Pattern Recognition 140 (2023) 109595

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Semantics-enhanced early action detection using dynamic dilated convolution

Matthew Korban[a], Xin Li[b,*]

[a] Room 328, Rice Hall, 85 Engineer's Way, Charlottesville, VA 22903, United States
[b] Section of Visual Computing and Computational Media, School of Performance, Visualization, and Fine Arts, Texas A&M University, TX 77845, United States

ARTICLE INFO

Article history:
Received 18 September 2022
Revised 19 March 2023
Accepted 3 April 2023
Available online 5 April 2023

Keywords:
Early action detection
Action semantics
Dilated convolutional network

ABSTRACT

This paper proposes a new pipeline to perform early action detection from skeleton-based untrimmed videos. Our pipeline includes two new technical components. The first is a new Dynamic Dilated Convolutional Network (DDCN), which supports dynamic temporal sampling and makes feature learning more robust against temporal scale variance in action sequences. The second is a new semantic referencing module, which uses identified objects in the scene and their co-existence relationship with actions to adjust the probabilities of inferred actions. Such semantic guidance can help distinguish many ambiguous actions, which is a core challenge in the early detection of incomplete actions. Our pipeline achieves state-of-the-art performance in early action detection in two widely used skeleton-based untrimmed video benchmarks. The source codes are available at: https://github.com/Powercoder64/DDCN_SRM.

1. Introduction

Early action detection is to identify an action before the action is completed. It has many applications in smart surveillance systems, robotics, and autonomous driving [1]. In practical action detection tasks, the detection needs to run on a streaming, untrimmed video to identify both the action type and its starting/ending frames. Such a detection is challenging, as different actions may have different paces and lengths, and different actions could have similar beginning motions, making reliable prediction difficult. Because of these issues, standard action recognition techniques, which run on complete, trimmed video clips, often do not apply effectively here.

Temporal Scale Variance. In different videos, similar or the same actions can have different lengths, as different people may perform actions (or different portions of action) at different paces. Such a property of actions, so-called "temporal scale variance", has been shown to hamper the accuracy of many action detection systems [2]. An example is shown in Fig. 1. When conventional action detection networks use static temporal windows to process a certain number of frames and detect actions, a long action such as the "triple jump" only gets partially sampled, and not all the key poses in the key stages of the run, hop, step, and jump will be observed. And furthermore, if the pace of the video changes, the predetermined sampling of frames is often not optimal. Consequently, this action is often identified as a "sprint" or "long jump" action.

One way to tackle long videos with temporal scale variance is to capture the most representative frames, or keyframes, from videos. However, existing key frame extraction approaches need a separate module to detect keyframes. Training such a module is non-trivial [3], and often rely on a large amount of expensive manually labeled data. Some recent methods propose to develop self-supervised keyframe detection such as Self-attentive networks [4] or Collaborative Learning [5] to circumvent expensive manual labeling. But these methods still have two limitations: (1) they are *computationally inefficient*, and (2) they capture features from individual key-frames ignoring the *temporal dependencies* among them. First, having a *computationally efficient* system is critical since the prediction is expected to be made as early as possible and before the action is completed. Second, modeling temporal dependencies is important in action analysis [6] as actions are about the changes in a person's movements in the temporal dimension.

Another strategy to handle temporal scale variance is through using Temporal Dilated Convolutional Networks (TDCN) [7]. The main idea of TDCN is to use a hierarchical temporal structure. Different time intervals, as sub-parts of the hierarchy, are assigned to different convolutional layers. With different layers of the TDCN network extracting features using different temporal intervals, TDCN can better detect incomplete sequences with missing frames. Recent TDCN design often adopts an exponential dilated structure, and this allows it to capture *long-term temporal de-*

* Corresponding author.
E-mail addresses: acw6ze@virginia.edu (M. Korban), xinli@tamu.edu, xin.shane.li@ieee.org (X. Li).

https://doi.org/10.1016/j.patcog.2023.109595
0031-3203/

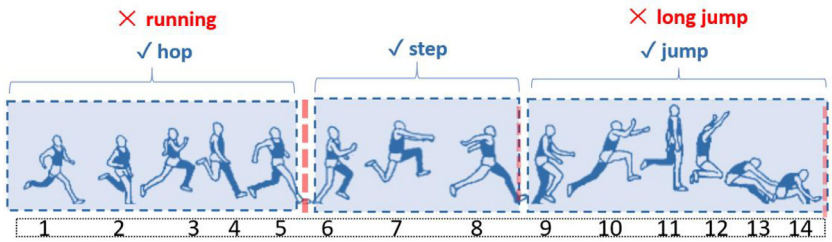

Fig. 1. An example of the temporal scale variance issue: A long action class "triple jump", and the consequent incorrect action detection.

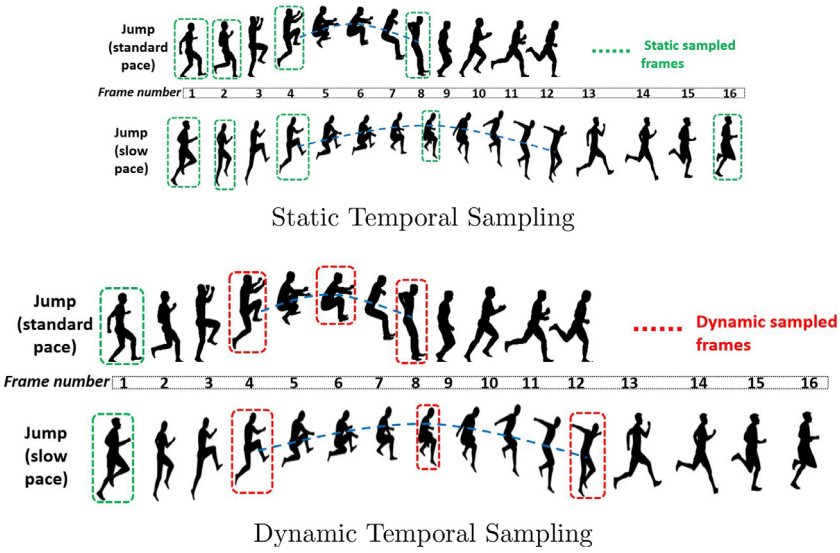

Fig. 2. Static temporal sampling: different sampled frames will be extracted from similar action sequences with different paces. Dynamic temporal sampling: frames are more consistently sampled according to the paces.

pendencies. It also shares weights between different convolutional layers, and this makes it efficient in processing long untrimmed temporal sequences [8]. TDCN has been used to achieve state-of-the-art performance in handwriting recognition [9], sign language recognition [8], and action recognition [10]. However, most existing TDCN adopts a *(static) temporal sampling*, where the intervals are pre-designed in a heuristic manner, based on important frames of actions following standard paces. Consequently, when the given action has a different pace, existing TDCN could fail to recognize them correctly.

We propose a new Dynamic Dilated Convolutional Network (DDCN) to tackle the temporal scale variance in untrimmed actions. Specifically, we enhance the TDCN with a new *dynamic temporal sampling* scheme. Without needing to perform an extra keyframe extraction, the DDCN is an end-to-end network without bringing in much latency during online prediction when processing long untrimmed videos.

Through training, DDCN aims to find the optimal temporal sampling distribution and store them in a set of channels, so that actions with temporal scale variance (e.g., different paces, framerates) can be better modeled. Fig. 2 illustrates the differences between the static and dynamic temporal sampling in dealing with videos with temporal scale variance. When using the static temporal sampling, from two similar actions with different paces, we get quite different sampled frames (in green). Consequently, building a stable and robust action feature becomes harder. On the other hand, when utilizing our dynamic temporal sampling, a more adaptive sampling on frames (in red) can be obtained. Such a better temporal sampling makes the robust feature modeling of action under temporal scale variance noticeably easier.

We redesigned the standard DCN's hierarchical dilated structure to make it more effective in dealing with temporal scale variance. Our new design includes our novel dynamic temporal sampling, dynamic dilated layer aggregation algorithm, and a new loss function to accommodate the dynamic temporal updates in our DDCN.

