International Journal of Computer Vision (2023) 131:659–679
https://doi.org/10.1007/s11263-022-01695-5

Automatic Modelling for Interactive Action Assessment

Jibin Gao[1] · Jia-Hui Pan[1] · Shao-Jie Zhang[1] · Wei-Shi Zheng[1,2]

Received: 30 August 2021 / Accepted: 24 September 2022 / Published online: 10 December 2022

Abstract
Action assessment, the task of visually assessing the quality of performing an action, has attracted much attention in recent years, with promising applications in areas such as medical treatment and sporting events. However, most existing methods of action assessment mainly target the actions performed by a single person; in particular, they neglect the asymmetric relations among agents (e.g., between persons and objects), limiting their performance in many nonindividual actions. In this work, we formulate a framework for modelling asymmetric interactions among agents for action assessment, considering the subordinations among agents in many interactive actions. Specifically, we propose an asymmetric interaction learner consisting of an automatic assigner and an asymmetric interaction network search module. The automatic assigner is designed to automatically group agents within an action into a primary agent (e.g., human) and secondary agents (e.g., objects); the asymmetric interaction network search module adaptively learns the asymmetric interactions between these agents. We conduct experiments on the *JIGSAWS* dataset containing surgical actions and additionally collect two new datasets, *TASD-2* and *PaSk*, for action assessment on interactive sporting actions. The experimental results on these three datasets demonstrate the effectiveness of our framework in achieving state-of-the-art performance. The extensive experiments on the *AQA-7* dataset also indicate the robustness of our model in conventional action assessment settings.

Keywords Action assessment · Interactive action · Video understanding

1 Introduction

Action assessment aims to assess the quality of an action from visual features of a video (Doughty et al., 2018; Malpani et al., 2014; Ilg et al., 2003; Parmar and Tran Morris, 2019; Bertasius et al., 2017; Tang et al., 2020; Gao et al., 2020; Zeng et al., 2020) and it has drawn increasing attention in recent decades. There are many practical scenarios where action assessment may play a promising role in assisting humans in making decisions. For example, in sports, action assessment models can be designed to help referees score sporting events and to assist athletes in training (Parmar and Tran Morris, 2019; Pirsiavash et al., 2014; Bertasius et al., 2017; Parmar and Morris, 2019; Gao et al., 2020; Tang et al., 2020; Zeng et al., 2020). Referees could show evidence by utilizing feedback from the action assessment model, while with comprehensive feedback, athletes could make reasonable corrections to their motions. In addition, in modern medical treatment, especially in rehabilitation treatment, action assessment models could be deployed in the rehabilitation training of patients (Malpani et al., 2014; Sharma et al., 2014; Zhang and Li, 2011). An assessment report of a patient's daily rehabilitation training can be generated, and the doctor can give corresponding follow-up treatment suggestions based on the report.

Most existing methods for action assessment (Pirsiavash et al., 2014; Pan et al., 2019; Parmar and Tran Morris, 2017; Tang et al., 2020; Liu et al., 2021) have been designed for individual actions performed by a single person (e.g., diving and vaulting). However, many nonindividual actions can be observed in our daily life. Such actions are always con-

Communicated by Dima Damen.

✉ Wei-Shi Zheng
zhwshi@mail.sysu.edu.cn

Jibin Gao
gaojb5@mail2.sysu.edu.cn

Jia-Hui Pan
panjh7@mail2.sysu.edu.cn

Shao-Jie Zhang
zhangshj56@mail2.sysu.edu.cn

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China

[2] Peng Cheng Laboratory, Shenzhen 518055, China

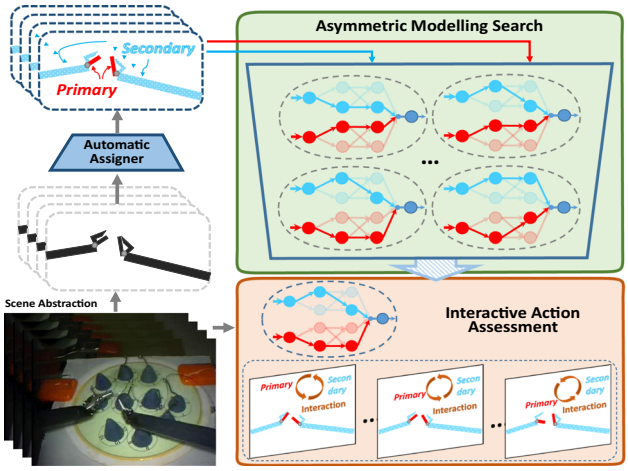

Fig. 1 Our asymmetric interaction learner module is designed to assess action performance. For egocentric surgical videos, we regard the motions of the master tool manipulator as the primary agent (in red) and those of the patient-side tool manipulators and handles, which are relatively inactive, as the secondary agent (in blue) (Color figure online)

structed by interaction, and in particular, there exists direct or indirect subordination between agents (e.g., humans and objects) in an interaction case. For instance, as shown in Fig. 1, the egocentric surgical action (Gao et al., 2014; Liu et al., 2021) involves four agents, (i.e., the master tool manipulators and the patient-side tool manipulators of both left and right sides). In this case, the interactions between these four agents should be explicitly modelled for the performance assessment. In addition, in pair figure skating, two players cooperate to complete the action. Thus, accordingly, the actions involving interactions between the two players should be modeled directly. Moreover, it is obvious that the interactions mentioned above are asymmetric in terms of agents' roles, which can be naturally grouped into the primary agent and the secondary agents. The primary is the predominant agent in an interactive action, and the secondaries are relatively inactive but have latent interactions with the primary. The skills of the primary agents are often deterministic to the performance score, and the secondary agents help by supporting or interacting with the primary. Although many existing works, such as Pan et al. (2019), are employed to address the performance assessment of interactive actions, they consider all agents equally, without explicitly modelling the asymmetric subordination between agents (e.g., between humans and objects).

In this work, we propose a new framework to automatically model asymmetric action interactions for human action assessment. Specifically, we develop an asymmetric interaction learner that models the interactions among multiple action agents (e.g., humans and objects) for action assessment. Our asymmetric interaction learner consists of two parts: an automatic assigner and an asymmetric interaction network search module. The automatic assigner recognizes the agents as primary (e.g., a human) and secondary (e.g., objects) automatically, and the asymmetric interaction network search module develops operations for interactive learning automatically. In the automatic assigner, we develop a trainable indicator matrix that learns to compute the probabilities of each agent being the primary agent. The asymmetric interaction network search module first constructs an interaction network prototype to exploit the interactions between the *primary* and the *secondary* in the latent space. Specifically, the prototype consists of a transformation module to transform the feature subspace, a difference module to measure the motion difference between the primary and the secondary agents, and a temporal interaction module to perform temporal fusion. In particular, the operations of the three modules can be learned automatically with a network search strategy and instantiated for various action tasks. Finally, we construct an attentive assessor to learn the contextual interaction between the asymmetrical interaction feature and the whole-scene feature to obtain the performance score of an action.

In addition, our method can not only assess interactive actions in cases of strong subordination (primary-secondary relations) among the parts but can also be employed for interactive actions whose agents are in weak primary-secondary or equal relations, such as synchronous sports (e.g., synchronized diving). To better adapt our model to the interactive action assessment with multiple criteria (e.g., both execution and synchronization scores), we generalize our model with multi-task learning.

To the best of our knowledge, there are few datasets strictly for action assessment of interactive actions other than *JIGSAWS* (Gao et al., 2014). Therefore, we additionally collected two new datasets, named the Two-person Action Synchronized Diving dataset (*TASD-2*) and the Pair Skating dataset (*PaSk*), for evaluating asymmetrically interactive actions.

In summary, our contributions to this work are fourfold:

1. An automatic assigner is proposed to assign the agents of any action as the primary and the secondary agents automatically.
2. An asymmetric interaction network search module, which can adaptively search proper operations, is built to compose the asymmetric interaction module for various actions.
3. A general framework for interactive action assessment is constructed that can be readily generalized to various kinds of action assessment tasks.

4. Two new datasets, called *TASD-2* and *PaSk*, are collected in our work, as there are few datasets strictly used for action assessment of interactive actions.

In the experimental section, we report the experimental results of the comparisons and ablation studies to validate the effectiveness of our proposed method, and the results also demonstrate the superiority of our framework on the actions in both strong and weak asymmetric relations among different agents.

2 Related Work

2.1 Action Quality Assessment

Action quality assessment, also known as skill assessment or skill determination, evaluates how well an action is performed. Existing works on action quality assessment are divided into three branches or categories: (1) some regard the task as a classification problem in which action executions are sorted into various skill levels (e.g., expert, intermediate and novice) (Zia et al., 2016, 2018); (2) some regard the task as a regression problem that predicts an exact performance score for each action (Pirsiavash et al., 2014; Xu et al., 2018; Parmar and Tran Morris, 2017, 2019; Zia & Essa, 2018; Zeng et al., 2020; Liu et al., 2021); and (3) some regard it as a pairwise ranking task that ranks the actions pair-by-pair according to their performance skills (e.g., action X is better than Y) (Doughty et al., 2018, 2019; Bertasius et al., 2017; Zhang & Li, 2015). To consider more fine-grained assessment, our work follows the second branch. However, few existing works have assessed action performance by explicitly exploiting the interaction in actions; in particular, modelling asymmetric interactions for assessment has been neglected. To learn the interaction among the joints of performers' skeletons, Pan et al. (2019) assessed action performance based on joint relation modelling by a graph neural network (Scarselli et al., 2009). Unlike (Pan et al., 2019), we explore action interaction with asymmetric modelling (i.e., we treat the primary and secondary nonequally, while JRG (Pan et al., 2019) treats them equally). Nevertheless, many existing methods can be employed to model actions in joint-based interactions but overlook subordination modelling in asymmetric interactions. Moreover, our framework provides automatic and adaptive modelling for asymmetric interactions, which existing works have not explored.

2.2 Group Activity Recognition

Group activity recognition is the task of recognizing an activity performed by a group of people. Different from single-person action recognition, group activity recognition requires interaction modelling among multiple moving agents. Recently, with the rapid development of deep learning in video analysis, some deep models (Chang et al., 2015; Wang et al., 2017; Shu et al., 2017; Yan et al., 2018a; Zhang et al., 2019; Lu et al., 2019; Azar et al., 2019; Wu et al., 2019) have been proposed to address this problem. Some works (Wang et al., 2017; Shu et al., 2017) introduce using temporal modelling for individual-level actions and extract the group-level feature with simple pooling functions. To further explore the impact of key persons on a group activity, some recent methods propose deep relational modelling (Azar et al., 2019; Wu et al., 2019) and attention modelling (Yan et al., 2018a; Lu et al., 2019) to model the interactions between persons in a group activity. Similar to the group activity recognition task, our asymmetric modelling targets actions with multiple agents. However, we focus on the task of interactive action assessment to evaluate how well the action is performed with interactions. Therefore, we focus more on the motion fluency and coordination among the agents in interactions instead of the type of action.

