Action Assessment by Joint Relation Graphs

Jia-Hui Pan[1], Jibin Gao[1], Wei-Shi Zheng[1,2,3,*]
[1]School of Data and Computer Science, Sun Yat-sen University, China
[2]Peng Cheng Laboratory, Shenzhen 518005, China
[3]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
panjh7@mail2.sysu.edu.cn, gaojb5@mail2.sysu.edu.cn, wszheng@ieee.org

Abstract

We present a new model to assess the performance of actions visually from videos by graph-based joint relation modelling. Previous works mainly focused on the whole scene including the performer's body and background, yet they ignored the detailed joint interactions. This is insufficient for fine-grained and accurate action assessment, because the action quality of each joint is dependent of its neighbouring joints. Therefore, we propose to learn the detailed joint motion based on the joint relations. We build trainable Joint Relation Graphs, and analyze joint motion on them. We propose two novel modules, the Joint Commonality Module and the Joint Difference Module, for joint motion learning. The Joint Commonality Module models the general motion for certain body parts, and the Joint Difference Module models the motion differences within body parts. We evaluate our method on six public Olympic actions for performance assessment. Our method outperforms previous approaches (+0.0912) and the whole-scene analysis (+0.0623) in the Spearman's Rank Correlation. We also demonstrate our model's ability to interpret the action assessment process.

1. Introduction

Action performance assessment is a task of assessing how well an action is performed. Action assessment techniques are rather important in some real-world applications. For example, in medical treatment, action assessment systems can help to monitor and evaluate the patients as they perform daily tasks. In sports, using automatic assessment techniques, we can build universal scoring systems for each Olympic event, helping athletes to improve their performance.

The problem of automatically assessing the performance of actions using has recently been explored in the computer vision community. Several works attempted action performance assessment in videos [6, 7, 24, 10, 21, 20]. However, they analyze the actions in a coarse manner. Many methods [6, 7, 21, 20] simply extract features of the whole scene to assess the performance of actions. Although some works [24, 10] have analyzed the motion of each joint in an attempt to better discover the fine-grained clues for assessment, they analyze the locations of each joint individually. Differently, for evaluating the fine-grained performance, we consider the interactive motion pattern of several locally connected joints rather than looking at every joint individually. For example, in diving a bending knee is normal when the ankle and hip are also bending (e.g. at the rolling stage), whereas it is probably a flaw if both the ankle and hip are straight at a stretching posture (e.g. at the water-entering stage). Therefore, it is more appropriate to focus on the locally connected joints instead of a single joint for assessing the fine-grained performance.

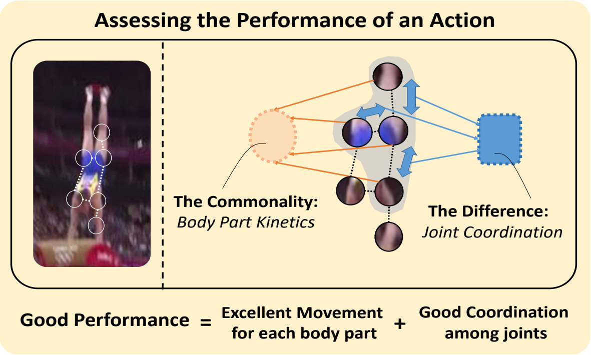

Assessing the Performance of an Action
The Commonality:
Body Part Kinetics
The Difference:
Joint Coordination
Good Performance = Excellent Movement for each body part + Good Coordination among joints

Figure 1. **Learning joint motion on relation graphs for action assessment.** We learn motion commonalities and differences on the joint relation graphs. The motion commonalities reflect the general motion of a body part, while the motion differences indicate the motion coordination. A well-performed action must have both excellent part-based motion and good coordination among joints.

[*] Corresponding author

In this work, we explore the *joint relations* among the motion of locally connected joints for action assessment. In particular, we focus on the body part kinetics modelled via the *motion commonality*, and the joint coordination modelled via the *motion difference* (See Figure 1). The motion commonalities of neighbouring (locally connected) joints indicate the general motion of a certain body part, while the motion differences among those neighbouring joints reflect the action coordination. A well-performed action must have skilled detailed motion and good coordination among joints all together.

To model the relations among the joint motion, we propose a graph-based action assessment network in which the nodes of the graphs correspond to the joint motion. We define two learnable relation graphs: the spatial relation graph to model the joint relations within a time step, and the temporal relation graph to model the joint relations across two immediate time steps. Based on these two graphs, we develop two motion learning modules, namely the joint commonality module and the joint difference module. The joint commonality module extracts the body part kinetics information at a specific time step by aggregating the joint motion in the spatial graph. The joint difference module extracts the coordination information by comparing each joint to its locally connected neighbours in the spatial graph as well as the temporal graph. Our model not only exploits the joint motion to improve the action assessment, but also learns the relation graphs for interpreting the evaluation process, because the trainable joint relation graphs show how much impact the neighbouring joints have on each other for action assessment.

In summary, our contributions are as follows. 1) We present a novel framework for action performance assessment that learns detailed joint motion. 2) We propose a Spatial Relation Graph and a Temporal Relation Graph to model the relations between neighbouring joints. 3) We propose a Joint Commonality Module and a Joint Difference Module to learn the joint motion on the joint relation graphs. We demonstrate that our proposed method outperforms previous works on existing datasets and that it can provide an understanding on the assessment process.

