Unified Keypoint-based Action Recognition Framework via Structured Keypoint Pooling

Ryo Hachiuma*, Fumiaki Sato*, Taiki Sekii
Konica Minolta, Inc.
{rhachiuma,fumiaki.sato.jp,taiki.sekii}@gmail.com

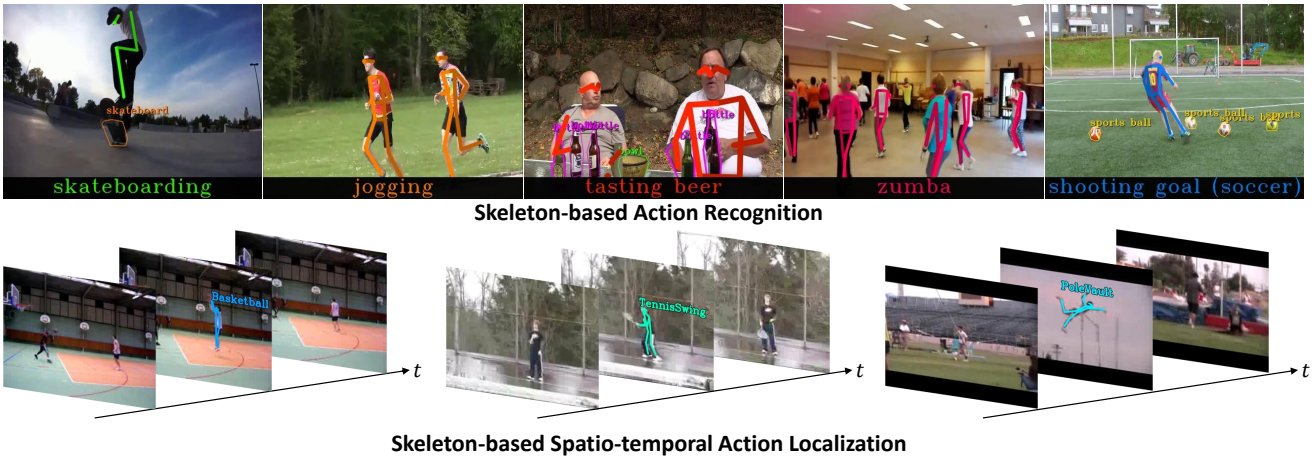

Figure 1. Qualitative results of the proposed framework for the skeleton-based action recognition (top) and spatio-temporal localization task (bottom). The input keypoints and the estimated action labels are visualized in the figure. We achieve state-of-the-art accuracy for the recognition task while it runs ∼1800FPS on a single RTX 3080Ti GPU. In addition, the proposed method outperforms the state-of-the-art weakly supervised spatio-temporal localization methods. See the website for the demo video.

Abstract

This paper simultaneously addresses three limitations associated with conventional skeleton-based action recognition; skeleton detection and tracking errors, poor variety of the targeted actions, as well as person-wise and frame-wise action recognition. A point cloud deep-learning paradigm is introduced to the action recognition, and a unified framework along with a novel deep neural network architecture called Structured Keypoint Pooling *is proposed. The proposed method sparsely aggregates keypoint features in a cascaded manner based on prior knowledge of the data structure (which is inherent in skeletons), such as the instances and frames to which each keypoint belongs, and achieves robustness against input errors. Its less constrained and tracking-free architecture enables time-series keypoints consisting of human skeletons and nonhuman object contours to be efficiently treated as an input 3D point cloud and extends the variety of the targeted action. Furthermore, we propose a* Pooling-Switching Trick *inspired by Structured Keypoint Pooling. This trick switches the pooling kernels between the training and inference phases to detect person-wise and frame-wise actions in a weakly supervised manner using only video-level action labels. This trick enables our training scheme to naturally introduce novel data augmentation, which mixes multiple point clouds extracted from different videos. In the experiments, we comprehensively verify the effectiveness of the proposed method against the limitations, and the method outperforms state-of-the-art skeleton-based action recognition and spatio-temporal action localization methods.*

1. Introduction

Recognizing the actions of a person in a video plays an essential role in various applications such as robotics [28, 41] and surveillance cameras [11, 25, 49]. The approach to the action recognition task differs depending on whether leveraging appearance information in a video or human skeletons[1] detected in the video. The former appearance-based approaches [2,7,11,18,20–23,25,32,45,51,52,56,58] directly use video as an input to deep neural networks

* Equal contribution.

[1] Joints or keypoints specific to a person are referred to as skeletons for clarity, although some are not actual human joints.

(DNNs) and thus even can recognize actions with relatively small movements. However, they are less robust to appearances of the people or scenes that differ from the training data [34, 55]. On the other hand, the latter skeleton-based approaches [5,9,10,13,17,29,33,34,49,57,60] are relatively robust to such appearance changes of a scene or a person because they only input low-information keypoints detected using the multi-person pose estimation methods [6, 42, 50].

Starting from ST-GCN [57], various skeleton-based approaches employing graph convolutional networks (GCNs) have emerged [5, 9, 10, 13, 33, 44]. These approaches model the relationship among keypoints by densely connecting them in a spatio-temporal space using GCNs, which treat every keypoint as a node at each time step. However, most approaches exhibit low scalability in practical scenarios, and further performance improvement is required since they exhibit three limitations regarding network architectures or their problem settings, as described below.

Skeleton Detection and Tracking Errors. Conventional GCN-based methods heavily rely on dense graphs, whose node keypoints are accurately detected and grouped by the same instance. These methods assume that the DNN features are correctly propagated. Therefore, if false positives (FPs) or false negatives (FNs) occur during keypoint detection, or if the multi-person pose tracking [39, 47] fails, such assumptions no longer hold, and the action recognition accuracy is degraded [17, 62].

Poor Variety of the Targeted Actions. Conventional approaches limit the number of input skeletons to at most one or two. Therefore, the recognition of actions performed by many people or those interacting with nonhuman objects is an ill-posed problem. On the other hand, for a wide range of applications, it is desirable to eliminate such restrictions and target a variety of action categories.

