2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Action Segmentation Using 2D Skeleton Heatmaps and Multi-Modality Fusion

Syed Waleed Hyder Muhammad Usama Anas Zafar Muhammad Naufil Fawad Javed Fateh
Andrey Konin M. Zeeshan Zia Quoc-Huy Tran

***Abstract*— This paper presents a 2D skeleton-based action segmentation method with applications in fine-grained human activity recognition. In contrast with state-of-the-art methods which directly take sequences of 3D skeleton coordinates as inputs and apply Graph Convolutional Networks (GCNs) for spatiotemporal feature learning, our main idea is to use sequences of 2D skeleton heatmaps as inputs and employ Temporal Convolutional Networks (TCNs) to extract spatiotemporal features. Despite lacking 3D information, our approach yields comparable/superior performances and better robustness against missing keypoints than previous methods on action segmentation datasets. Moreover, we improve the performances further by using both 2D skeleton heatmaps and RGB videos as inputs. To our best knowledge, this is the first work to utilize 2D skeleton heatmap inputs and the first work to explore 2D skeleton+RGB fusion for action segmentation.**

I. Introduction

With the arrival of advanced deep networks and large-scale datasets, action recognition [1], [2], which aims to classify a trimmed video into a single action label, has achieved considerable progress and maturity. However, action segmentation [3], [4], [5], which seeks to locate and classify action segments of an untrimmed video into action labels, remains a challenging problem. Action segmentation underpins a number of robotics and computer vision applications. Notable examples include human-robot interaction [6], [7], [8] (i.e., recognizing human actions to facilitate interactions between humans and robots), ergonomics studies [9], [10] (i.e., extracting action segments in videos for ergonomics analyses), and visual analytics [11], [12] (i.e., conducting time and motion studies on video recordings).

A majority of action segmentation methods, e.g., [3], [13], [14], [15], [16], take RGB videos as inputs, which are then passed through TCNs to capture long-term action dependencies and predict segmentation results (see Fig. 1(a)). Since the introduction of TCNs to action segmentation in the pioneering work of Lea et al. [3], several improvements have been proposed. For example, multi-stage TCNs [15], [16] operate on the full temporal resolution of the videos, as compared to downsampling the temporal resolution of the videos as in Lea et al. [3].

In contrast with the above RGB-based methods, skeleton-based alternatives [9], [10], [4] have attracted research interests only recently because of their action focus and compact representation. These skeleton-based methods [9], [10], [4] take 3D skeleton sequences as inputs and employ GCNs to process 3D skeleton sequences directly due to their irregular graph structures (see Fig. 1(b)). As a result, it is difficult to incorporate other modalities that have regular grid structures (e.g., RGB, depth, flow) into existing skeleton-based methods [9], [10], [4].

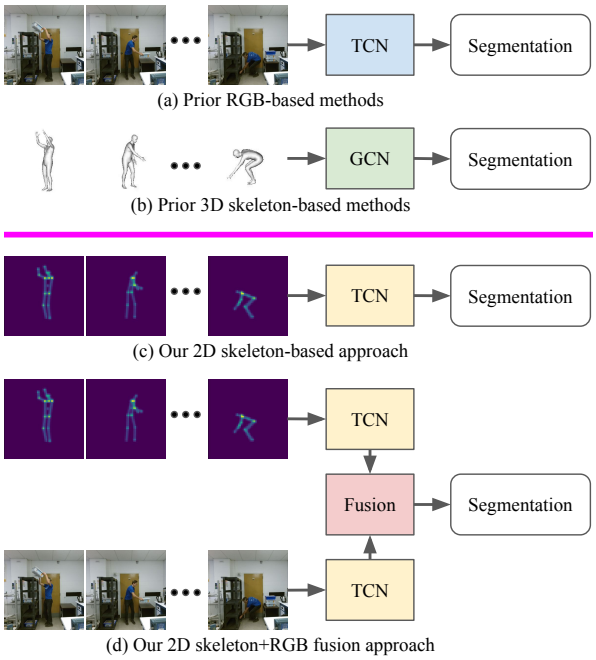


Fig. 1: Prior methods either take sequences of RGB frames (a) or sequences of 3D skeletons (b) as inputs. We propose a new approach which relies on sequences of 2D skeleton heatmaps (c). We further explore 2D skeleton+RGB fusion (d) for action segmentation, leading to performance gains.

In this work, we present a novel skeleton-based action segmentation approach, which relies only on 2D skeleton sequences. Inspired by the success of Duan et al. [17] in skeleton-based action recognition, we transform sequences of 2D skeletons into sequences of heatmaps. Since our heatmaps have image-like structures, i.e., $W \times H \times C$ with $W = H = 56$, $C = 3$, we can extract features from the heatmaps by using pre-trained ResNet [18]/VGG [19] before feeding the features to TCNs for action segmentation (see Fig. 1(c)), in a similar manner as employing multi-stage TCNs [15], [16] on RGB videos. Experiments on action segmentation datasets demonstrate that our 2D skeleton-based approach achieves similar/better performances and

All authors are with Retrocausal, Inc., Redmond, WA 98052, USA.
Website: www.retrocausal.ai
Email: {waleed,shaik,anas,naufil,fawad,andrey,zeeshan,huy}@retrocausal.ai

higher robustness against missing keypoints than previous 3D skeleton-based methods. In addition, we boost the performances further via multi-modality fusion, i.e., by combining RGB inputs with 2D skeleton inputs (see Fig. 1(d)). More specifically, we introduce fusion modules at multiple stages of [15], [16] to facilitate deep supervision [20], [21], [22].

