paper is organized as follows: Section II briefly reviews the related work, Section III details the core components of the proposed method, Section IV shows quantitative and qualitative experiments, and Section V concludes the whole paper.

II. Related Work

This section first introduces some classical frameworks for AQA, and then reviews the video representation learning methods and structured video analysis technology, respectively.

A. Action Quality Assessment

AQA can date back to 1995 by Gordon [26], aiming at producing scores or ranks as output based on the analysis of a given input video instance. We divide existing AQA methods into pose-based methods and vision-based methods according to different input types.

Early pose-based AQA frameworks [26], [4] consist of three stages: location tracking, feature extraction, and score prediction. Through tracking body parts such as hands, feet, and waist, features such as position, speed, and direction can be extracted. The action quality score or grade is finally calculated by manually designed rules or machine learning methods. For example, Pirsiavash *et al.* [4] have first estimated poses of athletes and then encoded them using discrete cosine transform. Finally, a support vector regression model is used to map these pose-based features to action quality scores. Due to illumination, view changes, occlusion, *etc*, the obtained pose parameters are not accurate enough [15], which severely affects the action assessment performance and limits the spread of applications. With the rapid development of computer vision, complex actions can be efficiently modeled and evaluated in detail. Similar to early frameworks, we also break vision-based ones into three phases: feature extraction, aggregation, and score regression.

In terms of the input format, these frameworks include exemplar-based [3], [12], [7] and exemplar-free [4], [11], [15] methods depending on whether exemplars are used or not. The former usually involves selecting a set of reference exemplars along with the target example as input, while the latter does not. For example, Yu *et al.* [12] have proposed a novel exemplar-based method for AQA using group-aware contrastive regression. This method can improve the performance to a certain extent by regressing relative action quality scores. However, manual exemplars selection brings personal bias and the feature learning of exemplars requires an extra computational burden. To avoid such a problem, the proposed method falls into the latter.

In terms of the output format [27], there are quality score regression [11], grading [28], [3], and pairwise-sorting [29], [13], [30] methods. For the quality score regression methods, a specific score of an action is predicted, while the grading ones divide the action quality into different levels. For example, Parmar *et al.* [28] classify the levels of cerebral palsy rehabilitation exercises as *good* or *bad*. The pairwise sorting methods take any two videos to evaluate the action quality. For example, Doughty *et al.* [14] have trained temporal attention modules using a novel rank-aware loss function. This work outputs quality scores and adopts rank coefficients for evaluation.

B. Video Representation Learning

The human action analysis is closely related to video representation learning, which provides spatial-temporal features for downstream tasks, such as regression and classification. We mainly review early and deep learning-based methods for video representation.

Early methods describe the video by extracting the feature of key points, represented by space-time interest points [31], dense trajectories [32], [33], *etc*. These descriptors are then aggregated into the video-level representation using encoding methods like BoW and Fisher vector. These methods cannot fully extract discriminative representative features from the video, so it is difficult to capture subtle differences between actions from the same class for AQA. This is one of the main reasons that AQA is slow to develop in the early phase.

Deep learning-based methods have achieved better performance in video representation learning than the early ones. There are clip-level [34], [35] and video-level [36] representation forms based on deep learning-based methods. One of the most common and effective spatial-temporal feature extractors is 3D CNNs. However, due to their memory and computationally intensive nature, current 3D CNNs, such as C3D [21] and I3D [6], are not suitable for processing long videos including over 100 frames. Different from action recognition which can be performed by seeing as little evidence as some key video frames, accessing the quality of action requires processing a full action sequence [19]. Therefore, almost all AQA approaches have built clip-level features instead of video-level representation. For example, to extract spatial-temporal features, Pan *et al.* [37] have uniformly sampled 16 frames in each sequence as the input to the I3D network. This uniform sampling is first presented in [38] for video-level representation learning. However, such a trade-off strategy suffers from the intra-clip confusion problem. In this work, we devise a novel clip refinement module to handle this problem.

C. Structured Video Analysis

Since the raw video is the unstructured data stream, efficient analysis and access is not easy task. The structured video analysis [39] describes videos with a hierarchical structure, which provides convenience for content-based video processing [25].

The hierarchical video structure consists of shots, groups, and scenes. A shot contains several consecutive frames captured once by the camera. Similar shots can be grouped into one group. Semantically related shots can be merged into one meaningful scene, which depicts and conveys a high-level concept. Generally, shots can be divided from the whole video sequence by shot boundary detection methods [40], [24]; major visual content of shots can be represented by key-frames [41]. Previous work [24] has shown that high-level features are powerful and suitable to distinguish transitions. In this work, we regard the shot as the basic motion unit and different shots are complete and dependent, such as the high-level semantic groups of jumping and running, composed of fine-grained motion primitives. The scene represents a meaningful procedure and a scene may include several shots, such as that of take-off, flight, *etc*. Based on the hierarchical structure of the video, Wang *et al.* [42] have developed a method for automatically dividing complex activities into sub-activities within a specific video, achieving excellent performance in the classification of complex activities. Different from this latent hierarchical model using SVM for action classification, we design a deep hierarchical model using GCNs for AQA based on the video hierarchy.

III. The Hierarchical Graph Convolutional Network

In this section, we start with the clip feature extraction in Section III-A. Then to handle the intra-clip confusion and inter-clip incoherence issues, we introduce our hierarchical GCN with three basic modules to obtain the enhanced video-level representation: (a) clip refinement, (b) scene construction, and (c) action aggregation. The clip refinement module (Section III-B) aims to correct the semantic information confusion caused by the uniform clip division before training; the scene construction module (Section III-C) aims to obtain the refined clips and construct the scene graph by combining several consecutive clips into one scene; the action aggregation module (Section III-D) aims to obtain the video-level representation. In Section III-E and Section III-F, we present the score distribution regression module and the loss function respectively. The overview of the proposed framework is shown in Figure 2. Unless otherwise specified in this paper, the normal bold $\mathbf{symbol}$ indicates a matrix or tensor, the italic bold $\boldsymbol{symbol}$ indicates a vector, the italic-only $symbol$ indicates a variable, and the normal symbol indicates a constant.

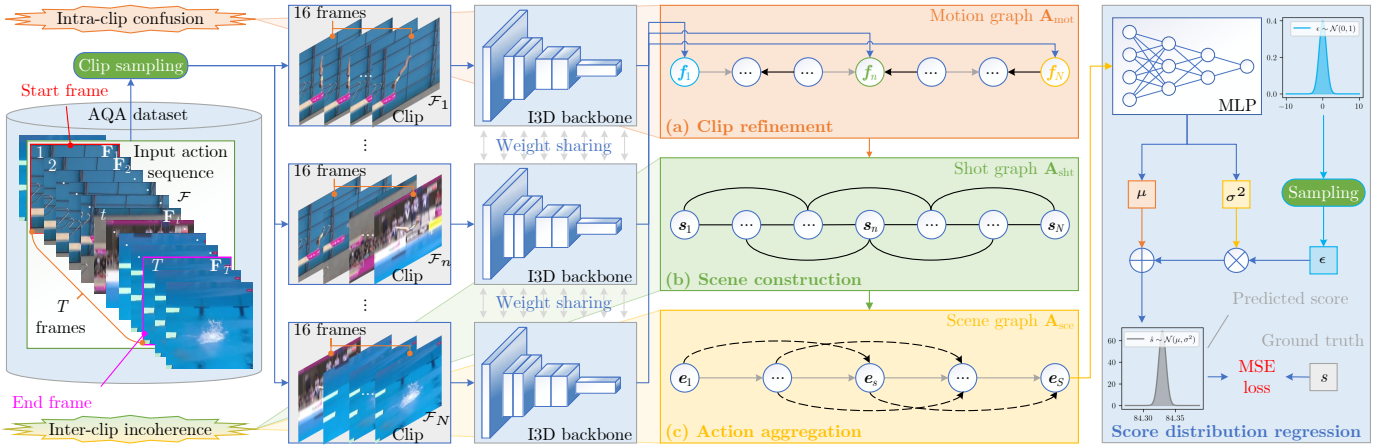

Fig. 2. The framework of the proposed hierarchical GCN for AQA: an input video sample $\mathcal{F}$ with T frames from the AQA dataset is first taken. Then, $\mathcal{F}$ is divided into N clips $\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_N$ by the uniform clip division strategy. Through a weight-sharing I3D backbone network [6], $\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_N$ are encoded into high-level clip features $\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_N$. To address the intra-clip confusion problem, $\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_N$ are refined to obtain the shot features $\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_S$ by (a) clip refinement. To address the inter-clip incoherence problem, the (b) scene construction module is performed by reducing several shots into one scene. Next, the (c) action aggregation module is used to aggregate the video-level representation. Finally, the final score $\hat{s}$ is predicted by the score distribution regression network.

A. Clip Feature Extraction

As can be seen in Figure 2, the input of the system is a video from an AQA dataset. For a video with T frames, we denote it as $\mathcal{F} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_T\}$ and its t-th frame of size $W \times H$ can be represented as a tensor $\mathbf{X}_t \in \mathbb{R}^{W \times H \times 3}$. We divide the whole video sequence $\mathcal{F}$ into N clips spanning 16 frames each, which are denoted as $\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_N$, in accordance with the majority of earlier efforts [11], [12], [38]. Through a weight-sharing I3D backbone network [6], these clips $\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_N$ are separately encoded into the high-level clip features $\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_N \in \mathbb{R}^{C_1}$ where C_1 represents the feature dimension. The encoded clip features are combined to the matrix $\mathbf{F} \in \mathbb{R}^{N \times C_1}$.

B. Clip Refinement

The clip refinement module aims to correct the semantics information confusion of the clip features $\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_N$, *i.e.*, moving redundant information from a clip to the corresponding incomplete clip. There are two possible scenarios:

- The semantic information contained by the i-th clip is insufficient to represent one motion, and the necessary part must be acquired from the ending of its precursors or the beginning of its successors;
- The semantic information contained by the i-th clip belongs to more than one motion, and the unnecessary part must be sent to the ending of its precursors or the beginning of its successors.

