Selective Spatio-Temporal Aggregation Based Pose Refinement System: Towards Understanding Human Activities in Real-World Videos
(Supplementary Material)

Di Yang[1,2] Rui Dai[1,2] Yaohui Wang[1,2]
Rupayan Mallick[1] Luca Minciullo[3] Gianpiero Francesca[3] François Brémond[1,2]

[1]Inria [2]Université Côte d'Azur [3]Toyota Motor Europe
{di.yang, rui.dai, yaohui.wang, rupayan.mallick, francois.bremond}@inria.fr
{luca.minciullo, gianpiero.francesca}@toyota-europe.com

A. Details of AGCNs

Adaptive Graph Convolutional Network [11] (AGCNs) is our backbone network for action recognition. It is based on the relation between the joints along the spatial as well as temporal domain without a predefined graph as mentioned for GCNs. Shi et al. [11] propose an adaptive model in which two types of graph are optimized individually, one for representation of the global pattern of the data and the other for the unique pattern of each data. In addition, AGCNs [11] use second order information *i.e*. bones between two joints in the place of vectors representing the joints for late fusion, so that the performance for the action recognition task can be improved as the joints and bones information complement each other. In this section, we shortly summarize the formulations related to AGCNs and our implementation for experiments.

A.1. Spatio-Temporal Graph Convolution

Yan et al. [12] propose a spatio-temporal graph to model the sequence of skeletons *i.e*. human body joints (represented as 2D or 3D coordinates) where these skeleton sequences are represented in the form of vectors. For constructing spatio-temporal graph, both the spatial as well as temporal dimensions are considered. Intra-frame vertex connections are formed according to human body joints and for temporal dimension each joint is connected to the same joint in successive frames.

The formulation to compute the output feature map for the node or the vertex v_{ti} can be seen resembling that of 2D convolution operation (1). The sampling function for the same node v_{ti} is given in (2). The sampling function represents the neighbors at vertex v_{ti}. f_{in} and f_{out} denotes the input and output feature maps. Z_{ti} is normalizing term, $w(\cdot)$ and $l_{ti}(\cdot)$ is denoting the weight function and the label map at vertex v_{ti} function respectively.

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(v_{tj}).\mathbf{w}(l_{ti}(v_{tj})) \tag{1}$$

$$B(v_{ti}) = \{v_{ti} | d(v_{tj}, v_{ti}) \leq D\} \tag{2}$$

The extension to the temporal domain is by considering the same joints for successive frames. Yan et al. [12] modeled by introducing a parameter Γ i.e the kernel size in the temporal dimension. The kernel size is the range in the temporal dimension. The label map l_{ST} for spatial-temporal dimension is given in (3) where $l_{ti}(v_{tj})$ is the label map at vertex v_{ti} for 1 frame. The label map is used for the convolution operation as it maps a particular node with a unique weight vector.

$$l_{ST}(v_{qj}) = l_{ti}(v_{tj}) + (q - t + \lfloor \Gamma/2 \rfloor) \times K \tag{3}$$

The sampling area for the temporal dimension to include the joints connected temporally is given as:

$$B(v_{ti}) = \{v_{qj} | d(v_{tj}, v_{ti}) \leq K, |q - t| \leq \lfloor \Gamma/2 \rfloor\} \tag{4}$$

A.2. Adaptive Graph

The adaptive graph layer derived by Shi et. al. [11], where the difference from Spatio-Temporal Graph lies in the way the adjacency matrix is created. The strategy to make the graph adaptive is to divide the adjacency matrix into three matrices (A_k, B_k and C_k). The A_k matrix is represented for the human body structure whereas the B_k represents the matrix which facilitates the graph for totally learning from the training data as B_k is parameterized and optimized together with the learning process, and C_k represents a similarity matrix learning a specific graph for each training sample. The equation (1) for ST-GCN can be modeled as:

$$f_{out} = \sum_{k}^{K_v} W_k (f_{in} A_k) \odot M_k \tag{5}$$

The equation (6) shows the implementation of the adaptive graph convolutional network. In the equation(5), A_k represents connections between the joints in the physical graph whereas M_k represents the mask which is an attention mechanism, stating the strength of the connection. Connections which are absent in the original graph cannot be regenerated as mathematically the dot product between A_k and M_k are zero if any value in A_k is 0, irrespective of the value in M_k.