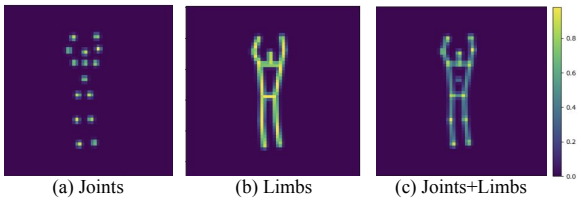
In summary, our contributions include:

- We propose a 2D skeleton-based action segmentation method, which takes 2D skeleton heatmaps as inputs and utilizes TCNs to capture spatiotemporal features. Our approach achieves comparable/better results and higher robustness against missing keypoints than 3D skeleton-based methods which operate directly on 3D skeleton coordinates and employ GCNs for spatiotemporal feature extraction.
- We further fuse 2D skeleton heatmaps with RGB videos, yielding improved performances. To the best of our knowledge, our work is the first to utilize 2D skeleton heatmap inputs and the first to explore 2D skeleton+RGB fusion for action segmentation.
- We annotate framewise action labels and obtain estimated 2D/3D skeletons for TUM-Kitchen, which are available at https://bitly.ws/3eWhV.

II. Related Work

RGB-Based Action Segmentation. RGB-based methods, e.g., [3], [13], [14], [15], [16], typically employ TCNs to learn long-term action dependencies. TCNs are first applied in Lea at al. [3], which perform temporal convolutions and deconvolutions. TricorNet [13] further utilizes bi-directional LSTMs, while TDRN [14] uses deformable temporal convolutions instead. One drawback of the above methods is that they temporally downsample videos. To address that, multi-stage TCNs [15], [16] are designed to preserve temporal resolutions. Recently, refinement techniques, e.g., [23], [24], are developed to reduce over-segmentations. Instead of RGB inputs, in this paper we use 2D skeleton inputs and introduce 2D skeleton+RGB fusion for action segmentation.

Skeleton-Based Action Segmentation. Skeleton-based approaches [9], [10], [4] have emerged only recently. In particular, Parsa et al. [9] propose a GCN architecture for skeleton-based action segmentation, while a GCN backbone is employed in their succeeding work [10] for joint skeleton-based action segmentation and ergonomics analysis. Recently, Filtjens et al. [4] extend multi-stage TCNs [15], [16] for RGB inputs to multi-stage GCNs for skeleton inputs. All of the above methods use 3D skeleton inputs and GCNs, which makes it hard to combine with other modalities (e.g., RGB, depth, flow). In contrast, our approach employs 2D skeleton inputs and TCNs, yielding comparable/superior results and better robustness against missing keypoints. Further, our 2D skeleton+RGB fused version leads to performance gains. Note that 2D skeletons have been used in previous methods [25], [26]. However, Kobayashi et al. [25] compute hand heatmaps for reweighting image features only, while Ma et al. [26] operate directly on 2D skeleton coordinates.



(a) Joints (b) Limbs (c) Joints+Limbs

Fig. 2: Examples of 2D skeleton heatmaps.

Skeleton-Based Action Recognition. Action recognition with skeleton inputs can roughly be grouped into GCN-based methods, e.g., [27], [28], [29], [30], and CNN-based ones, e.g., [31], [32], [33], [34]. The former process skeleton sequences directly with GCNs, whereas the latter aggregate them into regular grid inputs that are suitable for CNNs. While GCN-based methods are difficult to fuse with other modalities (e.g., RGB, depth, flow), CNN-based ones suffer from information loss during aggregation. To address the above, Duan et al. [17] model skeleton sequences as heatmap sequences and pass the heatmap sequences to CNNs, yielding superior results. Motivated by Duan et al. [17], we exploit heatmap representations and heatmap+RGB fusion for the fine-grained action segmentation task.

III. Our Approach

Below we present our main contributions, including a 2D skeleton-based approach and a multi-modality approach with both RGB and 2D skeleton inputs for action segmentation.