Semantic Ambiguity from Similar Motions. Many actions have intrinsically similar motions. Some examples can be seen in Fig. 3, where "opening a cabinet" (row-1), "opening a fridge" (row-2), and "opening a microwave" (row-3) all involve similar motions. These similar motions are hard to differentiate in current action recognition or detection systems, especially if an early prediction on incomplete actions is needed. Our observation is that relevant objects in the scene, or we call *semantic references*, can provide useful information to help tackle such ambiguity. For example, here by using semantic references, "cabinet", "fridge", and "microwave", these similar motions can be distinguished.

Existing action detection/recognition strategies have not effectively modeled and used semantic references. Current strategies can be classified into two main categories: skeleton-based and image-based methods. (1) The skeleton-based methods predict actions using skeleton joints. Hence, semantic references from background contents are not considered. (2) The image-based approaches use 2D convolutions to encode action-related information in the images. Certain background information may be encoded into action features, but they are still insufficient to effectively model semantic reference information. This is because they are not designed to explicitly model semantic relevance between reference objects and actions; consequently, they often include a lot of semantically irrelevant background information, which actually reduces the expressive power of the action features [4].

Fig. 3. An example of ambiguity in detecting similar-looking actions: "opening a cabinet" (above row), "opening a fridge" (middle row), and "opening a microwave" (bottom row). The actions movements look similar without the corresponding semantic references, "cabinet", "fridge", and "microwave".

In this work, we propose to develop a new "Semantic Referencing Module" (SRM) to learn and utilize semantic reference information to help reduce ambiguity from similar-looking incomplete actions. Unlike most traditional image-based strategies, our method can better detect/select action-relevant semantic information in images and discard irrelevant information. This can significantly increase early action detection accuracy.

Our SRM includes several components to capture the correlations between semantics and actions. We first capture an initial set of informative features from action videos and then use a recommendation system (i.e., Implicit Matrix Factorization (IMF) algorithm) to discard less important features. We redesign the IMF algorithm, commonly used for recommendation tasks in online stores, and integrate it into our SRM to be effective for early action detection. Finally, we convert the recommendation system output to action detection scores.

The main **contributions** of this paper are:

(1) We propose a novel Dynamic Dilated Convolutional Network (DDCN) to handle the temporal scale variance in incomplete actions from untrimmed videos.
(2) We design a new Semantic Reference Module (SRM) to suggest relevant semantic objects to distinguish similar-looking actions.
(3) We conducted thorough experiments on untrimmed action benchmarks, PKU-MMD and OAD. Combining the above two new designs, our proposed pipeline outperforms the state-of-the-art early action detection systems.

2. Previous work

This section discusses the existing strategies related to our work and the challenges we aim to overcome. First, we explain the online action detection work, followed by early action detection. Then we discuss the previous strategies proposed to handle the two main challenges in action detection and early action detection, which are "Temporal Scale Variance" and "Semantic Ambiguity".

Online Action Detection. Online action detection aims to identify actions in untrimmed videos in real-time. Online action detection is often performed on long videos that introduces new challenges. To handle these challenges, researchers proposed different strategies. For example, [11] used unidirectional and bidirectional LSTMs to deal with short sequences in online early action detection and long sequences in offline action detection. [12] suggested a Knowledge distillation strategy that transfers the knowledge from a teacher in offline shorter clips to online longer student clips [13]. proposed a method based on Recurrent Neural Network (RNN) namely temporally smoothing network to smooth per-frame of long videos. [14] introduced an online temporal classification model, that jointly with an action inference graph can detect human action from long videos more efficiently. [15] modeled human appearance based on the regions associated with human skeleton joints and evaluated the temporal consistency of human poses in different frames in real-time. [16] suggested two novel modules, Temporal Label Aggregation and Dense Probabilistic Localization (DPL) to handle the uncertain action annotations, which is a common issue in annotating long videos.

Early Action Detection. Early action detection aims to identify incomplete actions in streaming video sequences. Several strategies have been suggested to deal with incomplete actions.

One strategy is modeling temporal information hierarchically. Hierarchical information can be levels of movement [17], hierarchical correlations between partial sequences and corresponding temporal segments [18], and hierarchical temporal correlations among various action classes [19]. These algorithms however are computationally expensive and mostly require a separate step to extract temporal hierarchical correlations in actions. A more efficient algorithm to capture temporal information online is desirable. Another strategy to enhance the detection of incomplete actions studies is using information captured from full videos. To achieve this, several algorithms have been proposed such as deep sequential context network (DeepSCN) [20] and teacher-student learning scheme [21]. While using information from full videos can be beneficial for early action detection, besides its computational complexity issue, it often does not handle long ambiguous actions, as an example given in Fig. 2.

Temporal Scale Variance in Action Segmentation. Temporally localizing the action in untrimmed videos is referred to as *action segmentation*. A major challenging issue towards reliable action segmentation is handling temporal scale variance. [22] padded the videos to a fixed size and utilized an Autoencoder to segment actions in videos [2]. suggested a Deformable Temporal Residual Module to learn multi-scale temporal information through multi-scale pooling layers. [23] introduced a graph convolutional network to convert the varied-size graphs to fixed-size feature maps [24]. proposed to restrict the duration of actions to reasonable lengths. These existing action segmentation meth-

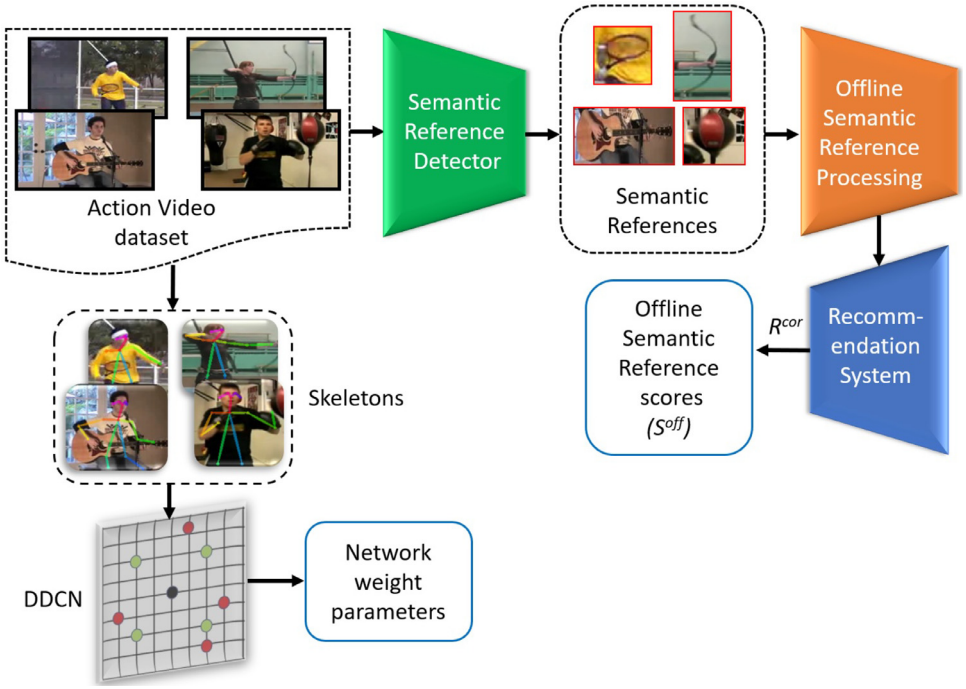

Fig. 4. Offline phase: in this phase, we first use the "Semantic Reference Detector" identify semantic references in the scene. Next, in the "Offline Semantic Reference Processing" step we extract the initial set of semantic features. Next, we use a recommendation system to select the most informative semantic features that describe the co-existence between semantic references and actions. Then we convert the recommendation system outputs to the "Offline Semantic Reference scores" which are used to predict the actions. On the other side, we trained our DDCN and dynamic sampling based on skeleton pose sequences and store the network weight parameters.

ods have some limitations, including (1) expensive computational cost for real-time applications, (2) only capable to handle complete action sequences, and (3) failing to handle significant temporal scale variance. For early action prediction, we need an efficient pipeline that runs on incomplete sequences with temporal scale variance.

Temporal Scale Variance in Action Detection. Performing both action segmentation and classification in untrimmed videos is often referred to as *action detection* in literature.