$$f_{out} = \sum_{k}^{K_v} W_k f_{in} (A_k + B_k + C_k) \tag{6}$$

A.3. Combination of AGCNs with SSTA-PRS

As shown in Fig. 1, we propose a framework extended by SSTA-PRS and AGCNs [11] for not only the action classification, but also temporal action detection. Both tasks use only pose data instead of any RGB information. For trimmed video, we directly feed the pose sequence to AGCNs for classification while for the untrimmed video, it is divided into several non-overlapping segments, each segment consisting of 16 continues poses. These segments are sent to the fine-tuned AGCNs model to extract the segment-level features. The feature map (*i.e.* video representation) is processed by a temporal action detection model like [7, 8, 4].

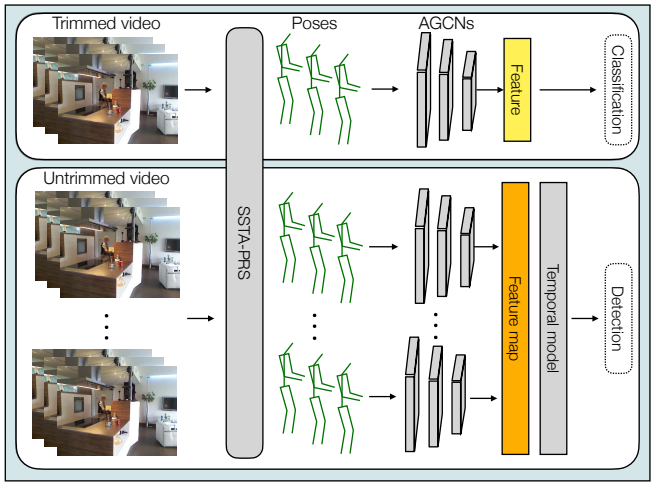

Figure 1. Overview of action recognition framework. The pose sequence of the trimmed video (top) from SSTA-PRS is fed into AGCNs and the feature is used directly for classification. For the long-term untrimmed video (bottom), the extracted segment-level features are fed into the temporal model [3, 4, 8] for detection.

A.4. 5-Channels AGCNs

Most GCN-based methods use high-quality 3D human poses obtained from a Motion Capture system and these methods have also been applied to 2D poses. 3D poses are usually more informative than 2D poses which are inherently ambiguous and view-invariant while the 3D pose centered by torso in the pre-processing only has relative movement information between

the joints of the body and lacks of global movement information of the body in the image, such as leaving, entering, walking, which are indistinguishable from the 3D pose. The same cases are sitting down, bending over, etc. In the main paper, we evaluate our model with refined 2D pose while we also conduct an additional experiment on Toyota Smarthome by expanding the channels of 2s-AGCN [11] to five as 5C-AGCN, which concatenate both 3D skeletons to be unambiguous and better to handle motion dynamics and 2D skeletons to obtain the global trajectory information in the RGB videos. It can achieve the best pose-based recognition performance. The 3D pose is from VideoPose3D [6] over the refined 2D pose.

A.5. Training Details of AGCNs

Action classification: For AGCN [11], we apply a SGD with Nesterov momentum for optimization strategy with momentum 0.9, an initial learning rate 0.1 for 50 and 65 epochs with step LR decay with a factor of 0.1 at epochs $\{30, 40\}$, and $\{45, 55\}$ for Smarthome and Kinetics-50 respectively. We follow the same pre-processing, data-augmentation and hyper parameters setting as in 2s-AGCN [11] for fair comparison. We report the best result after performing the weighted-score-level fusion on a two-stream (joints and bones).

Action detection: On top of the segment-level feature, Bidirectional-LSTM [3], we have two opposite direction 512 hidden units LSTM layers. The features from LSTMs are concatenated before the classifier. For TGM [8], on InHouse dataset, we utilize 4 layer structure. For Dilated-TCN [4], we adopt a 4 layer structure. A residual link connects the input of the first layer and the output of the last layer. We increase the number of filters to 512 per layer to model complex temporal relations. The unspecified parameters are similar to the original papers.