A. 2D Skeleton Heatmap

Given video frames $\boldsymbol{F}$, a well-known human detector, e.g., Faster-RCNN [35], and a recent top-down 2D human pose estimator, e.g., HRNet [36], can first be applied for extracting 2D skeletons $\boldsymbol{S}$ with high quality. Here, we model a 2D skeleton $\boldsymbol{s}$ by a set of 2D joint triplets $\{(x_k, y_k, c_k)\}$, where the k-th joint has 2D coordinates (x_k, y_k) and (maximum) confidence score c_k. In the following, 2D skeletons $\boldsymbol{S}$ are transformed into heatmaps $\boldsymbol{H}$. For a 2D skeleton $\boldsymbol{s}$, we derive a heatmap $\boldsymbol{h}$ of size $W \times H \times K$ (with the width W and height H of the video frame and the number of 2D joints K). Specifically, given a set of 2D joint triplets $\{(x_k, y_k, c_k)\}$, we derive a *joint* heatmap $\boldsymbol{h}^J$, which includes K Gaussian distributions centered at every 2D joint, as:

$$\boldsymbol{h}^J_{ijk} = e^{-\frac{(i-x_k)^2+(j-y_k)^2}{2*\sigma^2}} * c_k, \tag{1}$$

with standard deviation $\sigma = 0.6$. Alternatively, we derive a *limb* heatmap $\boldsymbol{h}^L$ of size $W \times H \times L$ (with the number of limbs L) as:

$$\boldsymbol{h}^L_{ijl} = e^{-\frac{dist((i,j),seg(a_l,b_l))}{2*\sigma^2}} * min(c_{a_l}, c_{b_l}). \tag{2}$$

Here, the l-th limb denotes the segment $seg(a_l, b_l)$ between the joints a_l and b_l, and the $dist$ function denotes the distance from the location (i, j) to the segment $seg(a_l, b_l)$. As we will show in Sec. IV-A, combining joint and limb heatmaps yields the best results. Thus, for each 2D skeleton $\boldsymbol{s}_i$, we derive the *combined* heatmap $\boldsymbol{h}_i = \boldsymbol{h}_i^{J+L}$. In contrast with

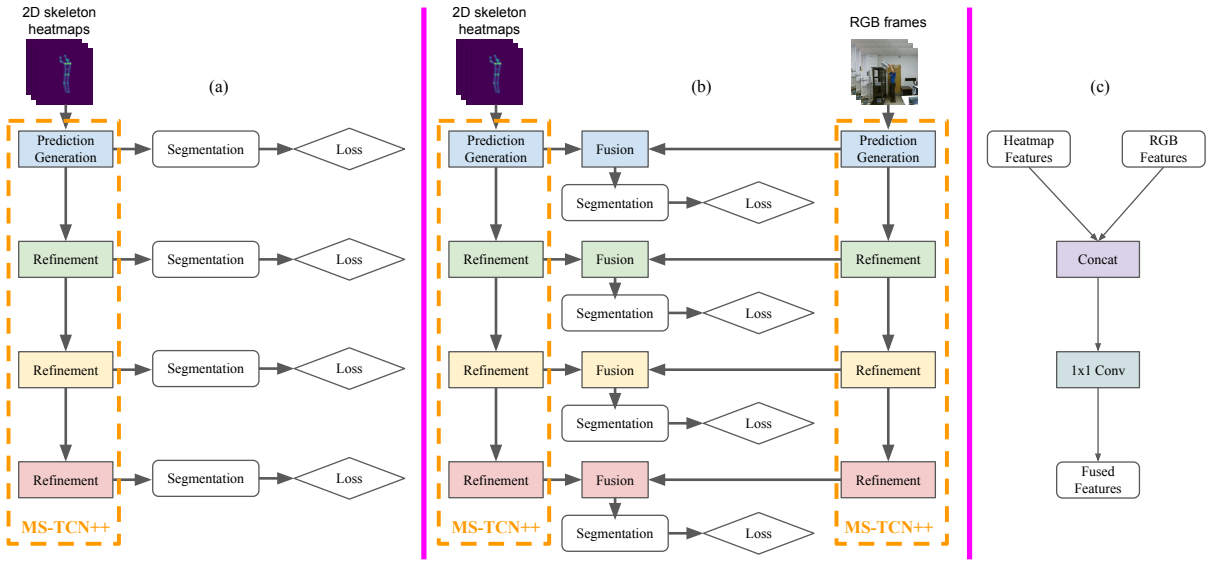


Fig. 3: (a) 2D skeleton-based action segmentation. We convert 2D skeletons into image-like heatmaps, which are passed to an RGB-based network for action segmentation, i.e., MS-TCNN++ [16]. (b) 2D skeleton+RGB-based action segmentation. During training, we propose fusion modules at various stages of MS-TCN++ [16] for deep supervision [20], [22]. At testing, the segmentation predicted by the last refinement stage is considered as our output. (c) 2D skeleton+RGB fusion module.

all previous skeleton-based methods [9], [10], [4], which use 3D skeletons, we rely only on 2D skeletons. Also, accurate 2D skeletons are easier to obtain than accurate 3D skeletons, as accurate depths are not required. Lastly, as we will show in Sec. IV-C, 2D skeleton heatmaps are more robust than 3D skeleton coordinates as joints/limbs are represented by Gaussian distributions.

