Understanding Spatio-Temporal Relations in Human-Object Interaction using Pyramid Graph Convolutional Network

Hao Xing and Darius Burschka

***Abstract*— Human activities recognition is an important task for an intelligent robot, especially in the field of human-robot collaboration, it requires not only the label of sub-activities but also the temporal structure of the activity. In order to automatically recognize both the label and the temporal structure in sequence of human-object interaction, we propose a novel Pyramid Graph Convolutional Network (PGCN), which employs a pyramidal encoder-decoder architecture consisting of an attention based graph convolution network and a temporal pyramid pooling module for downsampling and upsampling interaction sequence on the temporal axis, respectively. The system represents the 2D or 3D spatial relation of human and objects from the detection results in video data as a graph. To learn the human-object relations, a new attention graph convolutional network is trained to extract condensed information from the graph representation. To segment action into sub-actions, a novel temporal pyramid pooling module is proposed, which upsamples compressed features back to the original time scale and classifies actions per frame.**

We explore various attention layers, namely spatial attention, temporal attention and channel attention, and combine different upsampling decoders to test the performance on action recognition and segmentation. We evaluate our model on two challenging datasets in the field of human-object interaction recognition, i.e. Bimanual Actions and IKEA Assembly datasets. We demonstrate that our classifier significantly improves both framewise action recognition and segmentation, e.g., F1 micro and F1@50 scores on Bimanual Actions dataset are improved by 4.3% and 8.5% respectively.

I. INTRODUCTION

As part of human activities, *human-object interactions* (HOIs) are closely related to the surrounding environment and the objects in the scene. Recognizing HOI in videos is a fundamental task in understanding human activities, in which the sub-activities are segmented and recognized per frame by analyzing the interactive relations between human and objects [1]. When human and objects are simply represented by skeleton and center points, these relations naturally form a relation graph in both spatial and temporal dimensions, which can describe their relative positions and dynamic interactions during the activity. Benefiting from the development of deep learning in the field of vision, we can easily build a spatial relation graph by detecting human and object in scenes. However, it is still challenging to discover the temporal structure of sub-actions in a complex task.

Currently, the available graph convolutional networks (GCN) [2], [3] primarily focus on the overall prevalent action being executed, in which only a single action is performed in one set of clips. These methods typically exploit the cascaded structures and can successfully extract and concentrate spatio-temporal features. However, they limit the action recognition task to assigning action labels to the given segments [4], [5]. Can the extracted spatio-temporal information be used for exploring the temporal structure of activities, i.e., action segmentation?

Regarding this question, we find that it is similar to the difference between image classification and segmentation, where image classification usually adopts a cascade structure, extracts high-level features globally and classify the whole image [6], and image segmentation focuses on the distinction between pixels by upsampling the cascaded features back to the original scale [7].

Hence, in this paper, we propose Pyramid Graph Convolution Network (PGCN) to improve HOI recognition and segmentation by combining the cascaded graph convolutional network with a novel temporal upsampling module, namely temporal pyramid pooling (TPP). Due to the dynamic interactive relations between human and objects, we introduce a novel spatial attention mechanism in GCN to adaptively generate new edges between strongly correlated vertices throughout the activity. The framewise recognition and segmentation capabilities of PGCN are demonstrated with superior quantitative and qualitative performance on two challenging human-object interaction datasets.

Overall, the technical contributions of the paper are:

- We propose Pyramid Graph Convolution Network that utilizes a novel temporal pyramid pooling module to extend the capabilities of GCNs for action segmentation.
- We present a new spatial attention mechanism that can improve action recognition by adaptively generating spatial relation graph in dynamic human-objects interaction scenes.
- We examine our model on two challenging HOI datasets, the Bimanual Actions and IKEA Assembly datasets. Compared to other good action recognition and segmentation approaches, our model achieves the best quantitative and qualitative performance on both datasets.

The rest of the paper is organized as follows: in section II, we briefly review existing approaches of GCNs, action segmentation and HOI recognition. Section III introduces the proposed PGCN. Section IV reports experimental results and discussions. Section V concludes the paper.

