Spatio-Temporal Dynamic Inference Network for Group Activity Recognition

Hangjie Yuan[1] Dong Ni[1,2*] Mang Wang[3]
[1]College of Control Science and Engineering, Zhejiang University, Hangzhou, China
[2] State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, China
[3]DAMO Academy, Alibaba Group, China
{hj.yuan,dni}@zju.edu.cn, wangmang.wm@alibaba-inc.com

1. Supplementary Material

In this supplementary material, we will detail several aspects which are omitted in the main paper due to length limits.

- Section 1.1 Implementation details for the codebase.
- Section 1.2 More qualitative analysis of the results.

1.1. Implementation Details for the Codebase

Previous methods may claim their state-of-the-art performances while varying in backbones and input modalities, which influence the results a lot. Just like other video-related tasks (*e.g.*, video action classification [5] and video detection [4]), we think it to be more appropriate to set backbones and input modalities the same while comparing.

Generally, our codebase follows the pipeline outlined in the main paper: spatial-temporal feature extraction and reasoning. The feature extraction stage composes of the backbone inference, the RoIAlign cropping and the feature embedding. All methods use the tracklets from [1]. The RoIAlign extracts a feature map of size 5×5 for every person feature. The embedding layer is instantiated by a fully-connected layer. For fair comparison, we all follow [7] to initialize the backbone parameters with ones from the base model. We use a batch size of 2 due to high image resolution. The codes are implemented using Python 3.6, Pytorch 1.2 and Torchvision 0.4 on the CentOS7 system. We conduct our experiments on NVIDIA TITAN RTX.

Furthermore, we provide more details for implementing reasoning modules.

- **PCTDM** [8] The original instantiation of PCTDM[1] is a bit different from our pipeline, which uses cropped and resized person images (224×224) to get person feature embeddings. In this codebase, we use RoIAlign features instead and change its backbone to ResNet-18.
- **ARG** [7] We strictly follow its original setting and publicly available codes[2]. The only adaptation is that we change its backbone to ResNet-18 following [9].
- **AT** [3] The original paper uses the backbone of I3D and HRNet to obtain person features. In this codebase, we change its backbone to ResNet-18 and use only RGB images as input.
- **HiGCIN** [9] The original instantiation of HiGCIN[3] is similar to [8], which uses cropped person images. Although we try using RoIAlign features, the experiment shows slow convergence. So, we only provide results using cropped person images. The #Params and FLOPs for HiGCIN are 1.05M and 184.99G for the reasoning module per video.
- **SACRF** [6] Also, we strictly follow the inference scheme from the original paper: a Self-attention Augmented Conditional Random Field (SACRF) and a Bidirectional UTE (BiUTE). The original paper only provides results with I3D+FPN+Alphapose[2] using RGB+Flow input. To seek for a fair comparison, we change its backbone to ResNet-18 and input modality to RGB images.

1.2. More Qualitative Analysis

Due to the length limits, we only illustrate one example in the main paper. In this supplementary material, we provide more visualization results to understand the model better.

In Figure 1, we illustrate one example of *right spike* activity from VD. Specifically, we visualize the group interaction graph in the upper right image of Figure 1. As we

[*] Corresponding author.

[1] Original PCTDM codes are available at https://github.com/ruiyan1995/Group-Activity-Recognition.

[2] Original ARG codes are available at https://github.com/wjchaoGit/Group-Activity-Recognition.

[3] Original HiGCIN codes are available at https://github.com/ruiyan1995/HiGCIN.