B. Additional Results

NTU-Pose: Effect of Occlusion and Resolution. We wonder in which scenario our method is the most effective. To do so, we repeated the experiments on NTU-Pose with three different levels of resolution: High-1920×1080 without occlusion (original video), High with occlusion (High+Occ), and Low-320×180 with occlusion (Low+Occ) as the experiment in the main paper. The results are shown in the Tab. 1. We conclude that our method aims to make the three expert estimators complementary to each other. the performances of all of the experts are good enough with high-resolution without occlusion and our method is more adaptive for complex real-world scenarios (low-resolution and occlusion), which is corresponding to the motivation of this work.

Methods	**PCKh** @2.0 (%) on **NTU-Pose**		
	High	High+Occ	Low+Occ
LCRNet++ [9]	80.9	73.1	54.1
AlphaPose [2]	81.2	53.4	53.2
OpenPose [1]	**81.3**	46.0	45.4
SST-A (ours)	81.2	**73.3**	**61.8**

Table 1. PCKh of poses from expert estimators and proposed SSTA-PRS using SST-A (Sec. 3.1) on NTU-Pose with different scenarios.

Toyota Smarthome: Why 2s-AGCN? (Comparison of state-of-the-art backbones) . We compare the performances among three state-of-the-art GCNs-based backbone networks [11, 10, 5] with our refined skeleton data using the same parameter settings. (official default settings) on Smarthome. The results are shown in Tab. 2. It suggests that the MS-AAGCN [10] and MS-G3D [5] do not boost the performance, that's why we select 2s-AGCN [11] as our backbone network.

Methods	**Smarthome** CS(%)
MS-G3D-J [5]	47.6
2s-AGCN-J [11]	55.7
2s-AGCN-B [11]	60.2
MS-AAGCN-J [10]	55.0
MS-AAGCN-B [10]	56.4
MS-AAGCN-JB [10]	58.9
2s-AGCN-JB [11]	**60.9**

Table 2. Mean per-class accuracy with state-of-the-art methods on the Toyota Smarthome dataset with refined pose data.

Toyota Smarthome: 5C-AGCNs. We expand the channels of 2s-AGCN [11] from 3 to 5 to concatenate the 2D and 3D pose data. For 2D, we normalize the coordinates in $[-1, 1]$ so that we can preserve the global trajectory. Note that we use SSTA-PRS pose data of Smarthome. The result in Tab. 3 shows that concatenating the 2D and 3D poses is significantly better ($+2.5\%$ for mean accuracy, $+1.8\%$ for total accuracy from 2D on CS). Compared with late fusion, we still get a better performance ($+0.3\%$ for mean accuracy) without the need to train two models. Therefore, the 5 channels concatenating is effective.

2s-AGCN-Joint [11]	**Smarthome CS**	
	Mean Acc (%)	TOT Acc (%)
+2D	55.7	77.1
+3D	54.0	73.4
+5D (late fusion)	57.9	78.5
+5D (5 channels)	**58.2**	**78.9**

Table 3. Mean per-class accuracy and total accuracy on Smarthome dataset using 2s-AGCN [11] with 2D or 3D skeleton data and 2D, 3D combination by channels concatenating and late fusion.

Charades: 31 semantic verbs. As most action classes in Charades are relevant to the objects. To evaluate the pose quality for action detection on this dataset, we ignore the object information and keep only the semantic verbs as the new action classes. The list of the verbs is: *Hold, Put, Sit, Take, Tide, Wash, Close, Close off, Comb hair, Fix object, Open, Turn on, Work, Play, Smile, Watch, Stand, Eat, Cook, Snuggle, Walk, Reach for, Drink, Pour, Lie, Awake, Grasp, Dress, Run, Sneeze, Undress, Talk.*

Charades: Results in Original Setting. We also evaluate our pose data with 157 action classes as the original setting. As shown in Tab. 4, all the methods can not achieve satisfying results, as object information lacks in pose modality. However, in this setting, our SSTA-PRS still outperforms the other pose estimation methods.