Some action detection systems extract keyframes/keyposes from untrimmed action videos and utilize them to perform classification and tackle temporal scale variance. Self-attention networks were adopted in some pipelines [25] to detect keyframes in untrimmed action videos. While [4] only used a temporal attention network, [25] utilized two spatial and temporal attention networks to detect temporal keyframes and spatial key-pixels. The spatial features are extracted using a Convolutional Neural Network (CNN), and temporal features are computed using a Recurrent Neural Network (RNN). [26] suggested an approach to detect early actions in continuous untrimmed videos by selecting keyposes. To pick the keyposes, each pose is matched with those template poses in the datasets using the Dynamic Time Warping (DTW) algorithm; then the Markov chain is exploited to classify the features extracted from keyposes.

Semantic Ambiguity in Early Action Detection. The ambiguity of similar-looking actions is a critical challenge in early action detection. To resolve this issue, several solutions have been suggested such as using action-specific intermediate features [27] and remembering hard-to-predict actions [28]. While these algorithms are effective in certain scenarios, they use only temporal information to resolve the ambiguity in similar-looking actions. However, many ambiguous actions can be only distinguishable by relevant spatial semantic information. An example of such similar-looking actions is shown in Fig. 3.

3. Methodology

3.1. Overview and terminologies

Our proposed pipeline consists of an offline (Fig. 4) and an online (Fig. 5) phase.

Offline Phase. During the offline phase, the input of our pipeline is video sequences $V_D = \{V^k, k = 1, 2, \ldots, I\}$ from the action dataset. Our DDCN module takes as input the skeleton data, which can be obtained from the action videos through pose estimation (using, e.g., OpenPose [29] in our work), and then extracts different combinations of dynamic sampling points to enhance the temporal samplings for effective classification of different action classes $c^k \in C$, where C is the set of action classes. The extracted sampling positions and weights are stored in multiple channels. More details of this proposed dynamic sampling are discussed in Section 3.2. In the SRM module, first, we train an *object detector* using a *Semantic Reference Detector* [30] (green box in Fig. 4) to identify reference objects that co-exist with corresponding actions in V_D. Next, we extract a list of *offline semantic reference attributes* X^{off}, which consists of *occurrence frequency of reference objects* (number of occurrence) p, *movement* (shift of reference objects) z, and *object* detection confidence c. From these attributes X^{off} we can calculate initial semantic correlation ratings $R^{\text{cor}} = r_{m,n}$ between each pair of reference object m and action class n. A recommendation system is then adopted to refine X^{cor} to get the final *offline semantic reference scores* S^{off}.

Online Phase. During the online phase, in the SRM module, given an action video sequence $V^k = \{v_t, t = 1, 2, \ldots, T\}$, we first compute the *online semantic reference attributes* X^{on} which consists of similar items of X^{off} for each frame $t \in T$. X^{on} helps the pipeline better predict the action in each time step $0 \leq t \leq T$. Then, from X^{on} and S^{off}, we compute the *online semantic reference scores*

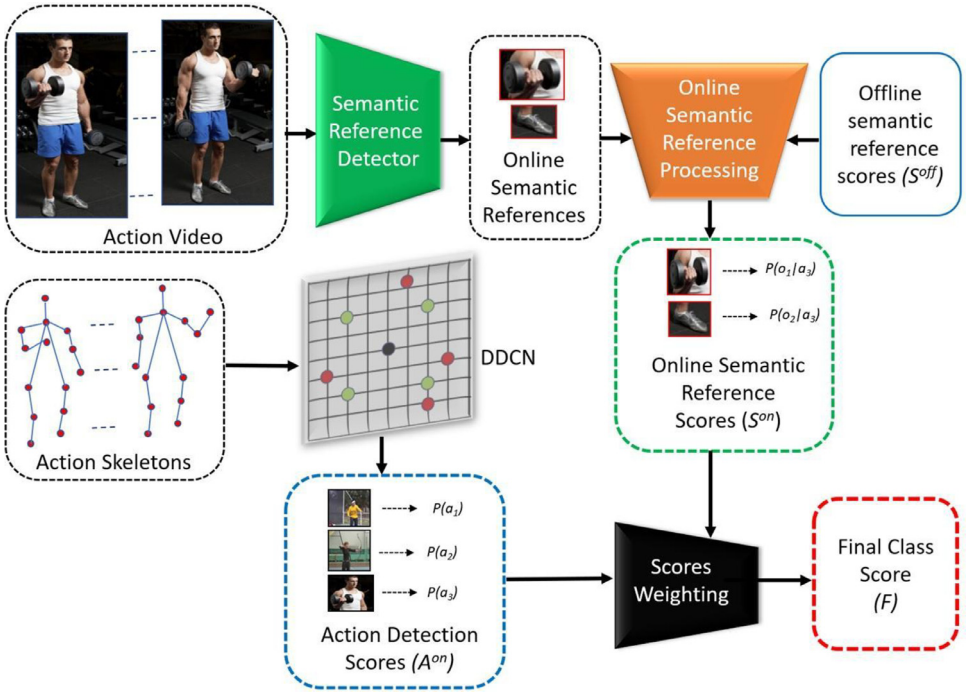

Fig. 5. Online phase: in this phase, we first use the "Semantic Reference Detector" to identify semantic references in the scene. Next, "Online Semantic Reference Processing" extracts the set of semantic features which are compared to the previously calculated offline semantic reference scores to compute the "Online Semantic Reference Scores". These scores indicate the joint probability of detected semantic reference occurrence with different action classes. Simultaneously, our DDCN produces the "Action Detection Scores" by processing online skeleton sequences and trained dynamic sampling. Finally, we use a "Score Weighting" strategy to compute the "Final Class Score" based on the two described scores.

S^{on} for each action class. S^{on} indicates the correlation scores of online detected objects with respect to each action class. Concurrently, in the DDCN module, given the action skeleton sequence $Y^k = \{y_t, t = 1, 2, \ldots, T\}$ whose joints are computed from V^k, the DDCN uses the dynamic samplings learned in the offline stage to calculate the final action detection scores A^{on}.

The final action detection scores F is the weighted summation of the two aforementioned scores. In the following, we will elaborate on the two main components of this pipeline.

3.2. Dynamic dilated convolutional network (DDCN)

A Temporal Dilated Convolutional Network, often abbreviated as Dilated Convolutional Network (DCN), has recently been developed as a temporal deep-net structure and shown to outperform RNN and Bi-LSTM [6] in processing action data [6] and some other temporal sequences. Another benefit of using DCN is that it allows us to build deeper networks of less computational cost to capture long-range temporal dependencies and obtain better reliability on small-size datasets [10]. By expanding the receptive field, the DCN can capture long-term temporal dependencies effectively and increase the performance and efficiency of dense prediction architectures [31].

A DCN is a stack of dilated convolutions whose kernel elements expand consecutively. A DCN utilizes *temporal sliding windows* on temporal sequences to capture temporal information. The sliding windows are defined within the fixed temporal lengths and are a collection of *temporal samplings* whose inputs are consecutive time frames. In a DCN, the temporal windows follow fixed structures and are stacked as a power of 2 of dilated steps d^s (i.e., 1, 2, 4, 8, ...). Consecutive temporal samplings from different sliding windows are assigned to different convolutional layers.

In a conventional DCN, the convolutions are applied over two time steps, t and $t - s$. Thus the kernel weights can be represented as $W = \{W^{(1)}, W^{(2)}\}$. Therefore, we can define the dilated convolution [10] for layer l at time step t as

$$q_t^l = f(W^{(1)} q_{t-s}^{l-1} + W^{(2)} q_t^{l-1} + b), \tag{1}$$

where q_t^l is the dilated convolution result on time t in layer l, s is the dilation rate increasing as the power of two in consecutive layers, and b is the bias vector. The output of the network Q_t at time t is the concatenated tensor of all the dilated output layers as $Q_t = [q_t^l, q_t^{l-1}, \ldots, q_t^0]$.

To compute the feature maps, the convolutional operator is usually applied to the neighboring temporal points. So, to extract convolutional feature maps effectively, specifying appropriate temporal neighboring points is crucial. The effectiveness of the neighboring temporal points, and consequently, feature maps, dictate the accuracy of a temporal network when modeling temporal sequences. In a conventional DCN, the neighboring temporal points are defined in static positions. However, such a static temporal structure in DCN is not always effective as it may not accommodate the temporal variance in action sequences well. When dealing with spatial and temporal variance in images or videos, dynamic networks [32] have been shown to be more effective than static networks. Therefore we propose a new dynamic temporal sampling to make the DCN adaptive to varying temporal information. Specifically, the DDCN can more adaptively capture temporal neighboring points according to the data.