1) Analysis: We argue that the feature of one action can be represented as the combination of a group of orthogonal motion primitives $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_B \in \mathbb{R}^{D_1}$ where D_1 is the embedding dimension. In this work, shots are defined as dependent and complete individuals in terms of semantic information, containing only one complete motion primitive. By contrast, different clips may share motion primitives due to semantic information confusion. To eliminate the intra-clip information confusion problem, we first detect transitions by motion decomposition and then transfer information from one shot with redundant primitives to another shot with insufficient primitives. Our method implicitly explores the relationships between different shots, which is different from the shot boundary detection [24] that requires finding boundaries between different shots. The network architecture of the clip refinement module is depicted in Figure 3, which consists of three steps: motion decomposition, motion graph construction, and information transfer. The following provides a detailed introduction to them.

2) Motion Decomposition: Based on the above analysis, we decompose the i-th clip feature $\boldsymbol{f}_i$ into the superposition of different motion primitives in a latent manifold space. This process can be logically represented as: $\boldsymbol{m}_i = \lambda_i^1 \boldsymbol{b}_1 + \lambda_i^2 \boldsymbol{b}_2 + \cdots + \lambda_i^B \boldsymbol{b}_B$, where $\boldsymbol{m}_i$ is the combination of motion primitives and $\lambda_i^1, \lambda_i^2, \cdots, \lambda_i^B$ are component coefficients. Because our goal is to indirectly explore the relationship between different clips with the help of motion decomposition, we do not need to explicitly obtain motion primitives. Instead, we can model it by neural networks easily, which can be implemented by:

$$\boldsymbol{m}_i = \mathrm{ReLU}\left(\mathrm{Conv1D}_{\text{group}}\left(\boldsymbol{f}_i, g\right)\right), \tag{1}$$

where $\mathrm{Conv1D}_{\text{group}}(\cdot)$ denotes the 1D convolution layer with groups of size g and kernels of size 1, and $\mathrm{ReLU}(\cdot)$ represents the rectified linear unit activation function. The ReLU activation function is chosen over logistic activation functions (*e.g.*, softmax) due to its robustness to vanishing gradients and its computational efficiency.

In this way, it is convenient to determine the direction and the magnitude of information transfer by comparing the corresponding primitives within any two clips.

3) Motion Graph Construction: To facilitate information transfer, we construct a motion graph as $\mathcal{G}_{\text{mot}} = (\mathcal{V}_{\text{mot}}, \mathcal{E}_{\text{mot}})$, including $|\mathcal{V}_{\text{mot}}|$ motion nodes and $|\mathcal{E}_{\text{mot}}|$ links. The motion graph $\mathcal{G}_{\text{mot}}$ is a direct graph and $\mathbf{A}_{\text{mot}} \in \mathbb{R}^{N \times N}$ is the corresponding adjacent matrix.

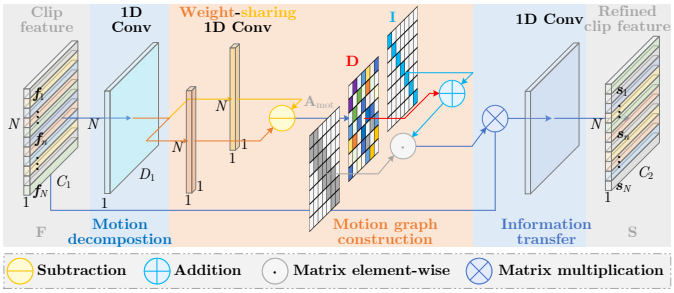

Fig. 3. The network architecture of the clip refinement module: the clip feature $\mathbf{F}$ is refined to the corresponding shot feature $\mathbf{S}$ through motion decomposition, motion graph construction, and information transfer.

As shown in Figure 2, one clip may contain information of several shots or lack information, thus the information transfer may occur in r-adjacent clips. In this way, the adjacent matrix $\mathbf{A}_{\text{mot}}$ can be represented as:

$$A_{\text{mot}}^{ij} = \begin{cases} 1, \text{ if } |i-j| \leq r \\ 0, \text{ otherwise} \end{cases}, \tag{2}$$

where A_{mot}^{ij} controls the connections between the clip features $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, and r indicates the minimum distance between two clips with neighboring relationship. In particular, r is set to 1 in Figure 1.

When there is information interference in a certain motion primitive of two clips, the information corresponding to such motion primitive needs to be transferred from one clip to another. Next, we need to identify *which* clip the information corresponding to the motion primitive belongs to. In other words, we need to determine the direction and the magnitude of information transfer. To this end, we learn a distance function d_{ij} to measure the distance from the i-th primitives combination to that of the j-th. Accordingly, the learnable distance d_{ij} should maintain both direction and magnitude.

In detail, the direction $\text{sign}(d_{ij})$ measures whether $\boldsymbol{f}_j$ contains the motion primitive that belongs to $\boldsymbol{f}_i$ and the magnitude $|d_{ij}|$ weighs the amount of transferred information. To measure the distance between the motion features $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$, we map $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$ simultaneously from the feature space to the metric space using a weight-sharing 1D convolution with kernel size 1 shown in Figure 3. Next, we calculate the distance direction and magnitude between $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$ by a non-linear transformation. Thus, the process can be represented as:

$$d_{ij} = \tanh\left(\left(\boldsymbol{m}_i - \boldsymbol{m}_j\right)\mathbf{W}_{\text{mot}}\right), \tag{3}$$

where $\mathbf{W}_{\text{mot}} \in \mathbb{R}^{D_1 \times 1}$ represents a linear transformation matrix and $\tanh(\cdot)$ denotes the hyperbolic tangent activation function.

The element d_{ij} ranges from -1 to 1, which controls both the direction and the magnitude of information transfer between adjacent clips.

- If $d_{ij} > 0$, the clip feature $\boldsymbol{f}_i$ need to receive $|d_{ij}|$ information from the clip feature $\boldsymbol{f}_j$. This case means that the transition boundary between the shot i and j locates in the i-th clip.
- If $d_{ij} = 0$, the clip feature $\boldsymbol{f}_i$ does not need to receive information from the clip feature $\boldsymbol{f}_j$. This case means that the transition boundary between the shot i and j is exact.
- If $d_{ij} < 0$, the clip feature $\boldsymbol{f}_j$ need to be removed $|d_{ij}|$ information of the clip feature $\boldsymbol{f}_j$. This case means that the transition boundary between the shot i and j locates in the j-th clip.

4) Information Transfer: After we get the information transfer relationship of different clips, we can correct the information confusion accordingly. In this work, we implement the information transfer process between two adjacent clip features $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ by a GCN layer:

$$\boldsymbol{s}_i = \text{ReLU}\left(\boldsymbol{f}_i\mathbf{W}_{\text{tra}} + \sum_{j\in\mathcal{N}_i} B_{\text{mot}}^{ij}\boldsymbol{f}_j\mathbf{U}_{\text{tra}}\right), \tag{4}$$

where $B_{\text{mot}}^{ij} = A_{\text{mot}}^{ij} \cdot d_{ij}$, $\mathbf{W}_{\text{tra}}, \mathbf{U}_{\text{tra}} \in \mathbb{R}^{C_1 \times C_2}$ are linear transformation matrices, $\mathcal{N}_i$ is neighbors of $\boldsymbol{f}_i$, $\boldsymbol{s}_i \in \mathbb{R}^{C_2}$ is the refined clip (shots) features, and C_2 is the dimension of shot features.

Through the information transfer process, the inter-clip confusion is corrected and we can obtain the corresponding shot features $\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_N$. These shots are dependent and complete in terms of semantic information, which represent the basic units of action procedures.

5) Sequential Property: Previous works [43], [44], [45] have shown that an encoder is an effective mapping from the raw data space to a motion manifold. As a spatial-temporal feature extractor, I3D can be seen as a manifold projection. However, separate clip encoding cannot preserve the sequential property. That is, the distance between adjacent clip features is small, while the distance between non-adjacent clips is large. Different from Laplace regularization methods [46], [47] that consider spatial or temporal closeness, solving this problem requires clip-level closeness:

$$\begin{aligned} O &= \min_{\boldsymbol{h}} \sum_{i=1}^{N}\sum_{j=1}^{N} A_{\text{mot}}^{ij}\|\boldsymbol{h}_i - \boldsymbol{h}_j\|_2^2 \\ &= \min_{\mathbf{H}} \text{tr}\left(\mathbf{H}\mathbf{L}\mathbf{H}^\top\right), \end{aligned} \tag{5}$$

where $\boldsymbol{h}_i \in \mathbb{R}^{C_2}$ represents the embedding of the clip feature $\boldsymbol{f}_i$, $\mathbf{H} \in \mathbb{R}^{N \times C_2}$ denotes the matrix format of $\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_N$, and $\mathbf{L} \in \mathbb{R}^{N \times N}$ indicates the Laplace matrix of $\mathbf{A}_{\text{mot}}$.

Theorem 1: When $\mathbf{H} = \mathbf{F}\mathbf{W}$, the propagation process of GCNs is equivalent to optimizing the above clip-level regularization in Equation (5).

Proof 1: Set derivative of the objective function in Equation (5) *w.r.t.* $\mathbf{H}$ to zero, and then we can obtain

$$\frac{\partial \text{tr}\left(\mathbf{H}\mathbf{L}\mathbf{H}^\top\right)}{\partial \mathbf{H}} = 0 \Rightarrow \mathbf{L}\mathbf{H} = 0 \Rightarrow \mathbf{H} = \hat{\mathbf{A}}\mathbf{H}. \tag{6}$$

where $\hat{\mathbf{A}}$ is the normalized adjacent matrix.