2. Related Work

Action Performance Assessment. Several works attempted action performance assessment in videos [11, 14, 16]. Gordon [11] was the first to explore the viability of automated action assessment from videos. It addresses several issues concerning the application of automated video assessment and demonstrates the application by assessing the performance of gymnastic vaults from skeleton trajectories. Then due to the intensive training needs in medical domain, many works [18, 26, 36, 39, 40, 10, 19] have focused on assessing skills in surgical tasks. However, they design specific features for each surgical maneuver, and thus the methods are difficult to generalize.

Regarding the task formulation, some methods [41, 40] formulate action assessment as a level classification task, splitting participants into categories of novice and expert. Some other methods [37, 6, 7, 1] formulate skill determination in videos as a pair-wise ranking task. Instead, following some methods [24, 32, 21, 20], we exploit reliable scores given by expert judges to guide the learning process and then formulate it as a regression task.

Most existing works on action assessment analyze the whole scene in a coarse manner[6, 7, 21, 20], without further modelling the detailed joint motion. Pirsiavash has taken into consideration the joint location sequences [24]. It extracts the DCT features of the joint location sequences and uses Support Vector Regression model to assess Olympic events. However it models each joint individually, without considering the relations among joints.

Relation Models. There are a few works on modelling different kinds of relations in the computer vision community, such as semantic relations [4], spatial relations[5, 35, 13, 29] and temporal relations [30, 38]. Some works aimed at modelling the spatial-temporal relations on human skeleton structure in action analysis [17, 34, 23, 15, 3]. In modelling the skeleton structure, a few works [17, 34] construct the human skeleton as a tree, which has actually removed some edges from the original skeleton graph, such as waist-to-waist. Others [23, 15] take out neighbouring joints, and tile them up to form an image. This has, in fact, added some irrelevant relations, since some adjacent joints in the image can be unrelated at all in the original skeleton structure. Another work of Çeliktutan's [3] has tried to model skeleton graphs for action analysis. However this work focuses on aligning skeleton dynamic sequences, and has not modelled the joint relations on the skeleton graph. Instead we aim at modelling the joint relations for action analysis, preserving the graph structure of the skeleton joints.

Graph-based Joint Relations. Some works on action recognition but not assessment also model spatial-temporal joint relations through graphs [28, 33, 13]. They simply connect the same joints individually across time. This could work well for action recognition but is not enough in action assessment where the short-term, local fluency and proficiency is of great importance. In comparison, we focus on solving the action assessment problem and argue that a more fine-grained modelling on the joint relations is important. What is more, we particularly propose to aggregate the kinetics differences of each joint from its neighbours on both the spatial and the temporal relation graphs, which has not been attempted on graph modelling.

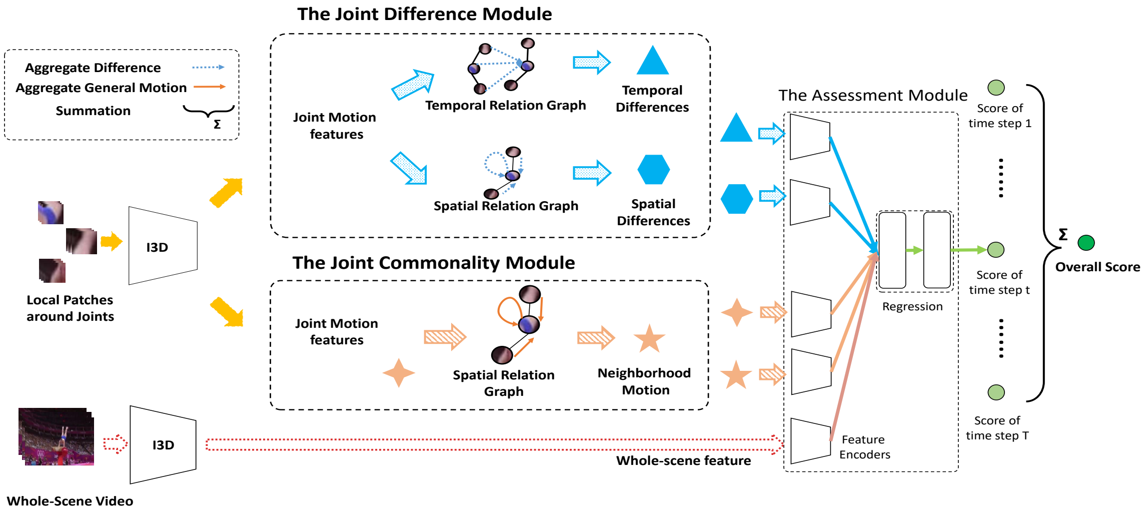


Figure 2. **The overall structure of our model.** An input video is uniformly divided into T time steps. Our model gives assessment results for each time step. We take whole-scene and local-patch videos as input, where the local patches are cropped around joints. We extract the features of the whole-scene videos and the local-patch videos. Then the proposed Joint Commonality Module and the Joint Difference Module learn joint motion on the relation graphs, giving four learned features. The learned features are then fed to the Regression Module. Our model gives a partial result for each time step and an overall result for the whole video.

3. Approach

We consider the interactive motion pattern of several locally connected joints for assessing the performance of an action. For this purpose, we propose to learn both the detailed joint motion and the coordination within a *joint neighbourhood*, which consists of a certain joint and its neighbours. The overall structure of our model is shown in Figure 2.

3.1. Learning the motion of joint neighbourhoods

When assessing the action performance, we concern the motion of joint neighbours. We design a Joint Commonality Module to learn the general motion of a joint neighbourhood.