Authors are with Machine Vision and Perception Group, Department of Computer Science, Technical University of Munich, Arcisstraße 21, 80333 Munich, Germany hao.xing@tum.de, burschka@cs.tum.edu

II. RELATED WORK

A. Graph convolution networks

Recently, Graph Convolution Networks (GCNs) designed for representation of structured data raise the attention. The GCNs can be categorized into two classes: spatial and spectral. The spatial GCNs operate the graph convolutional kernels directly on spatial graph nodes and their neighborhoods [8]. Yan et al. [2] proposed a Spatial-Temporal Graph Convolutional Network (ST-GCN), which extracts spatial feature from the skeleton joints and their naturally connected neighbors and temporal feature from the same joints in consecutive frames. Shi et al. [3] introduced a two stream Graph Convolutional Network (2s-GCN) based on ST-GCN, which not only extracts features from skeleton joints but also considers the direction of each joint pair (bone information). Chen et al. [9] proposed a Channel-wise Topology Refinement Graph Convolution Network (CTR-GCN) that refines a spatial attention mechanism on channel dimension to efficiently learn dynamical features in different channels.

The spectral GCNs consider the graph convolution in form of spectral analysis [10]. Henaff et al. [11] developed a spectral network incorporating with graph neural network for the general classification task. Kipf and Welling [12] extends the spectral convolutional network further in the field of semi-supervised learning on graph structured data.

This work follows the spatial GCNs that operate nodes and edges on spatial domain.

B. Action segmentation

Action Segmentation aims to segment activity by exploring the temporal structure [13]. As part of earlier works, the hidden Markov model (HMM) is often used to find activity temporal structure. Pantic et al. [14] introduced a facial profile recognition scheme combining with HMM to segment facial actions. Some other approaches [15], [16] segment action using a sliding window and comparing the similarity between multiple temporal scales. More recently, convolutional neural networks (CNNS) and recurrent neural networks (RNNs) were main streams for action segmentation. For instance, Shou et al. [13] proposed a multi-stage CNN model to classify and localize sub-actions in untrimmed long sequence. Fathi et al. [17] segment human activities by identifying state changes of objects and materials in the environment using a RNN model. Motivated by the success of temporal convolution in Nature Language Process (NLP) area, many works applied various temporal convolution networks for action segmentation task, such as dilated temporal convolution [18], encoder-decoder temporal convolution [19]. Very recently, attention mechanism from transformer has been successfully applied to action segmentation [20], due to its strong ability of extracting global information. However, the attention mechanism requires known number of involved objects and subjects to define the size of adjacent matrix.

In this work, we take advantage of the attention mechanism to adaptively extract human-objects spatial relations with single subject and known object number and classes.

C. Human-object interaction recognition

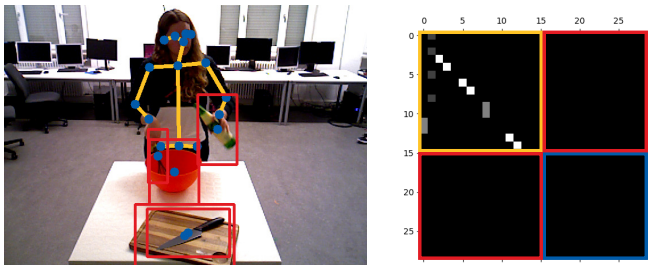
Different from action segmentation task, the HOI recognition task aims at detecting HOI label for whole trimmed action clip. Feichtenhofer et al. [21] introduced a two-stream 2D CNN that utilizes features from both appearance in still images and stacks of optical flow. In a more recent work [22], authors proposed a two-stream inflated 3D CNN (I3D) that improves the ability of 2D CNNs in extracting spatial-temporal features. Dreher et al. [23] presented a graph network that uses three multilayer perceptron (MLP) blocks to update nodes, edges and aggregation features from graph representation of HOI. Authors also published their HOI dataset, namely Bimanual Actions dataset. Asynchronous-Sparse Interaction Graph Networks (ASSIGN) [1] is a recent attempt on the HOI recognition task. It used a recurrent graph network that automatically detect the structure of interaction events associated with entities of a sequence of interaction, which are defined as human and objects in a scene. However, the short-term memory of recurrent networks limits their performance in analyzing global temporal structures. In order to expand the receptive field, we adopt dilated convolution layers [24] in the head of our temporal pyramid pooling module, which constrains the implementation in real-time scenarios as it requires relations from future.

III. PYRAMID GRAPH CONVOLUTIONAL NETWORK

The idea of pyramid graph convolutional network is inspired by upsampling methods for solving image semantic segmentation tasks. A common purpose of both image segmentation and action segmentation is to predict every single elemental unit of the input data by extracting different levels of semantic features and corresponding such features back to the input data to build a segment map. The basic idea of PGCN is to downsample the large-scale data to distill helpful spatial information, which is normally with a smaller temporal scale, and then upsample the distilled information back to the same temporal scale as the input. This is also known as an encoder-decoder structure.