The above equation can be explained as a limit distribution [48], and we use the following iterative form to approximate the limit of $\mathbf{H}$ with $l \to \infty$:

$$\lim_{l\to\infty} \mathbf{H}^{(l)} = \hat{\mathbf{A}}\mathbf{H}^{(l-1)}. \tag{7}$$

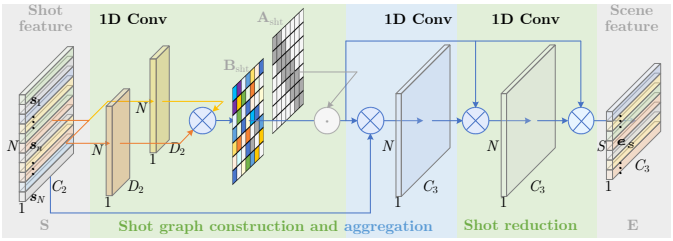

Fig. 4. The network architecture of the scene construction module: the shot feature $\mathbf{S}$ is transformed to $\mathbf{E}$ through the shot graph construction, aggregation, and the shot reduction.

When the initial representation $\mathbf{H}^{(0)}$ is initialized as $\mathbf{FW}$, we can obtain:

$$\mathbf{H}^{(l)} = \hat{\mathbf{A}}\mathbf{H}^{(l-1)} = \cdots = \hat{\mathbf{A}}^l\mathbf{H}^{(0)} = \hat{\mathbf{A}}^l\mathbf{X}\mathbf{W}, \tag{8}$$

which matches the matrix form of GCNs by ignoring the non-linear transformation.

In conclusion, Theorem 1 can be proved.

Since the information transfer in Equation (4) also adopts a GCN layer, the conclusion of Theorem 1 can hold for it. Therefore, the clip refinement can simultaneously preserve the sequential nature between different clips so that we do not need to add additional penalties.

C. Scene Construction

The scene construction module aims to combine several consecutive shots into a meaningful scene to address the inter-clip incoherence problem.

1) Analysis: A meaningful scene represents a crucial procedure and thus is key to evaluating the performance of local action details. For example, the *diving* action is usually filmed in a similar environment and all the videos contain the same set of action procedures, including *take-off*, *flight*, and *entry*. The subtle differences mainly appear in the numbers of both somersault and twist, flight positions as well as their executed qualities. To capture these subtle differences for AQA, it is vital to parse the procedures of actions and quantify the executed qualities of these procedures. Through the scene construction module, we can capture the full dynamics of each action procedure, which is beneficial to assess action details and is key to the final score. The network architecture of the scene construction module is depicted in Figure 4, which consists of three steps: shot graph construction, shot graph aggregation, and shot reduction. The following provides a detailed introduction to them.

2) Shot Graph Construction: To model shot relationships within the scene, we construct a shot graph as $\mathcal{G}_{\text{sht}} = (\mathcal{V}_{\text{sht}}, \mathcal{E}_{\text{sht}})$, including $|\mathcal{V}_{\text{sht}}|$ shot nodes and $|\mathcal{E}_{\text{sht}}|$ connections. The shot graph $\mathcal{G}$ is an undirected graph and $\mathbf{A}_{\text{sht}} \in \mathbb{R}^{N\times N}$ is the adjacent matrix.

Different from the motion graph in Section III-B that focuses on the difference between clips, the shot graph aims to explore the similarity of shots within scenes. Since several consecutive shots constitute a meaningful scene, we first set a neighborhood of size K for each shot to dredge the information flow between different shots within a single scene. Thus, the adjacent matrix $\mathbf{A}_{\text{sht}}$ can be calculated by:

$$A_{\text{sht}}^{ij} = \begin{cases} 1, & \text{if } |i-j| \le K \\ 0, & \text{otherwise} \end{cases}, \tag{9}$$

where the element A_{sht}^{ij} controls the connections between the shot features $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$.

We argue that the connection magnitude of different shots in the same scene is stronger than that of different scenes. However, only the static graph $\mathbf{A}_{\text{sht}}$ cannot achieve this. In addition, we do not know exactly how many neighbors each shot has and different shots may have different numbers of neighbors. To make the topology more adaptive, we acquire the learnable adjacent matrix by using the self-attention mechanism [49]. The process can be represented as:

$$B_{\text{sht}}^{ij} = \text{softmax}\left(\frac{(\boldsymbol{s}_i\mathbf{W}_1)\cdot(\boldsymbol{s}_j\mathbf{W}_2)^\top}{\sqrt{D_2}}\right), \tag{10}$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{C_2\times D_2}$ denote linear transformation matrices, D_2 is the embedding dimension, and $\text{softmax}(\cdot)$ represents the normalized exponential activation function. The element B_{sht}^{ij} measures the connection magnitude between the shot features $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$.

3) Shot Graph Aggregation: In this work, the shot graph aggregation operation can be also implemented by a basic GCN layer:

$$\boldsymbol{s}_i' = \text{ReLU}\left(\boldsymbol{s}_i\mathbf{W}_{\text{sht}} + \sum_{j\in\mathcal{N}_i} A_{\text{adp}}^{ij}\boldsymbol{s}_j\mathbf{U}_{\text{sht}}\right), \tag{11}$$

where $\mathbf{W}_{\text{tra}}, \mathbf{U}_{\text{tra}} \in \mathbb{R}^{C_2\times C_3}$ denote linear transformation matrices, $\boldsymbol{s}_i' \in \mathbb{R}^{C_3}$ represents the i-th updated feature, and C_3 is the dimension size of the updated feature. In this work, we set the adaptive weight A_{sht}^{ij} as $A_{\text{sht}}^{ij} \odot B_{\text{sht}}^{ij}$ or $A_{\text{sht}}^{ij} + B_{\text{sht}}^{ij}$ to explore the optimal way.

4) Shot Reduction: Though different actions in the same class have the same procedures, each procedure for different samples contains different shots. The recognition of such subtle differences is important and challenging for AQA.

To address the above challenge, we adopt a differential graph transformation mechanism to generate the transformation matrix $\mathbf{T} \in \mathbb{R}^{N\times S}$ by learning a direct mapping from N shots to S scenes using a GCN layer:

$$\mathbf{T} = \text{softmax}\left(\bigoplus_{i=1}^{N} A_{\text{adp}}^{ij}\sum_{j=1}^{N}\boldsymbol{s}_j'\mathbf{W}_{\text{tft}}\right), \tag{12}$$

where $\oplus$ denotes the concatenation operation and $\mathbf{W}_{\text{tft}} \in \mathbb{R}^{C\times S}$ indicates the linear transformation matrix. Then, we can get scenes by:

$$\mathbf{E} = \mathbf{S}'\mathbf{T}, \tag{13}$$

where $\mathbf{E} \in \mathbb{R}^{S\times C_3}$ represents the scene matrix with S scenes $\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_S$ and $\mathbf{S}' \in \mathbb{R}^{N\times C_3}$ denotes the feature matrix composed of the updated shot features $\boldsymbol{s}_1', \boldsymbol{s}_2', \cdots, \boldsymbol{s}_S'$.

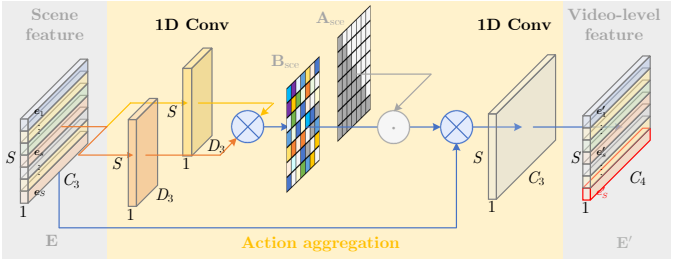

Fig. 5. The network architecture of the action graph aggregation module: the scene feature $\mathbf{E}$ is encoded to the video-level representation $\mathbf{E}'$ through action graph aggregation.

D. Action Aggregation

The action aggregation aims at capturing the global dynamics of an action, which reflects the comprehensive execution qualities of all procedures and is critical for the final score.

1) Analysis: Considering the successive nature between different scenes, the action information of the former scene can only be transmitted to its latter scenes, while the latter cannot. Identifying such property is important for AQA, as any change in the action procedures will affect the final score. Therefore, we need to explore the dependencies between different action procedures and aggregate them to obtain the video-level representation for score distribution regression. The network architecture of the action aggregation module is depicted in Figure 5, which consists of two steps: scene graph construction, and video-level aggregation. The following provides a detailed introduction to them.

2) Scene Graph Construction: To explore dependencies between action procedures, we construct a scene graph as $\mathcal{G}_{\text{sce}} = (\mathcal{V}_{\text{sce}}, \mathcal{E}_{\text{sce}})$, including $|\mathcal{V}_{\text{sce}}|$ nodes and $|\mathcal{E}_{\text{sce}}|$ connections. The scene graph $\mathcal{G}_{\text{sce}}$ is a direct graph and $\mathbf{A}_{\text{sce}} \in \mathbb{R}^{N \times N}$ is the adjacent matrix.

The successive nature of actions controls the flow of information from front to back, so the adjacent matrix $\mathbf{A}_{\text{sce}}$ can be calculated by:

$$A_{\text{sce}}^{ij} = \begin{cases} 1, & \text{if } i - j > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{14}$$

where A_{sce}^{ij} controls connections between two scenes. Similar to the shot graph construction in Section III-C, we need to learn an adaptive weight B_{sce}^{ij} for modeling the relationships and the adaptive connection magnitude between different scenes, which can be calculated by:

$$B_{\text{sce}}^{ij} = \text{softmax}\left(\frac{(\boldsymbol{e}_i \mathbf{U}_1) \cdot (\boldsymbol{e}_j \mathbf{U}_2)^\top}{\sqrt{D_3}}\right), \tag{15}$$

where $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{C_3 \times D_3}$ denote linear transformation matrices, and D_3 is the embedding dimension.

3) Video-level Aggregation: We need to aggregate discriminative features to recognize the subtle differences between different actions for score regression. Different from time-consuming LSTMs, we implement action aggregation by the efficient GCN layer:

$$\boldsymbol{e}_i' = \text{ReLU}\left(\boldsymbol{e}_i \mathbf{W}_{\text{act}} + \sum_{j \in \mathcal{N}_i} \left(A_{\text{sce}}^{ij} \cdot B_{\text{sce}}^{ij}\right) \boldsymbol{e}_j \mathbf{U}_{\text{act}}\right), \tag{16}$$

where $\mathbf{W}_{\text{act}}, \mathbf{U}_{\text{act}} \in \mathbb{R}^{C_3 \times C_4}$ denote linear transformation matrices, $\boldsymbol{e}_i' \in \mathbb{R}^{C_4}$ is the corresponding updated feature of $\boldsymbol{e}_i$, and C_4 is the dimension of updated features. The enhanced scene feature $\boldsymbol{e}_S'$ is regarded as the final video-level representation $\boldsymbol{v} \in \mathbb{R}^{C_4}$ for the score regression operation.