In addition, we crop $\boldsymbol{h}_i$ along the spatial dimensions by using the smallest bounding box containing all 2D skeletons in the entire video sequence. Next, we sum all component (joint/limb) heatmaps within $\boldsymbol{h}_i$ into a single heatmap and resize it to 56×56. Finally, we replicate the heatmap and stack the copies, yielding an image-like heatmap $\boldsymbol{h}_i$ of size $56 \times 56 \times 3$, which can be used for extracting pre-trained features. Fig. 2 illustrates examples of 2D skeleton heatmaps. Unlike Duan et al. [17], which are mainly interested in video-level cues for coarse-grained action recognition (predicting a *single* label for a video) and perform temporal sampling to reduce computation costs, our work addresses fine-grained action segmentation (predicting *framewise* labels) where frame-level cues are important and hence temporal sampling is not employed in our method.

B. 2D Skeleton-Based Action Segmentation

As described above, our 2D skeleton heatmaps resemble RGB images. Thus, we can extract features from the 2D skeleton heatmaps by using ResNet [18]/VGG [19] pre-trained on ImageNet [37]. Next, an RGB-based action segmentation network can be employed to perform action segmentation on the extracted features. Here, we choose MS-TCN++ [16] due to its accuracy and efficiency. Fig. 3(a) shows an overview of our 2D skeleton-based action segmentation method. The network includes one prediction generation stage, which computes the initial segmentation, and three refinement stages, which improve the initial segmentation. The prediction generation stage consists of eleven dilated temporal convolutional layers, while the refinement stages share the same ten dilated temporal convolutional layers.

To train MS-TCN++ [16], deep supervision [20], [22] is employed. Particularly, all four stages are trained with the same combination of classification loss and smoothness loss. The classification loss is computed as the cross-entropy loss:

$$\mathcal{L}_{class} = \frac{1}{M} \sum_{i} -\log y_{i,c}, \tag{3}$$

with the number of video frames M and the probability $y_{i,c}$ of assigning frame $\boldsymbol{f}_i$ to ground truth action class c. To reduce over-segmentation, the smoothness loss is added:

$$\mathcal{L}_{smooth} = \frac{1}{MC} \sum_{i,c} \tilde{\Delta}_{i,c}^2, \tag{4}$$

$$\tilde{\Delta}_{i,c} = \begin{cases} \Delta_{i,c}, & \Delta_{i,c} \leq \tau \\ \tau, & \Delta_{i,c} > \tau \end{cases}, \tag{5}$$

$$\Delta_{i,c} = \left| \log y_{i,c} - \log y_{i-1,c} \right|, \tag{6}$$

with the number of action classes C and thresholding parameter $\tau = 16$. The final loss is written as:

$$\mathcal{L} = \mathcal{L}_{class} + \alpha \mathcal{L}_{smooth}, \tag{7}$$

with balancing parameter $\alpha = 0.15$. At testing, the prediction by the last refinement stage is used as our output.

C. 2D Skeleton+RGB-Based Action Segmentation

In the following, we explore multi-modality fusion to improve the performance. Fig. 3(b) shows an overview of our 2D skeleton+RGB-based action segmentation method.

The network consists of two branches for processing 2D skeleton heatmaps and RGB frames respectively. Since our 2D skeleton heatmaps have the same structures as the RGB frames, we use the same MS-TCN++ [16] architecture for both branches. To perform 2D skeleton+RGB fusion, we introduce fusion modules at all stages from prediction generation stage to refinement stages. In particular, our fusion module first concatenates the heatmap feature vector with the RGB feature vector before passing the concatenated feature vector through a 1×1 convolutional layer to reduce the fused feature vector to the original size of 64. Fig. 3(c) illustrates the above steps. We follow MS-TCN [16] to use the same losses as described in the previous section. At testing, our output is the prediction by the last refinement stage.

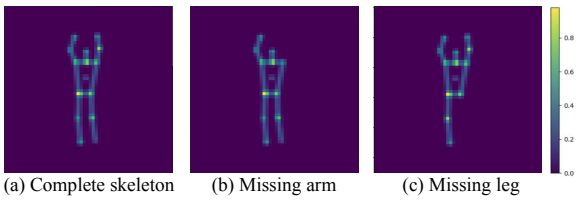
In addition, we have experimented with adding augmentation to 2D skeleton heatmaps (i.e., temporal augmentation, position augmentation, orientation augmentation, and horizontal flipping) and RGB frames (i.e., temporal augmentation, brightness augmentation, contrast augmentation, and horizontal flipping). However, the performance gain is marginal while the computational cost is increased significantly. Therefore, we do not apply data augmentation in this work. In contrast with Parsa et al. [9], which operate directly on 3D skeleton coordinates, our 2D skeleton heatmap representation makes it easier to fuse with RGB frames. Furthermore, while Parsa et al. [9] conduct fusion at the final level only, we perform fusion across multiple levels.

IV. Experiments

Datasets, Annotations, and Poses. We use three datasets, which capture a diverse set of human activities:

- *UW-IOM* [38] includes 20 videos of a warehouse activity, which consists of 17 actions (e.g., *"box_bend_pick_up_low"*). Each video is $\sim$3 minutes long. We use both framewise action labels and 2D/3D human poses (estimated with [39]) released by [10].
- *TUM-Kitchen* [40] consists of 19 videos of a kitchen activity, which comprises of 21 actions (e.g., *"walk,hold-both-hand"*). The duration of each video is $\sim$2 minutes. We manually annotate framewise action labels and obtain 2D/3D human poses with [39], as they are not provided by [10].
- *Desktop Assembly* [41], [42] includes 76 videos of an assembly activity. The activity consists of 23 actions (e.g., *"tighten_screw_4"*) and the video length is $\sim$1.5 minutes. We use framewise action labels of [41], while estimating 2D human poses with [36] and 3D human poses with [43].