2.3 Network Architecture Search

Network architecture search (NAS), the task of searching optimized network architectures for specific tasks, has attracted increasing interest in recent years. Existing works (Pham et al., 2018; Liu et al., 2018; Guo et al., 2019; Dong and Yang, 2019; Hu et al., 2020; Xie et al., 2018; Cai et al., 2018) have demonstrated that NAS automatically designs effective neural networks for various kinds of tasks. Most of the methods (Pham et al., 2018; Liu et al., 2018; Guo et al., 2019; Dong and Yang, 2019; Hu et al., 2020; Xie et al., 2018; Cai et al., 2018) formulate the NAS by stacking neural networks with several repeated blocks, and the search space for these blocks is viewed as a directed acyclic graph, where the edges between the computational nodes represent candidate operations (e.g., max-pooling and convolution). Pham et al. (2018) used a recurrent neural network (RNN) to construct the controller for the network architecture search, which determined the candidate path between two nodes in the search space that would form the final optimal neural network. Liu et al. (2018) built a differentiable search model called DARTS by parameterizing each candidate edge of the network architecture graph. In DARTS, after the search phase, the final model is determined according to the greatest possibilities represented by the parameters of the edges in the architecture. Inspired by DARTS, Guo et al. (2019) designed a supernet, a tiny search space similar to that of DARTS, in which they sampled only one candidate edge for each node pair to train all possible models; thus, compared to DARTS, their method more tightly constrained the models in the search process and ensured that the searched model was contained in the search space. In addition, many methods (Dong and Yang, 2019; Hu

et al., 2020; Xie et al., 2018; Cai et al., 2018) have attempted to approximate the architecture parameters of node pairs as one-hot vectors.

Since agents are different kinds of entities(e.g., humans, objects) and the representations of agents are various (e.g., kinetic feature, pose), adaptive modelling for asymmetric interactions is needed, which learns to search for a proper network for asymmetric interaction modelling rather than using a specific manually designed network. Inspired by the differentiable network architecture search (Liu et al., 2018), we use the search mechanism to choose the proper operation for each module in the progressive asymmetric interaction learning network for various actions automatically.

In our preliminary work, we attempt to form an asymmetric modelling for action assessment (Gao et al., 2020). By comparison, there are several significant differences. (1) The current framework can learn an automatic assigner to assign the agents of any action as the primary agents and the secondary agents automatically, while our preliminary version (Gao et al., 2020) requires manually assigning them according to the empirical observations. (2) In contrast to the asymmetric interaction module in our preliminary work (Gao et al., 2020), we build an asymmetric interaction network search module that can adaptively search proper operations to compose the asymmetric interaction module for various actions. (3) A new dataset, called *PaSk*, is collected to evaluate our framework, and the results demonstrate the superiority of our framework on the actions in strong asymmetric relations. Consequently, the current framework is more feasible and general for different kinds of actions and gains state-of-the-art performances on three datasets, in particular, with remarkable improvements of 0.17 on *JIGSAWS*, compared to our preliminary version (Gao et al., 2020).

3 Approach

In this section, we introduce our framework for automatically modelling asymmetric interaction actions in detail. Intuitively, actions can be defined by the interaction of agents. In our work, we model interactive agents asymmetrically. This is because asymmetric modelling on agents can explicitly explore interactions among agents whether they are in asymmetric interactions or equal relations since agents in equal relations can be considered weak asymmetric (a particular case of asymmetric modelling). Moreover, for the complicated scenarios of asymmetric interactions among different kinds of entities (e.g., humans, objects), we consider adaptive modelling for asymmetric interaction to better search for a proper network for each interactive action instead of using a manually designed network for all interactions. The overall framework of our model is presented in Fig. 2, with a video sample segmented into T clips. In this structure, there are two main components, the asymmetric interaction learner and the attentive assessor for action assessment. For the asymmetric interaction learner, an automatic assigner and a progressive search module are constructed. The automatic assigner recognizes the primary and the secondary agents;

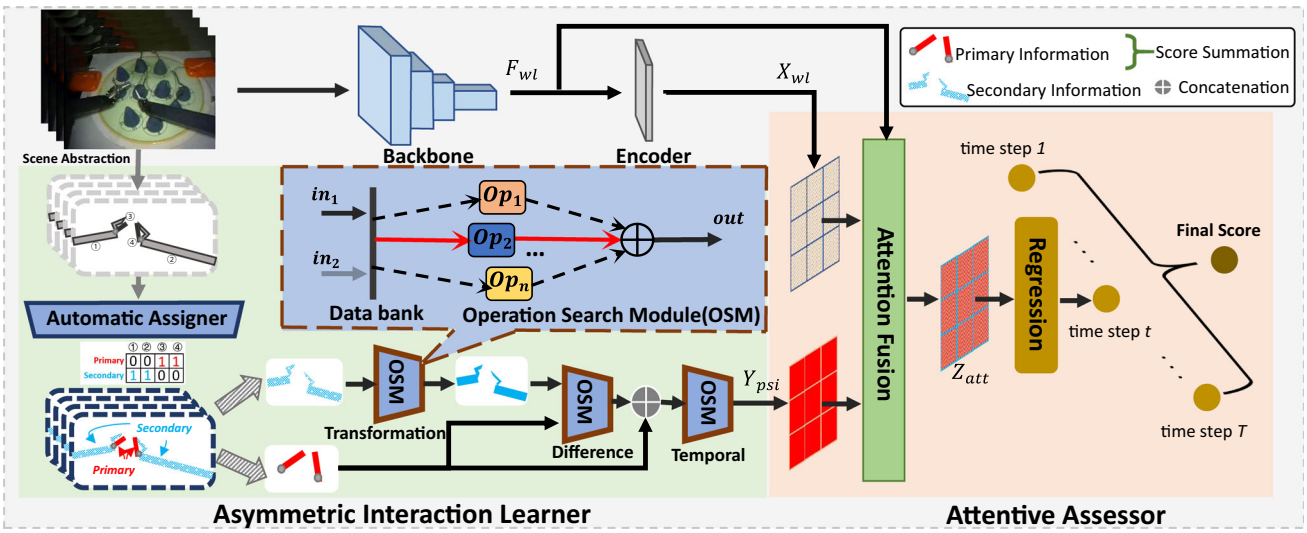

Fig. 2 An overview of our proposal. We uniformly divide an input video into T time steps and present the process of the asymmetric interaction learner module in a clear manner at time step t. The kinetic information of mobile objects is extracted and passed through an automatic assigner, and they are categorized into primary and secondary objects. We perform asymmetric interaction between the primary and secondary objects and obtain the asymmetric interaction feature. Then, we perform attentive contextual interaction between the whole-scene feature, which is extracted via I3D (Carreira and Zisserman, 2017), and the asymmetric interaction feature, extracted with attention fusion. Finally, a regression module is utilized to learn the regression of the action quality

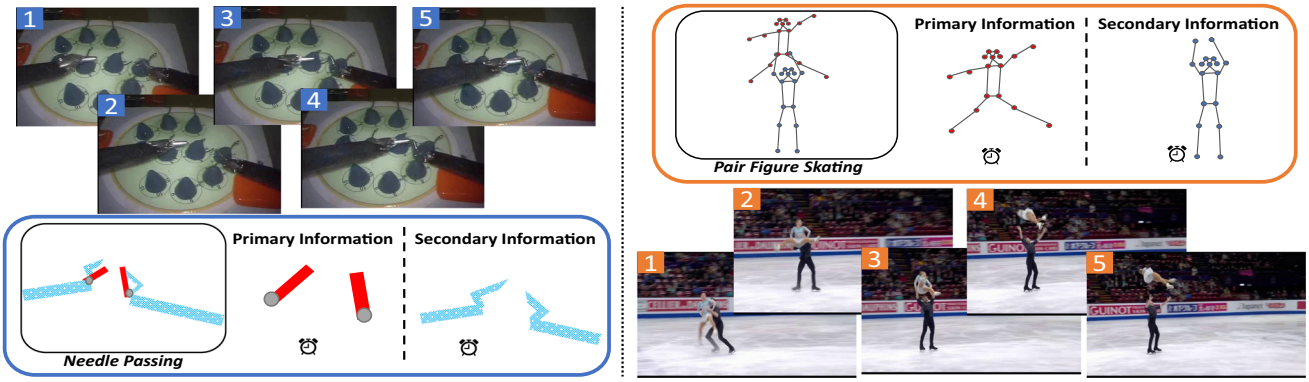

Fig. 3 Examples of primary and secondary information partitioning. The clock icon indicates the motion of an individual part

then, a progressive search module for asymmetric modelling of these intelligent agents is introduced to learn an adaptive model for various interactive actions. Finally, an appropriate module block selected for modelling specific asymmetric actions is utilized to learn the interactive action assessment with the assessor.

3.1 Asymmetric Interaction Learner

We construct an asymmetric interaction learner to model the asymmetric interaction in the actions. We hope this learner can automatically assign a suitable role to each agent (e.g., a primary or secondary role) and adaptively model the asymmetric interaction between primary and secondary roles. Correspondingly, we design an automatic assigner for agents in the scene as well as an asymmetric interaction network search module for asymmetric interaction modelling. The details of these two components are introduced below.

3.1.1 Automatic Assigner for Agents

It is ubiquitous that actions are performed with multiple people or multiple agents, and these are mostly asymmetric interactions. Thus, for action assessment modelling, it is necessary to consider asymmetric interactions among humans and (or) objects in actions. To reduce noise interference in videos, we extract a set of subtle but informative features at high-level and abstract semantics, denoted as A_a, which contains several agents (humans or objects); namely, only indispensable kinetic information for describing action is introduced, such as human pose and speed information. For instance, in surgery tasks whose samples are from an egocentric perspective, we assign A_a the kinetic information of the tool tips that contains the object information (e.g., tool orientations) and speed information; for most types of action performance where entire human bodies should be considered, A_a is the pose information detected by a pose estimator.

Before explicitly modelling asymmetric interactions, it is critical to determine the primary agent that is dominant with respect to others in the interaction, and the others are viewed as the secondary agents. Thus, according to the semantics in the interaction relation, we can divide A_a into two parts: the primary information (denoted as A_p) and the secondary information (denoted as A_s). See Fig. 3 for an example. The egocentric surgical action involves four agents, (i.e., the master tool manipulators and the patient-side tool manipulators of both left and right sides). Intuitively, for semantic consistency, we assign the motions of the master tool tips as the *primary* and those of the patient-side tool tips and handles, which are relatively inactive but have latent interactions with the primary, as the *secondary*. However, it is not practicable to manually assign the primary agents and the secondary agents of any action according to semantics and our experience. Thus, it is necessary to propose an automatic module to assign agents to the corresponding roles in an action. For instance, we hope this module can automatically assign the master tool tips as the *primary* and the patient-side tool tips and handles as the *secondary* among the four agents in the surgical action.

To automatically obtain these two parts (i.e., *primary* and *secondary*), we design an automatic assigner for agents. Since the raw data of each agent are likely to come from different subspaces (i.e., $A_a = [A_a(1), A_a(2), \ldots, A_a(n)]$, where n is the number of agents), we embed the raw data of each agent into the same format for further modelling, and this can be expressed as

$$\hat{A}_a = [\mathcal{E}_1(A_a(1)), \mathcal{E}_2(A_a(2)), \ldots, \mathcal{E}_n(A_a(n))], \tag{1}$$

where $\mathcal{E}_i(\cdot)$ is the embedding function for the i-th agent and $\hat{A}_a \in \mathbb{R}^{n*M}$ represents the fixed abstract feature of n agents.

To determine which agents are the primary, we use a trainable indicator matrix learned from $\hat{A}_a$. It is computed by

$$B = \mathcal{H}(\hat{A}_a), \tag{2}$$

where $B \in \mathbb{R}^n$ and $\mathcal{H}(\cdot)$ is a function that computes the probabilities that the agents will be assigned as the primary. The indicator matrix B is learnable from agents since the primary agents inherently contain a larger amount of information about action interaction than the secondaries.

Thus, we can obtain the primary information as follows:

$$A_p = \Gamma_k(B) \cdot \hat{A}_a, \tag{3}$$

where $A_p \in \mathbb{R}^{k*M}$ and $\Gamma_k(B)$ means that only the top k probabilities in matrix B will be fixed as 1 and the others will be 0 to serve as an indicator.