E. Score Distribution Regression

The score distribution regression module aims to predict the score distribution instead of directly regressing the final score, so that we can attend to the intrinsic ambiguity in the score labels caused by multiple judges or their subjective appraisals, thus improving the scoring performance.

Regarding AQA as a regression problem ignores the intrinsic ambiguity in the score labels caused by multiple judges or their subjective appraisals. The uncertainty-aware score distribution learning [11], [16] is usually used to address the above problem, and their main idea is to learn a score distribution that can automatically generate distinguishable variances for two different actions. Therefore, similar to the previous work [16], the action score in this paper is also represented as a random variable. Then, we need to learn its corresponding score distribution, which indicates the probability of different evaluated scores. Finally, the predicted score is sampled from the learned distribution.

For the video-level feature $\boldsymbol{v}$, a probabilistic encoder $\mathbb{R}^{C_4} \to \mathbb{R}$ is first used to encode $\boldsymbol{v}$ into a random score variable s. The encoded score random variable s is subject to the Gaussian distribution, as follows:

$$p(s; \boldsymbol{v}) = \frac{1}{\sqrt{2\pi\sigma^2(\boldsymbol{v})}} \exp\left(-\frac{(s - \mu(\boldsymbol{v}))^2}{2\sigma^2(\boldsymbol{v})}\right), \tag{17}$$

where the mean parameter μ and variance parameter σ^2 *w.r.t.* the feature representation $\boldsymbol{v}$ are used to quantify the quality and uncertainty of the action score, respectively. The re-parameterization trick [50] is then applied to the sample from the distribution to output the predicted score.

As illustrated in Figure 2, we do not directly sample from the score distribution, but the first sample from another random variable ϵ, which is distributed in the standard normal distribution $\mathcal{N}(0, 1)$. In this way, the score distribution sampling process is differentiable to ensure that the encoder training is feasible. Then, the predicted score $\hat{s}$ is calculated according to the independently sampled random variable ϵ, mean parameter $\mu(\boldsymbol{v})$ and variance parameter $\sigma^2(\boldsymbol{v})$ of the output.

$$\hat{s} = \mu(\boldsymbol{v}) + \epsilon \cdot \sigma(\boldsymbol{v}), \tag{18}$$

where the parameter $\sigma(\cdot)$ represents the standard variance. In this way, the score distribution sampling process is differentiable to ensure that the encoder training is feasible.

F. Loss Function

In this work, we use the MSE loss to supervise the score distribution regression.

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(s_i - \hat{s}_i)^2. \tag{19}$$

where $\hat{s}_i$ and s_i are denoted as the predicted score and the ground-truth score of the i-th sample, respectively. The MSE loss controls the training process directly so that we can progress toward convergence during the training procedure.

IV. Experiments

In this section, we briefly introduce the experimental setup, including datasets, evaluation metrics, and implementation details. Next, we present extensive qualitative and quantitative experiments and analyze the results of these experiments.

A. Datasets

We evaluate the performance of the proposed method on three public quality assessment datasets in the sports domain.

AQA-7 [18] contains a total of 1,189 samples from seven different actions, which are collected from winter and summer Olympic games. It is made of seven datasets, including single diving-10m platform (370 samples, previously released as NLV-Dive [22]), gymnastic vault (176 samples, previously released as UNLV-Vault [22]), big air skiing (175 samples), big air snowboarding (206 samples), synchronous diving-3m springboard (88 samples), and synchronous diving-10m platform (91 samples).

MTL-AQA [19] is the currently largest dataset for AQA. It contains the *diving* action with 1,412 samples including both male and female, both individual and synchronous divers, both 3m springboard and 10m platform, and different views. The various annotations consist of the *degree of difficulty* (DD), scores from 7 judges, the action type of the diver, and the final score. We adopt the evaluation protocol suggested in [19] in our experiments.

JIGSAWS [20] is a surgical action dataset that contains 3 types of the surgical task: *Suture* (S), *NeedlePassing* (NP), and *Knotted* (KT). For each task, each video sample is annotated with multiple annotation scores assessing different aspects of surgical actions, and the final score is the sum of those sub-scores. Every action in the dataset is recorded by the left and right cameras at the same time. We adopt a similar four-fold cross-validation strategy as previous works [37], [11].

B. Evaluation Metrics

We use two evaluation metrics to validate the performance of the proposed and other AQA methods.

Similar to previous works [14], [11], [3], we adopt the Spearman's rank correlation coefficient ρ to evaluate the performance of AQA methods. ρ is defined as the Pearson correlation coefficient between two rank vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ *w.r.t.* the predicted and ground-truth scores:

$$\rho = \frac{\sum_i (p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_i (p_i - \bar{p})^2 \sum_i (q_i - \bar{q})^2}}, \tag{20}$$

where $\bar{p}$ and $\bar{q}$ denote the average values of the rank vectors $\boldsymbol{p}$ and $\boldsymbol{q}$, respectively. The higher the value of the coefficient ρ, the higher the rank correlation between the predicted scores and ground-truth scores.

We also adopt a stricter metric to measure the performance of AQA models more precisely, which is called the relative ℓ_2 distance (R-ℓ_2) [12]. Given the highest and lowest scores for an action $s_{\max}$ and $s_{\min}$, the relative ℓ_2 distance R-ℓ_2 is defined as:

$$\text{R-}\ell_2 = \frac{1}{N}\sum_{n}^{N}\left(\frac{|s_n - \hat{s}_n|}{s_{\max} - s_{\min}}\right)^2 \times 100, \tag{21}$$

where s_n and $\hat{s}_n$ represent the ground-truth score and prediction for the n-th sample, respectively. Fisher's z-value is used to measure the average performance across actions.

C. Implementation Details

We have implemented the proposed hierarchical GCN with the PyTorch deep learning framework and accelerated the training process with two NVIDIA RTX 3090 GPUs.

For the AQA-7 and MTL-AQA datasets, we extract 103 frames for each video clip as same as previous works [11], [12], [19], [37], and then divide them into 10 overlapping clips, each containing 16 continuous frames. For the JIGSAWS dataset, we follow the previous work [11] to evenly sample out 160 frames and form 10 non-overlapping 16-frame clips. The channel numbers C_1, C_2, C_3, C_4 are set to 1024, 512, 526, and 128, respectively, and $D_1 = C_1/2, D_2 = C_2/2, D_3 = C_3/2$. We adopt the I3D model pre-trained on the Kinetics dataset [6] as the feature extractor. The learning rate is set to $1e^{-4}$ and the Adam optimizer is adopted with the weight decay 10^{-4}. The group number g of all convolution and MLP layers is set to 4. In practice, these hyper-parameters can be slightly adjusted for different datasets. The maximum number of training epochs is set to 200.

D. Results and Analysis

Firstly, we show the comparison results of three datasets with the state-of-the-art AQA methods, respectively. Then, we qualitatively verify different phases of the proposed method. Finally, extensive ablation studies are performed to explore the effectiveness of basic components.

1) Comparisons with the State-of-the-Art: Tables I to III are comparison results with state-of-the-art methods on the AQA-7, MTL-AQA, and JIGSAWS datasets, respectively. For example, on the MTL-AQA dataset in Table I, both ours with DD and ours without DD achieve the best assessment performance. With the DD information, ours achieves a Spearman's rank coefficient of 0.9563, which is 0.0170 higher than that of TSA-Net [15]. Additionally, although the proposed method does not perform as well as TSA-Net [15] in *Gym Vault*, *BigSnow.* and *Sync. 3m* of the AQA-7 dataset, it is generally better than that of TSA-Net [15] from Table II. All of the results in Tables I to III demonstrate that the proposed method outperforms the state-of-the-art methods.

TABLE I
COMPARISONS OF THE SPEARMAN'S COEFFICIENT ρ AND R-ℓ_2 DISTANCE WITH STATE-OF-THE-ART METHODS ON THE MTL-AQA DATASET.

DD	Methods	ρ	R-ℓ_2
w/o	Pose + DCT [4]	0.2682	–
	C3D-SVR [22]	0.7716	–
	C3D-LSTM [22]	0.8489	–
	MSCADC-STL [19]	0.8472	–
	C3D-AVG-STL [19]	0.8960	–
	MSCADC-MTL [19]	0.8612	–
	C3D-AVG-MTL [19]	0.9044	–
	USDL [11]	0.9066	0.654
	MUSDL [11]	0.9158	0.609
	I3D + MLP [12]	0.9196	0.465
	CoRe [12]	0.9341	0.365
	I3D + MLP	0.9301	0.424
	Ours	**0.9390**	**0.360**
w/	RGR [3]	0.7600	–
	USDL [11]	0.9231	0.468
	MUSDL [11]	0.9273	0.451
	I3D + MLP [12]	0.9381	0.394
	CoRe [12]	0.9512	0.260
	TSA-Net [15]	0.9393	–
	UD-AQA [51]	0.9545	0.259
	I3D + MLP	0.9452	0.371
	Ours	**0.9563**	**0.235**

2) Qualitative and Quantitative Results: Unless otherwise stated, all qualitative and quantitative experiments are conducted on the MTL-AQA dataset.