Person-wise and Frame-wise Action Recognition. Conventional approaches classify an entire video into actions, while practical scenes are complex and include multiple persons performing different actions in different time windows. Hence, recognizing each person's action for each frame (*spatio-temporal action localization*) is necessary.

In this paper, a unified action recognition framework and a novel DNN architecture called *Structured Keypoint Pooling*, which enhances the applicability and scalability of the skeleton-based action recognition (see Fig. 1), is proposed to simultaneously address the above three limitations. Unlike previous methods, which concatenate the keypoint coordinates and input them into a DNN designed on a predefined graph structure of a skeleton, the proposed method introduces a point cloud deep-learning paradigm [37, 38, 61] to the action recognition and treats a set of keypoints as an input 3D point cloud. PointNet [37], which was proposed in such a paradigm, is an innovative research, whose output is permutation-invariant to the order of the input points. It extracts the features for each input point and sparsely aggregates them to the output feature vector using *Max-Pooling*. Unlike PointNet, the proposed network architecture aggregates the features extracted from the point cloud in a cascaded manner based on prior knowledge of the data structure, which is inherent in the point cloud, such as the frames or the detection results of the persons (instances) to which each keypoint belongs. As a result, it is less constrained than conventional approaches and tracking-free. Also, its feature propagation among keypoints is relatively sparse. Therefore, the range of the DNNs affected by the keypoint errors (*e.g.*, FPs, FNs, and tracking errors) associated with the first robustness limitation can also be limited.

In addition, the permutation-invariant property of the input in the proposed network architecture eliminates the constraints of the data structure and size (*e.g.*, number of instances and pose tracking) found in the GCN-based methods. This property is exploited, and the *nonhuman object* keypoints[2] defined on the contour of the objects are used as an input in addition to human skeletons. Thus, the second target-action limitation mentioned above is addressed by increasing the input information without relying on the appearances while avoiding overfitting on them [14, 34, 55].

Finally, the third multi-action limitation is addressed by extending the proposed network architecture concept to a weakly supervised spatio-temporal action localization, which only requires a video-level action label during training. This is achieved using the proposed *Pooling-Switching Trick* inspired by Structured Keypoint Pooling, which switches the pooling structures according to the training and inference phases. Furthermore, this pooling-switching technique naturally enables the proposed training scheme to introduce novel data augmentation, which mixes multiple point clouds extracted from different videos.

In summary, our main contributions are three-fold: (1) We propose Structured Keypoint Pooling based on point cloud deep-learning in the context of action recognition. This method incorporates prior knowledge of the data structure to which each keypoint belongs into a DNN architecture as an inductive bias using a simple Max-Pooling operation. (2) In addition to the human skeletons, object keypoints are introduced as an additional input for skeleton-based action recognition. (3) A skeleton-based, weakly supervised spatio-temporal action localization is achieved by introducing a Pooling-Switching Trick, which exploits the feature aggregation scheme of Structured Keypoint Pooling.

2. Related Work

2.1. Action Recognition

Appearance-based Action Recognition. Numerous prior works rely on RGB images, which are used as inputs

[2]Nonhuman objects are referred to as objects for simplicity.

to DNNs [7, 20–23, 45, 51, 52, 56, 58]. In early deep-learning-based approaches, RGB and optical flow images are used as inputs to a 2D convolutional neural network (CNN) to explicitly model the appearance and motion features [45, 58]. The methods that extract spatio-temporal features using a 3D CNN obtain the motion feature extractors in a data-driven manner [7, 22, 51]. On the other hand, some studies have focused on reducing the computational cost and the number of parameters of a 3D CNN [20, 21, 52, 56]. Recently, methods that extract long-range features using the Transformer [53] have been proposed [2, 23, 32]. These appearance-based approaches have an advantage over skeleton-based methods because they use more detailed movement features.

Skeleton-based Action Recognition. Skeleton-based approaches have been actively investigated since ST-GCN [57], which models the relationships among time-series keypoints using GCNs. Upon the ST-GCN, the robustness and performance of these approaches have been improved by extracting the features from distant keypoints in the spatio-temporal space [9, 10, 29, 33] or by employing efficient graph convolution layers [5, 60]. In these methods, the input skeleton sequences can capture only motion information that is immune to contextual nuisances such as background variation and lighting changes [14, 34, 55]. Despite their significant success, GCN-based methods exhibit the three limitations mentioned in Sec. 1.

SPIL [49], which uses an attention mechanism among keypoints, also handles skeleton sequences as an input 3D point cloud and competes with the proposed method only with respect to the network architecture concept. Unlike SPIL, the proposed method does not rely on such a redundant attention module. Instead, it introduces a simple and sparse feature aggregation structure, which exploits prior knowledge of the data structure to which each keypoint belongs as an inductive bias.

2.2. Spatio-temporal Action Localization

When multiple persons appearing in a video perform different actions in different time windows, according to the third multi-action limitation mentioned in Sec. 1, this can be handled as a spatio-temporal action localization task. In the fully-supervised setting, appearance-based approaches [27, 30, 36] have been proposed but require dense instance-level annotations during the training. To reduce the annotation cost, weakly supervised methods [3, 12, 19] use only a single label for the video as supervision. These methods employ the multiple instance learning framework [16] for the weakly supervised setting to which this study also focuses on. Unlike such appearance-based approaches, our input keypoint information is less sensitive to the appearance changes. In addition, weakly supervised learning is achieved using a simple Pooling-Switching Trick, which exploits our point cloud-based setting and only changes the pooling kernels between the training and inference phases.

3. Proposed Framework

3.1. Overview

The proposed network architecture and its components are shown in Fig. 2. One of our core ideas is the feature aggregation by a Max-Pooling operator based on groups belonging to the same instance or the same frame (referred to as *local groups*). Limiting the feature-propagation range to the local group is essentially similar to the convolution operation, which extracts the pixel features locally; this is introduced as an inductive bias in our model. The proposed model essentially consists of only a few conventional DNN modules; nevertheless, its original design and inputs contribute to a significant performance improvement. In the following, we describe the network architecture along its process and each component in detail.