A. Graph construction

In order to accurately describe the relationship between people and objects without being affected by texture information, we represent people as skeletons and objects as center points. All skeleton joints and object points are vertexes and their connections are represented as edges. Each vertex has inward, outward and self-connecting edges [3]. The connections between skeleton joints are naturally defined by the pose architecture, with inward connections from each joint to adjacent joints that are closer to the center of the body, and outward connections in reverse. However, object-related connections (human-objects and objects-objects) are challenging due to the dynamic nature of the scene. In this work, we consider that there is no initial connections between objects-objects and between human-objects joints, see Fig. 1



(a) (b)

Fig. 1: The initial spatial relation graph: (a) Spatial graph with notes (blue) and edges (orange) on example of Bimanual Actions dataset [23]; (b) Initial inwards adjacent matrix with skeleton inward edges (orange block), empty human-objects (red blocks) and objects-objects edges (blue block).

(a). All edges form a binary adjacency matrix A, in which $a_{ij} = 1$ means vertexes v_i and v_j are connected from i to j. Since the initial connections between objects-related pairs of vertexes are not considered, both inward and outward edges are empty, as shown in the Fig 1 (b).

Given the adjacency matrix, a spatial scene graph feature map can be obtained by the following equation:

$$\mathbf{G} = \hat{\mathbf{A}} \cdot \mathbf{F}_{in} \tag{1}$$

where $\mathbf{F}_{in}$ is the input skeleton-objects feature map, $\mathbf{G}$ is the graph feature map, and $\hat{\mathbf{A}}$ is column-wise normalization of $\mathbf{A}$.

B. Attention based graph convolutional encoder

Since the important human-objects interaction information is still missing in the constructed graph, we propose an attention based graph network, which adaptively update the initial adjacent matrix through the attention score map. The attention score is calculated by the dot product between nodes as follows:

$$M_{ij} = \frac{f_i \cdot f_j^T}{\sqrt{n}} \tag{2}$$

where M is the attention mask map, f is the node feature vector and i, j are the indices of nodes. In this work, we find that feeding mask maps into a 1-dimensional convolution layer contributes to the relationship learning process. As shown in Fig. 2, the input feature map is fed into two 2D convolution layers in parallel to generate two output maps with the same size. Their dot product is then fed into a 1D convolution layer with a *sigmoid* activation function to extract the attention mask. The final attention map is generated by the combination of the attention mask with the adjacent matrix as follows:

$$\mathbf{A}_{final,i} = \mathbf{M}_i + \hat{\mathbf{A}}_i = \mathbf{W}_i(\mathbf{F}_{1,i}^T \cdot \mathbf{F}_{2,i}) + \hat{\mathbf{A}}_i \tag{3}$$

where $\mathbf{M}$ is the attention mask that is extracted by the 1D convolution kernel on the dot product of feature maps $\mathbf{F}_1$ and $\mathbf{F}_2$, $\mathbf{W}$ is the kernel weight, $\mathbf{A}$ is the adjacent matrices and i is the index of the three connection types (*inwards*, *outwards*, *self-connecting*). In order to give more flexibility to the spatial graph, we set adjacency matrices as learnable parameters with given initial values.

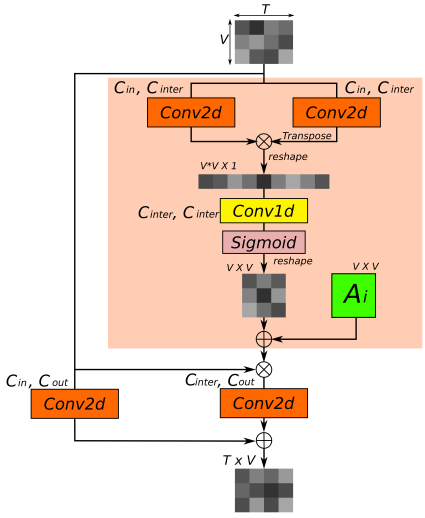


Fig. 2: Illustration of the attention unit (orange region) in a spatial convolutional layer

The output feature map of the spatial attention layer is extended to C_{out} output channels through an additional 2D convolution layer and is merged with the residual stream. The process can be mathematically expressed as follows:

$$\mathbf{G}_i = \text{Conv2d}(\mathbf{A}_{final,i} \cdot \mathbf{F}_{in}) + \text{res}(\mathbf{F}_{in}) \tag{4}$$

where $\mathbf{F}_{in}$ is the input feature map, res is the residual layer, and $\mathbf{G}$ is the $i-th$ output graph feature map. The final block output feature map is obtained by summing the outputs of all three types of connections as $\mathbf{G} = \sum_{i=1}^{3} \mathbf{G}_i$

In temporal dimension, we follow the ST-GCN [2], i.e., operating a 2D convolution kernel with size $K_t \times 1$ on $C \times T \times V$ feature maps, where K_t is set to 9 in this work.