Implementation Details. Our heatmap-based model is implemented in pyTorch. We randomly initialize our model and use ADAM optimization with a learning rate of 0.001. We train our model for 100 epochs. Furthermore, our heatmap+RGB-based model is trained in two stages: i) we first train the heatmap and RGB branches separately with the same losses in Eq. 7, ii) we then train our entire fusion model (with a learning rate of 0.0005) by using the weights from the first stage as initialization.



(a) Complete skeleton (b) Missing arm (c) Missing leg

Fig. 4: Examples of missing keypoints.

Competing Methods. We test against prior skeleton-based methods, which rely on 3D skeletons and GCNs. They include ST-PGN [9], MTL/STL [10], and MS-GCN [4]. Moreover, we evaluate the classical MS-TCN++ [16] with RGB inputs. Lastly, we compare with the 3D skeleton+RGB fusion version of ST-PGN [9]. If the results are reported in the original papers, we copy them (except for TUM-Kitchen, since we use different sets of labels/poses). Otherwise, we run the official implementations [1] to obtain the results.

Metrics. Following [9], [10], we use *F1* score (10% overlap), *Edit* distance, and mean Average Precision (*mAP*). In addition, we follow [4], [16] to add framewise Accuracy (*Acc*). Please refer to [9], [10], [4], [16] for detailed definitions.

A. Impacts of Different Heatmaps

We first study the performance of our 2D skeleton-based approach by using various inputs: joint heatmaps, limb heatmaps and integrated joint+limb heatmaps. We use the UW-IOM dataset and ResNet-50 features in this experiment. The ablation results are shown in Tab. I. It is evident that the best performance is achieved by combined joint+limb heatmaps, followed by limb heatmaps alone, whereas the worst performance is obtained by joint heatmaps alone.

B. Impacts of Different Features

We now investigate the performance of our 2D skeleton-based method by using various networks pre-trained on ImageNet for feature extraction: VGG-16, ResNet-18, and ResNet-50. The UW-IOM dataset and joint+limb heatmaps are used in this experiment. Tab. II presents the ablation results. It is clear that ResNet-50 features yield the best results, outperforming ResNet-18 features and VGG-16 features.

C. Robustness against Missing Keypoints

We conduct an experiment in which a certain component of the skeleton is removed to examine the robustness of our 2D skeleton-based approach. In particular, at testing (without any finetuning), we randomly drop a limb (i.e., arm or leg) with a probability p for each frame in the UW-IOM dataset. Fig. 4 shows examples of missing keypoints. The F1 scores of our method, MS-GCN [4], and STL [10] are illustrated in Tab. III. It can be seen that our approach demonstrates a high level of robustness against missing keypoints. For example, when one limb is dropped per frame (i.e., $p = 100\%$), our F1 score drop is 2.54%, as compared to 3.88% and 6.99%

[1] MTL/STL: https://github.com/BehnooshParsa/MTL-ERA
MS-GCN: https://github.com/BenjaminFiltjens/MS-GCN
MS-TCN++: https://github.com/sj-li/MS-TCN2

Heatmap	F1	Edit	mAP	Acc
Joint	89.87 ± 05.95	89.77 ± 04.86	85.07 ± 06.71	79.62 ± 06.93
Limb	*93.41 ± 02.85*	*92.20 ± 03.92*	*88.63 ± 03.66*	*83.15 ± 03.37*
Joint+Limb	**93.82 ± 02.90**	**92.88 ± 04.13**	**89.82 ± 03.83**	**85.04 ± 03.08**

TABLE I: Impacts of different heatmaps on UW-IOM. Best results are in **bold**. Second best results are *underlined*.

Feature	F1	Edit	mAP	Acc
VGG-16	91.58 ± 04.38	92.26 ± 04.91	87.20 ± 02.98	79.42 ± 04.22
ResNet-18	*93.76 ± 03.23*	*92.79 ± 04.24*	*88.01 ± 04.19*	*84.53 ± 04.07*
ResNet-50	**93.82 ± 02.90**	**92.88 ± 04.13**	**89.82 ± 03.83**	**85.04 ± 03.08**

TABLE II: Impacts of different features on UW-IOM. Best results are in **bold**. Second best results are *underlined*.