In our DDCN, the output of each layer, for each time step t is formulated in Eq. (2).

$$q_t^l = f(W^{(1)} q_{t-s}^{l-1} + W^{(2)} q_t^{l-1} + W^{(3)} q_{t-t_f}^{l-l_f} + b) \tag{2}$$

Compared with the static DCN in Eq. (1), the DDCN includes extra parameters of Dynamic samplings $\{t_f, l_f\}$ that are represented with

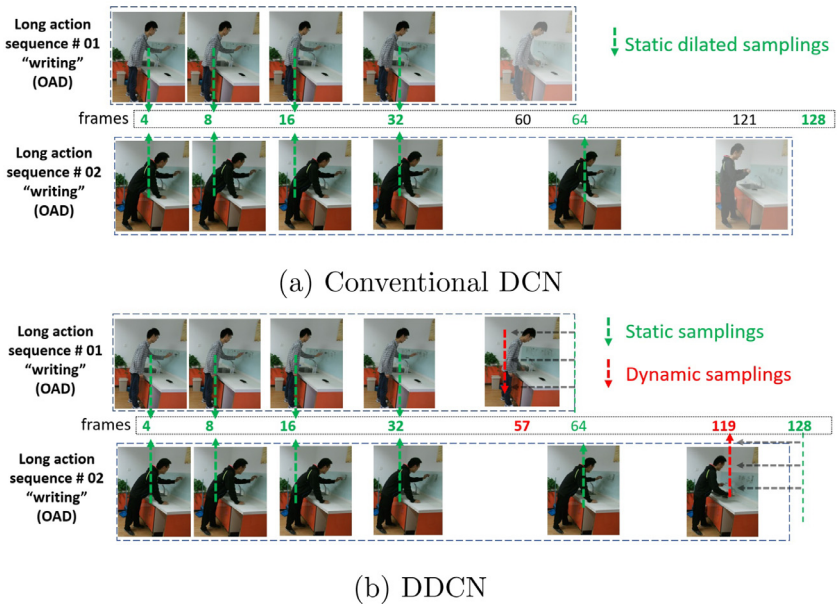

Fig. 6. The comparison of performance between a conventional DCN and our DDCN in dealing with temporal scale variance. Two similar actions "writing" with different paces are illustrated. The green and red arrows are static and dynamic temporal samplings in different frames, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

their weight $W^{(3)}q_{t-t_f}^{l-l_f}$. Here l_f and t_f are *dynamic layer offset* and *dynamic time step offset*, respectively. Specifically, t_f is calculated as

$$t_f = \arg\max_{\Delta} P(c^k|Y^k, t_f), \tag{3}$$

where $\Delta = \{0, 1, \dots \xi\}$ is the set of dilated integers (in our experiments $\xi = 8$), $c^k \in C$ is the action class, and Y^k is the sequence of input skeletons. Here

$$P(c^k|Y^k, t_f) = \arg\min_{C} Loss_A, \tag{4}$$

where $Loss_A$ is the loss function of action detection.

To ensure that the above equations are met, we train the DDCN to find the optimum $t_f \in \Delta$ so that the following $P(c^k|Y^k, t_f)$ is maximized:

$$P(c^k|Y^k, t_f) = softmax(f(W^A * Y^k + W^f * t_f)). \tag{5}$$

Here f is the activation function from the final layer, W^A is the kernel weights for the input skeleton data, and W^f is the 1D dilated kernel weights.

The dynamic layer offset, l_f is automatically updated based on t_f since, in a dilated network, different time steps are by default assigned to different layers. The dynamic samplings can be performed at any time steps to adaptively capture temporal information. Specifically, the dynamic time step offset $t_f \in \Delta$ enumerates in a 1D integer space $\mathbb{Z}$ and is updated to minimize $Loss_A$ during the training. So, the new dynamically sampled point can relocate to $t - t_f$ where more effective temporal features with respect to the previous temporal neighboring sampled points can be obtained. The output of the network at time t concatenates the outputs of all the layers.

Temporal Scale Variance. Unlike the conventional DCN that follows a static sampling structure (static temporal sliding windows), our DDCN can dynamically relocate temporal sampling structures to better accommodate temporal scale variance. An example for an action "writing" with two different paces is shown in Fig. 6. In this example, when using static temporal sampling, the sampled frames are different for the two similar actions with different paces. Such inconsistent sampling results in bigger intra-class variation, and, consequently, reduces the action detection accuracy. In contrast, our dynamic temporal sampling adjusts the sampling locations on these actions. As a result, it decreases the intra-class variation and enhances the action detection accuracy.

Identification of Similar Motions. In our pipeline, reference objects detected in the Semantic Referencing Module (SRM) help reduces the ambiguity in the detection of similar motions. But here the DDCN also helps alleviate this action ambiguity. This is because, during the training phase, DDCN selects the most distinctive temporal sampling patterns to differentiate actions in the dataset. An example that shows this between conventional DCN and DDCN is given in Fig. 7. With the static temporal sampling (green arrows), two different incomplete actions, "tearing up paper" and "reading" are similar throughout half of the video clips. In contrast, with DDCN, a more effective sampling (red arrows) helps distinguish them in the early stage of the actions.

3.3. Semantic referencing module (SRM)

Our proposed SRM contains an offline and an online phase.

3.3.1. Offline phase

Semantic Reference Attributes. The aim of the offline phase (Fig. 4) is to create the *offline semantic correlation scores* S^{off}. S^{off} is a $M \times N$ table storing the probability of correlation between the M reference objects and N action classes. S^{off} is used in the online stage to calculate the action class score S^{on}.

To this end, first, we train a trained reference object detector network [30]. This network calculates the offline semantic reference attributes $X^{\text{off}} = \{x^i_{mn}, m, n, i = 1, 2, \dots, M, N, I\}$ where x^i_{mn} is the reference object attribute m in action sample i of class n, and M, N, and I are the number of detected reference objects, number of action classes, and number of action samples in the action dataset respectively. We adopted [30] to perform reference object detection as it was reported to achieve the highest object detection accuracy (leader-board, Oct. 2020) on ImageNet [33], a widely used semantic objects dataset. Each $x^i_{mn} = \{p^i_{mn}, z^i_{mn}, c^i_{mn}\}$, and p^i_{mn}, z^i_{mn}, and c^i_{mn} are occurrence frequency, movement and detection score of reference object m in action sample i of class n.

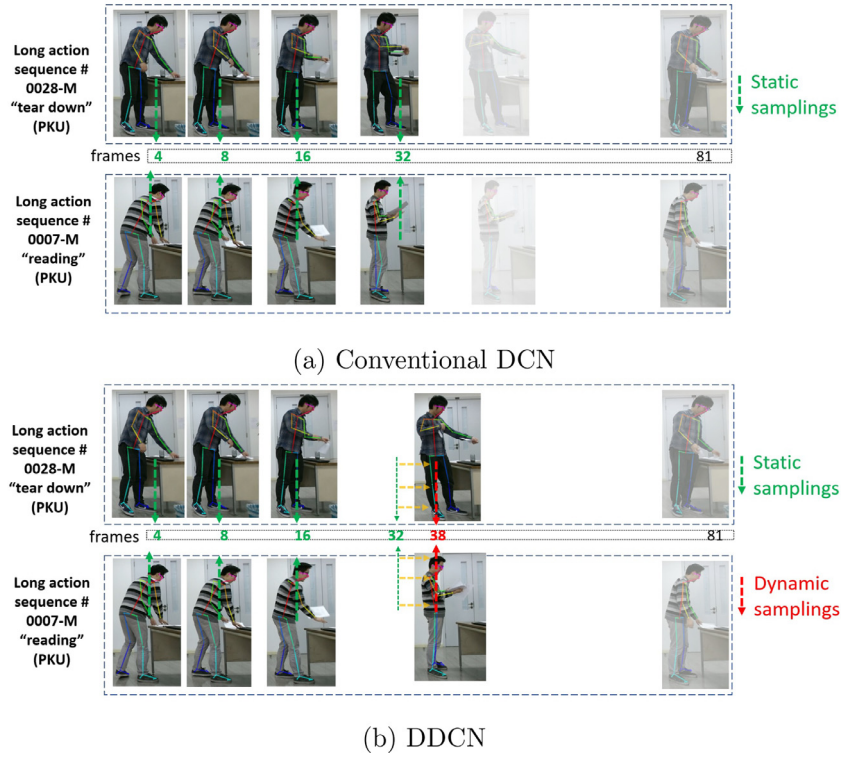

Fig. 7. The performance of a conventional DCN vs DDCN in handling the ambiguity of similar-looking actions. The conventional DCN fails, and DDCN succeeds in capturing distinguishing frames between two actions, "tear up paper" and "reading". The green and red arrows are the fixed and dynamic temporal samplings, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Recommended Semantic References. To rank the semantic references for different classes, we design a recommendation scheme based on the implicit matrix factorization algorithm [34]. This scheme is inspired by the common recommendation systems which suggest items to users by considering users' ratings through user-items interactions. In our scenario, we develop this recommendation system by replacing users with action classes and items with reference objects. First, we calculate the initial semantic correlation ratings of each semantic reference m in each sample i of action class n, $X^{\text{cor}} = \{r^i_{mn}, m, n, i = 1, 2, \ldots, M, N, I\}$, where each r^i_{mn} is obtained by