Before introducing the Joint Commonality Module, we first introduce a learnable **Spatial Relation Graph** which the module works on. The Spatial Relation Graph represents how much impact each neighbour has on the motion of a certain joint within each time step. Each node in the graph represents a certain joint. And each edge represents a relation between a pair of joints. An example of the spatial relation is shown in Figure 3. Note that not every pair of nodes is connected. A pair of nodes $(\mathbf{x}, \mathbf{y})$ is treated as irrelevant, if neither the corresponding joints are exactly the same, nor they are connected in the human skeleton. For example in Figure 3, the node of **a** and the node of **z** are irrelevant. We denote the adjacent matrices for the Spatial Relation Graph as $A_s \in \mathbb{R}^{J\times J}$, where J is the total number of skeleton joints. An element $A_s(i,j)$ in the adjacent matrix denotes how much impact the i^{th} joint has on the j^{th} joint. The elements in A_s are non-negative and learnable, except for those of the irrelevant joint pairs, which are set as zeros. The learnable elements are initialized randomly within $[0, 1)$ at the start of training.

The Joint Commonality Module preforms graph convolutions on the Spatial Relation Graph and learns the joint motion features within joint neighbourhoods, which is inspired by the Graph Convolution Networks [25]. The model outputs *the Commonality Features*, showing the general motion of joint neighbourhoods.

In an aggregation process, each node transmits the motion feature it possesses to its neighbours. We denote the feature matrices before and after the graph convolution as H_c^t, which contains hidden states of all nodes in the t^{th} time step. Here $c \in \{0, 1\}$ denotes whether the graph convolution has been performed. The graph convolution can be written as a matrix multiplication of the adjacent matrix and the hidden states matrix. The computation of H_1^t is as follows:

$$H_1^t = A_s \cdot H_0^t, \qquad (1)$$

where $H_0^t, H_1^t \in \mathbb{R}^{J\times M}$. Here J denotes the total number of joints and M denotes the feature dimension of the hidden states. Specially, the hidden states contain the motion features of the joints before the convolution, i.e. $H_0^t = F^t$, where $F^t \in \mathbb{R}^{J\times M}$ indicates the joint motion features at the t^{th} time step.

Then the module aggregates hidden states of all nodes as a Commonality Feature $\bar{h}_c^t$, where t is the time step number

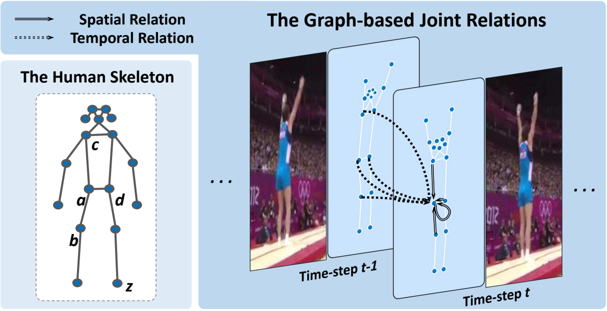


Figure 3. The spatial and temporal joint relation graphs of joint **a**, together with the human skeleton structure. Joint **b**,**c**,**d** are the neighbours of joint **a**, while joint **z** is irrelevant to joint **a**.

and $c \in \{0, 1\}$ denotes whether the graph convolution has been performed. The feature aggregation is a mean pooling, which can be written as:

$$\bar{h}_c^t = \frac{1}{N}({H_c^t}^T \cdot \mathbf{1}), \tag{2}$$

where $\mathbf{1} = [1, 1, ..., 1]^T$ is an all-ones vector.

The proposed Joint Commonality Module learns the motion of individual joints (before the convolution) and that of joint neighbourhoods(after the convolution) , which depict the local motion in smaller and larger granularity.

3.2. Learning coordination in joint neighbourhoods

Beside the general motion of joint neighbourhoods, the motion coordination is also important for action assessment. Great motion differences among joints within a neighbourhood indicate a lack of coordination. We introduce a Joint Difference Module that learns the motion differences of each joint in comparison to its spatial and temporal neighbours. For a certain joint, we now consider not only the motion of its neighbours at the present time step, but also the motion at the previous time step.

For the above purpose, we introduce **the Temporal Relation Graph** to model the joint relations across two immediate time steps. We represent the adjacent matrix for the Temporal Relation Graph as A_p, where $A_p \in \mathbb{R}^{J \times J}$. The Temporal Relation Graph also models the relations among neighbouring joints but across two immediate time step. The element of $A_p(i, j)$ denotes how much impact the i^{th} joint (at the previous time step $t - 1$) has on the j^{th} joint (at the present time step t). Similar to A_s, the adjacent matrix A_p is also non-negative and learnable. The trainable weights are initialized randomly over $[0, 1)$ at the start of training.

The Joint Difference Module learns the motion differences of each joint from its neighbours on both the Spatial Relation Graph and the Temporal Relation Graph. And it outputs *the Difference Features* $\bar{d}_s^t$ *and* $\bar{d}_p^t$ to depict the motion differences in joint neighbourhoods.

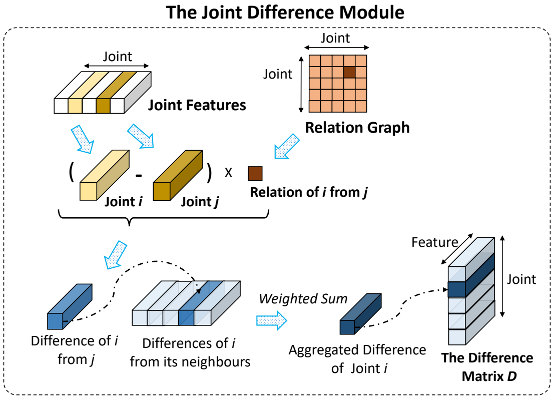


Figure 4. The computation in the Joint Difference Module. First, the motion difference between joint i and any neighbour j is computed. Then the difference is weighted by $A(i, j)$. An aggregated feature of joint i is formed by a weighted sum within its neighbourhood. The Difference Feature D^t for the relation graph A is constructed by the aggregated features of all joints.