Methods	**Charades**		
	Bi-LSTM [3]	Dilated-TCN [4]	TGM [8]
LCRNet 3D [9]	7.4	7.8	8.1
LCRNet 2D+VideoPose3D [6]	7.9	8.4	8.7
SSTA-PRS (ours)	**8.2**	**8.8**	**9.0**

Table 4. Action detection performances on Charades. The performance are evaluated using frame-based mAP.

C. Qualitative Evaluation

SST-A and Confidence Metric. Fig. 2 shows the distribution of the aggregated poses from proposed SST-A. The error decreases globally with the increase of confidence. According to this figure, we choose the threshold $\gamma = 0.18$ to make our self-training samples cleaner (keep enough samples with small errors).

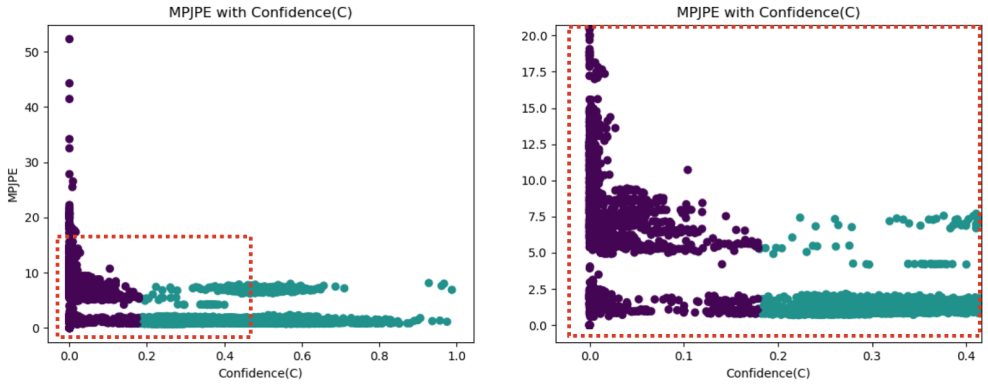

Figure 2. Distribution of aggregated poses with MPJPE and Confidence. (purple: high confidence with $\gamma >= 0.18$, green: low confidence with $\gamma < 0.18$) Zoom of the red bounding box is on the right.

Confusion Matrix on Toyota Smarthome. According to the confusion matrix (Fig. 3), we can see the impact of the pose refinement, which shows that owing to the improvement of the pose data, we can better distinguish actions that are closely related to skeleton like 'Cook.Stir', 'Walk', 'Sitdown', etc. In addition, in the living room (camera-5), the subject is often occluded by the sofa, and most of the actions that occur in this scene, such as 'Readbook' and 'WacthTV', are better classified. In the category of actions related to objects, we cannot improve too much. (*e.g.*'Drink.Fromcan', 'Drink.Frombottle', 'Drink.Fromglass')

Pose visualization of SSTA-PRS. In this section, we visualize some poses estimated from our proposed system for our experimental datasets. Note that the frames are randomly selected. Fig. 4 is the pose of the same frame with different systems. OpenPose [1] and AlphaPose [2] cannot provide correct pose for the occluded body-parts which LCRNet++ [9] can provide. For the upper-boddy, all of the three can work. The SST-A takes the upper-body parts from AlphaPose [2] while lower-body parts from LCRNet++ [9] to be the pseudo ground-truth pose. After self-training of the SSTA-PRS, We have a refined pose estimation model that combines the capabilities of three expert estimation systems. Fig. 5, Fig. 6 and Fig. 7 are the high-quality SSTA-PRS poses on Smarthome, Charades and Kinetics-50.

References

[1] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE TPAMI*, 2019.

[2] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017.

[3] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *IJCNN*, 2005.

[4] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *CVPR*, 2017.

[5] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *CVPR*, 2020.

[6] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, 2019.

[7] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *CVPR*, 2018.

[8] AJ Piergiovanni and Michael S. Ryoo. Temporal gaussian mixture layer for videos. In *ICML*, 2019.

[9] Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. *IEEE TPAMI*, 2019.

[10] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing LU. Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks. *arXiv:1912.06971 [cs]*, 2019.