$$r^i_{mn} = c^i_{mn} \cdot (p^i_{mn} + z^i_{mn}) \tag{6}$$

We calculate X^{cor} based on three terms. First, the importance of a reference object in an action class is related to the number of occurrences p^i_{mn}. Second, in many action classes, informative reference objects are often those that are dynamic or moved with the actions. Some examples can be found in Section 4.2.2. For example, objects with the most significant movement/shift in the action classes "baseball pitch", "biking", and "tennis swing" (selected from the UCF101 dataset [35]) are "baseball bat", "bicycle" and "tennis racket" respectively. Therefore, we define the object movement z^i_{mn} as the second term in X^{cor}. Third, we also consider (here by multiplying) the detection confidence score $0 \leq c^i_{mn} \leq 1$ to adjust the correlation.

The input of the recommendation scheme is the initial semantic correlation ratings X^{cor}; its output is the offline semantic reference scores $S^{\text{off}} = \{\hat{s_{mn}}, m = 1, 2, \ldots, M, n = 1, 2, \ldots, N\}$.

We choose the Implicit Matrix Factorization (IMF) recommendation system [34] to calculate semantic reference scores for different action classes. We observed that IMF is more effective than other collaborative-based recommendation systems such as Neural Collaborative Filtering (NCF) [36] in recommending semantic references for action detection (please see Section 4.2 for more details).

3.3.2. Online phase

In the online phase, first, the reference objects attributes (X^{on}) are detected using the same network [30] applied in the offline phase. Given S^{off} and the X_{on}, for each current action frame, we calculate the *online semantic reference scores*

$$S^{on} = \frac{1}{Z} \sum_{O_{m=0}}^{M'} \hat{s_{mn}}, \tag{7}$$

where O is the online detected reference object, M' is the number of online detected reference objects, Z is a normalizing factor, $\hat{s_{mn}}$ is derived form S^{off}. S^{on} is a $1 \times N$ vector indicating the probabilities of action frame related to different action classes $0 \leq n \leq N$ based on semantic references.

Simultaneously, DDCN calculates the action detection score A^{on}, a $1 \times N$ vector indicating the probabilities of action frame related to different action classes $0 \leq n \leq N$. The final detection score for the action class n is then calculated by

$$F_n = W_R \cdot S^{on} + W_A \cdot A^{on}, \tag{8}$$

where W_R and W_A are scalar weighting factors for the online semantic correlation and initial action detection scores.

3.4. Implementation details

Here, we summarize the implementation details of our proposed algorithm:

In our experiments, missing joints (due to occlusion or pose estimation errors) are set to zero.

All of our experiments are conducted using a Linux PC (Ubuntu 18.04) with an NVIDIA GTX 1070 graphic card, Intel Xeon 24-core processor, and 32 GB of RAM. Our deep learning networks are implemented using Pytorch.

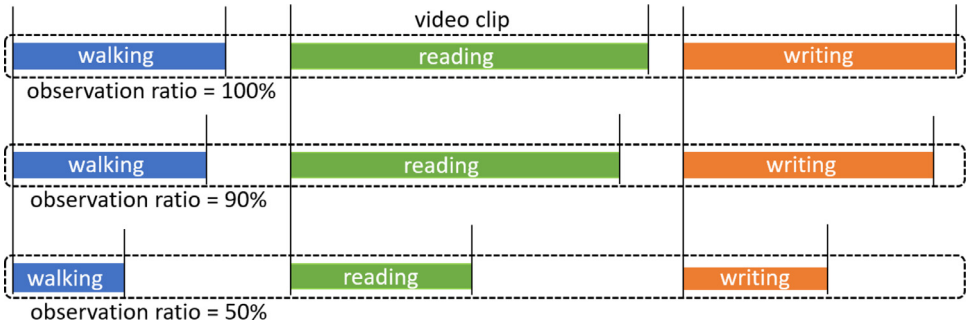

Fig. 8. An example of different observation ratios (100%, 90%, and 50%) on an untrimmed video clip used in our experiments. In this example, the untrimmed video clip includes three continuous actions:"walking", "reading", and "writing". The observation ratios are calculated based on the amount of observed part at the beginning of the actions, where the end of actions are cut..

4. Experimental results

4.1. Comparative results

We compared our algorithm with the state-of-the-art approaches. Our experiments are conducted on untrimmed continuous-streaming video clips. To test the prediction on an incomplete action, we cut the ending portion of an action video to mimic different Observation Ratio (OR). For example, 90% OR means the first 90 percent of the video sequence is available. We illustrated the above in Fig. 8, where three different observation ratios of 100%, 90%, and 50% are shown for an untrimmed video clip consisting of three continuous actions of "walking", "reading", and "writing".

In our experiments, we used the F1 score metric based on frame-level prediction. We evaluated the prediction results based on each frame to accommodate different observation ratios in untrimmed videos. The frame-level F1 score, following [1], is formulated as

$$F1(\theta) = 2 \cdot \frac{p(\theta) \times r(\theta)}{p(\theta) + r(\theta)} \tag{9}$$

where p, and r are precision and recall, which are for frame-level predictions in our experiments, and θ is the frame-level action detection threshold. For the OAD dataset, using the frame-level prediction is a standard metric. For the PKU-MMD dataset, however, rather than its original action-level metric using the F1 score, we adopted the frame-level metric using the F1 score. To evaluate the action prediction accuracy before the actions are completed, such a frame-level prediction metric is more suitable. We revised and ran the source codes of some representative full-sequence-based approaches and estimated the frame-level prediction accuracy under various observation ratios.

The experiments were conducted on two widely used datasets of continuous untrimmed actions videos, OAD [37], and PKU-MMD [38] datasets.

(1) **OAD dataset** [37] includes 59 long videos of continuous action sequences of 10 action classes. Skeletal joints are also provided together with the videos. The first 39 long videos are used for training, and the last 20 videos are used for testing. We compared our method with [37,39] (ECCV16), [40] (CVPR17), [41] (WACV17), [42] (WACV19), [43] (ICASSP17), [44] (TCSVT18), [6] (TPAMI19), [45] (TBCS2021), [46] (DT2021), and [47] (ICPR21). The experimental results for the OAD dataset are illustrated in Table 2. As shown in this table, our proposed DDCN + SRM outperforms the state-of-the-art methods on the accuracy of early action detection. Especially when the observation ratio is low (i.e., 10%), DDCN + SRM is significantly better than the others.

(2) **PKU-MMD dataset** [38] consists of 1076 long videos from 51 action categories, among which we used the cross-subject scenario

Table 1
Parameters used in our pipeline.

Parameter	Value	Section
Number of dilated integers (ξ)	8	3.2
Temporal receptive field for DDCN	1024	3.2
Dilated rate (s)	2	3.2
Number of dilated steps (d^s)	6	3.2
Number of dilated layers in DDCN	10	3.2
Number of channels in DDCN	32	3.2
Learning Rate	$1e^{-5}$	3.2
Number of skeleton joints (J)	25	3.2
Weight Decay	$1e^{-6}$	3.2
Maximum number of subjects for OAD and PKU-MMD	1 and 2	3.2
Maximum number of reference objects per frame	10	3.3
Total number of reference objects (M)	200	3.3
Number of classes for (N) for OAD and PKU-MMD	10 and 52	3.3
Final detection score weights ratio (W_A/W_R)	5.7	3.2.2

Table 2
Comparison of the proposed early action detection pipeline with existing methods on the OAD dataset under different observation ratios. The numbers with * are obtained by running the provided source codes, and the numbers without * are from the original papers, and the values are listed as "–" when unavailable or irrelevant.