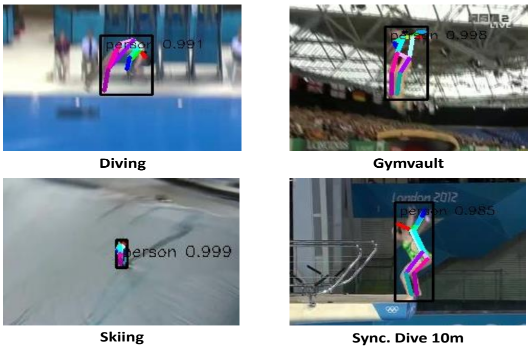
For computing the Difference Feature of a joint i, we first compute the motion differences between joint i and each of its neighbours j. The motion differences are attended by weights in $A_s(i, j)$ and $A_p(i, j)$. Then the joint i aggregates the motion differences within its neighbourhood with weighted sum. The aggregated neighbourhood differences for all joints at the present time step t form the matrices D_s^t and D_p^t. An illustration of the construction of D_s^t is shown in Figure 4. The computation of D_s^t and D_p^t can be written as:

$$\begin{gathered} D_s^t(i, m) = \sum_j (A_s(i, j) \cdot (F^t(i, m) - F^t(j, m))) \cdot w_j, \\ D_p^t(i, m) = \sum_j (A_p(i, j) \cdot (F^t(i, m) - F^{t-1}(j, m))) \cdot w_j, \\ 1 \leq i, j \leq J, 1 \leq m \leq M, \end{gathered} \tag{3}$$

where F^{t-1} is the joint features at time step $t - 1$ (the predecessor), and F^t is those at time step t (the predecessor). We use $F^t(i, m)$ to denote the m^{th} dimension of the joint feature of the i^{th} joint, which is a real number. Again, J is the total number of joints, and M is the dimension of the joint features. The weight in the neighbourhood aggregation is denoted as w_j, which is learnable and represents the influence joint j has on others for action assessment.

Then the aggregated motion differences for each joint are fused by mean pooling to form *the Difference Features* $\bar{d}_s^t$ *and* $\bar{d}_p^t$; that is $\bar{d}_s^t$ can be written as:

$$\bar{d}_s^t = \frac{1}{N}({D_s^t}^T \cdot \mathbf{1}), \tag{4}$$



Diving

Gymvault

Skiing

Sync. Dive 10m

Figure 5. Examples of the pose estimation results.

where $\bar{d}_k^t \in \mathbb{R}^M$ is the Difference Feature for time step t. The computation of $\bar{d}_p^t$ is the same as that of $\bar{d}_s^t$.

3.3. Assessing the performance

Regression Module. In the following, we introduce a Regression Module for computing assessment results. The features input to the Regression Module are *the whole-scene video feature* $q^t \in \mathbb{R}^M$, the Commonality Features ($\bar{h}_c^t$) and the Difference Features ($\bar{d}_s^t$ and $\bar{d}_p^t$). We take use of the whole-scene feature, because the athlete's position in the scene is also necessary for action assessment.

More specifically, firstly the features are encoded by into *feature encoders*. An input feature u_i^t can be q^t, $\bar{h}_0^t$, $\bar{h}_1^t$, $\bar{d}_s^t$ or $\bar{d}_p^t$. The encoding process is written as

$$\hat{u}_i^t = C_i(u_i^t), \tag{5}$$

where C_i is an encoding function and $\hat{u}_i^t$ is the corresponding encoded feature.

Secondly, they are aggregated by a *feature pooling layer* to form an overall feature v^t,

$$v^t = \sum_i \alpha_i \cdot \hat{u}_i^t + \beta_i, \tag{6}$$

where α_i is the scalar for a feature $\hat{u}_i^t$ and β_i is the corresponding bias.

In order to reduce the redundancy between different features, we apply an orthogonal regularization term in the feature pooling layer during training, as in [31]. The orthogonal regularization is written as

$$R_O = \sum_{i,j} \gamma \cdot (\hat{u_i^t}^T \cdot \hat{u}_j^t), \tag{7}$$

where $(\hat{u}_i^t, \hat{u}_j^t)$ is a pair of features to be orthogonalized, and γ is a predefined coefficient.

Finally we gain the assessment results with two fully connected layers. The overall assessment result is given by:

$$s = \sum_t S(v^t), \tag{8}$$

	Diving	Gymvault	Skiing	Snowboard	Sync. 3m	Sync. 10m	Avg. Corr.
Pose+DCT [24]	0.5300	–	–	–	–	–	–
ST-GCN [33]	0.3286	0.5770	0.1681	0.1234	0.6600	0.6483	0.4433
C3D-LSTM [21]	0.6047	0.5636	0.4593	0.5029	0.7912	0.6927	0.6165
C3D-SVR [21]	**0.7902**	0.6824	0.5209	0.4006	0.5937	0.9120	0.6937
Whole Scene	0.6339	0.6872	0.5179	0.5053	0.8783	0.8832	0.7226
Whole+Patch	0.7043	0.6758	0.5783	0.4547	0.8547	0.8766	0.7229
Ours	0.7630	**0.7358**	**0.6006**	**0.5405**	**0.9013**	**0.9254**	**0.7849**

Table 1. The results of our model in comparison with state-of-the-art methods and our baselines. Our model achieves state-of-the-art performance, and it outperforms the baselines in each of the six actions.