Clip refinement: To clearly understand the process of the clip refinement module, we show the learned heatmaps and illustrate the information transfer flow on the MTL-AQA dataset in Figure 6. Figure 6(a) and (b) show heatmaps of the adjacent matrix $\mathbf{A}_{\text{mot}}$ before and after the clip refinement when the neighborhood scope r is equal to 1. Since the information transfer flow between two clips is directed, we can see that $\mathbf{A}_{\text{mot}}$ is a negative symmetric matrix and is subject to $A^{ij}_{\text{mot}} = -A^{ij}_{\text{mot}}$. It can be seen from Figure 6(a) and (b) that the relationships between non-adjacent clips are zero. The information transfer flow *w.r.t.* Figure 6(a), is illustrated in Figure 6(c). For example, $d_{78} = 0.29$ means that the 7-th clip should receive 0.29 amount of information from the 8-th clip, indicating that the 7-th clip is incomplete; in contrast, $d_{87} = -0.29$ means that the 8-th clip should remove 0.29 amount of information from the 7-th clip, indicating that the 8-th clip is redundant. In Figure 6(b), all the information transfer strength is lower than 0.1 and can be viewed as noises, indicating that all of shots are relatively complete and dependent in terms of semantic information. In other words, it supports that the proposed clip refinement module is effective to handle the information confusion problem.

Scene construction: To clearly visualize the scene construction process, Figure 7 illustrates the heatmaps of the normalized transformation matrix $\mathbf{T}$ in Equation (12) when the scene number S is equal to 3, 5, and 7, respectively, conducted on the MTL-AQA dataset. T_{ij} indicates the contribution of the i-th shot to the j-th scene. It can be seen that the 10-th shot makes the biggest contribution to almost scenes. This

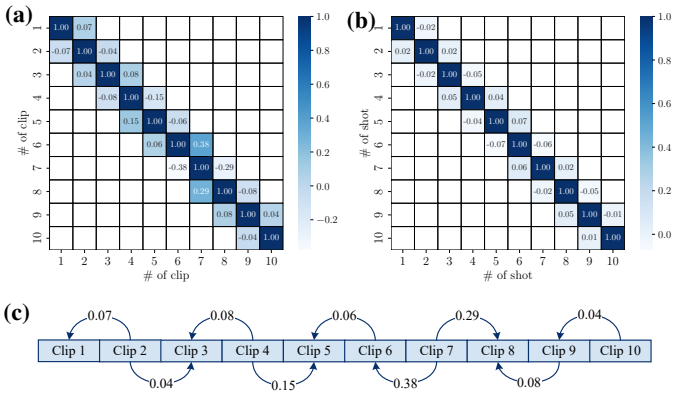

Fig. 6. The visualization of the clip refinement process: (a) and (b) are the heatmaps of the distance matrix with the self-connection before and after clip refinement, respectively; (c) is the diagram of information transfer flow between different clip features.

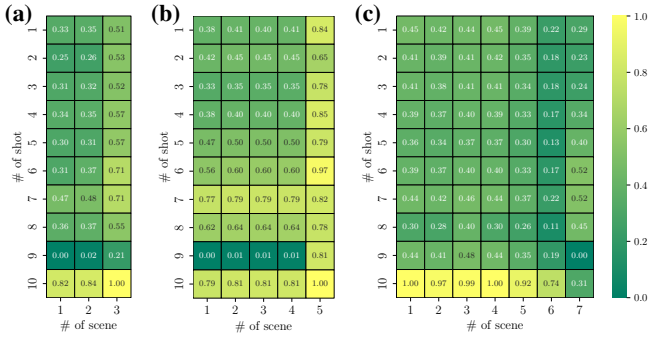

Fig. 7. The visualization of the scene construction process: (a), (b), and (c) are heatmaps of the normalized transformation matrix when the scene numbers are equal to 3, 5 and 7, respectively.

is because all the actions in MTL-AQA belong to the *diving* class, and the *splash* of the diver when falling into the water is usually located in the final shot (the ending of the video) and is an important clue to score. For example, as can be seen in Figure 7(b), the contribution of the 10-th shot to the 5-th scene is 1, indicating that the 10-th shot is considered as part of the final judging procedure.

Action aggregation: To best view the action aggregation process, the learned adjacent matrix $\mathbf{A}_{\text{sce}}$ in Equation (14) and the corresponding action information aggregation flow are shown in Figure 8, conducted on the MTL-AQA dataset. It can be seen from Figure 8(a) that $\mathbf{A}_{\text{sce}}$ is the adjacent matrix of a directed graph and the information can only pass to the current scene from its ahead neighbors. Non-adjacent relationships can effectively increase the receptive field of aggregation, which is conducive to the efficient generation of video-level features. Figure 8(b) shows the corresponding action information aggregation flow. For example, it can be seen that the final scene aggregates information from all the ahead scenes with equal weights, indicating that all action procedures contribute equally to the final scoring.

Predicted score distribution: The score error histogram of $s_{\text{err}} = |\hat{s} - s|$ on the MTL-AQA test set is shown in Figure 9, where blue, orange, and green bars are errors of I3D + MLP

TABLE II
COMPARISONS OF THE SPEARMAN'S COEFFICIENT ρ AND R-ℓ_2 DISTANCE WITH STATE-OF-THE-ART METHODS ON THE AQA-7 DATASET.

Metrics	Methods	Diving	Gym Vault	BigSki.	BigSnow.	Sync. 3m	Sync. 10m	Avg.
ρ	Pose + DCT [4]	0.5300	0.1000	–	–	–	–	–
	C3D-LSTM [22]	0.6047	0.5636	0.4593	0.5029	0.7912	0.6927	0.6165
	C3D-SVR [22]	0.7902	0.6824	0.5209	0.4006	0.5937	0.9120	0.6937
	ST-GCN [52]	0.3286	0.5770	0.1681	0.1234	0.6600	0.6483	0.4433
	JRG [37]	0.7630	0.7358	0.6006	0.5405	0.9013	0.9254	0.7849
	USDL [11]	0.8099	0.7570	0.6538	0.7109	0.9166	0.8878	0.8102
	I3D + MLP [12]	0.8685	0.6939	0.5391	0.5180	0.8782	0.8486	0.7601
	CoRe [12]	0.8824	0.7746	0.7115	0.6624	0.9442	0.9078	0.8401
	TSA-Net [15]	0.8379	**0.8004**	0.6657	**0.6962**	**0.9493**	0.9334	0.8476
	UD-AQA [51]	0.8532	0.7663	0.6836	0.5596	0.9281	**0.9438**	0.8318
	Ours	**0.8867**	0.7917	**0.7326**	0.6447	0.9213	0.9424	**0.8501**
R-ℓ_2	C3D-SVR [22]	1.53	3.12	6.79	7.03	17.84	4.83	6.86
	USDL [11]	0.79	2.09	4.82	4.94	0.65	2.14	2.57
	I3D + MLP [12]	0.81	2.54	6.06	5.31	1.41	3.08	3.20
	CoRe [12]	0.64	**1.78**	3.67	3.87	**0.41**	2.35	2.12
	Ours	**0.59**	1.85	**3.59**	**3.61**	0.82	**1.40**	**1.98**

TABLE III
COMPARISONS OF THE SPEARMAN'S COEFFICIENT ρ AND R-ℓ_2 DISTANCE WITH STATE-OF-THE-ART METHODS ON THE JIGSAWS DATASET.

Metrics	Methods	S	NP	KT	Avg.
ρ	ST-GCN [52]	0.31	0.39	0.58	0.43
	TSN [22]	0.34	0.23	0.72	0.46
	JRG [37]	0.36	0.54	0.75	0.57
	USDL [11]	0.64	0.63	0.61	0.63
	MUSDL [11]	0.71	0.69	0.71	0.70
	I3D + MLP [12]	0.61	0.68	0.66	0.65
	CoRe [12]	0.84	0.86	0.86	0.85
	UD-AQA [51]	0.87	**0.93**	0.86	0.89
	Ours	**0.89**	0.91	**0.90**	**0.90**
R-ℓ_2	I3D + MLP [12]	4.795	11.225	6.120	7.373
	CoRe [12]	5.055	5.688	**2.927**	4.556
	UD-AQA [51]	**3.444**	4.076	5.469	4.330
	Ours	4.784	**3.927**	3.380	**4.031**

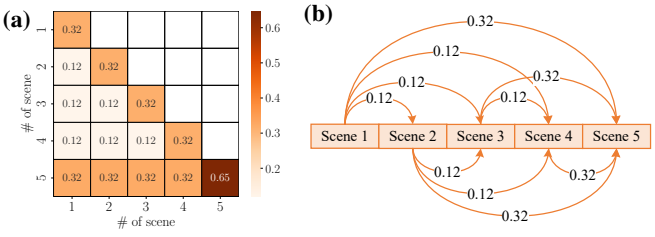

Fig. 8. The visualization of the action aggregation process: (a) denotes the heatmap of the learned adjacent matrix, and (b) illustrates the action information aggregation flow.

[16], CoRe [12] and ours, respectively. On the one hand, According to statistics, 70% and 55% of the sample errors of ours are less than that of I3D + MLP and CoRe, respectively, indicating that ours is more accurate than the others. On the other hand, 77% of the samples using our method have a small score error of less than 5, while that of CoRe [12] is only 71%. For example, the first row in Figure 10 shows the #013 sample and its predicted score distribution using the proposed method.

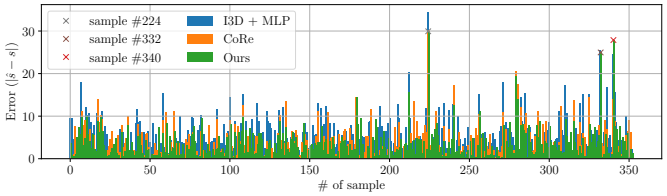

Fig. 9. The histogram of score error $|\hat{s} - s|$ on the MTL-AQA test set: blue, orange, and green are error bars of I3D + MLP, CoRe and ours, respectively.

The predicted score is 91.3, which is only 0.1 points off the ground-truth score of 91.2. The last two rows in Figure 10 show a gym vault sample and a skiing sample on the AQA-7 dataset, where the predicted results are close to their ground-truth scores.