[11] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[12] S. Yan, Yuanjun Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *AAAI*, 2018.

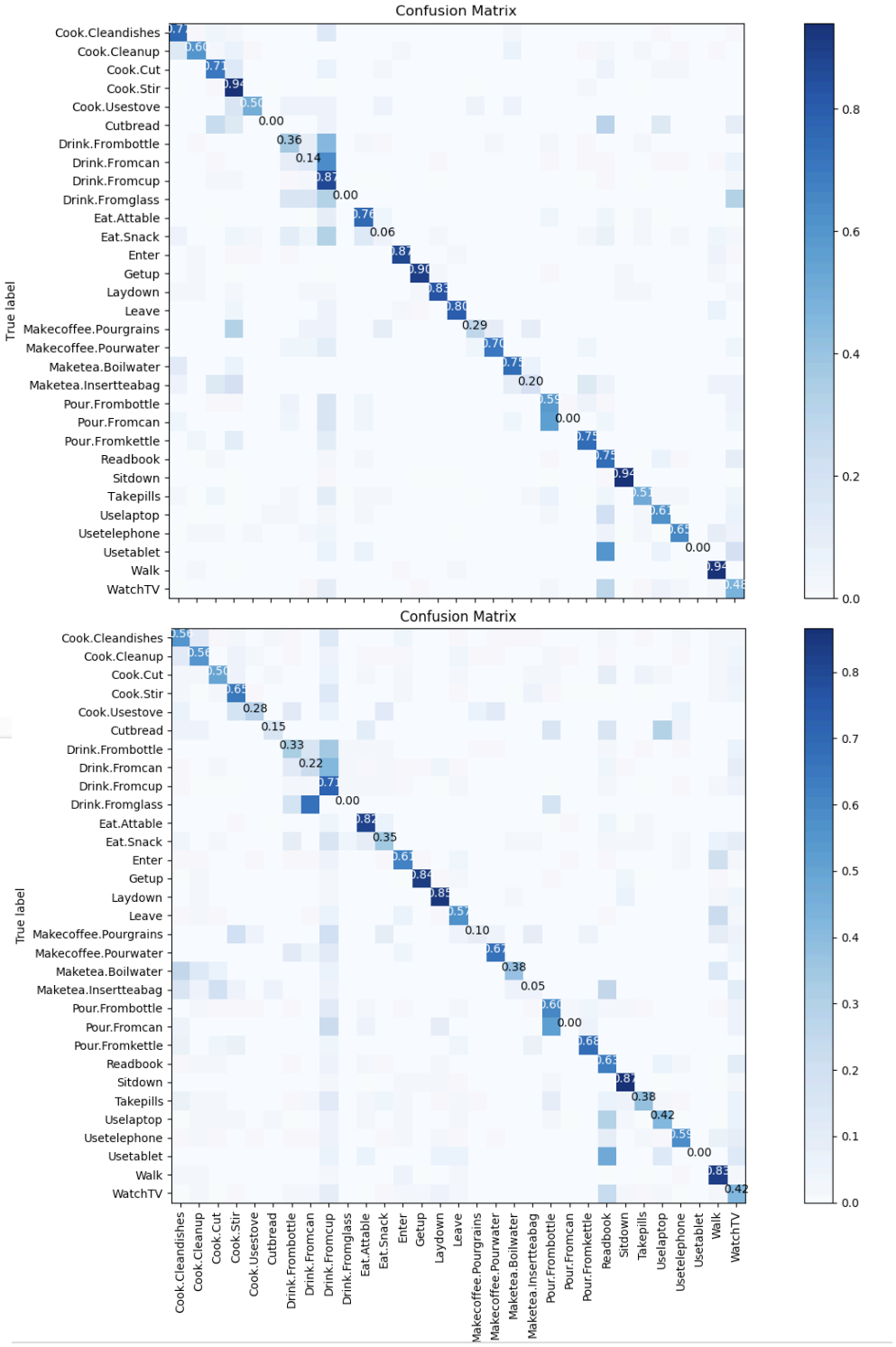

Figure 3. **Confusion matrix** of (top) 2s-AGCN with SSTA-PRS pose data and (bottom) with original pose data on Smarthome (CS Protocol).