First, multi-person pose estimation and object keypoint detection are applied to the input video, and human joints as well as object contour points (collectively denoted as keypoints) are obtained. Then, the keypoints extracted from all frames in the video are treated as a point cloud and used as inputs to the network. Each keypoint is represented by a four-dimensional vector, which consists of the two-dimensional image coordinates, the confidence score, and the category index of the instance in which the keypoint belongs (*e.g.*, 0 denotes *person*, 1 denotes *car*, etc.). Each element of the input vector is normalized between 0 and 1.

Structured Keypoint Pooling f_θ predicts logit z, where $z \in \mathbb{R}^C$ for the action recognition task and for the training phase in a weakly supervised action localization task. As discussed later, $z \in \mathbb{R}^{FI \times C}$ for the inference phase in the action localization task. F denotes the number of frames in the video clip, I denotes the number of instances per frame, and C denotes the number of target classes. θ in f_θ represents the trainable parameters, and f_θ mainly consists of MLP Blocks and Grouped Pool Blocks (GPB). During the training phase, the cross-entropy loss $L_\theta\left(\mathrm{softmax}(z), l\right)$ is computed using the softmax layer and the ground-truth action label l; θ is updated via backpropagation.

The Point embedding layer embeds the input vector into a high-dimensional feature vector using multilayer perceptrons (MLPs). The weights of the MLP are shared across all keypoints. We adopt keypoint index encoding, which replaces the position in the original sinusoid positional encoding [53] with a keypoint index. The keypoint index represents its type, for example, 0 for the *left shoulder* and 1 for the *right shoulder* regarding the skeleton keypoints; also, it is 0 for *up left* and 1 for *up right* regarding these objects.

The MLP Block computes the feature vectors considering the sparse relationships among them via Max-Pooling,

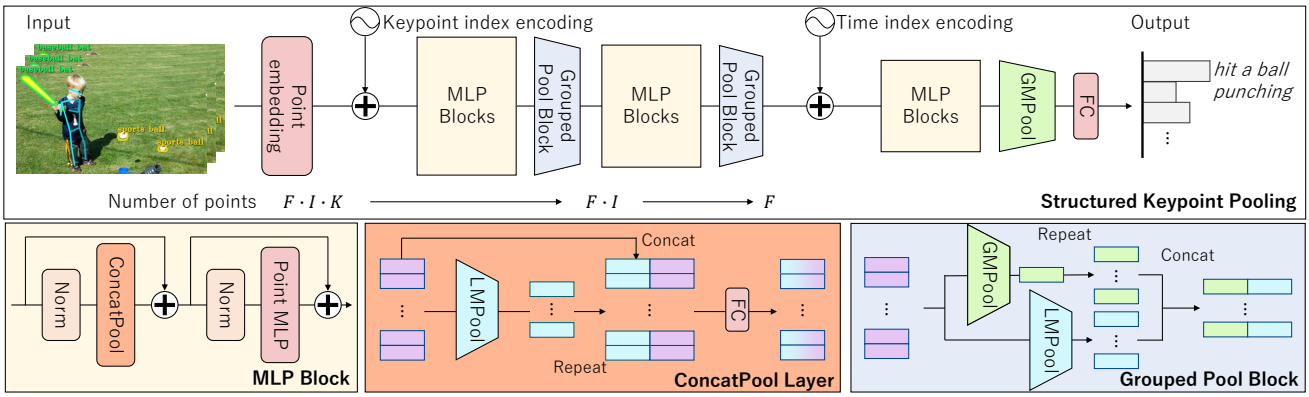

Figure 2. Overview of the Structured Keypoint Pooling network architecture (top) and its original components (bottom).

and the GPB aggregates such feature vectors into local groups. Similar to keypoint index encoding, we adopt time index encoding, which encodes the frame index in the video clip. The feature vectors are finally aggregated by global max-pooling (GMPool) to generate a single feature vector for the entire video. The logit is predicted via the fully-connected (FC) layers.

The reduction in the number of feature vectors by the GPB is described in the following. We denote K as the number of keypoints per instance, in addition to F and I defined above. The number of keypoints input to the network is $F \cdot I \cdot K$, which is reduced to $F \cdot I$ points by the first Grouped Pool Bock that aggregates K keypoint-wise features into a single vector. Then, the second GPB that aggregates I instance-wise features in a single vector reduces the number of points from $F \cdot I$ to F.

The Max-Pooling operator outputs a feature vector by selecting a maximum value for each dimension from N input vectors. Therefore, elements from maximum D points are selected (D is the feature dimension size of input vectors). As $N \gg D$ (*e.g.*, $N = F \cdot I \cdot K = 300 \cdot 2 \cdot 18$, $D = 512$) for the skeleton-based action recognition task, most points will be disregarded for the GMPool (pooling across all input points). Reducing the number of points (N) by cascaded feature aggregation and limiting the pooling range using local max-pooling (LMPool), which applies Max-Pooling to each local group to which the input feature vectors belong, are helpful to generate informative and robust feature vectors. The effect of using this cascaded reduction during the feature extraction will be quantitatively verified in Sec. 4.6.

The process in each block is invariant to the position and order of the input feature vectors, and the entire network can handle permutation-invariant inputs.

3.2. Grouped Pool Block (GPB)

The GPB consists of GMPool ϕ_G and LMPool ϕ_L. The first GPB outputs feature vectors containing the number of instances in the video, and the subsequent GPB outputs feature vectors containing the number of frames.