Accordingly, the secondary information can be assigned as

$$A_s = \Gamma_{n-k}(1 - B) \cdot \hat{A}_a, \tag{4}$$

where $A_s \in \mathbb{R}^{(n-k)*M}$.

Then, we obtain the motion features of the primary agent and the secondary agent from the abstract information of all agents in a learnable manner that automatically divides the agents into primary and secondary according to the corresponding case. Next, we introduce a progressive search module for asymmetric interaction modelling.

3.1.2 Asymmetric Interaction Network Search Module

To make our model more adaptive for various asymmetric interaction types, we design a progressive network architecture search module with operation search modules (OSMs) to search for a relatively optimal framework for general asymmetric interaction modelling, as shown in Fig. 2. As mentioned above, with an automatic assigner for agents that aims to learn the primary and secondary semantics in a specific scene, we automatically divide A_a into two parts, where the primary information is denoted as A_p and the secondary information is denoted as A_s. An example diagram is shown in Fig. 3. After dividing the agents into two parts, it is not reasonable for us to coarsely fuse them and extract deeper features with an agnostic network because it may make the process of automatic assignment for agents meaningless. Thus, we attempt to construct an effective network prototype to progressively assist in modelling asymmetric interactions between the primary and the secondary.

Interaction Network Prototype of Asymmetric Modelling We introduce a network module prototype, and an illustration of the construction of asymmetric interaction features between the primary and the secondary (denoted as Y_{psi}) is shown in Fig. 4. Although we separate the primary and secondary motion agents, it is difficult to manually model their asymmetric interactions, which vary under different scenarios, as shown in Fig. 3. Thus, with different semantics in an interaction action, we can generally assume that the primary and secondary information comes from different subspaces, and the primary one is dominant relative to the secondary in the action. Therefore, to properly explore the potential relation and asymmetric interaction between the *primary* and the *secondary*, we first use a transformation module to map the secondary information into a latent space, the same as that of the primary, to bridge the domain gap between the primary and the secondary. When the *primary* and *secondary* are from the same space, the transformation module will tend to learn an identity function (Chen et al., 2017); that is, the transformation module will degenerate to do nothing in this case. Next, to extract the coordination between the primary and the secondary agents in the action, we measure the difference between the *primary* and the *secondary* after the transformation, where it is demonstrated that the difference operation is an effective operation to use to explore the relations between visual instances (Pan et al., 2019). This process can be expressed as

$$\begin{aligned} \widetilde{A}_s^{(t)} &= \mathcal{T}(A_s^{(t)}), \\ I_d^{(t)} &= \mathcal{D}(A_p^{(t)}, \widetilde{A}_s^{(t)}), \end{aligned} \tag{5}$$

where $I_d^{(t)} \in \mathbb{R}^N$, and N denotes the dimensions of $I_d^{(t)}$ and $A_p^{(t)}$. The index t denotes the time step t of a certain feature in Fig. 2, $\mathcal{T}(\cdot)$ is a module that conducts the transformation operation, and $\mathcal{D}(\cdot)$ is a module to measure the difference between the *primary* and the *secondary*, which directly explores the relations between them.

As discussed above, the primary agent leads the action execution, while the secondary agent cooperates with it to perform the action. To incorporate the superiority of the primary information, we then concatenate the difference feature (i.e., I_d) and primary information (i.e., A_p), called the primary-secondary information (denoted as M_{ps}). It can be written as

$$M_{ps}^{(t)} = A_p^{(t)} \oplus I_d^{(t)}, \tag{6}$$

where $M_{ps}^{(t)} \in \mathbb{R}^{2N}$ and the operator $\oplus$ represents the concatenation operator.

The above-mentioned process can model the interactions between the primary and secondary in the spatial domain. Moreover, since the interactions occur over time, temporal relations for asymmetrically interactive action assessment among agents (i.e., the primary and the secondaries) are essential. Therefore, we utilize a temporal network to learn the pattern of temporal interactions and obtain the rich and complete spatial-temporal features, called asymmetric interaction features. We represent this process as

$$Y_{psi}^{(t)} = \mathcal{P}(M_{ps}^{(t)}), \tag{7}$$

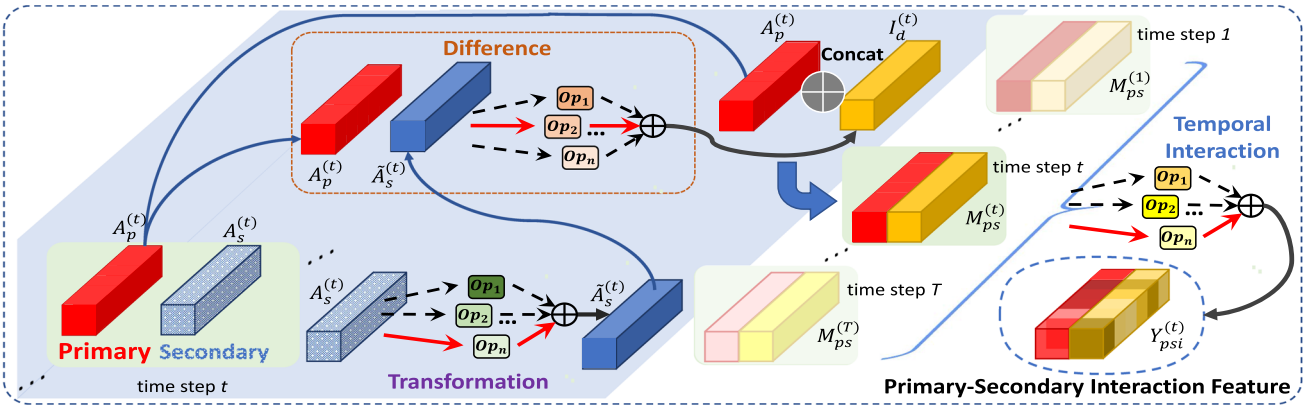

Fig. 4 The computation of the primary–secondary interaction feature Y_{psi}. The transformation of the secondary information maps the secondary into the latent space of the primary information. Difference and concatenation operations are performed on the primary information, proceeding on the interactions in the temporal domain to obtain the final asymmetric interaction feature Y_{psi}

where $Y_{psi}^{(t)} \in \mathbb{R}^d$, $\mathcal{P}(\cdot)$ is a temporal network and d is the dimension of the hidden layer of the temporal network.

As shown in Fig. 4, we construct a network prototype for our asymmetric interaction modelling. In this prototype, there are some submodules (i.e., transformation, difference and temporal modules) that need to be considered regarding which operations will be selected in the corresponding module. Inspired by related works on NAS (Liu et al., 2018), we propose our asymmetric interaction network search module, which consists of an effective prototype for progressive learning and a general OSM for adaptively searching the proper operations for each module in the prototype.

Operation Search Module In the operation search module, there are some candidate operations that are common for the corresponding module to compose the search spaces for *the transformation*, *the difference* and *the temporal interaction*. After constructing the search space, we use a differentiable architecture search mechanism (DARTS (Liu et al., 2018)) for each module in our prototype to search and select the most effective operation for asymmetric interaction modelling.

(1) Operation Search Space Since each module contains its own specific semantics in our prototype, different candidate operations are considered for each module. To make the process of searching for operations for each module easier and more subtle, we consider only general and simple operations, as shown in Fig. 5, which are common in the corresponding semantics of modules, rather than complicated manual operations. Therefore, for the transformation module, convolutions with different kernel sizes (Conv. 1*1 and Conv. 3*3) and a fully connected layer (FC layer) with an activation function are used as the candidate operations, and the identity function (Identity) is also considered in the search space of the transformation since the choice of identity function indicates that the transformation will do nothing. For the difference module, pairwise $\mathscr{L}_2$ distance, cosine distance, kernel distance (element-wise distance with an exponential kernel) and element-wise subtraction compose the operation candidate set. For the temporal interaction module, we construct the search space with LSTM, GRU, an FC layer, an attention network [ResAttention (Zhu & Wu, 2021)] and learnable pooling layers [NetVLAD (Arandjelovic et al., 2016), adaptive maximum pooling and adaptive average pooling] since they explore relations among temporal interactions with different focuses. In addition, we add a zero operation (Zero), setting all elements of feature to zero,

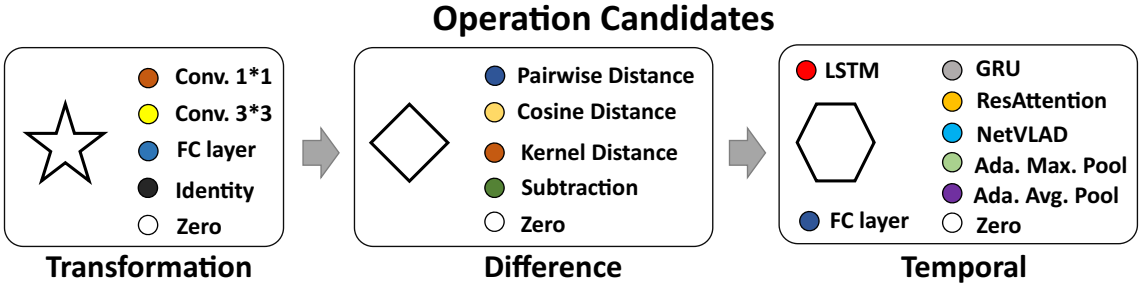

Fig. 5 Operation candidates of search space for the transformation, difference and temporal interaction

to the search space of each module, which is meaningful since the module will search a zero operation if it is deactivated.

(2) Operation Selection , the DARTS mechanism utilizes the *softmax* function as a continuous relaxation condition to adaptively select the operations from the candidate operation set (denoted as Op). The selection in the training phase can be written as

$$\begin{aligned} Q &= \sum_i \alpha_i^U * Op_i^U(P), \\ \alpha^U &= softmax(\delta^U), \end{aligned} \tag{8}$$

where $U \in \{\mathcal{T}, \mathcal{D}, \mathcal{P}\}$ indicates a module that is required to be performed with an OSM (i.e., transformation, difference and temporal interaction modules), Op_i^U denotes the ith candidate operation in the candidate operation set of module U, and δ^U denotes the weights of the edges for computing the probabilities that the OSM will select this operation with the *softmax* function. Here, $(P, Q) \in \{(A_s, \widetilde{A}_s), ((A_p, \widetilde{A}_s), I_d), (M_{ps}, Y_{psi})\}$, corresponding to U.

After the convergence of the progressive operation search, the operation with the highest weight, the most valuable operation in the corresponding module, will be selected in the final framework of the asymmetric interaction learner model. Therefore, we can rewrite Eq. (8) in the final model for evaluation as

$$\begin{aligned} Q &= Op_m^U(P), \\ m &= \arg\max_i \ \alpha_i^U. \end{aligned} \tag{9}$$

Specifically, the operation search module is only employed in modelling the asymmetric interactions of agents adaptively, since different actions always follow their specific criteria. It is not suitable to model the entire network in a searching manner due to its expensive computational consumption.

3.2 Attentive Assessor with Contextual Interaction

As shown in Fig. 2, we obtain the asymmetric interaction feature (i.e., Y_{psi}) through the asymmetric interaction learner. Even if we model the asymmetric interaction of agents in videos by extracting the high-level feature, there is rich whole-scene information left in the RGB images. To better assess how the actions perform, we build an attentive assessor with an attentive contextual interaction mechanism to attentively fuse the asymmetric interaction feature with the whole-scene feature. Finally, we use the fusion feature to score the performance of actions with a general assessment head.