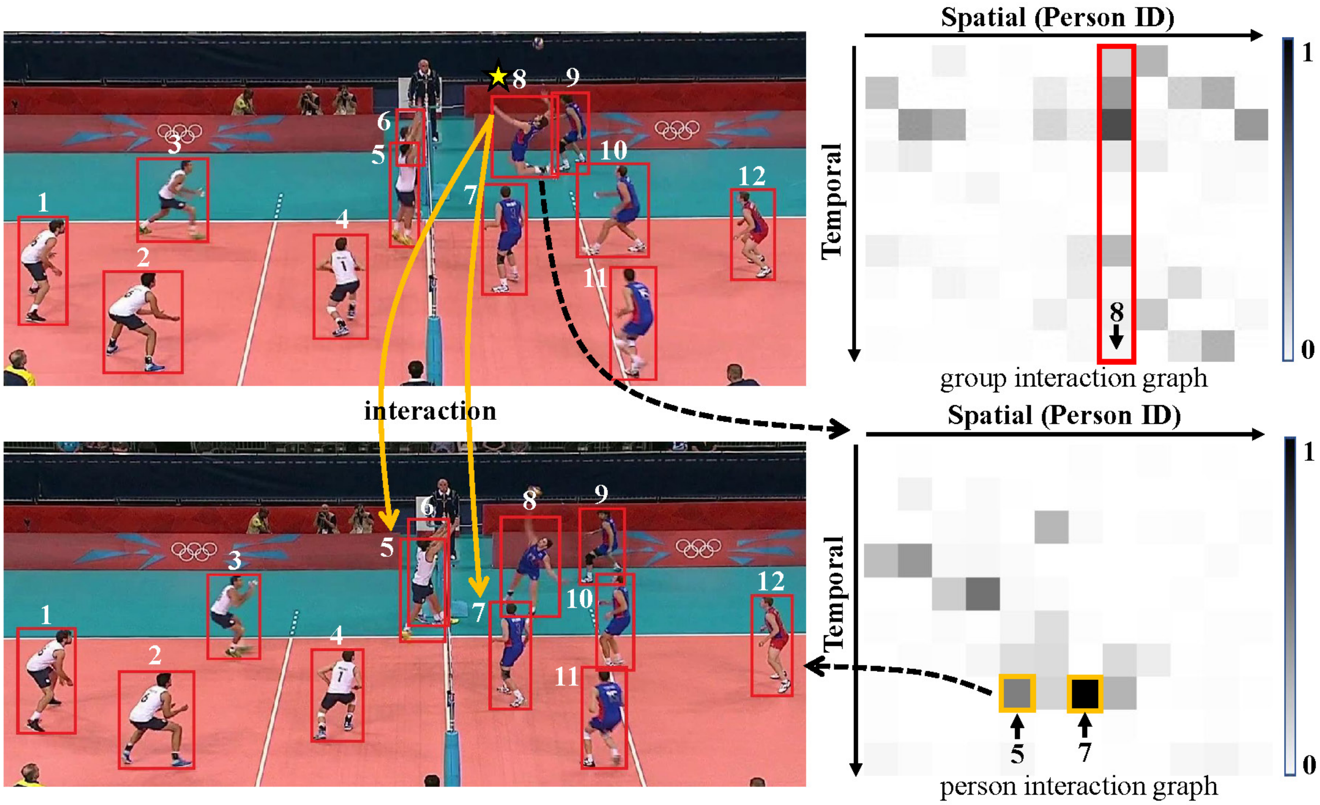

Figure 1. **Visualizations of a *right spike* activity example.** The upper left image is the starting image of the video clip. The upper right is the corresponding group interaction graph. The lower right is the interaction graph of the 8th person (key person, red boxes in the group interaction graph). The lower left illustrates two of the 8th person's key interactions (yellow boxes in the 8th person's interaction graph).

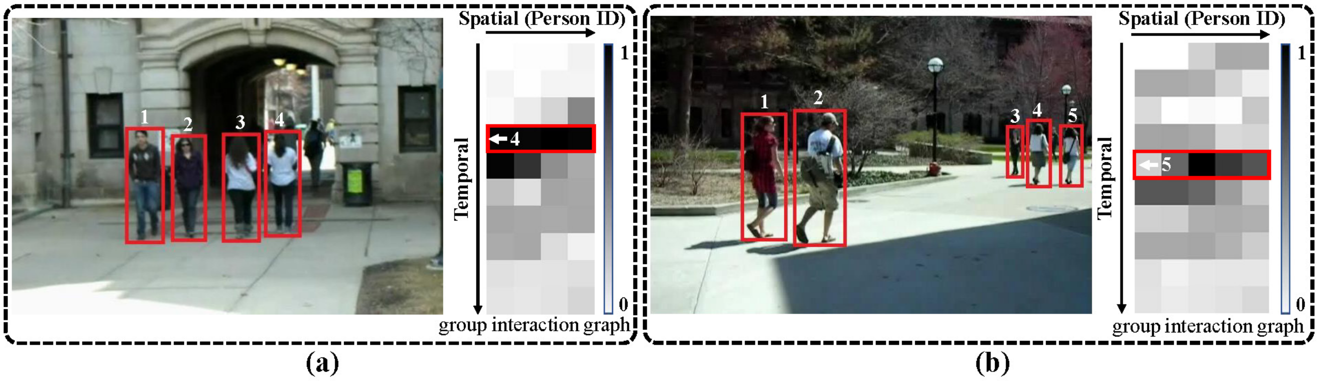

Figure 2. **Visualizations of two *moving* activity example.** For each example, we illustrate its group interaction graph on the right and its key frame (red boxes in the group interaction graph) on the left.

sum along the temporal axis in this group interaction graph, we can obtain that 8th person is the key person in this video clip. The 8th person in this video clip is performing the *spiking* action, which is semantically important for recognizing *right spiking* activity. Then we visualize the 8th person's person interaction graph. Although we initialize the interaction field locally, it still facilitates global-level interactions with the help of DIN. Moreover, we can observe which person interacts with the 8th person more in the spatio-temporal domain. We illustrate two of his key interactions (yellow boxes) in the 8th frame. The 5th person in the 8th frame is the one who is trying to block the ball. The 7th person in the 8th frame is the one who set the ball to the 8th person. They semantically interact with the key person.

In Figure 2, we illustrate two *moving* examples from

CAD. We visualize their group interaction graphs on the right of each example. Compared to VD's group interaction graphs, CAD's group interaction graphs have denser connections, indicating CAD needs more global interactions. Moreover, CAD's group interaction graphs show a more obvious temporal-independency than VD's. For the first presented example, the 4th frame is the key frame. For the second presented example, the 5th frame is the key frame.

References

[1] Timur Bagautdinov, Alexandre Alahi, François Fleuret, Pascal Fua, and Silvio Savarese. Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4315–4324, 2017.

[2] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017.

[3] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees GM Snoek. Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 839–848, 2020.

[4] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[5] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[6] Rizard Renanda Adhi Pramono, Yie Tarng Chen, and Wen Hsien Fang. Empowering relational network by self-attention augmented conditional random fields for group activity recognition. In *European Conference on Computer Vision*, pages 71–90. Springer, 2020.

[7] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu. Learning actor relation graphs for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9964–9974, 2019.

[8] Rui Yan, Jinhui Tang, Xiangbo Shu, Zechao Li, and Qi Tian. Participation-contributed temporal dynamic model for group activity recognition. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1292–1300, 2018.

[9] Rui Yan, Lingxi Xie, Jinhui Tang, Xiangbo Shu, and Qi Tian. Higcin: Hierarchical graph-based cross inference network for group activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.