The GPB can be expressed as follows:

$$Y = \left\{ \left[\phi_L \left(X \right)_j ; \phi_G \left(X \right) \right] \right\}_{j \in \{1,\ldots,M\}}. \tag{1}$$

X and Y are the matrices of the input and output feature vectors, respectively, as described below. M denotes the number of local groups in X; $M = F \cdot I$ in the first block; $M = F$ in the second block. Therefore, the input feature vector $x_i \in X$ is grouped into M local groups, and X can be expressed by a concatenated matrix as follows:

$$X = (x_1, \ldots, x_N)^T = (X_1; \ldots; X_M)^T. \tag{2}$$

Consequently, Y is computed using the output vector y_j as follows:

$$Y = (y_1, \ldots, y_M)^T. \tag{3}$$

In Eq. (1), we concatenate each feature vector $\phi_L \left(X \right)_j$ computed for the local group j and the global feature vector $\phi_G \left(X \right)$ in a channel dimension. Also, LMPool $\phi_L(\cdot)$ can be expressed as follows:

$$\phi_L \left(X \right) = \{ \mathrm{MaxPool}(X_j) \}_{j=\{1,\ldots,M\}}, \tag{4}$$

where $\mathrm{MaxPool}(\cdot)$ is the operation used to obtain the max value for each channel from the feature vectors. GMPool $\phi_G(\cdot)$ is expressed as follows:

$$\phi_G(X) = \mathrm{MaxPool}(X). \tag{5}$$

3.3. MLP Block

The MLP Block consists of two residual blocks. The first block models the relationship among feature vectors within each local group. The subsequent block applies MLPs for each feature vector. Each MLP block is repeated r times.

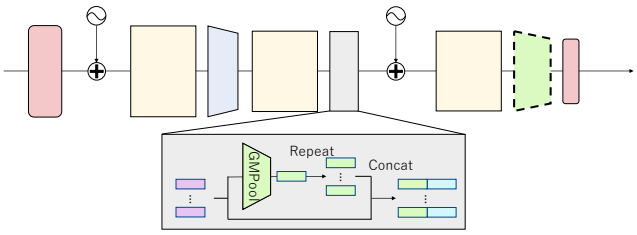

Figure 3. Pooling-Switching Trick for point cloud-based spatio-temporal action localization. The modules same as those in Fig. 2 are abbreviated with the same color. The dotted GMPool layer is only applied during the training phase.

The first residual block can be written using the input and output matrices X and Y, respectively, as follows:

$$Y = \mathrm{ConcatPool}(\mathrm{Norm}(X)) + X, \tag{6}$$

where $\mathrm{Norm}(\cdot)$ is the normalization layer, and $\mathrm{ConcatPool}(\cdot)$ is the learnable layer represented as

$$\mathrm{ConcatPool}(X) = \left\{\sigma\left(\left[x_i; \phi_L\left(X\right)_{j_i}\right] W_1\right)\right\}_{i\in\{1,\ldots,N\}}, \tag{7}$$

where $\sigma(\cdot)$ is a nonlinear activation function and $j_i \in \{1,\ldots,M\}$ is the local group index of the i-th feature vector. $W_1 \in \mathbb{R}^{2D\times D}$ is a learnable weight matrix, and D is the number of channels of X.

The second residual block can be expressed as follows:

$$Y = \sigma\left(\mathrm{Norm}\left(X\right) W_2\right) W_3 + X, \tag{8}$$

where $W_2 \in \mathbb{R}^{D\times\alpha D}$ and $W_3 \in \mathbb{R}^{\alpha D\times D}$ are learnable weight matrices, and α is the MLP expansion ratio.

3.4. Pooling-Switching Trick for Spatio-Temporal Action Localization

The proposed network architecture of the spatio-temporal action localization is shown in Fig. 3. To avoid aggregating instance-level features into frame-level features, the second GPB in Fig. 2 is changed. We propose a Pooling-Switching Trick, which switches the group of the pooling (kernel) from the training to the inference phases. This trick naturally enables our weakly supervised training scheme to introduce the proposed batch-mixing data augmentation.

Weakly Supervised Training. During the training, the loss is computed between the ground-truth action label assigned to the input video and the video-level logit predicted by aggregating instance-level features using the last GMPool. During the inference, the proposed method estimates the actions against targets different from the training, such as each instance, each frame, or each video, by switching the pooling kernel (target local group) at the last GMPool operation. For the spatio-temporal action localization task, the GMPool operation is simply removed from the network architecture (Fig. 3) to estimate the instance-level logit. The weights of the FC layer are shared across all targets.

Batch-Mixing Augmentation. To improve the localization robustness, we propose a novel data augmentation technique in the Pooling-Switching Trick. This technique mixes the point clouds extracted from different videos and promotes classifying multiple actions. Let $X \in \mathbb{R}^{FI\times D}$ and l denote instance-level features and the corresponding ground-truth one-hot label, respectively. Two training samples (X^a, l^a) and (X^b, l^b) are mixed for augmentation.

First, we mask two training samples as follows:

$$\hat{X}^a = B \odot X^a, \hat{X}^b = (1 - B) \odot X^b, \tag{9}$$

where $B \in \mathbb{R}^{FI\times D}$ denotes a binary mask indicating which keypoint is used in the two samples. Each column vector in B is 0 or 1, and $\odot$ denotes the element-wise multiplication. Also, the ground-truth label is mixed with a certain ratio λ as follows:

$$\hat{l} = \lambda l^a + (1 - \lambda) l^b. \tag{10}$$

A random sampling of the mixing ratio λ and the binary mask is followed to the CutMix strategy [59].

Instead of aggregating a set of feature vectors in the global feature using GMPool *within* each training sample (intra-sample), GMPool (two green boxes in Fig. 3) aggregates *between* two training samples (inter-samples) to the global feature vector during the training phase as follows:

$$\phi_G\left(\hat{X}^a, \hat{X}^b\right) = \mathrm{MaxPool}(\hat{X}^a; \hat{X}^b). \tag{11}$$

Finally, the mixed logit $\hat{z}$ is predicted, and the cross-entropy loss $L_\theta\left(\mathrm{softmax}\left(\hat{z}\right), \hat{l}\right)$ is computed.

4. Experiments

4.1. Datasets

Kinetics-400. The Kinetics-400 [7] dataset is a large-scale video dataset collected from YouTube videos with 400 action classes. It contains 250K training and 19K validation 10-second video clips.

UCF101 and HMDB51. The UCF101 [48] and HMDB51 [26] datasets contain 13K YouTube videos with 101 action labels and 6.7K videos with 51 action labels, respectively. We employ *split1* for training and test data splitting, according to the previous work [17].