3.2.1 Attentive Contextual Interaction

Although noise exists, the whole-scene feature contains additional information that can complement our asymmetric interaction feature. To help the learned asymmetric interaction feature in modelling the global scene, we further apply a 3D convolutional neural network as the backbone [i.e., I3D (Carreira and Zisserman, 2017)] to extract the whole-scene feature of videos, denoted as F_{wl}. Then, we obtain two streams of features, the asymmetric interaction feature and the whole-scene feature (i.e., Y_{psi} and F_{wl}). Before fusing these two features, we map the whole scene F_{wl} into the latent space, do the same for the asymmetric interaction feature Y_{psi} through an encoder, and obtain the encoded feature, denoted as X_{wl}, where $X_{wl}^{(t)} \in \mathbb{R}^d$ and d is the dimension of the encoded feature.

In our attentive assessor, we perform attentive contextual interaction between the whole-scene feature and the asymmetric interaction feature. Specifically, we utilize the whole-scene feature F_{wl} to learn a key map as the attention for fusing the whole-scene encoded feature (X_{wl}) and our asymmetric interaction feature (Y_{psi}). This is because the whole-scene feature F_{wl} contains the whole-scene context. Moreover, we regard $(X_{wl}^{(t)} \oplus Y_{psi}^{(t)})$ as the queries and values of the attention mechanism, inspired by self-attention (Vaswani et al., 2017). Therefore, we form the fusion process with our attentive contextual interaction as follows:

$$\begin{aligned} Z_{att}^{(t)} &= W^{(t)} \circ (X_{wl}^{(t)} \oplus Y_{psi}^{(t)})', \\ W^{(t)} &= softmax((X_{wl}^{(t)} \oplus Y_{psi}^{(t)})' \circ O_{key}^{(t)}), \\ O_{key}^{(t)} &= \mathcal{FC}_{key}(F_{wl}^{(t)}), \end{aligned} \tag{10}$$

where $\oplus$ is a concatenation operator for the dimension of features, $\circ$ represents matrix multiplication, and $softmax(\cdot)$ is the *softmax* function. $\mathcal{FC}_{key}(\cdot)$ is a fully connected layer used to learn the key mapping. Here, $X_{wl}^{(t)}, Y_{psi}^{(t)}, Z_{att}^{(t)}, O_{key}^{(t)} \in \mathbb{R}^d$, and A' is the transpose of matrix A.

3.2.2 Scoring for Action Assessment

Finally, the attentive assessor of our framework outputs a final score for action performance through the regression module, as shown in Fig. 2. Since there are T clips segmented from an entire video, we will have T scores for a video, which can be used to analyze the impact of each performance segment for the whole performance. Thus, the final overall assessment result will be represented as a score given by

$$S = \sum^{T} \mathcal{R}(Z_{att}^{(t)}), \tag{11}$$

where S denotes the predicted score for the action performance, $Z_{att}^{(t)}$ is the output of the attentive contextual interaction, and $\mathcal{R}(\cdot)$ is the regression module implemented with two fully connected layers.

3.3 Loss Function

In our framework, we use the mean-squared error (MSE) as the loss function of our model, which is defined as

$$\mathcal{L}(y, \hat{y}) = \frac{1}{C}\sum_{i}^{C}(y_i - \hat{y}_i)^2, \tag{12}$$

where y and $\hat{y}$ represent the ground truth and the predicted value, respectively, and C denotes the number of samples.

3.4 Extension to Synchronized Interactive Action Assessment: Multi-task Training

For synchronized interactive action where performers try their best to perform the same (or symmetric) action, our asymmetric interaction learner can be generalized to synchronized interactive action assessment, even when there are no explicit primary and secondary roles between performers (i.e., there is an equal relation). As shown in Fig. 7, there is no strong primary-secondary relation between the two performers in synchronized diving.

Specifically, we generalize our framework with multi-task training to assist synchronized interactive action assessment where the actions are in equal relationships. In our framework, there are two-stream features before attentive contextual interaction, and we find that multiple tasks can naturally align the two-stream features with reasonable semantics. That is, the whole-scene feature can be exploited to learn action assessment from the overall performance, while the asymmetric interaction feature can be developed to learn action assessment from interactive actions. Take synchronized diving as an example. After a series of actions are performed, the execution score and synchronization score will be given by professional referees during scoring for the entire action performance. Accordingly, we can assess the execution of the action by utilizing the whole-scene feature, which several existing methods (Parmar and Tran Morris, 2017, 2019) have done, and the features extracted by our asymmetric interaction learner can be reasonably used for learning the synchronization of action since the asymmetric interaction learner mainly learns the interaction between the two players in synchronized diving. Consequently, we can directly use the whole-scene feature X_{wl} to learn the scoring for the execution and use the asymmetric interaction feature Y_{psi} for the synchronization of the action, as shown in Fig. 6. Correspondingly, the assessment results for the execution and synchronization are represented as

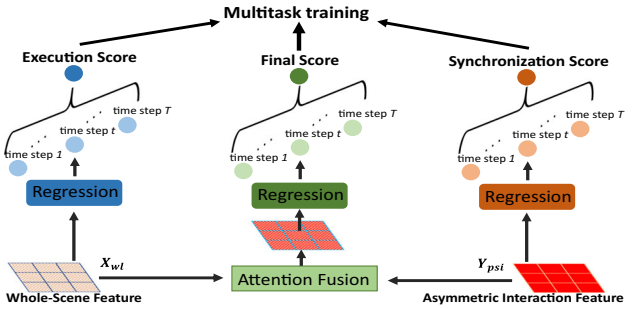

Fig. 6 Our framework with multi-task training on synchronized diving. We directly use the whole-scene feature X_{wl} to learn the scoring for the execution and use the asymmetric interaction feature Y_{psi} for the synchronization of the action

$$\begin{aligned} S_{ex} &= \sum^{T} \mathcal{R}_A(X_{wl}^{(t)}), \\ S_{sn} &= \sum^{T} \mathcal{R}_B(Y_{psi}^{(t)}), \end{aligned} \tag{13}$$

where S_{ex} and S_{sn} are the predicted execution score and synchronization score, respectively. $\mathcal{R}_*(\cdot)$ denotes the regression module, which is implemented with two fully connected layers in the experiments.

For multi-task training, the loss function is reformulated as

$$\mathcal{L} = \mathcal{L}_{fn} + \theta * \mathcal{L}_{ex} + (1 - \theta) * \mathcal{L}_{sn}. \tag{14}$$

Here, $\mathcal{L}_{fn}, \mathcal{L}_{ex}$ and $\mathcal{L}_{sn}$ represent the loss functions of regression for the final scores, execution scores and synchronization scores, respectively. θ is a trade-off weight for $\mathcal{L}_{ex}$ and $\mathcal{L}_{sn}$. Similarly, we use the mean-squared error (MSE) as the loss function, as presented in Eq. (12).

We believe that the overall loss function shown in Eq. (14) makes sense, since in synchronized diving, great performance should be excellent in both synchronization and execution. Therefore, in addition to the final score, the execution score and synchronization score, which are essential in synchronized diving, are utilized to perform multi-task training to assist in synchronized interactive action assessment.

4 Dataset

In this section, we introduce two new datasets, *TASD-2* and *PaSk*, which include two-person actions in weak primary-secondary relations and strong primary-secondary relations, respectively. These datasets are collected to explore the modelling of action quality assessment for asymmetric interactions between agents, which also offer the potential

to better study multi-agent action quality assessment in the future.

4.1 TASD-2 Dataset

We collected a new dataset, called *TASD-2*, for general interactive action assessment. Although *AQA-7* (Parmar and Tran Morris, 2019) contains a synchronized 3-m springboard (SyncDiving-3 m) and synchronized 10-m platform (SyncDiving-10 m), it is difficult to explore the interaction between two performers since the samples are captured in a side view, leading to performers overlapping most of the time. To provide a better view (i.e., a front view) to investigate the interaction between the two performers, we formed a new dataset, called *TASD-2*. Sample frames of the samples in *TASD-2* are shown in Fig. 7.

Fig. 7 Samples of the *TASD-2 dataset*. In this dataset, there are two sporting actions, synchronized 3-m springboard diving (SyncDiving-3 m) and synchronized 10-m platform diving (SyncDiving-10 m). The interactions of the two performers are captured in a front view

4.1.1 Dataset Construction Details

For *TASD-2*, we collected more than 600 samples from twenty accessible video recordings of synchronized diving events on YouTube, including four in the Olympic Games, three in FINA, nine in European diving competitions and four in the Southeast Asian Games, which could be categorized as synchronized 3-m springboard diving (SyncDiving-3 m) and synchronized 10-m platform diving (SyncDiving-10 m).

Table 1 Details of the *TASD-2* dataset compared to the *AQA-7* dataset

Dataset	TASD-2		AQA-7 (Parmar and Tran Morris, 2019)	
Sport	SD-3	SD-10	SD-3	SD-10
# Avg. Seq. Len	102	102	156	105
# Samples	238	368	88	91
# Training set	188	293	60	63
# Testing set	50	75	28	28
# Participants	2		2	
View	Front		Side	
Difficulty score	✓		–	
Execution score	✓		–	
Synchronization score	✓		–	
Final score	✓		✓	
Execution score_v2	✓		–	

'SD-3': SyncDiving-3 m. 'SD-10': SyncDiving-10. 'Front': view shot from a front perspective. 'Side': view shot from a side perspective

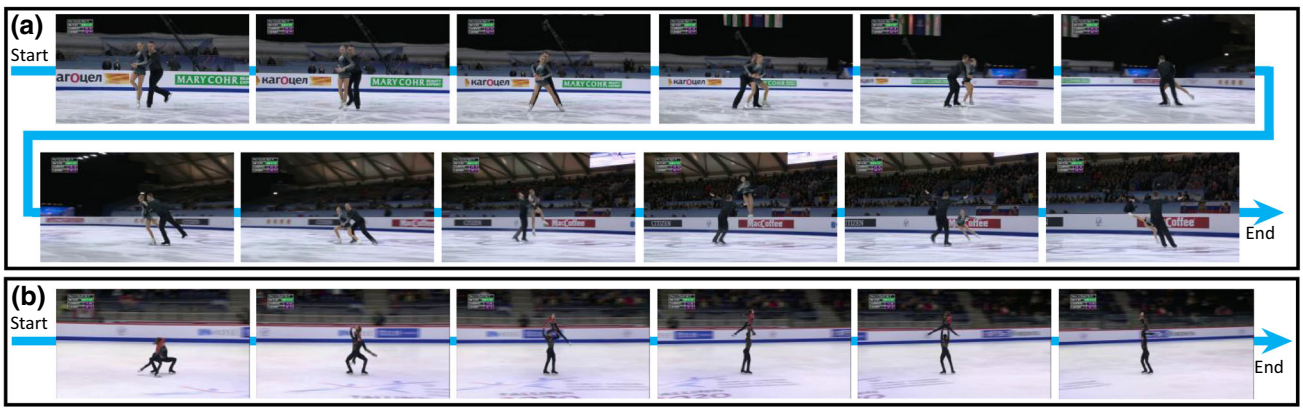

Fig. 8 Samples of the *PaSk dataset*. We visualize two samples, **a**, **b**, from the *PaSk dataset*. Since they contain different subactions, they have different execution time. **a** is longer than **b**

Although samples were clipped from video recordings of different international sporting events, referees are professional and strictly observe the judge handbooks of diving (International Swimming Federation, 2017). To determine whether a diving video was taken from the front view, we watched almost the entire video and recorded the starting frame and ending frame to separate a sample video. For the clipped sample, some labels should be annotated and recorded for further study, particularly, the final scores given by the professional referees for action quality assessment. The details of the dataset are presented in Table 1. It is worth noting that the "execution score_v2" is determined by calculating the "difficulty score" multiplied by the "execution score", since referees give only the "execution score", with a value ranging from 0 to 10, regardless of the difficulty of the action. Hence, in the individual analysis of the execution of synchronized diving, we prefer to use the "execution score_v2" rather than the "execution score" directly. The length of each video was uniformly modified to 102 frames with a format of 320×240 for each frame, referring to *AQA-7* (Parmar and Tran Morris, 2019). We augmented the videos by left-right flipping and split them into a training set and a testing set with a ratio of $4:1$ in a random fashion.