	Diving	Gymvault	Skiing	Snowboard	Sync. 3m	Sync. 10m	Avg. Corr.
Ours(Full)	**0.7630**	**0.7358**	0.6006	**0.5405**	**0.9013**	**0.9254**	**0.7849**
w/o Commonality	0.7020	0.7166	0.5222	0.5117	0.8632	0.9073	0.7423
w/o Difference	0.7469	0.7007	**0.6191**	0.4968	0.8651	0.8764	0.7455
w/o Spatial Relation	0.7193	0.6512	0.5059	0.4752	0.8725	0.8813	0.7229
w/o Temporal Relation	0.7273	0.6490	0.5186	0.5203	0.8824	0.9049	0.7423
w/o Feature Pooling	0.7288	0.7349	0.5504	0.4528	0.8640	0.9032	0.7451
w/o Feature Encoders	0.6504	0.6755	0.3088	0.3293	0.8421	0.8268	0.6512
Whole-scene (Baseline)	0.6339	0.6872	0.5179	0.5053	0.8783	0.8832	0.7226

Table 2. An ablation study showing the contributions of each component in our model. Both the Joint Commonality Module and the Joint Difference Module contribute to the model performance. The feature encoders and the feature pooling layer are necessary in our methods for fusing the learned features.

where $S(\cdot)$ is the regression function, and $s \in \mathbb{R}$ is the assessment result for a video. The video is divided into t segments, and for each segment a partial assessment result is given.

Optimization. We utilize the MSE Loss during training, together with the orthogonal regularization term (with a weight 0.8) and L2 regularization terms (with a weight 0.1) on the relation graphs.

4. Experiment

We first describe the implementation details of our model, and then we present the assessment results on six Olympic actions alongside baselines and analyze the contributions of each module with an ablation study. We also explore the robustness of our model to pose estimation methods and extend our methods to egocentric surgical tasks. Finally we present qualitative results of our method.

4.1. Implementation Details

Data Preprocessing. We extract human poses and bounding boxes with pose estimation method based on Mask-RCNN [12]. Examples of pose estimation results are shown in Figure 5. We utilize I3D pre-trained on Kinetics [2] to extract the joint features of RGB and optical flow(obtained by TV-L1 algorithm [22]). The whole-scene features are obtained with whole images, while the joint motion features are obtained with local patches cropped around joints. We divide the videos into 10 segments, and 16 frames are uniformly sampled out in each segment as the input to the I3D

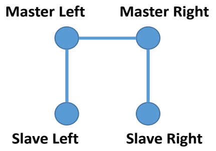


Figure 6. The relation graph for surgical actions.

Baseline	Ours:AlphaPose	Ours:Mask-RCNN
0.6339	**0.7558**	**0.7630**

Table 3. Robustness to pose estimation methods. Our model has performance gains on both pose estimation methods. The results show that our model is robust to the pose estimation methods.

	Suturing	Needle Passing	Knot Tying	Avg. Corr.
ST-GCN [33]	0.31	0.39	0.58	0.43
TSN [6]	0.34	0.23	0.72	0.46
Whole-scene	0.09	0.10	0.15	0.11
Joint Motion	0.17	0.25	0.55	0.34
Whole + Joint	0.17	0.37	0.73	0.46
Ours	**0.36**	**0.54**	**0.75**	**0.57**

Table 4. The results of our model in comparison with state-of-the-art methods and our baselines on the JIGSAWS Dataset through four-fold cross-validation.

network. We augment the videos by left-right flipping. The ground-truth scores published by the Olympic Committee are normalized to 0-100 as the supervision of our assessment model. More details can be found in our supplementary material.

Model Training. All models are trained using Adam Optimizer with a batch size of 64. We utilize cyclic learning rate [27] of $\{$1e-4, 1e-5 and 1e-6$\}$, changing every 500 iterations, starting from 1e-4. In implementation, C_i is an FC layer of shape 400×512 with ReLU activation. Then the first FC in the Scoring Module is of shape 512×128, with ReLU activation, and the second FC is a linear layer of shape 128×1. γ is set as 0.8. For stable training, we optimize the weights in multiple stages. Firstly, we separately train the baseline branch (Orange in Figure 2), the Joint Difference Module (Blue in Figure 2) and the Joint Commonality Module based on the A_s learned so far. In this stage, the feature pooling layer is not included and the Regression Block is not saved. All branches are trained for 4500 iterations. Then we train the full model based on the weights from the previous stage; the feature pooling layer is included, and the Regression Block is re-initialized. The second stage runs for 3000 iterations, while the loaded weights are fixed in the first 500 iterations. For fair comparison, we load weights of the 1500-iteration whole-scene checkpoints in this process. And we compare our final model (+3000 step) with the 4500-iteration whole-scene model and the 4500-iteration Whole+Patch model, which is denoted as the baselines in Table 1.

4.2. Quantitative Results

The Olympic Actions. We evaluate our assessment method on six actions of *the AQA-7 Dataset* [20] all collected from Summer and Winter Olympics, containing 1106 videos in total. We follow the experimental settings of [20] and exclude the trampoline, whose annotations are not released yet. The Diving was first collected by Pirsiavash [24] and then extended by [21]. The Gymvault was collected by [21], while the other four actions were first published in [20]. We followed the training and testing split in [20].

Evaluation Metric. For consistence with existing literature [24, 21, 20], the Spearman's rank correlation(ranging from -1 to 1, the higher the better), which shows the ranking correlations between two series, is used to evaluate the correlation between the predicted and ground-truth assessment results. It is defined as $\rho = \frac{\sum_i (p_i-\bar{p})(q_i-\bar{q})}{\sqrt{\sum_i (p_i-\bar{p})^2 \sum_i (q_i-\bar{q})^2}}$, where p and q indicate the ranking of the two series respectively. The average Spearman's Rank Correlation across actions is computed from individual per-action correlations using Fisher's z-value as in [20].