RWF-2000, Hockey-Fight, Crowd Violence, and Movies-Fight. The RWF-2000 [11], Hockey-Fight [4], Crowd Violence [24], and Movies-Fight [35] datasets are violence recognition datasets. These datasets contain two types of actions, violence and non-violence, with various people and backgrounds.

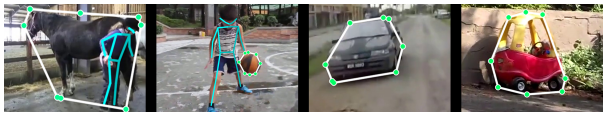

Figure 4. Examples of human skeletons (blue) and eight object contour keypoints (green).

Mimetics. The Mimetics dataset [55] contains 713 YouTube video clips of mimed actions that form a subset of 50 classes obtained from the Kinetics-400 dataset. This dataset evaluates human actions with out-of-context appearances different from the Kinetics-400 dataset, and thus the methods have been trained on only the Kinetics-400 dataset.

Mixamo. The Mixamo dataset [15] is an action recognition dataset that was proposed for the evaluation of domain adaptation tasks. This dataset is synthetically generated using the Mixamo library [1]. The 3D virtual avatars perform 14 different actions with various backgrounds and objects. The dataset contains 24K 2D-rendered videos.

UCF101-24. The UCF101-24 dataset [48] is a subset of the UCF101 dataset. Its 24 class action labels are annotated for each bounding box in the videos. Following the standard practice [3, 12], we use the corrected annotation [46].

4.2. Evaluation Metrics

We employ Top-1 Accuracy (%) (simply referred to as *accuracy*) as the evaluation metric for an action recognition task. For a spatio-temporal action localization task, we employ Video Average Precision (Video AP) (%) with different 3D IoUs (0.2 and 0.5) as the evaluation metrics. We use a machine equipped with Intel i7-10700K CPU, 32GB RAM, and GeForce RTX 3080Ti GPU to compute the speed. See the supplementary material for the implementation details, the hyperparameters of the training, and data augmentations pertaining to all experiments.

4.2.1 Keypoint Detectors

PPNv2. The pose proposal networks (PPNv2) [42, 43] simultaneously detect human skeletons and object keypoints located onto the object contours from an RGB image at a high speed. They are employed to generate keypoints in an experiment using object information as an input and consist of a Pelee backbone [54] trained on the MS-COCO dataset [31] with both human and object keypoint annotations. The definition of a human skeleton is the same as the OpenPose [6] definition. The object keypoints are defined as the eight extreme points on the contours with respect to the eight directions centered on the object (see Fig. 4). The input image is resized by 320×224 px^2.

HRNet. For a fair comparison with conventional skeleton-based approaches [17, 33, 34, 49], the HRNet [50] is also employed as the human keypoint detector. The HRNet is a Top-Down pose detector that achieves superior human pose estimation performance. However, its computational cost, which includes a human detector (Faster R-CNN [40]), is expensive. We use publicly available HRNet skeletons [17] for the Kinetics-400, UCF101, and HMDB51 datasets. With the same setting [17], HRNet skeletons are generated for the RWF-2000, Hockey-Fight, Crowd Violence, Movies-Fight, and Mimetics datasets.

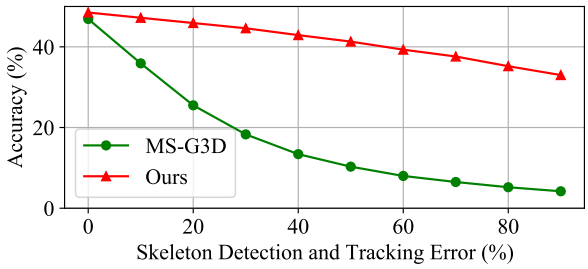

Figure 5. Comparison of the robustness against skeleton detection and tracking errors on the Kinetics-400 dataset. The methods are trained and evaluated using HRNet skeletons for a fair comparison.

4.3. Skeleton-based Action Recognition Performance Comparisons on the Kinetics-400

In Tab. 1, the action recognition accuracy and the speed between the proposed method and conventional skeleton-based approaches are compared on the Kinetics-400 dataset. It can be observed that the proposed method (Ours w/ objects), which inputs both the skeleton and object keypoints detected by PPNv2, outperforms the conventional methods. Moreover, its accuracy is improved by 9.2 percentage-point by introducing object keypoints in addition to the skeletons (Ours w/o objects vs. Ours w/ objects). The qualitative results are shown in Fig. 1 (top).

Compared with conventional methods that employ HRNet keypoints [50], the proposed method outperforms state-of-the-art (SoTA) methods [17, 33] (MS-G3D and PoseConv3D), while its runtime is 3x and 96x faster, respectively, than the runtime of these methods. Considering ablation studies, as discussed later, these results show that the proposed method overcomes both the first robustness and the second target-action limitations, mentioned in Sec. 1.

4.4. Robustness against Skeleton Detection and Tracking Errors

The robustness of the proposed method against skeleton detection errors (FPs, FNs, and tracking errors) is compared with that of the MS-G3D [33], which is the best-performing SoTA method considering both accuracy and runtime metrics, as shown in Tab. 1. Here, we synthetically generated three types of skeleton detection errors, FPs, FNs, and tracking errors. The FPs were generated by adding noise sam-

Table 1. Speed/Accuracy comparison of SoTA skeleton-based action recognition methods on the Kinetics-400 dataset. Column Runtime shows the computation time of only the action recognition model. Column Total FPS shows the speed, including keypoint detection and action recognition. The *joint-bone two-stream ensemble* framework is employed for a fair comparison with conventional methods [17, 33, 44]. Additionally, we combine HRNet human joints and PPNv2 object keypoints, and the result is 61.4% (+11.1 percentage-point by using objects).