Method	Input	Keypoint Missing Probability p			
		0%	25%	50%	100%
STL [10]	△	87.24 ± 01.50	86.19 ± 02.27	84.29 ± 03.71	80.25 ± 05.68
MS-GCN [4]	△	*91.93 ± 04.60*	*90.14 ± 04.63*	*89.33 ± 04.83*	*88.05 ± 04.64*
Ours	⋆	**93.82 ± 02.90**	**93.48 ± 02.95**	**93.43 ± 02.75**	**91.28 ± 02.65**

TABLE III: Robustness against missing keypoints on UW-IOM. Note that △ denotes 3D pose coordinate inputs and ⋆ denotes 2D pose heatmap inputs. Best results are in **bold**. Second best results are *underlined*.

of MS-GCN [4] and STL [10] respectively. Moreover, our F1 score remains stable for $p = 25\%$ and $p = 50\%$, whereas the F1 scores of MS-GCN [4] and STL [10] decrease up to 2.60% and 2.95% respectively. This is likely because our method models each joint as a Gaussian and uses TCNs for learning spatiotemporal features, whereas MS-GCN [4] and STL [10] represent each joint by its coordinates and use GCNs for spatiotemporal feature extraction.

D. Comparisons on UW-IOM

In this section, we compare our 2D skeleton-based and 2D skeleton+RGB-based approaches with state-of-the-art action segmentation methods, including ST-PGN [9], MTL/STL [10], MS-GCN [4], and MS-TCN++ [16], on UW-IOM. The quantitative results are shown in Tab. IV. It is evident that our 2D skeleton+RGB fusion approach achieves the best overall results, outperforming the 3D skeleton+RGB fusion version of ST-PGN [9] by large margins on F1 score, Edit distance, and mAP, e.g., 92.88% Edit distance for Ours (Fusion) vs. 80.90% for ST-PGN (Fusion). Next, our 2D skeleton-based approach obtains the second best overall results, outperforming previous 3D skeleton-based and RGB-based methods on F1 score, mAP, and Acc. The results in Tab. IV confirm the effectiveness of using 2D skeleton heatmaps and TCNs for capturing spatiotemporal features.

E. Comparisons on TUM-Kitchen

Tab. V presents the quantitative results on TUM-Kitchen. It is clear that our 2D skeleton-based approach outperforms previous 3D skeleton-based and RGB-based methods on all metrics, e.g., on Acc, 71.55% for Ours, as compared to 69.38%, 59.77%, and 69.28% for MS-GCN [4], STL [10], and MS-TCN++ [16] respectively. Further, fusing 2D skeleton inputs and RGB inputs leads to the best overall performance with the best numbers on Edit distance, mAP, and Acc. The above observations validate the use of 2D skeleton heatmaps for action segmentation.

F. Comparisons on Desktop Assembly

We now evaluate our approaches on Desktop Assembly. Tab. VI shows the quantitative results. It can be seen that our multi-modality fusion approach achieves the best overall performance, followed by our 2D skeleton-based approach, whereas STL [9] performs the worst, e.g., on mAP, 91.55% for Ours (Fusion) and 91.19% for Ours, as compared to 86.91%, 60.05%, and 91.00% for MS-GCN [4], STL [10], and MS-TCN++ [16] respectively. The above results show the advantages of using 2D skeleton heatmaps and TCNs for learning spatiotemporal features. Lastly, we present some qualitative results in Fig. 5, where our results match the ground truth better than MS-GCN [4] and MS-TCN++ [16]. Please see also our supplementary video [2].

G. Discussions

Numbers of Parameters. We discuss the numbers of parameters of our approaches and previous methods on UW-IOM. Our heatmap-only model with around 1M parameters is larger than MS-GCN [4] with around 650K parameters, but significantly smaller than STL and MTL [10] with about 20M and 40M parameters respectively. MS-TCN++ [16] has the same number of parameters (i.e., about 1M parameters) as our heatmap-only model, while the number of parameters of our fusion model roughly doubles that of our heatmap-only model (yielding about 2M parameters).

Run Times. We measure the run times of our approaches and previous methods on UW-IOM. Our heatmap-only approach has a similar run time as MS-GCN [4], i.e., 71ms and 72ms respectively. They are considerably more efficient than STL and MTL [10], which have run times of 183ms and

[2]Supplementary video: https://youtu.be/skx7rkkhcUw

Method	Input	F1	Edit	mAP	Acc
MS-TCN++ [16]	□	93.36 ± 03.35	92.17 ± 04.08	87.99 ± 05.10	82.29 ± 03.89
†ST-PGN [9]	△	87.95 ± 01.54	**97.86 ± 02.15**	87.03 ± 02.85	-
†ST-PGN (Fusion) [9]	△, □	88.08 ± 01.89	80.90 ± 02.06	87.05 ± 03.47	-
†MTL [10]	△	92.03 ± 02.54	91.59 ± 01.23	74.45 ± 10.36	-
†STL [10]	△	92.33 ± 00.78	92.08 ± 01.18	49.61 ± 00.17	-
MS-GCN [4]	△	91.93 ± 04.60	87.61 ± 05.87	87.52 ± 05.29	82.75 ± 05.04
Ours	⋆	*93.82 ± 02.90*	92.88 ± 04.13	*89.82 ± 03.83*	*85.04 ± 03.08*
Ours (Fusion)	⋆, □	**94.54 ± 02.75**	*93.25 ± 03.81*	**90.12 ± 03.65**	**85.84 ± 03.27**

TABLE IV: Quantitative comparisons on UW-IOM. Note that □ denotes direct RGB inputs, △ denotes 3D pose coordinate inputs, and ⋆ denotes 2D pose heatmap inputs. Also, † indicates that results are copied from the original papers. Best results are in **bold**. Second best results are *underlined*.