Given the defined spatial, temporal layer, an attention based graph convolutional block is formed. In the encoder, we employ 10 basic blocks and connect them through the usual cascade structure, as introduced in [2], [3].

C. Temporal pyramid upsampling decoder

Given the introduced encoder, three graph feature maps $\mathbf{G}_{in} = \{\mathbf{G}^4, \mathbf{G}^7, \mathbf{G}^{10}\}$ of 4-th, 7-th and 10-th blocks are jointly taken as input into the temporal upsampling module, which contain different level semantic information. Since these feature maps have different size, we unify the number of channels through a 2D convolution kernel and interpolate all feature maps to the initial time scale concatenate them along channel dimension. A segmentation feature extraction is performed through four parallel dilated convolution operations [24] as following:

$$\mathbf{G}_{out} = \sqcup_{i=1}^{4} \sigma(\mathbf{G}_{in,u}\mathbf{W}_i^d + \mathbf{B}_i^d) \tag{5}$$

where $\mathbf{G}_{out}$ is output graph feature map, $\mathbf{G}_{in,u}$ is upscaled input graph feature map, $\sqcup_{i=1}^{4}$ indicates the concatenation operation with 4 streams, σ is the *ReLU* activation function, $\mathbf{W}_i^d$ and $\mathbf{B}_i^d$ are parameters of i-th dilated convolutional kernel.

In order to extract a global contextual prior for prediction, a temporal pyramid pooling module is utilized before the

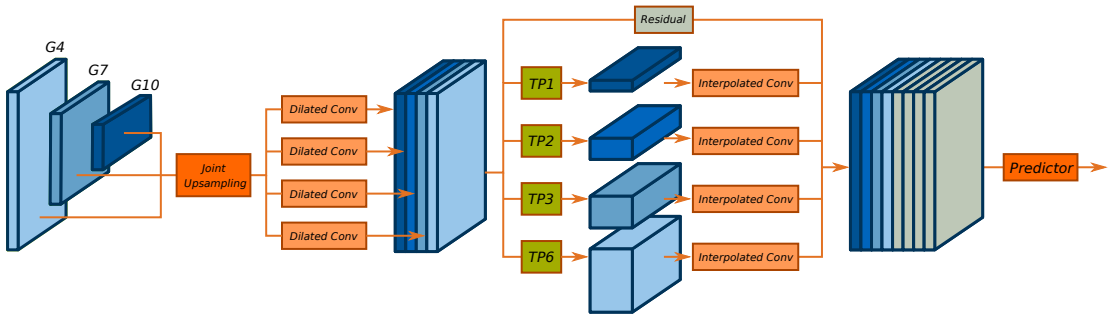


Fig. 3: Framework of temporal pyramid pooling decoder with three input graph feature maps: $\mathbf{G}^4$, $\mathbf{G}^7$ and $\mathbf{G}^{10}$, where $TP\ i$ is temporal pooling block with output size i.

predictor. In semantic segmentation tasks regarding images, global average pooling is a general choice as the global contextual prior. However, in the action segmentation, the features in temporal and spatial dimensions need to be handled differently. Since the final segmentation is in temporal dimension, i.e., the sequence of predicted labels per frame, four pyramid temporal average pooling blocks of different scales are first performed along the temporal dimension to extract a segment prior with multiple receptive fields.

Given the time series dilated graph feature map $\mathbf{G}_{out} \in \mathbb{R}^{N\times T}$ with N spacial nodes and T frames, it can be represented as a set of time segments at level i as $\mathbf{G}_{out} = \{\mathbf{G}_1, \cdots, \mathbf{G}_i\}$. A temporal filter with average pooling operator is applied to each time segment $[t_{min}, t_{max}]$ and provides a single feature vector for each segment as:

$$\mathcal{O}(\mathbf{G}_i) = \frac{\sum_{t_{min}}^{t_{max}} g_t^i}{t_{max} - t_{min}} \tag{6}$$

Then, a convolution layer is performed in spatial dimension to extract global spatial information with various temporal scales as following:

$$\mathbf{F}_{out} = \sigma(\mathbf{G}_{out}\mathbf{W}_s + \mathbf{B}_s) \tag{7}$$

where $\mathbf{W}_s \in \mathbb{R}^{k\times 1}$ and $\mathbf{B}_s \in \mathbb{R}^{k\times 1}$ are parameters of the spatial convolutional kernel, and $k \times 1$ indicates the kernel size.