Method	Acc. (%)	Keypoint Detector	COCO AP_{kp} (%)	Runtime (ms)	Total FPS
ST-GCN [57]	30.7	OpenPose [6]	56.3	4.0	85.4
2s-AGCN [44]	36.1			27.6	84.8
MS-G3D [33]	38.0			28.2	84.8
MS-G3D [33]	45.1	HRNet [50]	74.6	28.2	8.8
PoseConv3D [17]	47.7			960.0	8.5
Ours w/o objects	50.3			9.8	8.8
Ours w/o objects	43.1	PPNv2 [42]	36.4	9.8	1913
Ours w/ objects	**52.3**			11.2	1896

Table 2. Ablation study of the GPB on the Kinetics-400 dataset with HRNet skeletons.

Inst.	Frame	Acc. (%)	Runtime (ms)
-	-	47.3	89.5
✓	-	**48.6**	7.2
✓	✓	48.5	**4.9**

Table 3. Ablation study of the object keypoint input on the Kinetics-400 dataset with PPNv2 keypoints.

Category	Bbox	Contours	Acc. (%)
-	-	-	41.2
✓	✓	-	48.6
✓	-	✓	**49.2**

Table 4. Accuracy Comparison on UCF-101 (U), HMDB51 (HM), Mimetics (Mi), RWF-2000 (R), Hockey-Fight (Ho), Crowd Violence (C), and Movies-Fight (MF) datasets.

Method	Input	U	HM	Mi	R	Ho	C	MF
I3D [7]	RGB/ Flow	95.6	74.8	-	83.4	93.4	83.4	95.8
Flow Gated [11]		-	-	-	87.3	98.0	88.8	97.3
3D ResNext [55]		-	-	10.5	-	-	-	-
SlowOnly [21]		92.8	66.0	-	-	-	-	-
OmniSource [18]		**98.6**	**87.0**	-	-	-	-	-
SIP-Net [55]	Skeleton	-	-	14.2	-	-	-	-
IntegralAction [34]		-	-	15.3	-	-	-	-
PoseConv3D [17]		87.0	69.7	-	-	-	-	-
SPIL [49]		-	-	-	89.3	96.8	94.5	98.5
Ours		87.8	70.9	**21.2**	**93.4**	**99.5**	**94.7**	**99.0**

Table 5. Domain shift experiment on the Mixamo dataset for training and the Kinetics-400 dataset for evaluation. Unsupervised (US) and weakly supervised (WS) domain adaptation (DA) methods are employed as a comparison.

Method	DA	Input	Acc. (%)
I3D [7]	-	RGB	11.2
TA^3N [8]	US	RGB	10.0
CO^2A [15]			16.4
TA^3N [8]	WS	RGB	19.1
CO^2A [15]			20.1
Ours	-	Skeleton	27.6
		Skeleton+Object	**28.4**

Table 6. Ablation study of the overall framework on the Kinetics-400 dataset with HRNet skeletons.

Design of the GPB	Design of the MLP Blocks		
	Only MLPs	1st MLP Block → MS-G3D	Ours (MLP+ConcatPool)
w/o LMPool	30.3	45.5	47.3
Ours (w/ GPB)	44.5	45.7	**48.5**

Table 7. Comparison with SoTA weakly supervised spatio-temporal action localization methods on the UCF101-24 dataset.

Method	Input	AP@0.2	AP@0.5
Escorcia *et al.* [19]	RGB	45.5	-
Chéron *et al.* [12]		43.9	17.7
Anurag *et al.* [3]		61.7	35.0
Ours w/o Mix. Aug.	Skeleton	60.4	37.4
Ours w/ Mix. Aug.		**61.8**	**38.0**

pled from a normal distribution to the keypoint image coordinates. The FNs were generated by replacing the keypoint image coordinates and the confidence score with 0 using a certain ratio. The tracking errors were generated by switching the tracking indices with a certain interval. Note that the action recognition accuracy of GCN-based methods relies on tracking errors. On the other hand, the proposed method does not because the proposed network architecture is permutation-invariant for the input keypoints, and the tracking indices are not used.

Fig. 5 shows that the performance of the SoTA method (MS-G3D) is highly degraded by adding errors to the inputs. In contrast, since the performance degradation of the proposed method is relatively small, the proposed method is robust against skeleton detection and tracking errors, which are described as the first robustness limitation in Sec. 1.

4.5. Action Recognition Accuracy Comparison with Appearance-based Approaches

In Tab. 4, the performance of the proposed method is compared against that of both the SoTA skeleton-based and appearance-based approaches that use RGB and/or optical flow images as inputs. Here, for a fair comparison with SoTA skeleton-based approaches [17, 34, 49], only HRNet skeletons are used as input. HRNet exhibits a detection performance similar to that of conventional skeleton detection methods employed in these approaches.

The RWF-2000 dataset captures two actions (violence and non-violence) using surveillance cameras in a similar environment. In the Mimetics experiment, the DNNs are trained with the Kinetics-400 dataset, while the appearances of a person or a background are different from the Kinetics-400 in the evaluation videos. Hence, the skeleton-based approaches outperform the appearance-based approaches when the RWF-2000 and Mimetics datasets are

employed. The opposite occurs when the UCF101 and HMDB51 datasets are employed. Conclusively, the performance of each of the two approaches depends on the pair of datasets employed. This result, that the appearance-based approaches are highly biased to background or person appearances, is also mentioned in previous studies [14,34,55].

The proposed method outperforms the SoTA methods by a certain margin, except for UCF101 and HMDB51 datasets. In particular, the proposed method outperforms that of SPIL [49], which handles the sequential skeleton data as a point cloud on four violence recognition datasets.

4.6. Ablation Studies

Effect of the GPB. An ablation study of the GPB, which aggregates keypoint features using prior knowledge of the keypoint belongings, instances, or frames, is shown in Tab. 2. Three models are compared; the model where the GPB is not applied at the two stages (instance-level and frame-level), the model where the GPB is applied only at the first stage, and the proposed model. Instead of not using the GPB, we concatenate the feature vector from GMPool and each input vector. It can be observed that the first GPB, which aggregates the features into the instance level, significantly improves the accuracy and speed. The second GPB mainly improves the runtime.