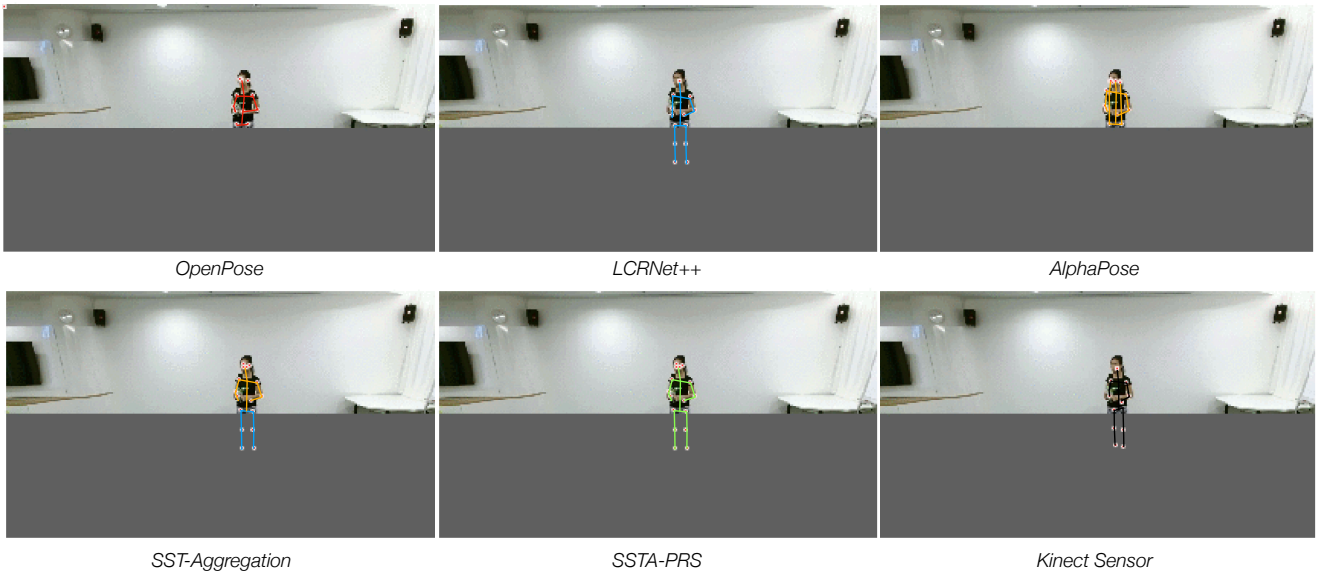

Figure 4. Visualization of poses from expert estimators, proposed SST-A, SST-PRS and Kinect Sensor v2 on **NTU-Pose**.

Figure 5. Visualization of poses from proposed SSTA-PRS on **Toyota Smarthome**.