The four low-dimension output feature maps are directly upsampled by bilinear interpolation to have the same temporal and spatial lengths as the original feature maps. At last, four different levels of features are concatenated with the residual feature map. After obtaining the feature map containing global contextual prior with various scales and framewise local features, a convolution based predictor is used to generate framewise interaction labels. The framework is illustrated in Fig. 3.

IV. EXPERIMENTS AND RESULTS

To evaluate the performance of proposed model, we experiment on two challenging human-object interaction recognition datasets: Bimanual Actions dataset [23] and IKEA Assembly dataset [25]. We first perform detailed ablation study on the Bimanual Actions dataset [23] to examine the contributions of the proposed model components. Then, we evaluate the final model on both datasets and compare the results with other state-of-the-art methods.

A. Dataset

Bimanual Actions Dataset [23] was build for human object interaction detection in third-person perspective. It contains 540 recordings with a total runtime of 2 hours 18 minutes. It has framewise predictions of 12 objects (3D bounding boxes) and 6 subjects (3D skeletons), including the both hands of subjects as one of 14 possible interaction categories. In each record, a single person is performing a complex daily task in one of the two set environments, namely kitchen and workshop. The authors of the dataset recommend a benchmark: **leave-one-subject-out** cross-validation that contains records from one subject for validation and the rest subjects for training.

IKEA Assembly Dataset [25] is a more challenging and complex human-object interaction dataset, which contains a total of 16,764 annotated actions with an average of 150 frames per action ($\sim$ 35.27h). The authors proposed a **cross-environment** benchmark, in which the test environments do not appear in the trainset and vise-versa. The trainset and testset consist of 254 and 117 scans, respectively.

B. Experimental settings

We evaluate the proposed model on the task: HOI framewise recognition and temporal segmentation. Two main evaluation metrics, i.e., F1-score and F1@k, are selected for framewise recognition and segmentation, respectively. the F1-score is formulated as: F1 $= tp/(tp + 0.5(fp + fn))$ where tp means true positive predictions, fp and fn refer to false positive and false negative predictions, respectively. For the F1@k score, common values of $k = 0.10$, 0.25, and 0.50 are used, in which the true or false positive for each predicted segment is determined by comparing the intersection over union (IoU) with threshold $\tau = k/100$. Incorrect predictions and missed ground-truth segments are counted as false positive and false negative, respectively. Moreover, for the multi-class prediction task, micro-average and macro-average over F1-scores of all classes are adopted as the framewise recognition metric. In the experiment comparing popular methods on IKEA Assembly dataset, we used top1 and macro-recall metrics to evaluate models.

For the Bimanual Action dataset [23], we use the center of 3D object bounding boxes and 3D human skeleton data released by authors [23], leave subject 1 out for validation in ablation study, and do leave-one-subject-out cross-validation

TABLE I: The F1 score of framewise prediction and F1@k score of action segmentation using original baseline model and models with different modifications in each unit

Encoder [a]			Decoder [b]		Evaluation Metrics [c]				
Spatial	Temporal	Channel	TPP	Fast-FCN	F1 macro (%)	F1 micro (%)	F1@10 (%)	F1@25 (%)	F1@50 (%)
-	-	-	-	✓	65.28	80.01	66.51	62.44	51.37
-	-	-	✓	-	65.17	81.80	86.36	83.66	71.84
✓	-	-	-	✓	70.92	83.09	70.38	66.26	66.27
✓	-	-	✓	-	**81.50**	**86.92**	**88.38**	**85.06**	73.88
-	✓	-	-	✓	75.93	83.42	67.83	63.51	52.68
-	✓	-	✓	-	77.26	84.94	78.77	75.46	61.43
-	-	✓	-	✓	70.16	83.56	74.16	70.55	58.92
-	-	✓	✓	-	67.39	82.50	88.23	84.86	**74.10**
✓	✓	-	✓	-	80.29	85.25	84.38	81.46	68.55
✓	-	✓	✓	-	69.65	80.57	84.09	81.07	66.99
-	✓	✓	✓	-	71.42	82.75	85.36	81.45	69.21
✓	✓	✓	✓	-	72.39	83.63	85.68	81.94	70.86

[a] We compare the performance of attention layer in the encoder setup on different dimensions, namely spatial, temporal and channel.
[b] The decoder is the common Fast-FCN [7] when there is no temporal pyramid pooling block.
[c] The best results comparing all modifications are in **bold**; The best results between TPP and Fast-FCN in the decoder setup are underlined

in comparison with other popular methods. For the IKEA Assembly dataset [25], we use the offered center of 2D object bounding boxes and 2D human skeleton data, and follow the cross-environment benchmark.