Effect of Object Keypoints. Tab. 3 shows the results obtained using object categories and eight object contour keypoints as an additional input with the Kinetics-400 dataset. It can be observed that compared to the accuracy of the skeleton-only input (41.2%), the accuracy of the proposed model is improved when using object categories and four bounding box points (48.6%) as an additional input. Furthermore, the accuracy is further improved by introducing eight object contour points (49.2%) instead of bounding box points, and both the category and object contour points are informative of the action recognition task.

Ablation Study of the Overall Framework. An ablation study is performed to verify the GPB and MLP Block contributions in Tab. 6. Also, Tab. 6 includes the results of the model replacing our first MLP Block with the GCN-based MS-G3D module [33]. The simplest baseline (top-left cell) extracts point-wise features and aggregates them using MLPs and GMPool, respectively, similar to PointNet [37]. This baseline performs poorly; the GPB yields significant enhancement in accuracy (30.3% vs. 44.5%). Moreover, our method outperforms the version employing the MS-G3D module, which models the temporal information among keypoints (45.7% vs. 48.5%).

4.7. Domain Shift by Introducing Object

An accuracy comparison with and without using object keypoint information is summarized in Tab. 5. Here, the models are trained using a synthetically-created Mixamo dataset and evaluated using a real Kinetics dataset to reproduce a challenging, cross-dataset domain shift. In addition, the proposed method is compared with an appearance-based method [7] and the SoTA unsupervised and weakly supervised domain adaptation methods [8, 15].

It can be observed that the accuracy is improved by introducing the proposed object keypoints (27.6% vs. 28.4%), and the variety of actions is expanded without overfitting (the second target-action limitation mentioned in Sec. 1). Furthermore, since the proposed method outperforms the appearance-based approaches without any domain adaptation or supervision of the target dataset (Kinetics-400), it is suitable for practical cases when the scene appearance differs between the training and inference phases.

4.8. Spatio-Temporal Action Localization

The weakly supervised spatio-temporal action localization accuracy is summarized in Tab. 7. Since no previous study addressed the task of using only skeletons as an input, the proposed method is compared against appearance-based approaches [3, 12, 19] and the evaluation protocol [3] is followed, although the UCF101 is an advantageous dataset for the appearance-based approaches shown in Tab. 4.

The proposed method without batch-mixing augmentation outperforms SoTA weakly supervised action localization methods with the AP@0.5 metric. Furthermore, the proposed method outperforms these methods with both AP@0.2 and AP@0.5 metrics by introducing batch-mixing augmentation. Therefore, as mentioned in the third multi-action limitation in Sec. 1, the proposed method localizes the actions of each person in each frame. The qualitative results of the action localization are shown in Fig. 1 (bottom).

5. Conclusion

In this paper, a novel framework with a DNN architecture, Structured Keypoint Pooling, was proposed to address the limitations of conventional skeleton-based action recognition methods. Time-series keypoints consisting of human skeletons and nonhuman object contours were treated as an input 3D point cloud of the Structured Keypoint Pooling, which sparsely aggregates keypoint features into a cascaded manner based on prior knowledge of the data structure to which the keypoints belong. A Pooling-Switching Trick, which switches the aggregation target in the phases, and novel data augmentation, which mixes multiple point clouds, were also proposed. We comprehensively verified the effectiveness against the limitations using several action recognition and localization datasets. The experimental results demonstrated that the proposed method outperforms SoTA methods regarding both skeleton-based action recognition and spatio-temporal action localization tasks.

References

[1] Adobe Systems Inc. 2021. Mixamo. https://www.mixamo.com. (Accessed on 03/20/2023). 6

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A Video Vision Transformer. In *ICCV*, 2021. 1, 3

[3] Anurag Arnab, Chen Sun, Arsha Nagrani, and Cordelia Schmid. Uncertainty-Aware Weakly Supervised Action Detection from Untrimmed Videos. In *ECCV*, 2020. 3, 6, 7, 8

[4] Enrique Bermejo Nievas, Oscar Deniz Suarez, Gloria Bueno García, and Rahul Sukthankar. Violence Detection in Video Using Computer Vision Techniques. In *CAIP*, 2011. 5

[5] Jinmiao Cai, Nianjuan Jiang, Xiaoguang Han, Kui Jia, and Jiangbo Lu. JOLO-GCN: Mining Joint-Centered Light-Weight Information for Skeleton-Based Action Recognition. In *WACV*, 2021. 2, 3

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *CVPR*, 2017. 2, 6, 7

[7] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 2017. 1, 3, 5, 7, 8

[8] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal Attentive Alignment for Large-Scale Video Domain Adaptation. In *ICCV*, 2019. 7, 8

[9] Tailin Chen, Desen Zhou, Jian Wang, Shidong Wang, Yu Guan, Xuming He, and Errui Ding. Learning Multi-Granular Spatio-Temporal Graph Network for Skeleton-Based Action Recognition. In *ACMMM*, 2021. 2, 3

[10] Zhan Chen, Sicheng Li, Bing Yang, Qinghan Li, and Hong Liu. Multi-Scale Spatial Temporal Graph Convolutional Network for Skeleton-Based Action Recognition. In *AAAI*, 2021. 2, 3

[11] Ming Cheng, Kunjing Cai, and Ming Li. RWF-2000: An Open Large Scale Video Database for Violence Detection. In *ICPR*, 2021. 1, 5, 7

[12] Guilhem Chéron, Jean-Baptiste Alayrac, Ivan Laptev, and Cordelia Schmid. A Flexible Model for Training Action Localization with Varying Levels of Supervision. In *NeurIPS*, 2018. 3, 6, 7, 8

[13] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. InfoGCN: Representation Learning for Human Skeleton-Based Action Recognition. In *CVPR*, 2022. 2

[14] Jinwoo Choi, Chen Gao, C. E. Joseph Messou, and Jia-Bin Huang. Why Can't I Dance in the Mall? Learning to Mitigate Scene Bias in Action Recognition. In *NeurIPS*, 2019. 2, 3, 8