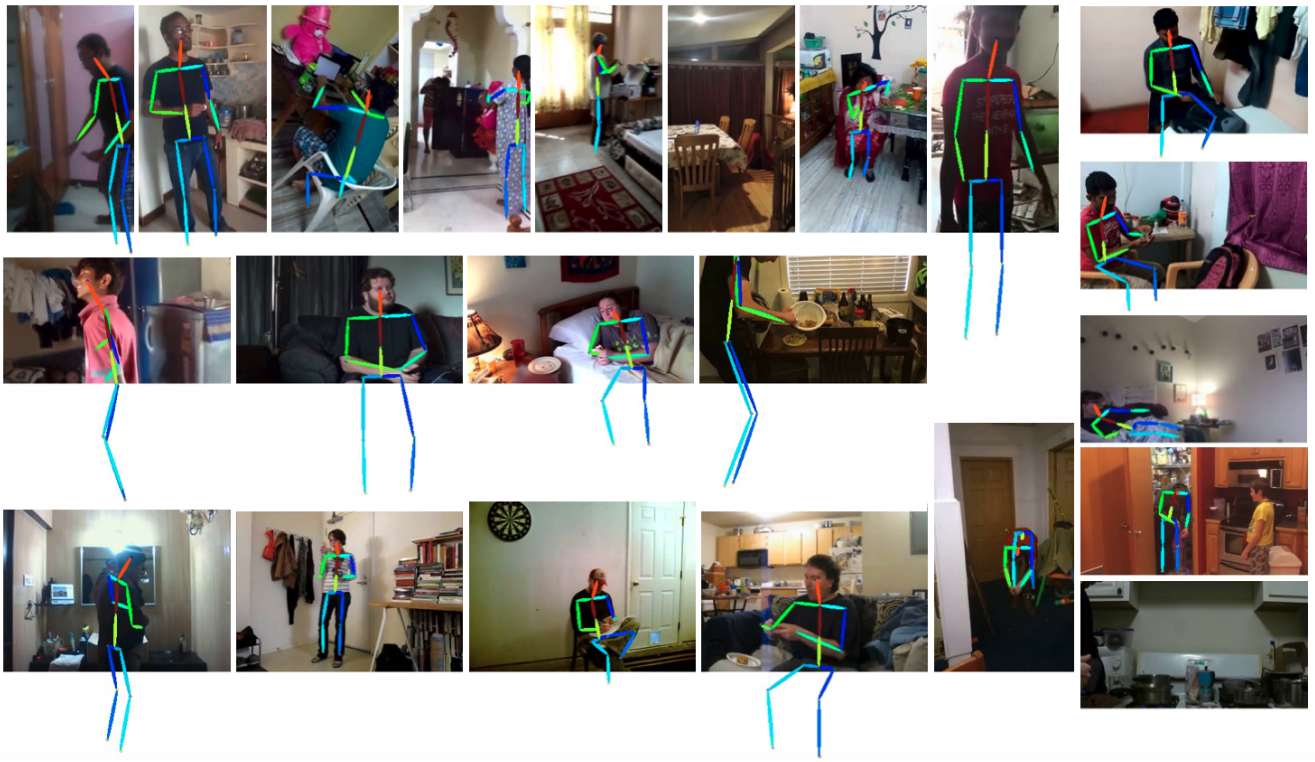

Figure 6. Visualization of poses from proposed SSTA-PRS on **Charades**.

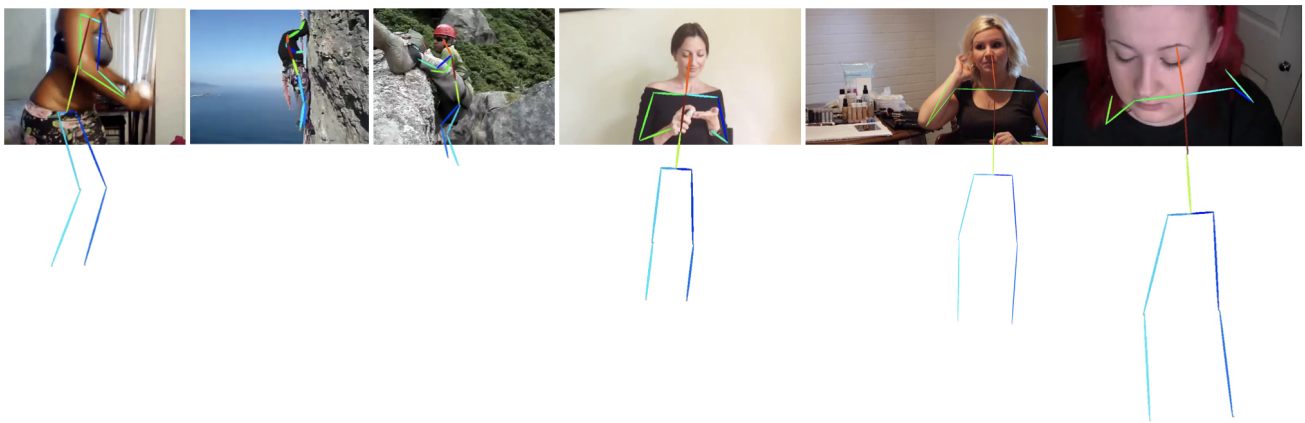

Figure 7. Visualization of poses from proposed SSTA-PRS on **Kinetics-50**.