4.2 PaSk Dataset

The *TASD-2* was collected to assess general interactive action. However, the actions in *TASD-2* are only in weak primary-secondary relations (equal relations). To complement the two-person actions in a strong primary-secondary relation, we collected another new dataset, called *PaSk*, containing two events involving two players, namely, pair free-skating and pair short programs. These events concern pair figure skating, and in pair figure skating, the two performers are in a clear primary-secondary relation because the male performer always assists the female performer in performing various kinds of actions in the air, and the overall performance effect can always be determined by the action execution of the female performer. In other words, the interaction between the male and female performers is asymmetric. Therefore, reasonable modelling for asymmetric interactions is necessary on the *PaSk* dataset.

4.2.1 Dataset Construction Details

To construct the new dataset *PaSk*, we collected more than 1000 samples from six valid videos of entire pair figure skating events on YouTube; the events were held by the International Skating Union (ISU), and they were categorized as pair free-skating and pair short programs. In a valid video of an entire pair figure skating event, there are generally more than ten pairs of participants, and every participant pair completes a series of performances in only approximately four minutes. In a complete series of performances, there are more than ten independent actions to be scored, and the referees score each of them after the action has been performed. Accordingly, in our dataset construction, we regard each independent action as a sample that is scored by the professional referees in the events. To record the starting frame and ending frame of each sample clip, we watched almost the entire video when annotating. Then, we split out a sample video and recorded the corresponding labels; see Table 2 for details. Note that the number of frames for each sample is different, ranging from 100 to 1000. In Table 2, "goe_score" (the gain or error score) is a subscore used to adjust the overall score for an action, which is a positive decimal for a gain or a negative decimal for an error. For this dataset, we randomly split all the samples into a training set and a testing set at a ratio of 4:1. Figure 8 presents the samples of our dataset, *PaSk*.

5 Experiments

The experiments are conducted on the assessment of interactive actions on *JIGSAWS*, whose actions mainly involve four agents, as well as *TASD-2* and *PaSk*, with two agents. Moreover, conventional action assessment involving a single person can be regarded as a special extension of our method. Thus, we also performed an evaluation of it on *AQA-7* (Parmar and Tran Morris, 2019).

5.1 Dataset Introduction

5.1.1 JIGSAWS

There are 206 samples in this dataset, which are egocentric videos of three surgical tasks, including 78 suturing samples, 56 needle passing samples and 72 knot tying samples. In our experiments, we used the scores given by experts according to the Objective Structured Assessment of Technical Skills (OSATS) grading scheme (Martin et al., 1997) as the ground-truth. Sample frames of these videos are presented in Fig. 9. Note that these videos are captured in stereo recordings with two different views (the left view and the right view) by deploying two cameras. Referring to existing methods (Pan et al., 2019; Doughty et al., 2018), we use all videos and evaluate our model with fourfold cross-validation in our experiments by following (Gao et al., 2020; Tang et al., 2020; Pan et al., 2019). The annotations for each video in *JIGSAWS* contain the 3D kinetics information of the master tool manipulators and patient-side manipulators, which will be used as the abstract information of the motions in our settings.

Table 2 Details of the *PaSk* dataset compared to the *UNLV-Skate* dataset (Parmar and Tran Morris, 2017)

Dataset Sport	PaSk Pair Fig. Skate	UNLV-Skate (Parmar and Tran Morris, 2017) Fig. Skate
# Avg. Seq. Len	500	4500
# Samples	1018	171
# Training set	811	100
# Testing set	207	71
# Participants	2	1
View	Single	Multiple
Final score	✓	✓
Gain or error score	✓	–

'Single': view shot from a single angle. 'Multiple': view shot from multiple angles

Fig. 9 Samples of the *JIGSAWS dataset*

5.1.2 TASD-2

There are 606 samples in this dataset, which include two kinds of actions, synchronized 3-m springboard (SyncDiving-3 m) and synchronized 10-m platform (SyncDiving-10 m) diving, captured in the front view. This dataset is introduced in Sect. 4.2, and the details of this dataset are presented in Table 1.

5.1.3 PaSk

There are 1018 samples in this dataset, which include two kinds of actions, pair free-skating and pair short programs. This dataset is introduced in Sect. 4.2, and the details can be found in Table 2.

5.2 Implementation Details

5.2.1 Data Preprocessing

On *JIGSAWS* (Gao et al., 2014), a dataset containing egocentric surgical videos, we extract the primary and secondary information from the 3D kinetics feature in the dataset. To map the different observed variables into a common space, DCT is performed on the 3D kinetics feature to obtain a 50-dimensional expanded A_a, where there are two master tool manipulators and two patient-side manipulators and these agents are assigned as the *primary* and *secondary* automatically in our asymmetric interaction module.

On sport action assessment tasks, we extract human poses (i.e., the coordinates of the key points of poses) with AlphaPose (Fang et al., 2017) to construct the initial abstract information A_a, with denoising and linear interpolation for completion. Figure 10 shows an example of the detection results of applying AlphaPose (Fang et al., 2017) to our *TASD-2*. In addition, for the actions in *PaSk*, we extract the whole-person feature (cropped by bounding boxes detected by FairMOT (Zhang et al., 2020)) via I3D pretrained on Kinetics (Carreira and Zisserman, 2017) and regard it as A_a, since the two performers in pair figure skating are usually in contorted postures that fool the existing pose estimation method (Fang et al., 2017).

Additionally, we extract the whole-scene feature via I3D pretrained on Kinetics (Carreira and Zisserman, 2017), with RGB and optical flow (Pérez et al., 2013) feature input. In contrast to previous works (Pan et al., 2019), we uniformly divide every video into 10 segments, corresponding to 10 time steps. For each segment, 16 frames from sports videos of *TASD-2* are uniformly sampled as the input of I3D; 24 frames are sampled for the videos of the pair figure skating in *PaSk*; in egocentric surgical videos, 64 frames are sampled

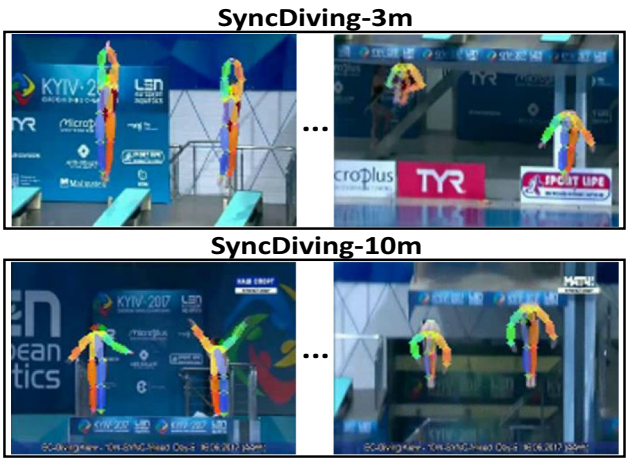

Fig. 10 Detection results by using AlphaPose (Fang et al., 2017) on *TASD-2*. The colored lines represent the skeletons of the players (Color figure onlne)

due to their longer duration than the sports videos used in our experiment. Except for *TASD-2* and *PaSk*, which were augmented during dataset construction, we augment the videos by the left-right flipping used in Pan et al. (2019). The scores in each dataset are normalized to [0, 100] as the labels used to supervise our model.

5.2.2 Model Training Setting

We implement our model using PyTorch. Our model uses the Adam optimizer with a weight decay rate of 0.2. We set the batch size to 16 in the searching and training phase. For the learning rates, we use a learning-rate schedule, *cosine annealing warm restarts*, in our model searching and training with an initial cycle of 10 epochs. In the operation searching phase, we train our model for 3000 epochs and determine the final operations according to the average rank of the frequencies with which each operation is chosen. After the model is fixed, we train our model for 50 epochs for *JIGSAWS* and 4000 epochs for the others. The number of time steps T is set to 10. We implement the encoder with an FC layer of dimensions 400×512 followed by a rectified linear unit (ReLU) activation. In the regression module, we utilize two FC layers, the first layer with dimensions of 512×128 with ReLU activation and the second layer with dimensions of 128×1 without an activation function to avoid a dead ReLU during score regression. The dropout parameter is set to 0.2, and θ is 0.4. The number of agents n is 4 for actions in *JIGSAWS* and 2 for actions in *TASD-2* and *PaSk*, and the number of *primary agents* k is set to half of n.

5.2.3 Evaluation Metric

To compare our method with existing works (Tang et al., 2020; Pan et al., 2019; Parmar and Tran Morris, 2017, 2019; Pirsiavash et al., 2014), we utilize Spearman's rank correlation coefficient as the evaluation metric of our model on action assessment, which is defined as

$$\rho = \frac{\sum_i (p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_i (p_i - \bar{p})^2 \sum_i (q_i - \bar{q})^2}}, \tag{15}$$

where p and q denote the rankings of two sequences. $-1 \leq \rho \leq 1$ (the higher the value is, the better). In our experiments, we use Spearman's rank correlation coefficient to evaluate the ranking relation between the predicted and ground-truth assessment results of our model, corresponding to p and q in Eq. (15). Spearman's rank correlation coefficient is preferable and frequently used for evaluating the regression setting in action assessment, since there exists subjective deviation in the ground-truth scores even though the rules of actions are strict and referees are professional. In addition, when there are multiple actions in a dataset, Fisher's z value[1] is computed as the average Spearman's rank correlation coefficient across actions from individual action correlations, as in Gao et al. (2020), Tang et al. (2020), Pan et al. (2019) and Parmar and Tran Morris (2019).

5.3 Comparison

5.3.1 Experiments on JIGSAWS

As shown in Table 3, we first evaluated our model on *JIGSAWS* by comparison with the state-of-the-art methods and our baseline (our full model without the asymmetric interaction learner and attentive contextual interaction). In comparison, while the TUSSA (Liu et al., 2021), USDL (Tang et al., 2020), JRG (Pan et al., 2019), TSN (Doughty et al., 2018), and ST-GCN (Yan et al., 2018b) approaches achieved state-of-the-art performance for action assessment on *JIGSAWS*, the results in Table 3 reveal that our model outperforms the previous state-of-the-art methods (TUSSA, USDL, JRG, TSN, ST-GCN) with an improvement of more than 0.08 on average. To obtain the baseline, we remove the asymmetric interaction learner and attentive contextual interaction module in Fig. 2. Namely, only the I3D feature of the whole scene is used for evaluating our baseline. Moreover, we concatenate the I3D feature and kinetics feature as a stronger baseline. As shown in the last three rows of Table 3, we observe improvements of more than 0.42 on average, which demonstrates that the asymmetric interaction learner is much more important in our model, and the effectiveness of the asymmetric interaction learner is confirmed.

Moreover, to further demonstrate the effectiveness of our proposal, we compared our method with the best non-deep learning approach reported in Zia and Essa (2018) by using leave-one-user-out (LOUO), another evaluation method with a different data construction, as shown in Table 4. We find that both JRG (Pan et al., 2019) and our method have their strengths. While the LOUO setting is demanding for the model's generalization ability, our model is slightly better than JRG (Pan et al., 2019) since our modelling is less specialized than that of JRG (Pan et al., 2019), in which each joint is modeled in a specialized manner.

5.3.2 Experiments on Sporting Actions

In addition to actions whose agents are parts, our framework can generalize to sporting actions involving two persons in both weak and strong primary-secondary relations. There-

[1] The Fisher Transform was proposed in 1921 to address a skewed distribution of the sample correlation (r) (PEARSON, 1913; Fisher, 1915); introducing it in the average correlation computation makes the result more reliable (Corey et al., 1998).