[15] Victor G. Turrisi da Costa, Giacomo Zara, Paolo Rota, Thiago Oliveira-Santos, Nicu Sebe, Vittorio Murino, and Elisa Ricci. Dual-Head Contrastive Domain Adaptation for Video Action Recognition. In *WACV*, 2022. 6, 7, 8

[16] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the Multiple Instance Problem with Axis-parallel Rectangles. *Artificial Intelligence*, 89(1):31–71, 1997. 3

[17] Haodong Duan, Yue Zhao, Kai Chen, Dian Shao, Dahua Lin, and Bo Dai. Revisiting Skeleton-based Action Recognition. In *CVPR*, 2022. 2, 5, 6, 7

[18] Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-Sourced Webly-Supervised Learning for Video Recognition. In *ECCV*, 2020. 1, 7

[19] Victor Escorcia, Cuong D. Dao, Mihir Jain, Bernard Ghanem, and Cees Snoek. Guess Where? Actor-supervision for Spatiotemporal Action Localization. *CVIU*, 192:102886, 2020. 3, 7, 8

[20] Christoph Feichtenhofer. X3D: Expanding Architectures for Efficient Video Recognition. In *CVPR*, 2020. 1, 3

[21] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. In *ICCV*, 2019. 1, 3, 7

[22] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal Residual Networks for Video Action Recognition. In *NeurIPS*, 2016. 1, 3

[23] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video Action Transformer Network. In *CVPR*, 2019. 1, 3

[24] Tal Hassner, Yossi Itcher, and Orit Kliper-Gross. Violent Flows: Real-Time Detection of Violent Crowd Behavior. In *CVPRW*, 2012. 5

[25] Zahidul Islam, Mohammad Rukonuzzaman, Raiyan Ahmed, Md. Hasanul Kabir, and Moshiur Farazi. Efficient Two-Stream Network for Violence Detection Using Separable Convolutional LSTM. In *IJCNN*, 2021. 1

[26] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A Large Video Database for Human Motion Recognition. In *ICCV*, 2011. 5

[27] Akash Kumar and Yogesh Singh Rawat. End-to-End Semi-Supervised Learning for Video Action Detection. In *CVPR*, 2022. 3

[28] Sang Uk Lee, Andreas Hofmann, and Brian Williams. A Model-Based Human Activity Recognition for Human–Robot Collaboration. In *IROS*, 2019. 1

[29] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition. In *CVPR*, 2019. 2, 3

[30] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as Moving Points. In *ECCV*, 2020. 3

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 6

[32] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video Swin Transformer. In *CVPR*, 2022. 1, 3

[33] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition. In *CVPR*, 2020. 2, 3, 6, 7, 8

[34] Gyeongsik Moon, Heeseung Kwon, Kyoung Mu Lee, and Minsu Cho. IntegralAction: Pose-Driven Feature Integration for Robust Human Action Recognition in Videos. In *CVPRW*, 2021. 2, 3, 6, 7, 8

[35] Enrique Bermejo Nievas, Oscar Deniz Suarez, Gloria Bueno Garcia, and Rahul Sukthankar. Movies Fight Detection Dataset. In *CAIP*, 2011. 5

[36] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-Context-Actor Relation Network for Spatio-Temporal Action Localization. In *CVPR*, 2021. 3

[37] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, 2017. 2, 8

[38] Charles R. Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, 2017. 2

[39] Umer Rafi, Andreas Doering, Bastian Leibe, and Juergen Gall. Self-supervised Keypoint Correspondences for Multi-person Pose Estimation and Tracking in Videos. In *ECCV*, 2020. 2

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015. 6

[41] I. Rodomagoulakis, N. Kardaris, V. Pitsikalis, E. Mavroudi, A. Katsamanis, A. Tsiami, and P. Maragos. Multimodal Human Action Recognition in Assistive Human-robot Interaction. In *ICASSP*, 2016. 1

[42] Taiki Sekii. Pose Proposal Networks. In *ECCV*, 2018. 2, 6, 7

[43] Taiki Sekii. Object Detection Method and Object Detection Device. In *Patent WO2021/117363*, 2021. 6

[44] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In *CVPR*, 2019. 2, 7

[45] Karen Simonyan and Andrew Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NeurIPS*, 2014. 1, 3

[46] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip H. S. Torr, and Fabio Cuzzolin. Online Real-Time Multiple Spatiotemporal Action Localisation and Prediction. In *ICCV*, 2017. 6

[47] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 Keypoints Is All You Need. In *CVPR*, 2020. 2

[48] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR*, abs/1212.0402, 2012. 5, 6

[49] Yukun Su, Guosheng Lin, Jinhui Zhu, and Qingyao Wu. Human Interaction Learning on 3D Skeleton Point Clouds for Video Violence Recognition. In *ECCV*, 2020. 1, 2, 3, 6, 7, 8

[50] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep High-Resolution Representation Learning for Human Pose Estimation. In *CVPR*, 2019. 2, 6, 7

[51] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features With 3D Convolutional Networks. In *ICCV*, 2015. 1, 3

[52] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *CVPR*, 2018. 1, 3

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*, 2017. 3

[54] Robert J. Wang, Xiang Li, and Charles X. Ling. Pelee: A Real-Time Object Detection System on Mobile Devices. In *NeurIPS*, 2018. 6

[55] Philippe Weinzaepfel and Grégory Rogez. Mimetics: Towards Understanding Human Actions out of Context. *IJCV*, 129(5):1675–1690, 2021. 2, 3, 6, 7, 8

[56] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification. In *ECCV*, 2018. 1, 3

[57] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *AAAI*, 2018. 2, 3, 7

[58] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond Short Snippets: Deep Networks for Video Classification. In *CVPR*, 2015. 1, 3

[59] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *ICCV*, 2019. 5

[60] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-Guided Neural Networks for Efficient Skeleton-Based Human Action Recognition. In *CVPR*, 2020. 2, 3

[61] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point Transformer. In *ICCV*, 2021. 2

[62] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust Graph Convolutional Networks Against Adversarial Attacks. In *SIGKDD*, 2019. 2