Method	Input	F1	Edit	mAP	Acc
MS-TCN++ [16]	□	81.75 ± 04.04	84.76 ± 02.90	56.44 ± 02.63	69.28 ± 03.98
STL [10]	□	78.81 ± 08.42	81.50 ± 07.57	46.24 ± 17.42	59.77 ± 15.76
MS-GCN [4]	△	76.30 ± 04.23	80.14 ± 03.36	57.20 ± 02.59	69.38 ± 03.70
Ours	⋆	**81.96 ± 03.72**	*85.14 ± 03.01*	*58.81 ± 03.95*	*71.55 ± 04.75*
Ours (Fusion)	⋆, □	*81.38 ± 02.86*	**85.33 ± 02.16**	**61.40 ± 01.93**	**72.89 ± 03.77**

TABLE V: Quantitative comparisons on TUM-Kitchen. Note that □ denotes direct RGB inputs, △ denotes 3D pose coordinate inputs, and ⋆ denotes 2D pose heatmap inputs. Best results are in **bold**. Second best results are *underlined*.

Method	Input	F1	Edit	mAP	Acc
MS-TCN++ [16]	□	97.24 ± 02.08	**98.05 ± 02.11**	91.00 ± 04.62	87.42 ± 03.35
STL [10]	△	87.16 ± 07.23	85.71 ± 06.43	60.05 ± 00.23	76.41 ± 15.23
MS-GCN [4]	△	95.81 ± 03.43	95.03 ± 04.04	86.91 ± 04.97	87.01 ± 04.08
Ours	⋆	*97.90 ± 02.48*	97.15 ± 02.22	*91.19 ± 03.88*	*88.85 ± 03.92*
Ours (Fusion)	⋆, □	**98.02 ± 01.71**	*97.75 ± 02.39*	**91.55 ± 03.58**	**89.40 ± 02.62**

TABLE VI: Quantitative comparisons on Desktop Assembly. Note that □ denotes direct RGB inputs, △ denotes 3D pose coordinate inputs, and ⋆ denotes 2D pose heatmap inputs. Best results are in **bold**. Second best results are *underlined*.

Fig. 5: Qualitative comparisons on Desktop Assembly (sequence *2020-04-02-150120*).

554ms respectively. The run time of our fusion approach (i.e., 147ms) roughly doubles that of our heatmap-only approach.

Limitations. Despite our state-of-the-art performances on standard action segmentation datasets, our 2D skeleton heatmap-based approach may suffer from a few drawbacks. In contrast to 3D skeletons, 2D skeletons do not include depth cues. Thus, our approach may fail in cases where depth cues are important, e.g., occlusions and viewpoint changes. However, depth cues can still be inferred implicitly from 2D skeleton heatmaps, similar to monocular depth prediction. In addition, both 3D and 2D skeletons do not contain context details, e.g., objects and background information, which are available in RGB videos. Therefore, our approach may suffer in scenarios where context details are crucial. Nevertheless, our fusion approach overcomes that by utilizing both 2D skeleton heatmaps and RGB videos as inputs.

V. Conclusion

We introduce a 2D skeleton-based action segmentation method, which uses 2D skeleton heatmap inputs and employs TCNs for spatiotemporal feature learning. This is in contrast with previous methods, where 3D skeleton coordinates are handled directly and GCNs are used to capture spatiotemporal features. Our approach achieves similar/better results and higher robustness against missing keypoints than previous methods on action segmentation datasets, without requiring 3D information. To further improve the results, we fuse 2D skeleton heatmaps with RGB videos. To our best knowledge, this work is the first to utilize 2D skeleton heatmap inputs and the first to perform 2D skeleton+RGB fusion for action segmentation. Our future work will study the generalization of our approach by evaluating it on Epic Kitchens [44], which has diverse hand-object interactions and camera viewpoints.

REFERENCES

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

[2] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6202–6211.

[3] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.

[4] B. Filtjens, B. Vanrumste, and P. Slaets, "Skeleton-based action segmentation with multi-stage spatial-temporal graph convolutional neural networks," *IEEE Transactions on Emerging Topics in Computing*, 2022.

[5] G. Ding, F. Sener, and A. Yao, "Temporal action segmentation: An analysis of modern technique," *arXiv preprint arXiv:2210.10352*, 2022.

[6] F. Rea, A. Vignolo, A. Sciutti, and N. Noceti, "Human motion understanding for selecting action timing in collaborative human-robot interaction," *Frontiers in Robotics and AI*, vol. 6, p. 58, 2019.

[7] H. Khan, S. Haresh, A. Ahmed, S. Siddiqui, A. Konin, M. Z. Zia, and Q.-H. Tran, "Timestamp-supervised action segmentation with graph convolutional networks," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10 619–10 626.