Table 3 The results of our proposal compared with those of the state-of-the-art methods and our baseline on *JIGSAWS*

	Input	Suturing	Needle passing	Knot tying	Avg. corr.
ST-GCN (Yan et al., 2018b)	𝕍𝕂	0.31	0.39	0.58	0.43
TSN (Doughty et al., 2018)	𝕍𝕂	0.34	0.23	0.72	0.46
JRG (Pan et al., 2019)	𝕍𝕂	0.36	0.51	0.75	0.57
USDL (Tang et al., 2020)	𝕍	0.71	0.69	0.71	0.70
TUSSA (Liu et al., 2021)	𝕍𝕂	**0.83**	0.76	0.82	0.80
Baseline	𝕍	0.05	0.09	0.11	0.08
Baseline+Kinetic	𝕍𝕂	0.17	0.37	0.73	0.46
Ours	𝕍𝕂	**0.83**	**0.83**	**0.95**	**0.88**

The best performance is marked in bold
𝕍: surgical videos. 𝕂: robotic kinematics

Table 4 The evaluation on *JIGSAWS* with LOUO

	Input	Suturing	Needle passing	Knot tying	Avg. corr.
DTC+DFT +ApEn (Zia & Essa, 2018)	𝕂	0.37	0.25	0.60	0.41
JRG (Pan et al., 2019)	𝕍𝕂	0.35	**0.67**	0.19	0.40
TUSSA (Liu et al., 2021)	𝕍𝕂	0.45	0.65	0.59	0.57
Ours	𝕍𝕂	**0.60**	0.66	**0.69**	**0.65**

The best performance is marked in bold
𝕍: surgical videos. 𝕂: robotic kinematics

fore, experiments on *TASD-2* and *PaSk* were conducted, and the results are shown in Table 5. While *TASD-2* and *PaSk* are our newly collected datasets, we utilize a naive model (RANDOM) that randomly predicts scores for action performance in the range of [0, 100] to observe the distribution of labels on these two datasets. The results in Table 5 indicate that the distribution of samples in *TASD-2* and *PaSk* is balanced. In addition, we evaluated C3D-LSTM (Parmar and Tran Morris, 2017) on *TASD-2*, but it did not work based on the experimental setting in Parmar and Tran Morris (2017, 2019). Then, we used I3D (Carreira and Zisserman, 2017) and SVR with different kernels, including linear, polynomial and radial basis function (RBF) kernels, on *TASD-2* and *PaSk*. The results in Table 5 show that the I3D-SVR models had great performance gains, which reflects the strong ability of I3D to some extent. By comparison, our method achieves state-of-the-art performance on both *TASD-2* and *PaSk*. In other words, our framework can model interactive actions well for both weak and strong primary-secondary relations, corresponding to the actions in *TASD-2* and *PaSk*, respectively.

5.4 Ablation Study

To investigate the contributions of each main module in our model, we conducted an ablation study on our framework by removing one of the components from our full model, including the attention fusion module and the asymmetric interaction learner module. Moreover, we further explore the contribution of each module in our asymmetric interaction learner, including the automatic assigner and the operation search module. The results are presented in Table 6.

Table 5 The results of our model on *TASD-2* and *PaSk*

	SyncDiving-3 m	SyncDiving-10 m	Pair figure skating	Avg. corr.
RANDOM	− 0.03	0.03	0.00	0.00
C3D-LSTM (Parmar and Tran Morris, 2017)	− 0.14	0.01	0.02	− 0.04
I3D (Carreira and Zisserman, 2017)-SVR-L	0.77	0.73	0.44	0.67
I3D (Carreira and Zisserman, 2017)-SVR-P	0.84	0.83	0.51	0.76
I3D (Carreira and Zisserman, 2017)-SVR-RBF	0.71	0.77	0.44	0.66
JRG (Pan et al., 2019)	0.89	0.81	0.60	0.79
Ours	**0.93**	**0.90**	**0.64**	**0.86**

The best performance is marked in bold

Table 6 The ablation study for exploring the effectiveness of each main module of our model on *JIGSAWS*

		Suturing	Needle passing	Knot tying	Avg. corr.
Asymmetric modelling	Full model	**0.83**	**0.83**	**0.95**	**0.88**
	W/o attention fusion module	0.61	0.55	0.80	0.67 (−0.21)
	W/o asymmetric interaction learner	0.07	0.41	0.64	0.40 (−0.48)
W/o auto-assigner	Manual assigner	0.76	0.72	0.87	0.79 (−0.09)
	Random assigner	0.72	0.83	0.72	0.76 (−0.12)
W/o OSM	W/o OSM for transformation	0.63	0.68	0.72	0.68 (−0.20)
	W/o OSM for Difference	0.68	0.75	0.68	0.70 (−0.18)
	W/o OSM for temporal interaction	0.65	0.54	0.59	0.60 (−0.28)
	W/o OSMs	0.37	0.59	0.64	0.54 (−0.34)

The best performance is marked in bold

5.4.1 Asymmetric Modelling

We attempt to remove the attention fusion module and asymmetric interaction learning. When replacing the attention fusion module with an average pooling, the model performance decreased by 0.21 on average, which implies that paying different amounts of attention to whole-scene features and asymmetric interaction features makes a positive impact. When we replace the asymmetric interaction learner with a single fully connected layer, the model performance decreases by 0.48. This result shows that our asymmetric leaner is indeed important for asymmetric action assessment.

5.4.2 Automatic Assigner

We attempt to remove the automatic assigner with a manual assigner or a random assigner. The manual assigner annotates the primary and the secondary agent by semantics that relies on heavy human annotations. In comparison, the random assigner does not require human annotation and it assigns the primary and the secondary agents randomly. Compared with our auto-assigner, using a manual assigner results in a decrease of 0.09 on average correlation, and using a random assigner results in a decrease of 0.48. The results show that our auto-assigner is useful for asymmetric modelling. Note that these two models are trained from scratch with the operation search module. To further analyze the effectiveness of our auto-assigner, we provide a further evaluation of the assigners with the same operations used in the corresponding modules in Table 7.

5.4.3 Operation Search Module

Then we analyze the effectiveness of our operation search module. We attempt to remove the transformation operations, the difference operations and the temporal interaction operations in our operation search module, respectively. Specifically, we implement a single FC layer as a basic transformation, feature subtraction as basic difference modelling, and an LSTM layer as a basic temporal interaction. From the results, we can see that removing the operation search results in a performance drop of at least 0.18, which demonstrates the importance of our operation search design.

5.5 Further Analysis

5.5.1 The Automatic Assignment for Agents

In the ablation study, although we replace the automatic assigner with a manual assigner or a random assigner, the operation search module (OSM) is re-trained and therefore different operations could be chosen for different assigners. To further analyze the effectiveness of our auto-assigner for the model without OSMs, we provide evaluations of the assigners using the same operations as the full model in Table 7. With the operations fixed, our auto-assigner outperforms the random assigner by 0.05 of the average correlation. It achieves similar results as the manual assigner, but the latter requires heavy semantics annotations. Moreover, with

Table 7 The further ablation study of the auto-assigner without OSM modelling on *JIGSAWS*

	Suturing	Needle passing	Knot tying	Avg. corr.
Full model	**0.83**	0.83	**0.95**	**0.88**
Manual assigner (fixed OSM)	0.80	**0.85**	**0.95**	**0.88**
Random assigner (fixed OSM)	0.76	0.73	0.93	0.83
Manual assigner (role exchange, fixed OSM)	0.72	0.64	0.65	0.67

The best performance is marked in bold

Table 8 The evaluation on the frequency of agent assignment on *JIGSAWS*

	Suturing	Needle passing	Knot tying	Avg. corr.
Per video segment (full model)	**0.83**	**0.83**	**0.95**	**0.88**
Per video	0.79	0.75	0.67	0.74

The best performance is marked in bold

the manual assigner, we exchange the roles of agents against their semantics, and a performance drop of 0.21 is obtained, which also demonstrates the effectiveness of our asymmetric modelling.

In addition, we evaluate the frequency of agent assignment in Table 8. Our auto-assigner helps to deal with the change of primary/secondary agents during the action by re-assigning agents per video segment in each video. If we only performed agent assignment once in each video, the average correlation would drop greatly by 0.14. Although per-frame assignment could be even more accurate, it requires too much computation effort and therefore, we perform agent assignments per video segment.

5.5.2 The Attentive Contextual Interaction

We compare using different features as the key of self-attention in our attentive contextual interaction (Eq. 10), and the results are presented in Table 9. In our full model, we implement the whole-scene feature $F_{wl}^{(t)}$ as the key since it encodes the original whole-scene video patterns. We also attempt to use the encoded whole-scene feature $X_{wl}^{(t)}$ or the asymmetric interaction feature $Y_{psi}^{(t)}$ as the key, but an inferior performance is observed.

5.5.3 Number of Primary/Secondary Agents

We also evaluate different numbers of primary/secondary agents on the *JIGSAWS* dataset, and the results are shown in Table 10. In the surgical actions, there are two master tool manipulators and two patient-side tool manipulators, 4 agents in total. Therefore, our model assigns 2 primary agents and 2 sary agents automatically. In this section, we additionally evaluate the performance of the model with different numbers of primary/secondary agents. The results indicate that assigning 2 primary and 2 sary agents achieves the best performance.

5.6 Training Time

The process of network searching and training is not time consuming. In the search phase, it takes approximately 0.05 s to train on a batch of 16 samples using one GTX 1080Ti GPU on *JIGSAWS*, while in the testing or training phase, it takes approximately 0.03 s. Moreover, we show the curves of the variance of the searching and training losses with the number of iterations in Fig. 11. This demonstrates that our method is convergent after effective training.

5.7 Visualization

5.7.1 Visualization of the Asymmetric Interaction Learner

To ensure that the automatic assigner for agents truly assigns the proper roles for actions in each video segment, we randomly sampled three results of the assignment on the actions suturing, SyncDiving-3 m and pair figure skating, as shown in Fig. 12. For pair figure skating, we find that the assigner tended to assign the female performer as the primary and the male performer as the secondary in every video segment, which corresponds to the real semantics (i.e., the female performer is relatively active compared with the male). This

Table 9 The evaluation of different key features for the attentive contextual interaction on *JIGSAWS*

	Suturing	Needle passing	Knot tying	Avg. corr.
Whole-scene feature as Key (full model)	**0.83**	**0.83**	**0.95**	**0.88**
Encoded whole-scene feature as Key	0.64	0.63	0.81	0.70
Asymmetric interaction feature as Key	0.71	0.65	0.78	0.72

The best performance is marked in bold

Table 10 The evaluation of different primary/secondary agent numbers on *JIGSAWS*

Agent Num. (primary/secondary)	Suturing	Needle passing	Knot tying	Avg. corr.
3/1	0.77	0.68	0.80	0.75
2/2 (full model)	**0.83**	**0.83**	**0.95**	**0.88**
1/3	0.73	0.78	0.83	0.78

The best performance is marked in bold

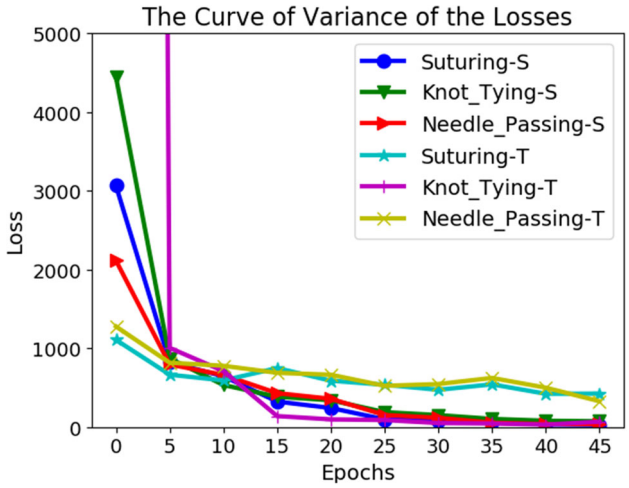

Fig. 11 The curves of the variance of the losses along with the training epochs on *JIGSAWS*. For the legend of this figure, the curves with the suffix '-S' denote that they are trained in the searching phase, while those with the suffix '-T' express they are trained in the training phase after network searching