The models and experiments are implemented and conducted on the PyTorch deep learning framework with a single NVIDIA-2070 GPU. The optimization strategy is selected to be the widely used stochastic gradient descent (SGD) with Nesterov momentum (0.9). Cross-entropy is applied as the loss function for the gradient back propagation. 32 batch size is applied for both training and testing. The weight decay is set to be 0.0001. The training process contains 60 epochs in total. The initial learning rate is set to be 0.1, and is divided by 10 at the 20-th and 40-th epoch.

C. Ablation studies

We examine the contribution of proposed components to the framewise HOI recognition and segmentation with leaving subject 1 on the Bimanual Actions dataset [23]. The baseline is the single joint stream of 2s-AGCN [3].

In order to get the best performance of the attention unit, we evaluate the unit on the dimension of spatial, temporal, channel and their combinations. The proposed temporal pyramid pooling module is compared with the baseline FastFCN [7].

The results of ablation studies are shown in Table I, where the performance of each setting is quantified by F1 and F1@k scores. From the F1@k score in the right column of the table, it is obviously that the temporal pyramid pooling block shows improvement on relation segmentation for each specific setting. Hence, we include the temporal pyramid pooling block for rest experiments on combined attention layers.

From the F1 scores in the middle column, we can see that all proposed components improve the performance of the baseline model AGCN [3] on framewise recognition. Moreover, since the spatial attention unit extracts the basic features representing spatial distribution and relations of nodes per frame, model with spatial attention unit has the best performance among all model settings. The temporal attention unit extracts temporal relations between consecutive

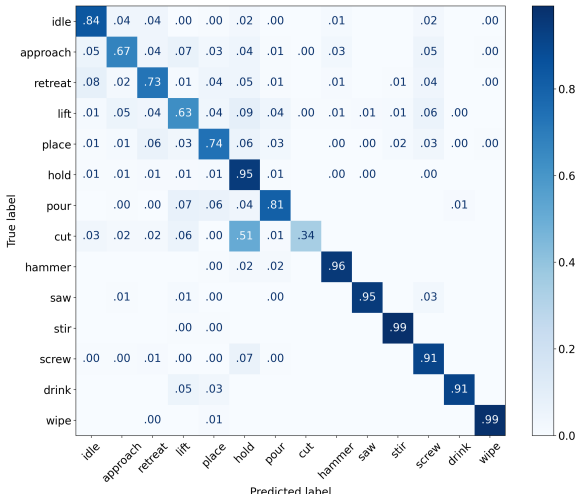


Fig. 4: Normalized confusion matrix for the top prediction of accumulative framewise classification correctness over all folds on Bimanual Actions dataset [23].

frames, which is beneficial for segmentation-known action recognition rather than the action segmentation. The channel attention layer focuses on the importance of distinguishing between channels, which is conducive to the classification of an entire clip of single action rather than the segmentation. The performance of combined models suffers from poor attention layers, namely temporal and channel attention layers.

Besides F1 scores, we also evaluate the Top-1 accuracy of the proposed model on Bimanual Actions dataset [23]. Fig. 4 depicts the normalized confusion matrices for the top prediction. A major confusion of the classifier is the prediction of *hold* while the true action is *cut*. The cause for the wrong prediction is that we use *wrist* joints to represent *hands*, which has small range of motion and is easily mistaken for *holding* a *knife*. Therefore, the prediction of *cut* usually has a large range of motion and rarely false recognized from the action of *hold*. There is also a number of confusions between actions *approach*, *retreat lift* and *place*. This is suffering from unstable object bounding box, namely the object detection method. Additional, *approach*

TABLE II: Comparison of framewise action recognition with state-of-the-art methods on Bimanual Actions dataset [23]

Model	F1 macro (%)	F1 micro (%)
Dreher et al. [23]	63.0	64.0
AGCN+FastFCN	65.3	80.0
AGCN+TPP	65.2	81.8
ST-GCN+FastFCN	68.7	82.5
ST-GCN+TPP	69.3	82.7
CTR-GCN+FastFCN	71.1	82.3
CTR-GCN+TPP	72.0	82.9
Independent BiRNN [1]	74.8	76.7
Relational BiRNN [1]	77.5	80.3
ASSIGN [1]	79.8	82.6
PGCN (Ours)	**81.5**	**86.9**