Method / OR	10%	50%	90%	100%
[43]	–	–	–	63.0%
[41]	–	–	–	67.0% *
[39]	60.0%	75.3%	77.5%	82.6% *
[40]	59.0%	75.8%	78.3%	82.9% *
[37]	62.0%	77.3%	78.8%	83.0% *
[6]	72.0%	81.2%	83.7%	85.8% *
[42]	65.5%	73.0%	81.5%	85.9% *
[46]	–	–	–	87.00% *
[47]	–	–	–	88.11%
DDCN+SRM (ours)	80.2%	86.1%	89.2%	**89.6%**

Table 3
Comparison of the proposed early action detection pipeline with existing methods on the PKU-MMD dataset under different observation ratios. The numbers with * are obtained by running the provided source codes, and the numbers without * are from the original papers, and the values are listed as "–" when unavailable or irrelevant.

Method / OR	10%	50%	90%	100%
[39]	22.9%	63.0%	74.5%	76.0% *
[37]	25.3%	64.0%	73.4%	75.9% *
[40]	19.8%	62.9%	74.9%	77.1% *
[48]	–	–	–	81.5% *
[52]	–	–	–	81.4% *
[6]	33.9%	74.1%	82.9%	85.2% *
[42]	26.3%	68.3%	80.1%	85.9% *
DDCN+SRM (ours)	**39.2%**	**80.7%**	**87.1%**	**88.0%**

Table 4
Comparison of the Dynamic DCN and a conventional DCN using SRM and different recommendation systems (NCF and IMF) on the *OAD* dataset under different Observation Ratios (ORs). In this table, the DCN and DDCN encode the pose features while SRM-NCF and SRM-IMF capture object contexts.

Modules/OR	10%	20%	30%	40%	50%	90%	100%
DCN	68.5%	70.7%	73.8%	74.8%	75.9%	80.3%	81.1%
DDCN	71.8%	73.5%	76.0%	77.9%	79.0%	83.4%	85.3%
SRM-NCF	60.1%	60.9%	62.0%	63.2%	64.7%	68.1%	72.0%
SRM-IMF	64.3%	65.0%	67.1%	67.9%	69.0%	73.3%	77.1%
DCN + SRM-NCF	68.6%	68.8%	71.0%	74.2%	76.1%	80.6%	81.6%
DCN + SRM-IMF	71.3%	73.0%	76.4%	79.4%	80.0%	84.7%	85.2%
DDCN + SRM-NCF	72.0%	74.0%	76.3%	78.5%	79.3%	83.8%	85.7%
DDCN + SRM-IMF	**80.2%**	**81.2%**	**83.9%**	**84.6%**	**86.1%**	**89.2%**	**89.6%**

Table 5
Comparison of the Dynamic DCN and a conventional DCN using SRM and different recommendation systems (NCF and IMF) on the *PKU-MMD* dataset under different Observation Ratios (ORs). In this table, the DCN and DDCN encode the pose features while SRM-NCF and SRM-IMF capture object contexts.

Modules/OR	10%	20%	30%	40%	50%	90%	100%
DCN	30.8%	48.6%	55.0%	66.5%	73.9%	80.0%	81.6%
DDCN	34.1%	51.9%	57.7%	70.4%	76.0%	83.3%	84.1%
SRM-NCF	22.4%	34.0%	44.8%	51.2%	59.3%	68.0%	69.9%
SRM-IMF	25.9%	36.3%	51.9%	58.9%	62.6%	73.7%	75.6%
DCN + SRM-NCF	30.9%	39.0%	58.3%	66.0%	73.9%	80.1%	81.7%
DCN + SRM-IMF	35.0%	47.1%	63.3%	70.2%	77.8%	83.9%	85.6%
DDCN + SRM-NCF	34.1%	45.0%	63.1%	69.7%	76.1%	83.5%	84.4%
DDCN + SRM-IMF	**39.2%**	**52.4%**	**60.0%**	**74.2%**	**80.7%**	**87.1%**	**88.0%**

with 946 and 130 videos for training and testing data, respectively. We compared our method with [37,39] (ECCV16), [40] (CVPR17), [48] (TIP18), [6] (TPAMI19), [42] (WACV19), [49] (EITCV20), [50] (ICPR21), [51] (IJARS21), [52] (ICMM), and [53] (CVPR22). The results are shown in Table 3. DDCN + SRM also outperforms existing methods.

4.2. Ablation study

We conducted several ablation studies on (1) different components of the proposed approach, (2) SRM for semantic references, (3) DDCN on temporal scale variance, and (4) weight weights in the loss function.

4.2.1. Analysis on different components

In Table 4 and Table 5, we illustrate the network performance when using dynamic sampling and using different recommendation systems for OAD and PKU-MMD datasets, respectively. These results show that with dynamic samplings, DDCN improves the performance of the conventional DCN in early action prediction. The Implicit Matrix Factorization (IMF) algorithm [34] is more effective than other collaborative-based methods, such as Neural Collaborative Filtering (NCF) [36] in sorting semantic references.

The traditional DCN and our DDCN are used to extract human skeleton pose information. Meanwhile, the SRM module, with IMF and NCF recommendation systems, are used to capture object contexts (semantics). The experimental results indicate that (1) using pose information (DDCN) alone contributes more to the performance gain, than using object contexts (SRM) alone; and (2) the combination of DDCN and SRM-IMF leads to the best overall performance.

4.2.2. Semantic references

Table 6 illustrates the normalized covariance matrix of the correlations between the semantic reference objects and actions. The semantic reference O_m for action class a_n is rated based on the sum of the recommended probability of semantic reference object occurring in action class sample i of class n:

$$\sum_{i=0}^{I} \hat{s_{mn}^{i}} = \sum_{i=0}^{I} P(O_m|a_n^i), \tag{10}$$

Where I is the number of samples in class n. In our experiments, for the OAD dataset (where actions are captured in a controlled environment), we have detected total of 29 semantic reference objects for 10 action classes.

As can be seen in Table 6, action classes are often correlated with unique combinations of semantic references. These combinations help distinguish different action classes. Many of these semantic references are logically connected to corresponding actions. For example, the action “wiping” in the kitchen in this dataset turns out to be highly correlated with “microwave”; “Eating” and “bowl” are highly correlated.

We also tested the recommendation system on the UCF101 dataset [35], a widely used action videos dataset that includes a great variety of semantic objects. To train the recommendation sys-

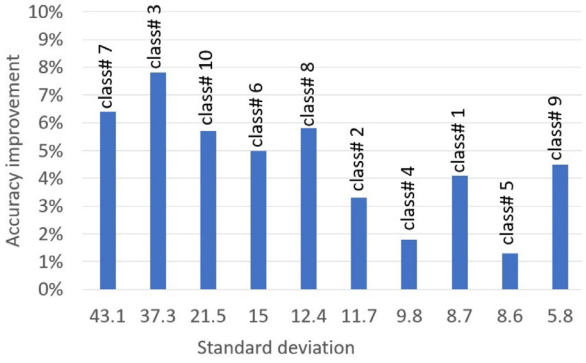

Fig. 9. DDCN's performance gain (y-axis) under different temporal scales. Here the temporal scale variance is described using the standard division (x-axis) of the video lengths (frame #) of each action class in the OAD dataset.