Comparison with state-of-the-art methods and our baselines. Table 1 shows the results of our model on the six Olympic actions, in comparison with state-of-the-art methods. We use the method proposed in [24, 21] as the current state-of-the-art performance for action assessment on the six actions. As Pirsiavash [24] performed experiments only on the diving action, we only present their results in Diving. Our proposal outperforms all state-of-the-art methods and baselines. Compared with the C3D-SVR method [21], our model achieves better performance in all but Diving, with an improvement of 0.0912 on average. The success of our model is partially because our method is based on the successful I3D [2] model in video feature extraction, and also partially because the use of our graph modelling for assessing the action performance. Therefore, we also evaluate two baselines using the I3D for video feature extraction without using our graph-based modelling, i.e. one uses only the I3D features for the whole scene to assess the action performance, and the other one using both the whole-scene and the local-patch features with mean pooling. Our method achieves better performance on each action as compared to the two baselines, and the results show that our graph-based joint motion learning provides significant improvement for action assessment. The proposed method also outperforms the ST-GCN [33] (the classification layer replaced with our Regression Module), showing the effectiveness of our graph modelling for action assessment.

Ablation Study. In Table 2 we present the results of a per-task ablation study. We evaluate the individual contributions of the Joint Commonality Module, the Joint Difference Module, the feature pooling layer and the feature encoders. We try to remove each of the component from

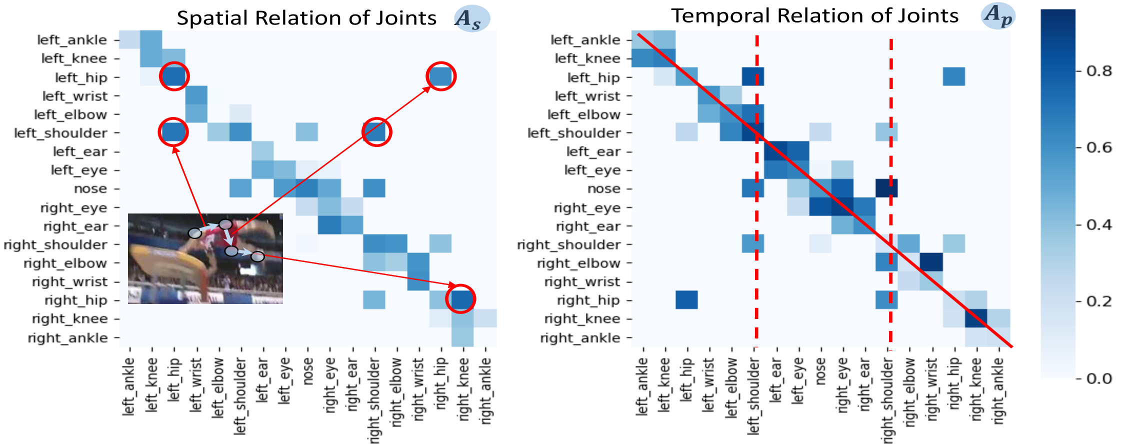


Figure 7. Visualizing the learned relation graphs for Gymvault. We visualize the adjacent matrix of the Spatial Relation Graph (on the left), and that of the Temporal Relation Graph (on the right). As shown, within a time step, our assessment model for Gymvault attaches great importance to the relations among some key joints, such as hips, shoulders, and knees. And our model pays much attention to the motion of the same joint and the shoulders from the previous time step.

our full model. On average, removing each of the component results in a fall in the model performance. Removing the Joint Commonality Module, the performance drop of 0.0426 on average. And the performance also drops about 0.04 if we remove the Joint Difference Module. This shows that both the Joint Commonality Module and the Joint Difference Module contribute to the performance of our model. Still, there are a few exceptions in some actions. In Skiing, only using the Joint Commonality Module already achieves a good result, because the assessment of skiing focuses more on the general motion of body parts rather than the motion differences, since all body part motion has similar high speed. What's more, the results drop when we remove the Spatial Relation(-0.062) or the Temporal Relation(-0.0426). This shows that both spatial and temporal relations are vital for action assessment. Also, removing the feature pooling layer and the feature encoders causes a drop of over 0.039 in the performance. Features from each branches have different distributions. They need to be mapped to a shared space at feature learning. The Encoders and the feature pooling layer are both necessary in our assessment framework.

Robustness to Pose Estimation Methods. Apart from the pose estimation methods based on Mask-RCNN [12], we also evaluate our model on the Diving based on AlphaPose [8]. The evaluation results of our model based on both pose estimation methods are shown in Table 3. As we can see, our method has improvement gains on both pose estimation methods. With AlphaPose, our method has a gain of 0.1219 on the correlation. With Mask-RCNN, our method has a performance gain of 0.1291. This shows that our method has robustness to both pose estimation methods.

Extension to Egocentric Surgical Videos. We additionally evaluate our model on *the JIGSAWS Dataset* [9], which contains egocentric videos on three surgical activities. The JIGSAWS Dataset contains stereo recordings captured by left and right cameras, and we consider both left-view and right-view recordings as individual samples. We regard the master tool manipulators (Masters) and the patient-side manipulators (Slaves) as nodes and extract DCT of the 3D kinetics as local features. Figure 6 shows the relation graphs for the surgical actions. We perform four-fold cross-validation as done in [6]. The results are shown in Table 4 and again demonstrates the better performance achieved by our model.