TABLE III: Cross validation results of action segmentation in comparison with state-of-the-art methods on Bimanual Actions dataset [23]

Model	F1@10 (%)	F1@25 (%)	F1@50 (%)
Dreher et al. [23]	40.6 ± 7.2	34.8 ± 7.1	22.2 ± 5.7
Independent BiRNN [1]	74.8 ± 7.0	72.0 ± 7.0	61.8 ± 7.3
CTR-GCN+FastFCN	74.9 ± 8.1	72.2 ± 8.7	66.6 ± 11.4
Relational BiRNN [1]	77.7 ± 3.9	75.0 ± 4.2	64.8 ± 5.3
ASSIGN [1]	84.0 ± 2.0	$81.2 \pm \mathbf{2.0}$	$68.5 \pm \mathbf{3.3}$
CTR-GCN+TPP	84.8 ± 3.2	82.1 ± 4.0	73.5 ± 5.6
PGCN (Ours)	$\mathbf{88.5 \pm 1.1}$	$\mathbf{85.5 \pm 2.0}$	$\mathbf{77.0} \pm 3.4$

and *retreat* are usually executed fast (sometimes within 5 frames). An example can be found in qualitative results. These problems can be mitigated by stable object detection and pose estimation methods. However, it is not main focus in this work and will not be addressed.

D. Comparison with the state-of-the-art

The proposed PGCN model is compared with the state-of-the-art action recognition and segmentation methods on Bimanual Actions [23] and IKEA Assembly datasets [25]. The methods used for comparison include the model proposed by Dreher et al. [23], Independent BiRNN, Relational BiRNN, ASSIGN [1] and several popular graph convolutional networks: ST-GCN [2], AGCN [3] and CTR-GCN [9] combining with two decoders, namely FastFCN [7] and the proposed temporal pyramid pooling (TPP) module.

The performance in terms of F1 score and F1@k on the Bimanual Actions dataset [23] are listed in Table II and Table III, respectively. The PGCN outperforms both the state-of-the-art and baselines in every configuration of the F1 and F1@k measure, e.g., the F1 macro and micro score are improved by 1.7% and 4.3% respectively. Moreover, it can be observed that the proposed temporal pyramid pooling block improves significantly the performance in terms of F1@k score (by 4.5%, 4.3% and 8.5% compared to ASSIGN, respectively), which again confirms its efficiency in action segmentation. The ASSIGN uses the Bi-directional Gated Recurrent Unit to combine information from consecutive frames, which limitedly enhances the extraction of temporal information and causes the shift-segmentation. Other methods employ separate segmentation label, which is lack of temporal information and leads to an over-segmentation case. More evidences can be found in the qualitative results.

Table IV presents the top-1 accuracy, micro-recall and F1@k score on IKEA Assembly dataset [25]. In terms of top-1, and all three F1@k scores, PGCN outperforms all other popular methods. This further demonstrates that our modeling entities with spatial attention and temporal pyramid pooling modules is a more competitive way to recognize and segment action per frame. Due to the uneven distribution of the dataset [25], the macro-recall of all methods is low.

TABLE IV: Framewise recognition and segmentation results in terms of top-1 accuracy, macro-recall, and F1@k on IKEA Assembly dataset [25]

Model	top 1	macro	F1@ (%) 10	25	50
HCN [26]	39.15	28.18	-	-	-
ST-GCN [2]	43.40	26.54	-	-	-
multiview+HCN [25]	64.25	**46.33**	-	-	-
ST-GCN+TPP	68.92	25.63	66.92	59.66	41.33
AGCN+TPP	70.53	27.79	76.32	69.85	52.14
CTR-GCN+TPP	78.70	37.98	78.84	72.68	54.40
PGCN (Ours)	**79.35**	38.29	**81.53**	**76.28**	**58.07**

E. Qualitative results

We present the detail outputs of PGCN model and related methods on examples from Bimanual Actions [23] and IKEA Assembly dataset [25]. Fig. 5 shows a simple example of *sawing* in Bimanual Actions [23], where both PGCN and ASSIGN have a stronger ability to prevent over-segmentation than Relational BiRNN. Our PGCN demonstrates more accurate segmentation than ASSIGN, which even recognizes correctly the frame index between *Approach* and *Hold* (0 frame error) in the example. As aforementioned, brief movement can easily lead to false predictions, see the end of the ground-truth and predictions.