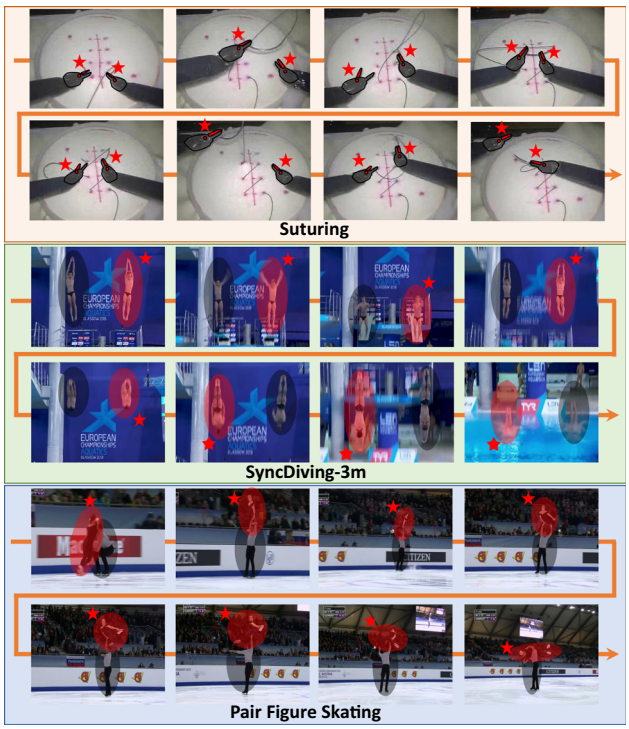

Fig. 12 Visualization of the automatic assignment of agents on samples of Suturing, SyncDiving-3 m and pair figure skating from video segments 2–9. Red denotes the primary agent (along with the red star), and blue denotes the secondary agent. Zoom in for the best view (Color figure online)

indicates that the automatic assigner could reasonably perform primary-secondary assignment for the agents of actions in strong primary-secondary relations, which is confirmed from the results on suturing as well. Moreover, for the results on SyncDiving-3 m, the ambiguous assignment also demonstrates that our model is adaptive in generalizing to actions in weak primary-secondary relations (equal relations). In addition, we present the results of a progressive operation search on various kinds of actions in Fig. 13, which demonstrates that our asymmetric interaction learner will adapt to asymmetric modelling along with actions.

5.7.2 Visualization of Attentive Contextual Interaction

As shown in Fig. 14, we visualized attention fusion by observing the computed results of Eq. (10). For knot tying, the attention fusion module always paid more attention to the asymmetric interaction feature because the motions of tool manipulators are more valuable. For the synchronized 3-m springboard, different amounts of attention were paid to different time steps in a sample, and the asymmetric interaction feature was more important after time step 8 because the interaction between the two actors when they were approaching entry was more important for synchronized diving assessment. It is obvious that our attention fusion module also made a difference by comparing different actions. This indicates that our method of attentive contextual interaction with attention fusion is effective.

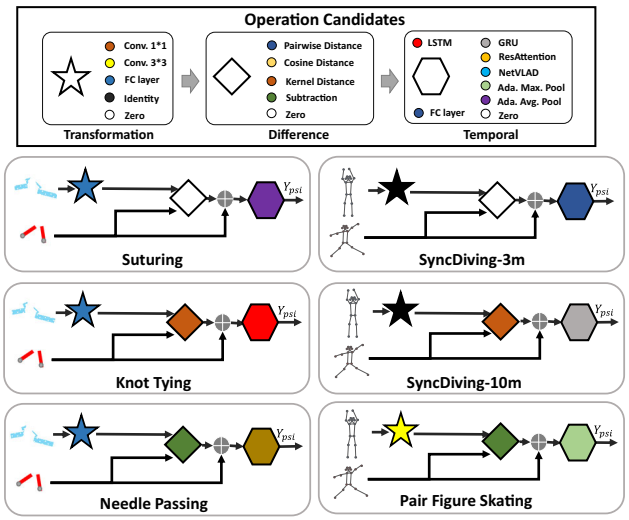

Fig. 13 The network search results of the progressive operation search module on the *JIGSAWS*, *TASD-2* and *PaSk* datasets

5.7.3 Visualization of the Subscore

To view the assessment process, we output the predicted subscores in different time steps defined in Eq. (11). Figure 15 presents an example of needle passing with scoring in each time step. From Fig. 15, we find that our model

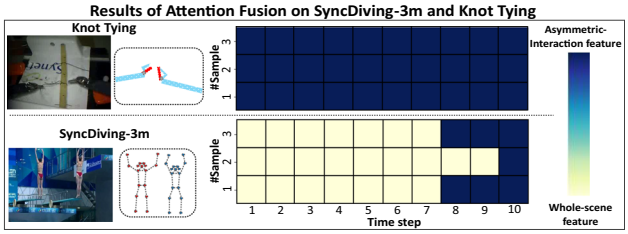

Fig. 14 Visualization of the attention fusion on samples of knot tying and synchronized 3-m springboard. "#Sample" represents the number of each of three randomly selected samples. The results indicate that our attention fusion method can pay different amounts of attention to different time steps

can give a reasonable score for each time step. Since there are no actions in the view in the first time step, our model gave a slightly negative judgement for it. Afterward, the first passing of the line used for needle passing is accomplished smoothly; thus, our model gave positive judgments in the next two time steps in Fig. 15 accordingly. However, in the middle stage of the needle passing case, we found that the two tool tips performed relatively abnormally and unskillfully, causing the surgical line to be staggered in the air; thus, poor judgements were obtained during this time. Correspondingly, when approaching finishing the needle passing task, our model gave relatively neutral judgements for simply ending the needle passing in this process. Therefore, the visualization also confirms that our framework is effective and interpretable.

5.8 Extended Experiment on Single-Person Actions

Unlike the multi-agent actions mentioned above, single-person actions contain only one agent. For generalization, we define the condition that if there is only one agent, we can use the motion of the camera capturing the action performance as a supplement. We additionally evaluated our framework on *AQA-7* (Parmar and Tran Morris, 2019) under this assumption; this dataset contains 1,106 videos in total composed of six actions. Unlike *TASD-2*, *AQA-7* (Parmar and Tran Morris, 2019) contains two-person actions, but they are captured only from the side view. The performers are not visually separable. Thus, visually, there is only one agent in the videos. Then, we extract the motion feature of the camera by computing the optical flow [using the TV-L1 algorithm (Pérez et al., 2013)] at the region near the edge of the images. In this task, we fix the weight decay rate of the Adam optimizer in our model at 0.8. With the experimental settings described in Gao et al. (2020), the performance results are reported in Table 11. The results demonstrate that our method is competitive with current state-of-the-art methods, with the best performances on the diving and the sync. 3 m action assessment. Our proposal outperforms most of the state-of-the-art methods on aver-

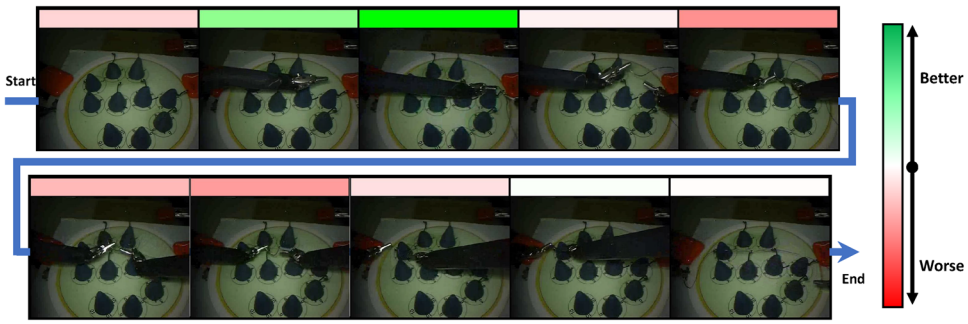

Fig. 15 The action assessment results of our model on a needle passing case. The assessment results of our model indicate good (green) and bad (red) action performance for each time step (Color figure online)

Table 11 The results of our model applied to *AQA-7*

	Diving	Gymvault	Skiing	Snowboard	Sync. 3 m	Sync. 10 m	Avg. corr.
Pose+DCT (Pirsiavash et al., 2014)	0.5300	–	–	–	–	–	–
ST-GCN (Yan et al., 2018b)	0.3286	0.5770	0.1681	0.1234	0.6600	0.6483	0.4433
C3D-LSTM (Parmar and Tran Morris, 2017)	0.6047	0.5636	0.4593	0.5029	0.7912	0.6927	0.6165
C3D-SVR (Parmar and Tran Morris, 2017)	0.7902	0.6824	0.5209	0.4006	0.5937	0.9120	0.6937
JRG (Pan et al., 2019)	0.7630	0.7358	0.6006	0.5405	0.9013	**0.9254**	0.7849
USDL-Regression (Tang et al., 2020)	0.7438	0.7342	0.5190	0.5103	0.8915	0.8703	0.7472
USDL (Tang et al., 2020)	0.8099	**0.7570**	**0.6538**	**0.7109**	0.9166	0.8878	0.8102
Ours	**0.8113**	0.7296	0.5925	0.5662	**0.9506**	0.9197	**0.8126**

The best performance is marked in bold and the second-best performance is underlined
This indicates that our framework can be generalized to traditional single-person action assessment; our performance is competitive, *i.e.,* second place on average and the best on *sync. 3 m*

Table 12 The results of our method on two datasets about daily actions, the *EPIC-Skills* and the *BEST* datasets

EPIC-skills dataset	Surgery	Dough-rolling	Drawing	Chopstick-using	Avg. ccc.
RankSVM (Joachims, 2006)	0.652	0.720	0.715	0.766	0.713
AlexNet+C3D (Yao et al., 2016)	0.661	0.781	0.720	0.703	0.716
Who's better (Doughty et al., 2018)	0.702	0.794	0.832	0.715	0.761
Rank aware attention (Doughty et al., 2019)	0.685	0.869	0.823	0.847	0.806
Ours	**0.754**	**0.877**	**0.851**	**0.855**	**0.834**

The best performance is marked in bold

Table 13 The results of our method on two datasets about daily actions, the *BEST* datasets

EPIC-skills dataset	Surgery	Dough-rolling	Drawing	Chopstick-using	Avg. ccc.	
BEST Dataset	**Apply-eyeliner**	**Braid-hair**	**Origami**	**Scrambled-eggs**	**Tie-Tie**	**Avg. Acc.**
Who's better (Doughty et al., 2018)	–	–	–	–	–	0.758
Rank aware attention (Doughty et al., 2019)	**0.855**	**0.765**	0.689	**0.877**	0.876	0.812
Ours	0.849	0.758	**0.784**	0.820	**0.882**	**0.819**

The best performance is marked in bold

age. Therefore, the extended experiment demonstrates that our framework can effectively generalize to common action assessment tasks.