4.3. Qualitative Results

The Relation Graphs. In Figure 7 we visualize A_s and A_p for the relation graphs G_s and G_p learned by our model on the Gymvault. From the learned A_s, we can see that our model pays more attention to the detailed motion among shoulders, hips and knees. This is not a surprising result, as they are traditional vital key-points in action analysis.

From the learned A_p, we can see the positions around the principal diagonal (the solid line) is in deeper color. We can see that our model attaches great importance to motion of the same joint from the previous time step.This is consistent with our perception, that judging the performance of an action depends highly on the previous motion of the same joints. And our model also attaches great importance to the shoulder's previous motion (the dashed lines) when analyzing related joints, because shoulders are important anchors in action analysis.

The results are very interesting, as they indicate how we

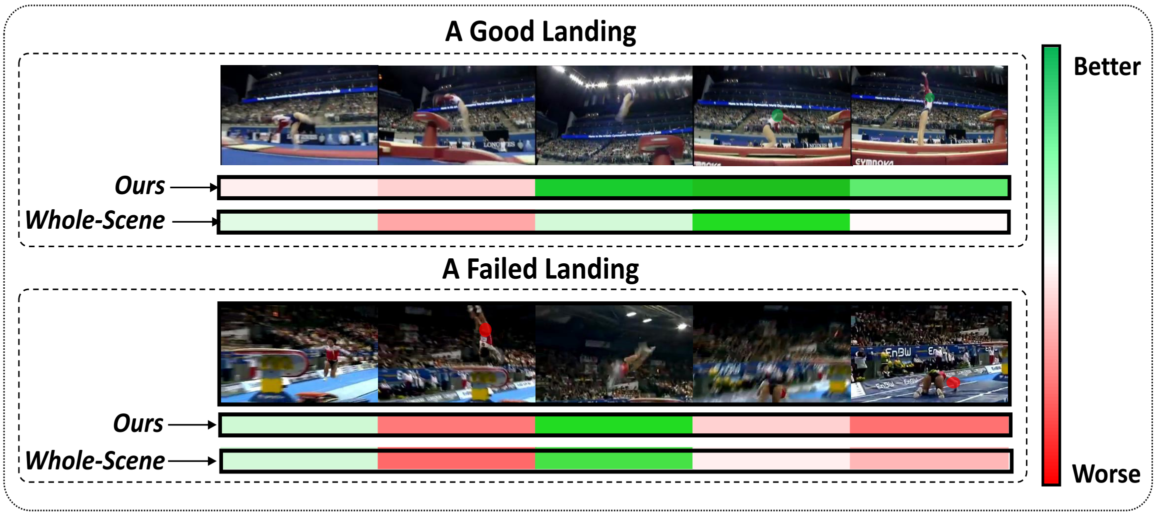


Figure 8. The action assessment results of our model on two Gymvault cases together with those of the whole-scene baseline. The assessment results of our model indicate good action performance (green) and flaws (red). Our model further gives assessment results on joints, showing joints with best and worst performance. Best view in color.

assess the present joint motion. When we evaluate the motion of each joint, we attach importance to the synchronous motion of several anchor joints, as well as the previous state of the same joint and the shoulders.

Fine-grained Assessment Results. In Figure 8, we show the assessment results of our model in comparison with those of the whole-scene baseline. We present the action assessment results of our model on two Gymvault cases. And we also show joints with highest and lowest scores. We obtain scores for a certain joint by retaining only the features of that joint at the Commonality and the Difference features, instead of pooling features of all joints together (Equation 2 and Equation 4).

We can see that the athlete above didn't perform well at first, but he successfully landed with a beautiful pose. Here, our model not only gives a scores throughout the landing process, but also successfully detects the joints with best performance. At the last two time steps, our model gives highest scores to the the left-shoulder. The whole-scene baseline, on the contrary, fails to recognize the good ending pose at the last time step. This is because the whole-scene method pays little attention to the human posture. In the second case, the athlete fell down at her landing. She also performed poorly at the second time step. This is also detected by both our model and the whole-scene baseline. At the second time step, our model gives the lowest score to the unstable right-hip. At the finishing posture, our model gives the lowest score to the right-elbow which bends abnormally.

5. Conclusion

In this paper we have presented a new model to assess the action performance through graph-based joint relation modelling. We build joint trainable joint relation graphs, and analyze joint motion on them. We propose two novel modules, the Joint Commonality Module and the Joint Difference Module for joint motion learning within body parts. The proposed method achieves state-of-the-art results for action performance assessment on Olympic actions, and can help to interpret the action assessment process.

Acknowledgement

This work was supported partially by the National Key Research and Development Program of China (2018YFB1004903), NSFC(U1611461), Guangdong Province Science and Technology Innovation Leading Talents (2016TX03X157), Guangzhou Research Project (201902010037) and the Royal Society Newton Advanced Fellowship (NA150459). We thank Hong-Xing Yu for useful feedback and suggestions.

References

[1] Gedas Bertasius, Hyun Soo Park, Stella X Yu, and Jianbo Shi. Am i a baller? basketball performance assessment from first-person videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2177–2185, 2017.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[3] Oya Çeliktutan, Ceyhun Burak Akgul, Christian Wolf, and Bülent Sankur. Graph-based analysis of physical exercise actions. In *Proceedings of the 1st ACM international workshop on Multimedia indexing and information retrieval for healthcare*, pages 23–32. ACM, 2013.

[4] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7239–7248, 2018.

[5] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3076–3086, 2017.