Additionally, there are three samples whose errors are more than 20, *i.e.*, the #224, #332, and #340 samples as shown in Figure 11. For example, for the sample #340, the diver has attempted to challenge the diving action at difficulty 3.1. It can be seen from the last row of Figure 11 that the diver has failed to control the run-up rhythm on the springboard, failed to complete required movements after taking off, and splashed heavily after falling into the water. Finally, the action is judged as a 0 score. However, our model predicts the action score as 27.5, which is far from the ground-truth score. In fact, not only does our model perform poorly for this type of action, but so do the others. At present, we lack an adequate solution in dealing with extreme cases such as fouls. Thus, foul detection before AQA will be a research topic worthy of future investigation.

Evaluation results: To intuitively observe the differences between the proposed hierarchical GCN and other methods, we visualize the prediction results on the MTL-AQA dataset in form of a scatter plot in Figure 12. The blue dashed line indicates the fitted predictions and the orange line represents the perfect predictions. The closer the two lines are, the more accurate the method is. It can be seen that our method is

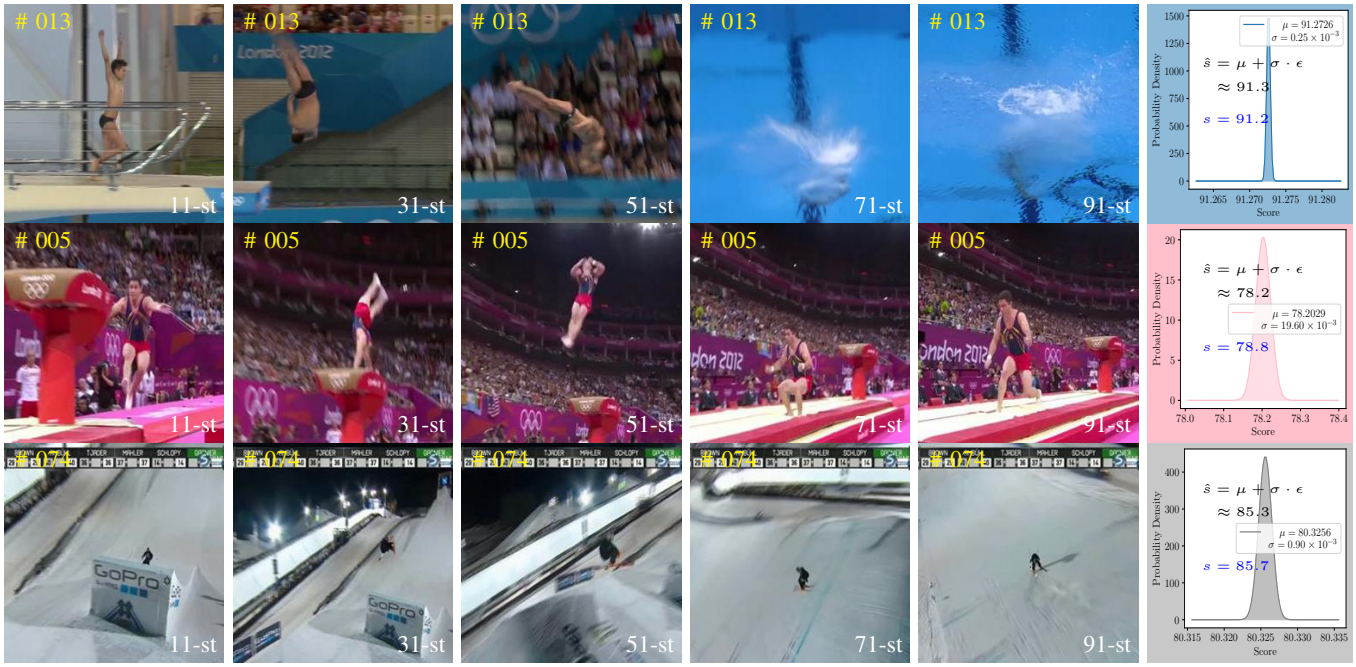

Fig. 10. Three successful action samples on the MTL-AQA and AQA-7 dataset and their predicted score distribution using the proposed hierarchical GCN: the first column to the fourth column denote the 11-st, 31-st, 51-st, and 71-st frames of samples #005 on the MTL-AQA dataset and #332 and #340 on the AQA-7 dataset, respectively; the final column represents the plot of the corresponding predicted score distribution.

much more accurate than other methods, which has a slope of 0.9 in Figure 12(d) and is closest to the perfect line. Without the DD information, the performance can be reduced, which is consistent with the previous works [12], [16]. This is mainly due to DD providing additional degree of difficulty information that is difficult to capture through visual features alone. For example, the same action is scored differently under different scoring standards like difficulty.

Furthermore, Figure 13(a) shows the cumulative score curves of the proposed methods and CoRe [12]. The larger the area under the curve indicates the better performance. Given an error threshold value s_{thr}, the absolute differences of samples between the predicted scores and the corresponding ground-truth scores that are less than s_{thr} will be regarded as positive samples. It can be observed that the red curve marked with the plus sign of the proposed hierarchical GCN shows a stronger ability to predict accurate scores than others under almost all the error thresholds. Notably, the parameter number of the CoRe model is 2.51M, while ours is as low as 0.41M. The training and testing results of 200 epochs on the MTL-AQA dataset are shown in Figure 13(b). The sharp R-ℓ_2 reduction proves that the proposed method converges fast during training. After 50 epochs, the correlation coefficient ρ changes slowly during testing.

3) Ablation Study: Unless otherwise stated, all of the relevant ablation studies are conducted on the MTL-AQA dataset.

Effectiveness of the DD label: Table IV shows the play-and-plug results on the baseline and Table V shows the ablation study of the Spearman's rank correlation coefficient ρ and the relative ℓ_2 distance R-ℓ_2 on the MTL-AQA dataset, where we use "$\star$" to indicate that we use DD in both training and testing. We choose I3D + MLP [16] as our baseline, which achieves the Spearman's rank coefficient of 0.9301 and the relative ℓ_2 distance R-ℓ_2 of 0.424. Notably, it can be seen that the proposed method with DD achieves the best performance where the Spearman's coefficient reaches up to 0.9563 and the relative R-ℓ_2 distance is as low as 0.235. We first verify the effectiveness of DD on the MTL-AQA dataset. By using the DD label in the baseline (I3D + MLP), it can be seen that the performance is improved; by removing the DD label from our

TABLE IV
PLAY-AND-PLUG COMPONENT RESULTS OF THE SPEARMAN'S RANK CORRELATION COEFFICIENT ρ AND THE RELATIVE ℓ_2 DISTANCE R-ℓ_2 ON THE MTL-AQA DATASET.

Protocols	ρ	**R-ℓ_2**
I3D + MLP	0.9301	0.424
I3D + MLP w/ DD	0.9452 $^{\uparrow 0.0151}$	0.371 $^{\downarrow 0.053}$
I3D + MLP w/ Section III-B	0.9343 $^{\uparrow 0.0042}$	0.413 $^{\downarrow 0.011}$
I3D + MLP w/ Section III-C	0.9371 $^{\uparrow 0.0070}$	0.396 $^{\downarrow 0.028}$
I3D + MLP w/ Section III-D	0.9364 $^{\uparrow 0.0063}$	0.409 $^{\downarrow 0.015}$

TABLE V
COMPONENTS ABLATION RESULTS OF THE SPEARMAN'S RANK CORRELATION COEFFICIENT ρ AND THE RELATIVE ℓ_2 DISTANCE R-ℓ_2 ON THE MTL-AQA DATASET.

Protocols	ρ	**R-ℓ_2**
Ours*	**0.9563**	**0.235**
Ours* w/o DD	0.9390 $^{\downarrow 0.0173}$	0.361 $^{\uparrow 0.126}$
Ours* w/o Section III-B	0.9527 $^{\downarrow 0.0036}$	0.287 $^{\uparrow 0.052}$
Ours* w/o Section III-C	0.9484 $^{\downarrow 0.0079}$	0.311 $^{\uparrow 0.076}$
Ours* w/o Section III-D	0.9507 $^{\downarrow 0.0056}$	0.291 $^{\uparrow 0.056}$

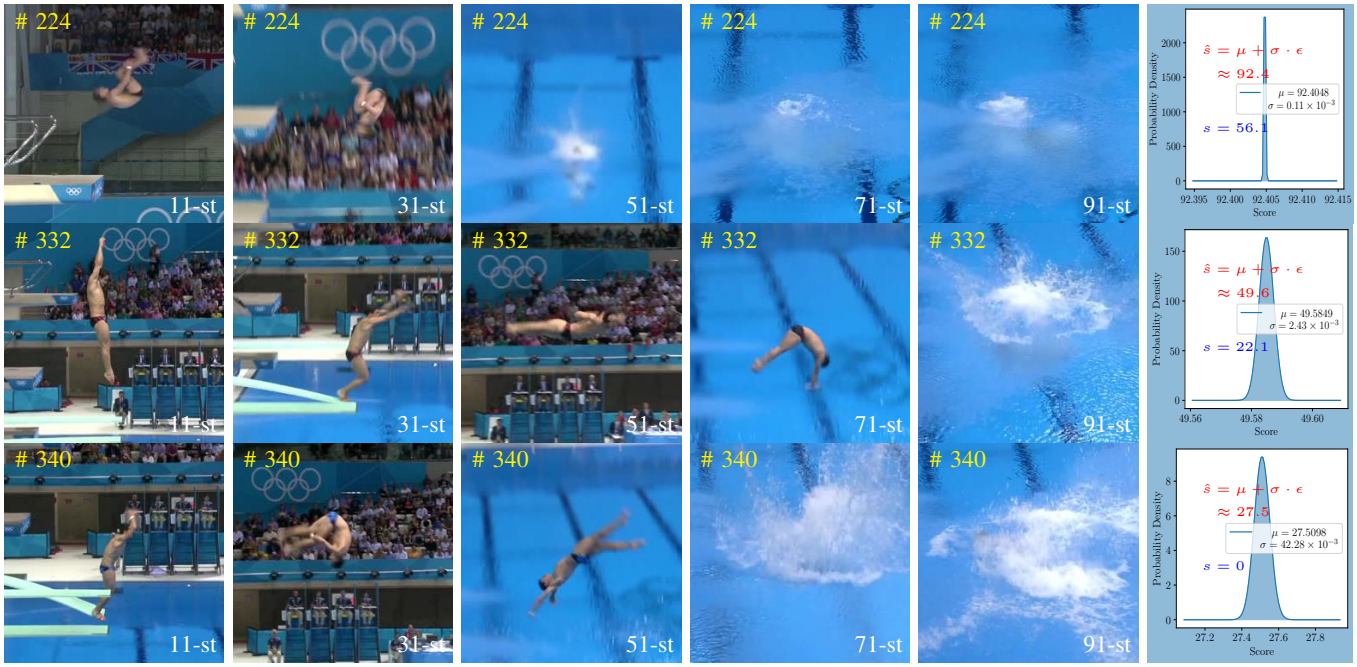

Fig. 11. Three failed diving action samples with large errors on the MTL-AQA dataset and their predicted score distribution using the proposed hierarchical GCN: the first column to the fourth column denote the 11-st, 31-st, 51-st, and 71-st frames of the #224, #332, and #340 samples, respectively; the final column represents the plot of the corresponding predicted score distribution.