In addition, we evaluate our method on daily actions with egocentric views on the *EPIC-Skills* dataset and the *BEST* dataset. These datasets formulate action assessment as a pairwise ranking problem, which aims to rank a pair of input videos according to the skills of the action execution. Some methods (Joachims, 2006; Yao et al., 2016; Doughty et al., 2018, 2019) have been proposed for pairwise ranking action assessment, including the SVM-based approach (Joachims, 2006) and deep models (Yao et al., 2016; Doughty et al., 2018, 2019). We extend our work to this setting to demonstrate the effectiveness of our method for egocentric single-person daily actions. To ensure a fair comparison with the existing works, we use the same whole-scene features released in Doughty et al. (2019). Since the camera motion is not directly accessible, we use a constant as the secondary feature alternatively to indicate an empty secondary agent. From the results shown in Table 12 and 13, we can see that our method achieves superior results in comparison with the existing works, demonstrating the effectiveness and generalization of our framework.

6 Conclusion

In this work, we have introduced a novel asymmetric interaction learner module for asymmetrically interactive action assessment. In our modelling, the roles in an asymmetrically interactive action are categorized as a primary agent and secondary agents, named the *primary* and *secondaries*, respectively. With the asymmetric interaction learner, we can model the interactive actions in many scenarios including strong asymmetric relations (e.g., pair figure skating), weak asymmetric relations (e.g., synchronized sports), actions with multiple primary agents (e.g., the surgical actions with left and right manipulators), actions with a single primary agent (e.g., single-person Olympics and daily actions). With the extensive experiments on *JIGSAWS* (Gao et al., 2014), *TASD-2* and *PaSk*, it is demonstrated that our asymmetric modelling can handle both the strong and the weak asymmetric relations automatically, corresponding to the case that the primary agents rarely change in strong asymmetric relations and the case that they change more frequently for weak asymmetric relations. Our automatic agent assigner can also adapt to a different number of primary agents. The extended experiments on *AQA-7* (Parmar and Tran Morris, 2019), *EPIC-Skills* (Doughty et al., 2018) and *BEST* (Doughty et al., 2019) also illustrated that our model can be adapted to perform conventional action assessment in which only one performer is considered. Nonetheless, several issues remain to be explored in the future, including the collection of action assessment datasets with more agents and the extension of our method to more complex multi-agent scenarios.

Acknowledgements This work was supported partially by the NSFC (U21A20471,U1911401,U1811461), Guangdong NSF Project (Nos. 2020B1515120085, 2018B030312002), Guangzhou Research Project (201902010037), the Key-Area Research and Development Program of Guangzhou (202007030004), and the Major Key Project of PCL (PCL2021A12). The corresponding author and principal investigator for this paper is Wei-Shi Zheng.

References

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR* (pp. 5297–5307).

Azar, S. M., Atigh, M. G., Nickabadi, A., & Alahi, A. (2019). Convolutional relational machine for group activity recognition. In *CVPR* (pp. 7892–7901).

Bertasius, G., Soo Park, H., Yu, S. X., & Shi, J. (2017). Am I a baller? Basketball performance assessment from first-person videos. In *ICCV* (pp. 2177–2185).

Cai, H., Zhu, L., & Han, S. (2018). Proxylessnas: Direct neural architecture search on target task and hardware. In *ICLR*.

Carreira, J., Zisserman, A. (2017). Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR* (pp. 6299–6308).

Chang, X., Zheng, W.-S., & Zhang, J. (2015). Learning person-person interaction in collective activity recognition. *TIP 24*(6), 1905–1918.

Chen, J., Wang, Y., Qin, J., Liu, L., & Shao, L. (July 2017). Fast person re-identification via cross-camera semantic binary transformation. In *CVPR*.

Corey, D. M., Dunlap, W. P., & Burke, M. J. (1998). Averaging correlations: Expected values and bias in combined Pearson RS and Fisher's Z transformations. *JGP, 125*(3), 245–261.

Dong, X., & Yang, Y. (2019). Searching for a robust neural architecture in four GPU hours. In *CVPR* (pp. 1761–1770).

Doughty, H., Damen, D., & Mayol-Cuevas, W. (2018). Whoś better, whoś best: Skill determination in video using deep ranking. In *CVPR*.

Doughty, H., Mayol-Cuevas, W., & Damen, D. (2019). The pros and cons: Rank-aware temporal attention for skill determination in long videos. In *CVPR* (pp. 7862–7871).

Fang, H.-S., Xie, S., Tai, Y.-W., & Lu, C. (2017). Rmpe: Regional multi-person pose estimation. In *ICCV* (pp. 2334–2343).

Fisher, R. A. (1915). Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika, 10*(4), 507–521.

Gao, J., Zheng, W.-S., Pan, J.-H., Gao, C., Wang, Y., Zeng, W., & Lai, J. (2020). An asymmetric modeling for action assessment. In *ECCV* (pp. 222–238), Springer.

Gao, Y., Vedula, S. S., Reiley, C. E., Ahmidi, N., Varadarajan, B., Lin, H. C., Tao, L., Zappella, L., Béjar, B., Yuh, D. D. et al. (2014). Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In *W2CAI* (Vol. 3, p. 3).

Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., & Sun, J. (2019). Single path one-shot neural architecture search with uniform sampling. In *ECCV* (pp. 544–560).

Hu, S., Xie, S., Zheng, H., Liu, C., Shi, J., Liu, X., & Lin, D. (2020). Dsnas: Direct neural architecture search without parameter retraining. In *CVPR* (pp. 12084–12092).

Ilg, W., Mezger, J., & Giese, M. (2003). Estimation of skill levels in sports based on hierarchical Spatio-temporal correspondences. In *JPRS* (pp. 523–531), Springer.

International Swimming Federation (FINA). Fina diving rules, 2017. URL https://resources.fina.org/fina/document/2021/01/12/916f78f6-2a42-46d6-bea8-e49130211edf/2017-2021_diving_16032018.pdf.

Joachims, T. (2006). Training linear SVMs in linear time. In *SIGKDD* (pp. 217–226).

Liu, D., Li, Q., Jiang, T., Wang, Y., Miao, R., Shan, F., & Li, Z. (June 2021). Towards unified surgical skill assessment. In *CVPR* (pp. 9522–9531).

Liu, H., Simonyan, K., & Yang, Y. (2018). Darts: Differentiable architecture search. In *ICLR*.

Lu, L., Lu, Y., Yu, R., Di, H., Zhang, L., & Wang, S. (2019). Gaim: Graph attention interaction model for collective activity recognition. *TMM 22*(2), 524–539.

Malpani, A., Vedula, S. S., Chen, C. C. G., & Hager, G. D. (2014). Pairwise comparison-based objective score for automated skill assessment of segments in a surgical task. In *IPCAI* (pp. 138–147), Springer.

Martin, J., Regehr, G., Reznick, R., Macrae, H., Murnaghan, J., Hutchison, C., & Brown, M. (1997). Objective structured assessment of technical skill (OSATS) for surgical residents. *BJS, 84*(2), 273–278.

Pan, J.-H., Gao, J., & Zheng, W.-S. (October 2019). Action assessment by joint relation graphs. In *ICCV*.

Parmar, P., & Morris, B. T. (June 2019). What and how well you performed? A multitask learning approach to action quality assessment. In *CVPR*.

Parmar, P., & Tran Morris, B. (2017). Learning to score Olympic events. In *CVPRW* (pp. 20–28).

Parmar, P., Tran Morris, B. (Jan 2019). Action quality assessment across multiple actions. In *WACV* (pp. 1468–1476). https://doi.org/10.1109/WACV.2019.00161.

Pearson, K. (1913). On the probable error of a correlation coefficient as found from a fourfold table. *Biometrika*. https://doi.org/10.1093/biomet/9.1-2.22

Pérez, J. S., Meinhardt-Llopis, E., & Facciolo, G. (2013). Tv-l1 optical flow estimation. In *IPOL* (pp. 137–150).

Pham, H., Guan, M. Y., Zoph, B., Le, Q. V., & Dean, J. (2018). Efficient neural architecture search via parameters sharing. In *ICML* (pp. 4092–4101).

Pirsiavash, H., Vondrick, C., & Torralba, A. (2014). Assessing the quality of actions. In *ECCV* (pp. 556–571), Springer.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *TNN, 20*(1), 61–80.

Sharma, Y., Bettadapura, V., Plötz, T., Hammerla, N., Mellor, S., McNaney, R., Olivier, P., Deshmukh, S., McCaskie, A., & Essa, I. (2014). *Video based assessment of OSATS using sequential motion textures*, Georgia Institute of Technology.

Shu, T., Todorovic, S., Zhu, S.-C. (2017). Cern: Confidence-energy recurrent network for group activity recognition. In *CVPR* (pp. 5523–5531).

Tang, Y., Ni, Z., Zhou, J., Zhang, D., Lu, J., Wu, Y., & Zhou, J. (2020) Uncertainty-aware score distribution learning for action quality assessment. In *CVPR* (pp. 9839–9848).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L.u., & Polosukhin, I. (2017). Attention is all you need. In *NeurIPS* (pp. 5998–6008). Curran Associates, Inc.,. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Wang, M., Ni, B., & Yang, X. (2017). Recurrent modeling of interaction context for collective activity recognition. In *CVPR* (pp. 3048–3056).

Wu, J., Wang, L., Wang, L., Guo, J., & Wu, G. (2019). Learning actor relation graphs for group activity recognition. In *CVPR* (pp. 9964–9974).

Xie, S., Zheng, H., Liu, C., & Lin, L. (2018). Snas: Stochastic neural architecture search. In *ICLR*.

Xu, C., Fu, Y., Zhang, B., Chen, Z., Jiang, Y.-G., & Xue, X. (2018). Learning to score the figure skating sports videos. arXiv preprint arXiv:1802.02774.

Yan, R., Tang, J., Shu, X., Li, Z., & Tian, Q. (2018a). Participation-contributed temporal dynamic model for group activity recognition. In *ACM MM* (pp. 1292–1300).

Yan, S., Xiong, Y., & Lin, D. (2018b). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

Yao, T., Mei, T., & Rui, Y. (2016). Highlight detection with pairwise deep ranking for first-person video summarization. In *CVPR* (pp. 982–990).

Zeng, L.-A., Hong, F.-T., Zheng, W.-S., Yu, Q.-Z., Zeng, W., Wang, Y.-W., & Lai, J.-H. (2020). Hybrid dynamic-static context-aware attention network for action assessment in long videos. In *ACM MM* (pp. 2526–2534).

Zhang, P., Tang, Y., Hu, J.-F., & Zheng, W.-S. (2019). Fast collective activity recognition under weak supervision. *TIP 29*, 29–43.

Zhang, Q. & Li, B. (2011). Video-based motion expertise analysis in simulation-based surgical training using hierarchical dirichlet process hidden Markov model. In *MMAR* (pp. 19–24), ACM.

Zhang, Q., & Li, B. (2015). Relative hidden Markov models for video-based evaluation of motion skills in surgical training. *TPAMI, 37*(6), 1206–1218.

Zhang, Y., Wang, C., Wang, X., Zeng, W., & Liu, W. (2020). Fairmot: On the fairness of detection and re-identification in multiple object tracking. arXiv preprint arXiv:2004.01888.

Zhu, K. & Wu, J. (2021). Residual attention: A simple but effective method for multi-label recognition. In *ICCV* (pp. 184–193).

Zia, A., & Essa, I. (2018). Automated surgical skill assessment in RMIS training. *IJCARS, 13*, 731–739.

Zia, A., Sharma, Y., Bettadapura, V., Sarin, E. L., Ploetz, T., Clements, M. A., & Essa, I. (2016). Automated video-based assessment of surgical skills for training and evaluation in medical schools. *IJCARS, 11*(9), 1623–1636.

Zia, A., Sharma, Y., Bettadapura, V., Sarin, E. L., & Essa, I. (2018). Video and accelerometer-based motion analysis for automated surgical skills assessment. *IJCARS, 13*(3), 443–455.