[6] Hazel Doughty, Dima Damen, and Walterio Mayol-Cuevas. Whoś better, whoś best: Skill determination in video using deep ranking. *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[7] Hazel Doughty, Walterio Mayol-Cuevas, and Dima Damen. The pros and cons: Rank-aware temporal attention for skill determination in long videos. *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[8] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017.

[9] Yixin Gao, S Swaroop Vedula, Carol E Reiley, Narges Ahmidi, Balakrishnan Varadarajan, Henry C Lin, Lingling Tao, Luca Zappella, Benjamın Béjar, David 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*, volume 3, page 3, 2014.

[10] Srujana Gattupalli, Dylan Ebert, Michalis Papakostas, Fillia Makedon, and Vassilis Athitsos. Cognilearn: A deep learning-based interface for cognitive behavior assessment. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 577–587. ACM, 2017.

[11] Andrew S Gordon. Automated video assessment of human performance. In *Proceedings of AI-ED*, pages 16–19, 1995.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[13] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.

[14] Marko Jug, Janez Perš, Branko Dežman, and Stanislav Kovačič. Trajectory based assessment of coordinated human activity. In *International Conference on Computer Vision Systems*, pages 534–543. Springer, 2003.

[15] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. Learning clip representations for skeleton-based 3d action recognition. *IEEE Transactions on Image Processing*, 27(6):2842–2855, 2018.

[16] Matej Kristan and Stanislav Kovačič. Automatic evaluation of organized basketball activity using bayesian networks. 2007.

[17] Jun Liu, Amir Shahroudy, Dong Xu, Alex C Kot, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3007–3021, 2018.

[18] Anand Malpani, S Swaroop Vedula, Chi Chiung Grace Chen, and Gregory D Hager. Pairwise comparison-based objective score for automated skill assessment of segments in a surgical task. In *International Conference on Information Processing in Computer-Assisted Interventions*, pages 138–147. Springer, 2014.

[19] Adeline Paiement, Lili Tao, Sion Hannuna, Massimo Camplani, Dima Damen, and Majid Mirmehdi. Online quality assessment of human movement from skeleton data. In *British Machine Vision Conference*, pages 153–166. BMVA press, 2014.

[20] Paritosh Parmar and Brendan Tran Morris. Action quality assessment across multiple actions. *arXiv preprint arXiv:1812.06367*, 2018.

[21] Paritosh Parmar and Brendan Tran Morris. Learning to score olympic events. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–28, 2017.

[22] Javier Sánchez Pérez, Enric Meinhardt-Llopis, and Gabriele Facciolo. Tv-l1 optical flow estimation. *Image Processing On Line*, pages 137–150, 2013.

[23] Huy-Hieu Pham, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, and Sergio A Velastin. Learning to recognize 3d human action from a new skeleton-based representation using deep convolutional neural networks. *arXiv preprint arXiv:1812.10550*, 2018.

[24] Hamed Pirsiavash, Carl Vondrick, and Antonio Torralba. Assessing the quality of actions. In *European Conference on Computer Vision*, pages 556–571. Springer, 2014.

[25] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[26] Yachna Sharma, Vinay Bettadapura, Thomas Plötz, Nils Hammerla, Sebastian Mellor, Roisin McNaney, Patrick Olivier, Sandeep Deshmukh, Andrew McCaskie, and Irfan Essa. Video based assessment of osats using sequential motion textures. Georgia Institute of Technology, 2014.

[27] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.

[28] Kalpit Thakkar and PJ Narayanan. Part-based graph convolutional network for action recognition. In *British Machine Vision Conference*, 2018.

[29] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 399–417, 2018.

[30] Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video. In *Advances in neural information processing systems*, pages 4539–4547, 2017.

[31] Wangmeng Xiang, Jianqiang Huang, Xianbiao Qi, Xiansheng Hua, and Lei Zhang. Homocentric hypersphere feature embedding for person re-identification. *arXiv preprint arXiv:1804.08866*, 2018.

[32] Chengming Xu, Yanwei Fu, Bing Zhang, Zitian Chen, Yu-Gang Jiang, and Xiangyang Xue. Learning to score the figure skating sports videos. *arXiv preprint arXiv:1802.02774*, 2018.

[33] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[34] Zhengyuan Yang, Yuncheng Li, Jianchao Yang, and Jiebo Luo. Action recognition with spatio-temporal visual attention on skeleton image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[35] Bangpeng Yao and Li Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1691–1703, 2012.

[36] Qiang Zhang and Baoxin Li. Video-based motion expertise analysis in simulation-based surgical training using hierarchical dirichlet process hidden markov model. In *Proceedings of the 2011 international ACM workshop on Medical multimedia analysis and retrieval*, pages 19–24. ACM, 2011.

[37] Qiang Zhang and Baoxin Li. Relative hidden markov models for video-based evaluation of motion skills in surgical training. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1206–1218, 2015.

[38] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018.

[39] Aneeq Zia, Yachna Sharma, Vinay Bettadapura, Eric L Sarin, Mark A Clements, and Irfan Essa. Automated assessment of surgical skills using frequency analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 430–438. Springer, 2015.

[40] Aneeq Zia, Yachna Sharma, Vinay Bettadapura, Eric L Sarin, and Irfan Essa. Video and accelerometer-based motion analysis for automated surgical skills assessment. *International journal of computer assisted radiology and surgery*, 13(3):443–455, 2018.

[41] Aneeq Zia, Yachna Sharma, Vinay Bettadapura, Eric L Sarin, Thomas Ploetz, Mark A Clements, and Irfan Essa. Automated video-based assessment of surgical skills for training and evaluation in medical schools. *International journal of computer assisted radiology and surgery*, 11(9):1623–1636, 2016.