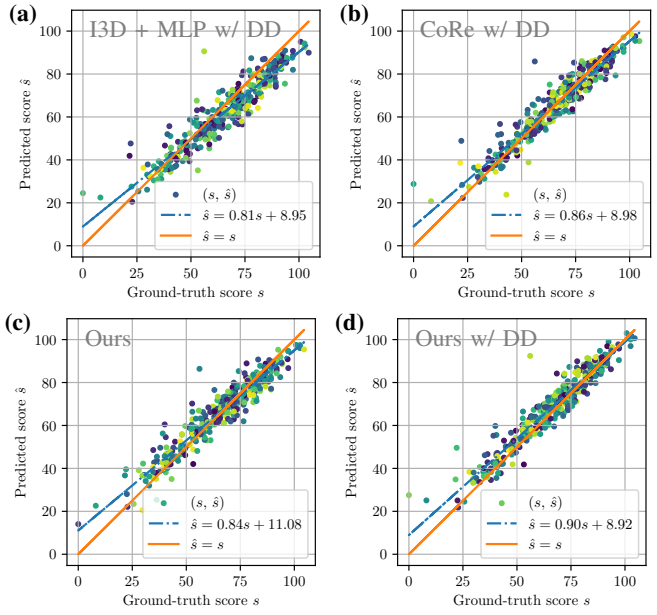

Fig. 12. Scatter plots of correlation for different methods between the ground-truth score s and the predicted score $\hat{s}$: (a) the implemented I3D + MLP method, (b) the CoRe method [12], and (c) the proposed hierarchical GCN.

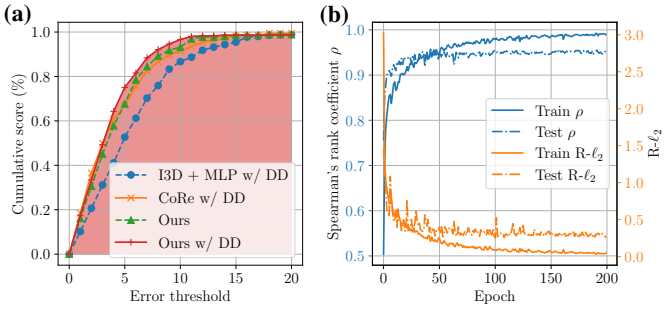

Fig. 13. Plots of (a) the cumulative score curves and (b) the Spearman's rank coefficient ρ and the relative ℓ_2 distance R-ℓ_2.

approach (ours*), the performance is weakened. For example, the Spearman's coefficient is reduced by 1.8% and 0.0173 for ours compared to ours*, indicating the effectiveness of the DD label. When the action is complex, the DD is high. Generally, as long as the diver completes the difficulty without mistakes, an innate advantage can be given.

Effectiveness of basic modules: On the one hand, we separately add the clip refinement module (Section III-B), the scene construction module (Section III-C), and the action aggregation module (Section III-D) into the baseline in order to verify their effectiveness. The corresponding results are reported in Table IV. It can be seen that the baseline performance is improved regardless of which module is added. For example, by adding the scene graph construction module, the Spearman's rank coefficient is increased by 0.75% and 0.007; the relative R-ℓ_2 distance is reduced by 0.028. Additionally, the result in Table IV also shows the play-and-plug advantage of the proposed method.

On the other hand, we remove one module at a time from ours* and the corresponding experiment results are shown in Table V. Similarly, it can be seen that the performance is reduced by removing any module, indicating that each proposed module is necessary and functional.

Effectiveness of different neighbors: For the scene construction phase, we first need to perform the shot graph construction. To explore the effectiveness of the different

TABLE VI
RESULTS ON THE EFFECTIVENESS OF DIFFERENT NEIGHBORS.

Metrics	$K=0$	$K=1$	$K=2$	$K=3$	$K=4$
$\boldsymbol{\rho}$	0.9510	0.9541	**0.9563**	0.9552	0.9522
R-ℓ_2	0.281	0.264	**0.235**	0.273	0.280

TABLE VII
RESULTS ON THE EFFECTIVENESS OF DIFFERENT SCENES.

Metrics	$S=3$	$S=5$	$S=7$	$S=9$	$\odot$
$\boldsymbol{\rho}$	0.9549	0.9542	**0.9563**	0.9514	0.9535
R-ℓ_2	0.279	**0.231**	0.235	0.283	0.275

neighbors in Equation (9), we set K with different values and the experimental results are shown in Table VI. When K is equal to 0, it means that there is no connection between different shots, and the performance of the model is poor. The larger K is, the larger the receptive field is, and the more efficiently the global information can be aggregated between different shots. However, it can also be observed that a large receptive field reduces model performance. This is because the model cannot pay attention to the local details of the action, which represents the basic procedure for the judge to give the score. Equation (9) shows that when K is equal to 2, the model performs best, indicating that 5 shots can be useful for capturing local dynamics of the motion dynamics for the *diving* action.

Effectiveness of different scenes: We need to determine how many different scenes we can divide for an action. The results of different scenes are reported in Table VII. A scene represents a scoring procedure, and these scoring procedures are the basic units of scoring. For the *diving* action, there are three basic procedures as mentioned above in reality. It can be seen that the performance is relatively good after the scene construction. The larger S is, the finer the granularity of action is and the more accurate the score is. However, a large number of scenes may inhibit the performance of the model, *e.g.*, when the number of scenes increases to 9, the performance deteriorates dramatically. This is because a scene contains several shots, and as the number of scenes to be divided increases, there are fewer shots in each scene and the semantics are incomplete. In terms of the relative ℓ_2 distance, the model performs better when the number of scenes is 5 and 7 than that of 3 and 9.

Effectiveness of the adaptive relationships: To avoid the negative effect of the static graph topology, we evaluate the performance of different adaptive strategies as stated in Equation (11). Table VII reports the corresponding results. The final column shows the element-wise strategy when S is equal to 7 and the others are that of the additive strategy. This element-wise strategy $A_{\text{sht}}^{ij} \odot B_{\text{sht}}^{ij}$ preserves the manually defined relationships where local neighbor relationship plays a major role, while the additive operation $A_{\text{sht}}^{ij} + B_{\text{sht}}^{ij}$ allows the network to learn potential connections and capture global dynamics of actions. The experimental results in Table VII show that the additive is much more effective than that of the element-wise one, indicating that undefined relationships also have the potential to improve AQA performance.

TABLE VIII
RESULTS ON THE EFFECTIVENESS OF DIFFERENT GROUPS.

Group Number	$\boldsymbol{\rho}$	**R-ℓ_2**	**Param.**	**GFLOPs**
$g=1$	0.9531	0.258	1.2500M	0.0123
$g=2$	0.9538	0.251	0.6935M	0.0068
$g=4$	0.9563	**0.235**	0.4149M	0.0041
$g=8$	**0.9564**	0.247	**0.2757M**	**0.0027**

Effectiveness of different groups: As shown in Equation (1), we use the group convolution to promote the computational performance for AQA. To explore the effectiveness of different groups, we have conducted experiments on the MTL-AQA dataset and the corresponding results are shown in Table VIII. It can be seen from Table VIII that with the increase of the number of groups, the number of parameters and computation (GFLOPs) of the model are reduced. For example, when the number of groups is equal to 1, the computation is 0.0123 GFLOPs, and when the number of groups is 8, the calculation is only 0.0027 GFLOPs. Notably, the performance of $g = 8$ is slightly better than that of $g = 1$. We have also noted that when the number of groups is 8, the Spearman's rank coefficient ρ is the largest, almost equal to that of 4, but the relative distance R-ℓ_2 is much larger than the number of groups is 4. This indicates that when the number of groups is greater than 4, the performance of the model is unstable and tends to decline. To balance the performance of the model with the computational overhead, we finally set the number of groups as 4 for the other experiments.

V. DISCUSSION AND CONCLUSION

This paper presents a hierarchical method using GCNs for AQA to address intra-clip confusion and inter-clip incoherence issues. To tackle semantic confusion, a clip refinement module is designed, which serves as a strong foundation for further hierarchical action analysis. Then, the shot reduction is used to detect meaningful scenes and score action in detail. The action aggregation module aggregates video-level representation, which enables better score distribution regression and improves the scoring performance among scenes. Experiments on AQA-7, MTL-AQA, and JIGSAWS prove that the proposed method outperforms the state-of-the-art.

Our hierarchical AQA network consists of four stages: feature extraction, clip refinement, scene construction, and action aggregation. The main difference between scene construction and action aggregation lies in the level of aggregation. While combining scene construction and action aggregation could provide a more comprehensive representation of the action and improve performance, it also has potential drawbacks such as increased model complexity and the need for careful design choices. Furthermore, the combined stage may require more data to achieve good generalization performance. In addition to scoring accuracy, real-time performance is also crucial for practical applications. Therefore, we have disentangled the

process into two stages to balance accuracy and efficiency. Further research is needed to determine the optimal design of an AQA system that combines the two stages, considering factors such as data availability, computational resources, and interpretability.