Table 6
Covariance matrix of the correlations between the actions (A) and semantic reference objects (O) in the OAD dataset. The action labels are abbreviated as: Drinking (Dr), Eating (Ea), Writing (WR), Opening Cupboard (Cu), Opening Microwave (Mi), Washing (Wa), Sweeping (Sw), Gargling (Ga), Throwing Trash (Tr), and Wiping (Wi).

microw	0.79	0.79	0.61	0.48	1	0.35	0.82	0.45	0.75	1
sink	0.8	1	0	0.33	0.48	0.99	0.54	0.94	0.76	0.29
chair	0.31	0	1	0.69	0.32	0.82	0.5	0.66	0.36	0.33
plant	0.9	0.52	0.79	0	1	0.57	0.42	0.87	0.37	0.88
table	0.83	0	0.77	0.3	0.1	0.54	0.88	0.48	0.76	1
mouse	0.35	0.11	0.19	0.74	0.12	0.18	0.47	0	1	0.4
vase	0.38	0.35	1	0.67	0.91	0.95	0	0.93	0.14	0.04
cup	0.94	0.11	0.87	0.61	0	0.53	1	0.6	0.31	0.88
remote	0.77	1	0.29	0.27	0.18	0.21	0	0.86	0.47	0.18
bowl	0.65	1	1	0.78	0.92	0.64	0.7	0.68	0.7	0.76
phone	0.77	0.79	0.09	0.69	0.25	0.11	0.91	0	0.97	1
knife	0.17	0	0.5	1	0.51	0.42	0.55	0.14	0.64	0.35
ball	0.26	1	0	0.34	0.11	0.26	0.23	0.32	0.26	0.14
toilet	0.27	1	0.99	0.37	0.82	0.9	0.98	0.64	0	1
suitcase	0	0.51	0.46	1	0.89	0.44	0.1	0.28	0.29	0.01
handbag	0.93	0.38	0.23	0.6	0.66	0.06	0	0.49	1	0.38
backpack	0.02	0.28	0.17	1	0.26	0.21	0.28	0	0.46	0
tv	0.08	0.35	0.38	1	0	0.28	0.41	0.02	0.59	0.23
bird	0	0.38	0.42	1	0.67	0.68	0.6	0.44	0.45	0.05
bottle	0.83	0.07	0.19	1	0.19	0.19	0.42	0	0.6	0.13
book	0	0.23	1	0.92	0.82	0.57	0.95	0.08	0.29	0.96
laptop	0.03	0.07	0.2	0.17	1	0.47	0.1	0.28	0	0.05
tie	0.48	0.2	0.16	0.37	1	0	0.26	0.17	0.37	0.53
clock	0.84	1	0.05	0.64	0.57	0.28	0	0.65	0.91	0.13
glass	0	0.58	0.48	1	0.97	0.45	0.07	0.32	0.28	0.02
spoon	0.63	0.99	0	0.58	1	0.5	0.18	0.51	0.79	0.13
bat	0	0.44	0.45	0.85	0.66	0.6	1	0.15	0.42	0.58
skboard	0	0.12	1	0.88	0.74	0.57	0.44	0.29	0.19	0.49
drier	0.45	0.89	0.7	1	0.77	0.75	0.23	1	0.5	0

tem model, we utilized all the videos in the dataset with a sampling rate of 5 frames per action clip. So, totally we created 63,689 semantic reference rating samples including 101 action classes and 79 semantic references. We show some of the recommended semantic reference objects for some classes in Table 7. Each row shows the "Score" values without normalization. From this table, we can see a strong correlation between top-rated semantic reference objects and corresponding action classes. For example, top-rated semantic reference objects for action class 11, "Biking", are "bicycle", "car", "chair", "motorcycle", "parking meter", and "traffic light", which are most commonly seen on the road.

4.2.3. Temporal scale variance

Fig. 9 illustrates the correlation between temporal scale variance and accuracy improvement achieved by the DDCN compared to a conventional DCN. Here, the standard deviation values are obtained from the set of video length values of the sample points on each action class. In this example, the standard deviation values show the temporal scale variance in action samples of 10 action classes from the OAD dataset. A higher standard deviation value indicates a higher variation in the number of frames (or temporal scale variance) for an action class. As shown in Fig. 9, there is a strong correlation between accuracy improvement and the stan-

Table 7
Top recommend semantic references for some example classes of UCF101 dataset.

class #	class name	top 1	top 2	top 3	top 4	top 5
4	Baby-Crawling	dog	chair	tv	sports ball	remote
Score:		104.5	92.9	82.1	74.9	70.7
7	Baseball Pitch	baseball bat	dog	car	baseball glove	horse
Score:		130.4	106.3	94.4	86.4	84.4
11	Biking	bicycle	car	chair	motor-cycle	parking meter
Score:		128.9	109.0	98.0	80.9	76.1
20	Brushing Teeth	tooth-brush	chair	tie	handbag	dog
Score:		107.4	99.8	90.9	84.5	80.6
49	Kayaking	boat	car	surf-board	cow	baseball bat
Score:		117.7	107.4	98.0	80.8	76.1
81	Skiing	skis	skate-board	snow-board	bird	dog
Score:		110.8	99.6	95.6	79.3	75.8
92	Tennis Swing	tennis racket	sports ball	car	potted plant	baseball bat
Score:		123.4	111.0	90.7	90.2	75.0
95	Typing	keyboard	laptop	cup	tv	book
Score:		102.9	84.6	80.9	73.5	70.5
99	Wall Pushups	refri-gerator	chair	tv	cell phone	remote
Score:		80.7	75.2	67.5	65.1	62.9

dard deviation values that indicates the impact of our DDCN in handling the temporal scale variance. For example, when the standard deviation for action classes are 37.3 and 43.1, DDCN achieves accuracy improvements of 7.8% and 6.4%, respectively.

5. Conclusions

We proposed a new pipeline to perform early action detection in untrimmed videos. Our pipeline contains two new technical components: (1) a Dynamic Dilated Convolutional Network (DDCN) to better handle the temporal scale variance, and (2) a Semantic Referencing Module (SRM) to reduce ambiguity in differentiating similar-looking actions. Our proposed pipeline outperforms state-of-the-art algorithms on two widely used untrimmed skeleton-based action datasets, PKU-MMD and OAD.

The main **strengths** of our work are: (1) our DDCN can detect actions on the videos with various temporal scales in an end-to-end and data-driven manner. Unlike most existing approaches that need a separate stage to handle temporal scale variance, our design improves the robustness and efficiency of the detection. (2) Our SRM effectively encodes semantic information about the scene, which cannot be captured using current skeleton-based approaches. This helps reduce ambiguity in early action detection. Both DDCN and SRM can be used as general modules in various recognition pipelines to deal with temporal scale variance and semantics enhancement.

Limitations and Future Work. Firstly, the SRM module relies on effective object detection in image frames. Detection errors caused by occlusion and noise could affect semantics inference. While the object detection confidence score was introduced in SRM to reduce its sensitivity against detection error, non-standard or incomplete objects in some frames can still cause detection failure and perturb the accuracy. Adopting some video-based object detection techniques could utilize temporal information from the video to reduce jettison. Secondly, while we believe our designs are general, our current pipeline was only tested on datasets that come with skeleton information. In the near future, we will explore the generalization of our pipeline to more general datasets such as ANET and THUMOS'14.

Declaration of competing interest

No conflict of interest exits in the submission of this manuscript, and manuscript is approved by both authors for publication. We declare that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part.

Data availability

No data was used for the research described in the article.

Acknowledgements

This work was partly supported by National Science Foundation of USA CBET-2115405.