Besides the simple example, Fig. 6 presents a segmentation example of complex *assembly side table* task on the IKEA Assembly dataset [25], where we compare the qualitative performance of methods with the same decoder and different encoders. It can be seen that our PGCN model prevents under-segmentation better than other models with the same decoder, which further demonstrates the effectiveness of our spatial attention unit. From simple to complex tasks, our model demonstrates strong and stable performance.

V. CONCLUSIONS

In this work, we introduce a novel pyramid graph convolutional network for understanding human-object interaction relation sequences via action recognition and segmentation, which includes a spatial attention graph convolutional encoder and a temporal pyramid pooling decoder.

The two components are complementary to each other, i.e., the spatial attention mechanism provides high-level spatial relations between human and objects to the decoder, and the temporal pyramid pooling decoder upsamples these spatial features to the original time-scale and predict framewise labels. Experimental analysis into PGCN's components shows that the new attention layer improves the accuracy of action recognition, further mitigate under-segmentation, and the new temporal pyramid block has strong ability to prevent action over- and shift-segmentation. Results on two HOI

Fig. 5: Comparison of the qualitative results on Bimanual Actions dataset [23] for a *sawing* example

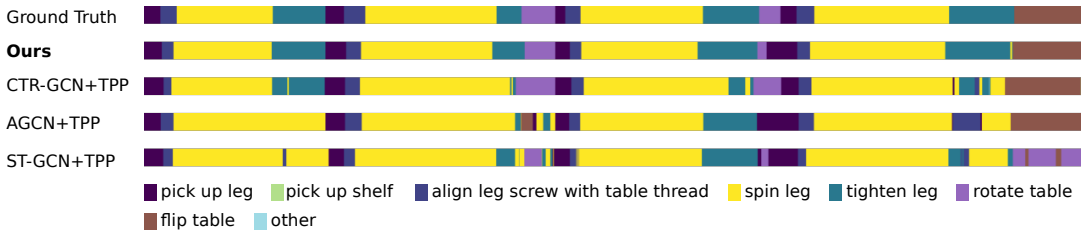


Fig. 6: Comparison of the qualitative results on IKEA Assembly dataset [25] for an *assembly side table* example

datasets with different input formats (2D and 3D) show that PGCN has a general capability that can be implemented on other structural-represented domains. Our future work will explore multi-persons involved HOI recognition and segmentation, and try to overcome the constrains of temporal pyramid pooling model and implement our model in real-time Human-Robot Collaboration tasks.

ACKNOWLEDGMENT

We gratefully acknowledge the funding of the Lighthouse Initiative Geriatronics by StMWi Bayern (Project X, grant no. 5140951) and LongLeif GaPa GmbH (Project Y, grant no. 5140953), and our special thanks goes to Xiongfei Ma for his help.

REFERENCES

[1] R. Morais, V. Le, S. Venkatesh, and T. Tran, "Learning asynchronous and sparse human-object interaction in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 041–16 050.

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

[3] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 026–12 035.

[4] 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.

[5] H. Xing and D. Burschka, "Skeletal human action recognition using hybrid attention based graph convolutional network," in *26th International Conference on Pattern Recognition (ICPR)*, 2022.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[7] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, "Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation." *CoRR*, vol. abs/1903.11816, 2019.

[8] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3595–3603.

[9] 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.

[10] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[11] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[13] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1049–1058.

[14] M. Pantic and I. Patras, "Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 2, pp. 433–449, 2006.

[15] L. Zelnik-Manor and M. Irani, "Statistical analysis of dynamic actions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1530–1535, 2006.

[16] H. Xing, Y. Xue, M. Zhou, and D. Burschka, "Robust event detection based on spatio-temporal latent action unit using skeletal information," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2941–2948.

[17] A. Fathi and J. M. Rehg, "Modeling actions through state changes," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2579–2586.

[18] N. Hussein, E. Gavves, and A. W. Smeulders, "Timeception for complex action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 254–263.

[19] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1003–1012.

[20] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[21] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[22] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[23] C. R. Dreher, M. Wächter, and T. Asfour, "Learning object-action relations from bimanual human demonstration using graph networks," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 187–194, 2019.

[24] P. Wang, Y. Cao, C. Shen, L. Liu, and H. T. Shen, "Temporal pyramid pooling-based convolutional neural network for action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2613–2622, 2016.

[25] Y. Ben-Shabat, X. Yu, F. Saleh, D. Campbell, C. Rodriguez-Opazo, H. Li, and S. Gould, "The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose," 2020.

[26] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2018, p. 786–792.