Despite the effectiveness of our approach in addressing the intra-clip confusion and inter-clip incoherence problem in AQA, there are s some limitations that need to be addressed in future research. Firstly, we acknowledge that our vision-based approach may not be able to effectively deal with chaotic environments characterized by large amounts of interference, such as background crowd interference. To mitigate the effects of background interference, we plan to explore data pre-processing operations such as denoising and contrast enhancement [53], [54], [55] to improve the accuracy of pose estimation. Additionally, we plan to leverage multi-modality inputs, incorporating both pose and visual information, to enhance the discriminative power of our model's features. Secondly, we recognize that our model may struggle to understand complex and rich semantics between entities due to implicit semantics mining. To address this limitation, we plan to investigate the use of action segmentation methods and attention mechanisms in combination with a hierarchical graph convolutional block to uncover the semantic relationships between action procedures. Lastly, we acknowledge the difficulty in accurately assessing the quality of extreme sports activities, such as fouls, which could limit the performance of the assessment. To overcome this limitation, we plan to explore the detection of foul actions before AQA to improve the assessment performance further. These limitations highlight the need for further research to improve the robustness and accuracy of AQA models in complex environments. We believe that addressing these limitations will enable our model to perform better in challenging scenarios, and we look forward to exploring these avenues of research in the future.

REFERENCES

[1] Z. Yang, Y. Li, J. Yang, and J. Luo, "Action recognition with spatio–temporal visual attention on skeleton image sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2405–2415, 2018.

[2] J. Kong, H. Deng, and M. Jiang, "Symmetrical enhanced fusion network for skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4394–4408, 2021.

[3] H. Jain, G. Harit, and A. Sharma, "Action quality assessment using siamese network-based deep metric learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 6, pp. 2260–2273, 2020.

[4] H. Pirsiavash, C. Vondrick, and A. Torralba, "Assessing the quality of actions," in *European Conference on Computer Vision*, 2014, pp. 556–571.

[5] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4305–4314.

[6] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4724–4733.

[7] J. Xu, Y. Rao, X. Yu, G. Chen, J. Zhou, and J. Lu, "Finediving: A fine-grained dataset for procedure-aware action quality assessment," *arXiv preprint arXiv:2204.03646*, 2022.

[8] A. Paiement, L. Tao, S. Hannuna, M. Camplani, D. Damen, and M. Mirmehdi, "Online quality assessment of human movement from skeleton data," in *BMVC*, 2014, pp. 153–166.

[9] M. Antunes, R. Baptista, G. Demisse, D. Aouada, and B. Ottersten, "Visual and human-interpretable feedback for assisting physical activity," in *European Conference on Computer Vision*. Springer, 2016, pp. 115–129.

[10] K. Zhou, R. Cai, Y. Ma, Q. Tan, X. Wang, J. Li, H. P. H. Shum, F. W. B. Li, S. Jin, and X. Liang, "A video-based augmented reality system for human-in-the-loop muscle strength assessment of juvenile dermatomyositis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2456–2466, 2023.

[11] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020.

[12] X. Yu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, "Group-aware contrastive regression for action quality assessment," in *Proceedings of the IEEE international conference on computer vision*, 2021, pp. 7919–7928.

[13] H. Doughty, D. Damen, and W. Mayol-Cuevas, "Who's better? who's best? pairwise deep ranking for skill determination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6057–6066.

[14] H. Doughty, W. Mayol-Cuevas, and D. Damen, "The pros and cons: Rank-aware temporal attention for skill determination in long videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 7862–7871.

[15] S. Wang, D. Yang, P. Zhai, C. Chen, and L. Zhang, "Tsa-net: Tube self-attention network for action quality assessment," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4902–4910.

[16] B. Zhang, J. Chen, Y. Xu, H. Zhang, X. Yang, and X. Geng, "Auto-encoding score distribution regression for action quality assessment," *arXiv preprint arXiv:2111.11029*, 2021.

[17] S.-J. Zhang, J.-H. Pan, J. Gao, and W.-S. Zheng, "Semi-supervised action quality assessment with self-supervised segment feature recovery," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[18] P. Parmar and B. Morris, "Action quality assessment across multiple actions," in *IEEE winter conference on applications of computer vision (WACV)*, 2019, pp. 1468–1476.

[19] P. Parmar and B. T. Morris, "What and how well you performed? a multitask learning approach to action quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 304–313.

[20] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Bejar, D. D. Yuh *et al.*, "Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling," in *MICCAI workshop: M2cai*, 2014, p. 3.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[22] P. Parmar and B. Morris, "Learning to score olympic events," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 76–84.

[23] X. Xiang, Y. Tian, A. Reiter, G. D. Hager, and T. D. Tran, "S3d: Stacking segmental p3d for action quality assessment," in *2018 25th IEEE International conference on image processing (ICIP)*. IEEE, 2018, pp. 928–932.

[24] J. Xu, L. Song, and R. Xie, "Shot boundary detection using convolutional neural networks," in *Visual Communications and Image Processing (VCIP)*. IEEE, 2016, pp. 1–4.

[25] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 797–819, 2011.

[26] A. S. Gordon, "Automated video assessment of human performance," in *Proceedings of AI-ED*, 1995, pp. 16–19.

[27] S. Wang, D. Yang, P. Zhai, Q. Yu, T. Suo, Z. Sun, K. Li, and L. Zhang, "A survey of video-based action quality assessment," in *2021 International Conference on Networking Systems of AI (INSAI)*. IEEE, 2021, pp. 1–9.

[28] P. Parmar and B. T. Morris, "Measuring the quality of exercises," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 2241–2244.

[29] J. Carvajal, A. Wiliem, C. Sanderson, and B. Lovell, "Towards miss universe automatic prediction: The evening gown competition," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 1089–1094.

[30] H. Doughty, W. W. Mayol-Cuevas, and D. Damen, "The pros and cons: Rank-aware temporal attention for skill determination in long videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 7862–7871.
[31] I. Laptev, "On space-time interest points," *IJCV*, vol. 64, no. 2, pp. 107–123, 2005.
[32] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *IJCV*, vol. 103, no. 1, pp. 60–79, 2013.
[33] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.
[34] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.
[35] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*. PMLR, 2015, pp. 843–852.
[36] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1510–1517, 2017.
[37] J. Pan, J. Gao, and W. Zheng, "Action assessment by joint relation graphs," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 6330–6339.
[38] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*. Springer, 2016, pp. 20–36.
[39] Y. Rui, T. S. Huang, and S. Mehrotra, "Exploring video structure beyond the shots," in *IEEE International Conference on Multimedia Computing and Systems*. IEEE, 1998, pp. 237–240.
[40] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang, "A formal study of shot boundary detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 2, pp. 168–186, 2007.
[41] G. Yan and M. Woźniak, "Accurate key frame extraction algorithm of video action for aerobics online teaching," *Mobile Networks and Applications*, pp. 1–10, 2022.
[42] L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 810–822, 2013.
[43] H. Wang, E. S. Ho, H. P. Shum, and Z. Zhu, "Spatio-temporal manifold learning for human motions via long-horizon modeling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 1, pp. 216–227, 2019.
[44] D.-K. Jang and S.-H. Lee, "Constructing human motion manifold with sequential networks," in *Computer Graphics Forum*, vol. 39, no. 6. Wiley Online Library, 2020, pp. 314–324.
[45] K. Zhou, Z. Cheng, H. P. Shum, F. W. Li, and X. Liang, "Stgae: Spatial-temporal graph auto-encoder for hand motion denoising," in *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2021, pp. 41–49.
[46] Z. Zhang and K. Zhao, "Low-rank matrix approximation with manifold regularization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1717–1729, 2012.
[47] S. Li, K. Li, and Y. Fu, "Temporal subspace clustering for human motion segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4453–4461.
[48] M. Zhu, X. Wang, C. Shi, H. Ji, and P. Cui, "Interpreting and unifying graph neural networks with an optimization framework," in *Proceedings of the Web Conference 2021*, 2021, pp. 1215–1226.
[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[50] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *arXiv preprint arXiv:1906.02691*, 2019.
[51] C. Zhou and Y. Huang, "Uncertainty-driven action quality assessment," *arXiv preprint arXiv:2207.14513*, 2022.
[52] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI conference on artificial intelligence*, 2018, pp. 7444–7452.
[53] S. P. Dakua and J. Abi-Nahed, "Patient oriented graph-based image segmentation," *Biomedical Signal Processing and Control*, vol. 8, no. 3, pp. 325–332, 2013.
[54] S. P. Dakua, J. Abinahed, and A. A. Al-Ansari, "Pathological liver segmentation using stochastic resonance and cellular automata," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 89–102, 2016.
[55] S. P. Dakua, J. Abinahed, and A. Al-Ansari, "A pca-based approach for brain aneurysm segmentation," *Multidimensional Systems and Signal Processing*, vol. 29, pp. 257–277, 2018.

Kanglei Zhou received a BSc degree from the College of Computer and Information Engineering at Henan Normal University in 2020. Now, he is pursuing a Ph.D. degree at the School of Computer Science and Engineering, Beihang University. His research interests include human motion analysis and augmented reality.

Yue Ma received a Bachelor's degree from Beijing University of Chemical Technology and received a Master's degree from Beihang University. Now, he is pursuing a Ph.D. degree at the School of Computer Science and Engineering, Beihang University. His research interests include human motion analysis and augmented reality.

Hubert P. H. Shum (Senior Member, IEEE) is an Associate Professor and the Deputy Director of Research in Computer Science at Durham University, researching on human-centric computer vision and graphics. Before this, he was an Associate Professor at Northumbria University, and a Postdoctoral Researcher at RIKEN Japan. He received his PhD degree from the University of Edinburgh. He chaired conferences such as Pacific Graphics, BMVC and SCA, and has authored over 150 research publications.

Xiaohui Liang received his Ph.D. degree in computer science and engineering from Beihang University, China. He is currently a Professor, working at the School of Computer Science and Engineering at Beihang University. His main research interests include computer graphics and animation, visualization, and virtual reality.