[8] F. Yang, S. Odashima, S. Masui, and S. Jiang, "Is weakly-supervised action segmentation ready for human-robot interaction? no, let's improve it with action-union learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[9] B. Parsa, B. Dariush, *et al.*, "Spatio-temporal pyramid graph convolutions for human action recognition and postural assessment," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1080–1090.

[10] B. Parsa and A. G. Banerjee, "A multi-task learning approach for human activity segmentation and ergonomics risk assessment," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2352–2362.

[11] J. Ji, W. Pannakkong, P. D. Tai, C. Jeenanunta, and J. Buddhakulsomsiri, "Motion time study with convolutional neural network," in *Integrated Uncertainty in Knowledge Modelling and Decision Making: 8th International Symposium, IUKM 2020, Phuket, Thailand, November 11–13, 2020, Proceedings 8*. Springer, 2020, pp. 249–258.

[12] J. Ji, W. Pannakkong, and J. Buddhakulsomsiri, "A computer vision-based model for automatic motion time study," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 73, no. 2, pp. 3557–3574, 2022.

[13] L. Ding and C. Xu, "Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation," *arXiv preprint*, 2017.

[14] P. Lei and S. Todorovic, "Temporal deformable residual networks for action segmentation in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6742–6751.

[15] Y. A. Farha and J. Gall, "Ms-tcn: Multi-stage temporal convolutional network for action segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3575–3584.

[16] S.-J. Li, Y. AbuFarha, Y. Liu, M.-M. Cheng, and J. Gall, "Ms-tcn++: Multi-stage temporal convolutional network for action segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[17] H. Duan, Y. Zhao, K. Chen, D. Lin, and B. Dai, "Revisiting skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2969–2978.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial intelligence and statistics*. Pmlr, 2015, pp. 562–570.

[21] C. Li, M. Zeeshan Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker, "Deep supervision with shape concepts for occlusion-aware 3d object parsing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5465–5474.

[22] C. Li, M. Z. Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker, "Deep supervision with intermediate concepts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1828–1843, 2018.

[23] Y. Ishikawa, S. Kasai, Y. Aoki, and H. Kataoka, "Alleviating over-segmentation errors by detecting action boundaries," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 2322–2331.

[24] Y. Huang, Y. Sugano, and Y. Sato, "Improving action segmentation via graph-based temporal reasoning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 024–14 034.

[25] T. Kobayashi, Y. Aoki, S. Shimizu, K. Kusano, and S. Okumura, "Fine-grained action recognition in assembly work scenes by drawing attention to the hands," in *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2019, pp. 440–446.

[26] H. Ma, Z. Yang, and H. Liu, "Fine-grained unsupervised temporal action segmentation and distributed representation for skeleton-based human motion analysis," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 13 411–13 424, 2021.

[27] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[28] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition," in *proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1625–1633.

[29] J. Cai, N. Jiang, X. Han, K. Jia, and J. Lu, "Jolo-gcn: mining joint-centered light-weight information for skeleton-based action recognition," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 2735–2744.

[30] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 359–13 368.

[31] H. Liu, J. Tu, and M. Liu, "Two-stream 3d convolutional neural network for skeleton-based action recognition," *arXiv preprint*, 2017.

[32] C. Caetano, J. Sena, F. Brémond, J. A. Dos Santos, and W. R. Schwartz, "Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition," in *2019 16th IEEE international conference on advanced video and signal based surveillance (AVSS)*. IEEE, 2019, pp. 1–8.

[33] Z. Lin, W. Zhang, X. Deng, C. Ma, and H. Wang, "Image-based pose representation for action recognition and hand gesture recognition," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*. IEEE, 2020, pp. 532–539.

[34] S. Asghari-Esfeden, M. Sznaier, and O. Camps, "Dynamic motion representation for human action recognition," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 557–566.

[35] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[36] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5693–5703.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[38] B. Parsa, E. U. Samani, R. Hendrix, C. Devine, S. M. Singh, S. Devasia, and A. G. Banerjee, "Toward ergonomic risk prediction via segmentation of indoor object manipulation actions using spatiotemporal convolutional networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3153–3160, 2019.

[39] G. Rogez, P. Weinzaepfel, and C. Schmid, "Lcr-net: Localization-classification-regression for human pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3433–3441.

[40] M. Tenorth, J. Bandouch, and M. Beetz, "The tum kitchen data set of everyday manipulation activities for motion tracking and action recognition," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, 2009, pp. 1089–1096.

[41] S. Kumar, S. Haresh, A. Ahmed, A. Konin, M. Z. Zia, and Q.-H. Tran, "Unsupervised action segmentation by joint representation learning and online clustering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 174–20 185.

[42] Q.-H. Tran, A. Mehmood, M. Ahmed, M. Naufil, A. Zafar, A. Konin, and Z. Zia, "Permutation-aware activity segmentation via unsupervised frame-to-segment alignment," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 6426–6436.

[43] W. Li, H. Liu, H. Tang, P. Wang, and L. Van Gool, "Mhformer: Multi-hypothesis transformer for 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 147–13 156.

[44] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, *et al.*, "The epic-kitchens dataset: Collection, challenges and baselines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 4125–4141, 2020.