References

[1] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, T. Tuytelaars, Online action detection, in: ECCV, Springer, 2016, pp. 269–284.
[2] P. Lei, S. Todorovic, Temporal deformable residual networks for action segmentation in videos, in: CVPR, 2018, pp. 6742–6751.
[3] L. Jiao, R. Zhang, F. Liu, S. Yang, B. Hou, L. Li, X. Tang, New generation deep learning for video object detection: a survey, NNLS (2021).
[4] X. Zhang, H. Shi, C. Li, K. Zheng, L. Zhu X.and Duan, Learning transferable self-attentive representations for action recognition in untrimmed videos with weak supervision, in: AAAI, volume 33, 2019, pp. 9227–9234.
[5] L. Sheng, D. Xu, W. Ouyang, X. Wang, Unsupervised collaborative learning of keyframe detection and visual odometry towards monocular deep slam, in: ICCV, 2019, pp. 4302–4311.
[6] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, A.C. Kot, Skeleton-based online action prediction using scale selection network, TPAMI 42 (6) (2019) 1453–1467.
[7] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, Wavenet: a generative model for raw audio, arXiv (2016).
[8] J. Pu, W. Zhou, H. Li, Dilated convolutional network with iterative optimization for continuous sign language recognition, in: IJCAI, volume 3, 2018, p. 7.
[9] A. Sharma, D.B. Jayagopi, Towards efficient unconstrained handwriting recognition using dilated temporal convolution network, ESAA 164 (2021) 114004.
[10] B. Xu, H. Ye, Y. Zheng, H. Wang, T. Luwang, Y. Jiang, Dense dilated network for video action recognition, TIP 28 (10) (2019) 4941–4953.
[11] F. Carrara, P. Elias, J. Sedmidubsky, P. Zezula, Lstm-based real-time action detection and prediction in human motion streams, Multimed. Tools Appl. 78 (19) (2019) 27309–27331.
[12] P. Zhao, L. Xie, J. Wang, Y. Zhang, Q. Tian, Progressive privileged knowledge distillation for online action detection, Pattern Recognit. 129 (2022) 108741.
[13] Y.H. Kim, S. Nam, S.J. Kim, Temporally smooth online action detection using cycle-consistent future anticipation, Pattern Recognit. 116 (2021) 107954.
[14] G.M.E. Elahi, Y.-H. Yang, Online temporal classification of human action using action inference graph, Pattern Recognit. (2022) 108972.
[15] K. Soomro, H. Idrees, M. Shah, Online localization and prediction of actions and interactions, TPAMI 41 (2) (2018) 459–472.
[16] J. Xu, G. Chen, N. Zhou, W.-S. Zheng, J. Lu, Probabilistic temporal modeling for unintentional action localization, TIP 31 (2022) 3081–3094.
[17] T. Lan, T.-C. Chen, S. Savarese, A hierarchical representation for future action prediction, in: ECCV, Springer, 2014, pp. 689–704.
[18] Y. Kong, Y. Fu, Max-margin action prediction machine, TPAMI 38 (9) (2015) 1844–1858.
[19] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, J. Lai, J. Zhang, Early action prediction by soft regression, TPAMI 41 (11) (2018) 2568–2583.

[20] Y. Kong, Z. Tao, Y. Fu, Deep sequential context networks for action prediction, in: CVPR, 2017, pp. 1473–1481.
[21] X. Wang, J.-F. Hu, J.-H. Lai, J. Zhang, W.-S. Zheng, Progressive teacher-student learning for early action prediction, in: CVPR, 2019, pp. 3556–3565.
[22] K. Gavrilyuk, A. Ghodrati, Z. Li, C. Snoek, Actor and action video segmentation from a sentence, in: CVPR, 2018, pp. 5958–5966.
[23] P. Ghosh, Y. Yao, L. Davis, A. Divakaran, Stacked spatio-temporal graph convolutional networks for action segmentation, in: WCACV, 2020, pp. 576–585.
[24] A. Richard, H. Kuehne, J. Gall, Action sets: Weakly supervised action segmentation without ordering constraints, in: CVPR, 2018, pp. 5987–5996.
[25] W. Du, Y. Wang, Y. Qiao, Recurrent spatial-temporal attention network for action recognition in videos, TIP 27 (3) (2017) 1347–1360.
[26] V. Bloom, V. Argyriou, D. Makris, Linear latent low dimensional space for online early action recognition and prediction, Pattern Recognit. 72 (2017) 532–547.
[27] H. Zhao, R.P. Wildes, Spatiotemporal feature residual propagation for action prediction, in: CVPR, 2019, pp. 7003–7012.
[28] Y. Kong, S. Gao, B. Sun, Y. Fu, Action prediction from videos via memorizing hard-to-predict samples, AAAI, volume 32, 2018.
[29] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, Y. Sheikh, Openpose: realtime multi-person 2d pose estimation using part affinity fields, TPAMI 43 (1) (2019) 172–186.
[30] H. Touvron, A. Vedaldi, M. Douze, H. Jégou, Fixing the train-test resolution discrepancy, in: NIPS, 2019, pp. 8252–8262.
[31] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, arXiv (2015).
[32] M. Korban, X. Li, Ddgcn: A dynamic directed graph convolutional network for action recognition, in: ECCV, Springer, 2020, pp. 761–776.
[33] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: CVPR, Ieee, 2009, pp. 248–255.
[34] X. He, H. Zhang, M. Kan, T. Chua, Fast matrix factorization for online recommendation with implicit feedback, in: SIGIR CRDIR, 2016, pp. 549–558.
[35] K. Soomro, A. Zamir, M. Shah, Ucf101: a dataset of 101 human actions classes from videos in the wild, arXiv (2012).
[36] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. Chua, Neural collaborative filtering, in: ICWWW, 2017, pp. 173–182.
[37] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, J. Liu, Online human action detection using joint classification-regression recurrent neural networks, ECCV (2016).
[38] L. Chunhui, H. Yueyu, L. Yanghao, S. Sijie, L. Jiaying, Pku-mmd: a large scale benchmark for continuous multi-modal human action understanding, arXiv (2017).
[39] J. Liu, A. Shahroudy, D. Xu, G. Wang, Spatio-temporal lstm with trust gates for 3d human action recognition, in: ECCV, Springer, 2016, pp. 816–833.
[40] J. Liu, G. Wang, P. Hu, L. Duan, A. Kot, Global context-aware attention lstm networks for 3d action recognition, in: CVPR, 2017, pp. 1647–1656.
[41] S. Baek, K. Kim, T. Kim, Real-time online action detection forests using spatio-temporal contexts, in: WACV, IEEE, 2017, pp. 158–167.
[42] J. Kundu, M. Gor, P. Uppala, V. Radhakrishnan, Unsupervised feature learning of human actions as trajectories in pose embedding manifold, in: WACV, IEEE, 2019, pp. 1459–1467.
[43] C. Liu, Y. Li, Y. Hu, J. Liu, Online action detection and forecast via multitask deep recurrent neural networks, in: ICASSP, IEEE, 2017, pp. 1702–1706.
[44] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, W. Zeng, Multi-modality multi-task recurrent neural network for online action detection, TCSVT 29 (9) (2018) 2667–2682.
[45] J. Yin, J. Han, R. Xie, C. Wang, X. Duan, Y. Rong, X. Zeng, J. Tao, Mc-lstm: real–time 3d human action detection system for intelligent healthcare applications, IEEE Trans. Biomed. Circuits Syst. 15 (2) (2021) 259–269.
[46] Q. Hong, Y. Sun, T. Liu, L. Fu, Y. Xie, Tad-net: an approach for real-time action detection based on temporal convolution network and graph convolution network in digital twin shop-floor, Digit. Twin 1 (10) (2021) 10.
[47] Y. Zhu, D. Doermann, Y. Zhang, Q. Liu, A. Girgensohn, What and how? jointly forecasting human action and pose, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 771–778.
[48] H. Wang, L. Wang, Beyond joints: learning representations from primitive geometries for skeleton-based action recognition and detection, TIP 27 (9) (2018) 4382–4394.
[49] R. Cui, A. Zhu, J. Wu, G. Hua, Skeleton-based attention-aware spatial–temporal model for action detection and recognition, IET CV 14 (5) (2020) 177–184.
[50] F.M. Thoker, C.G. Snoek, Feature-supervised action modality transfer, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 3751–3758.
[51] K. Wang, X. Li, J. Yang, J. Wu, R. Li, Temporal action detection based on two-stream you only look once network for elderly care service robot, Int. J. Adv. Rob. Syst. 18 (4) (2021). 17298814211038342
[52] F.M. Thoker, H. Doughty, C.G. Snoek, Skeleton-contrastive 3d action representation learning, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 1655–1663.
[53] O. Moliner, S. Huang, K. Åström, Bootstrapped representation learning for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4154–4164.

Matthew Korban received his BSc and MSc degree in Electrical Engineering in 2013 from the University of Guilan, where he worked on video processing related to sign language recognition. He received his Ph.D. in Computer Engineering from Louisiana State University. He is currently a Postdoc Research Associate at the University of Virginia, working with Dr. Scott T. Acton. His research interest includes Human Action Recognition, Early Action Recognition, Motion Synthesis, and Human Geometric Modeling in Virtual Reality environments.

Xin Li is a professor of Visual Computing and Computational Media at School of Performance, Visualization, & Fine Art, Texas A&M University, USA. He got his B.E. degree in Computer Science from University of Science and Technology of China in 2003, and his M.S. and Ph.D. degrees in Computer Science from State University of New York at Stony Brook in 2005 and 2008. His research interests are in Visual Computing, Geometric and Visual Data Processing, and Understanding, Computer Vision, and Computer Graphics. For more detail, please see https://people.tamu.edu/~xinli/.