# Detector-Free Weakly Supervised Group Activity Recognition
## *Supplementary Materials*

Dongkeun Kim[1] Jinsung Lee[2] Minsu Cho[1,2] Suha Kwak[1,2]
Department of CSE, POSTECH[1] Graduate School of AI, POSTECH[2]
https://cvlab.postech.ac.kr/research/DFWSGAR/

## 1. Experimental details

### 1.1. Implementation of reproduction

**NBA dataset.** We reproduce GAR [1, 6, 9, 12] and WSGAR [11] methods following the official code of DIN[1] [12] and the implementation description illustrated in its original paper, repectively. For a fair comparison, segment-based sampling [8], batch size of 4, the number of bounding boxes $N = 12$, and the number of frames $T = 18$ are applied for all methods. The only difference from the implementation of DIN is that all methods are trained in an end-to-end manner. Unless specified, all other hyperparameters are identical to those of the code from DIN. We provide more implementation details of each model below.

- **ARG** [9] ResNet-18 backbone replaces the original backbone of Inception-v3.
- **AT** [1] A single RGB branch is utilized and ResNet-18 backbone replaces the backbone of I3D and HRNet.
- **SACRF** [6] The backbone is replaced to ResNet-18, and its multiple modalities are substituted to single RGB input. Since NBA dataset does not have individual action labels, we remove the unary energy term.
- **DIN** [12] The experiment is conducted following its official implementation.
- **SAM** [11] The number of proposals ($N^p$) and the number of selected proposals ($K^p$) are set to 14 and 8, respectively.

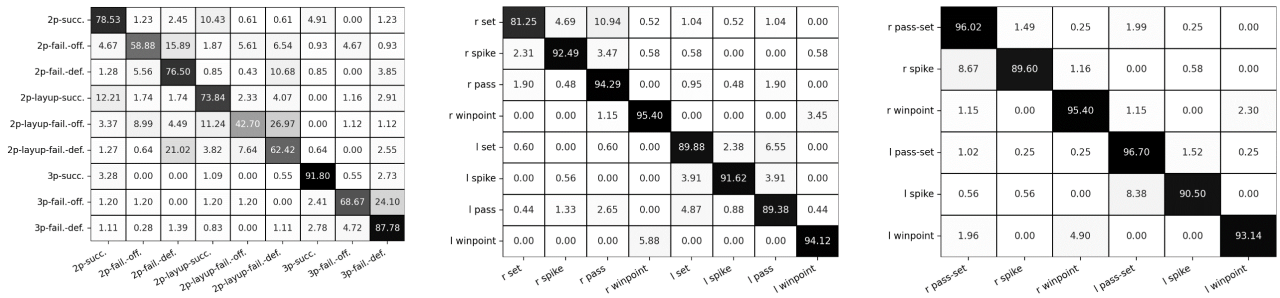
**Volleyball dataset.** Reproduction of the following models is also based on the code of DIN[1] [12]. Each reproduction first goes through backbone training process regarding the number of categories, then proceeds to train each inference module afterward. Note that MCA values of the following models under fully supervised setting are brought from DIN [12], hence we provide implementation detail of experiments under **A)** fully supervised setting with the aim of classifying actions into 6 (merged) labels, **B)** weakly supervised setting/8 labels, and **C)** weakly supervised setting/6 labels. In weakly supervised setting, actor bounding boxes are replaced to proposal boxes generated by Faster R-CNN [7] pretrained on COCO dataset [4] and individual action annotations are eliminated. In general, we use a batch size of 2 and the number of frames $T = 10$ for the following work. Unless mentioned, other hyperparameters are set based on the code provided by DIN. Followings are further implementation details of each model.

- **PCTDM** [10] ResNet-18 is applied instead of AlexNet, and RoIAlign features replace cropped/resized individual images of the original paper. Furthermore, weight decay rate of $1 \times 10^{-4}$ is applied to **C)**.
- **ARG** [9] Likewise, its backbone is changed to ResNet-18. Unlike its original setting, the backbone training is allowed in the model training process for a fair comparison with other models.
- **AT** [1] A single RGB branch is utilized and ResNet-18 backbone replaces the backbone of I3D and HRNet.
- **SACRF** [6] The backbone is replaced to ResNet-18, and its multiple modalities are substituted to single RGB input. Due to the removal of individual action labels, the unary energy term is removed.
- **DIN** [12] The experiment is conducted following its official implementation.

SAM [11] is reproduced following the method described in the original paper. The major difference with DIN-based reproductions is that it occupies a batch size of 8, a dropout rate of 0.1, and $T = 3$ frames.

- **SAM** [11] Note that SAM itself is a WSGAR work, so **A)** is disregarded. Since **C)** is already conducted in the original paper, we only reproduce **B)**. The number of proposals ($N^p$) and the number of selected proposals ($K^p$) are set to 16 and 12, respectively.

[1]Original DIN codes are available at https://github.com/JacobYuan7/DIN_GAR.




(a) (b) (c)

Figure 1. The confusion matrix (a) on the NBA dataset, (b) of the original 8 class classification on the Volleyball dataset, and (c) of the merged 6 class classification (merge *pass-set* class) on the Volleyball dataset.

## 1.2. Implementation of video backbones

We reproduce recent video backbones, ResNet-18 TSM [3] and VideoSwin-T [5] following the official codes. For a fair comparison, sampling strategy and training details are the same as ours.

## 1.3. Motion-augmented backbone

We use ResNet-18 [2] backbone in our experiment. We provide details of the backbone architectures to understand to which place the motion feature modules are inserted. Table 1 shows the ResNet-18 backbone architectures. For NBA dataset, we insert two motion feature modules after 4th and 5th residual block. For Volleyball dataset, we insert one motion feature module after the last residual block.

# 2. More ablation studies

In this section, we provide additional ablation on NBA dataset. Note that we do not adopt motion feature computation module in this additional ablations and use plain ResNet-18 backbone as a feature extractor.

**Effects of the temporal convolution layers.** Table 2 summarizes the performance according to different numbers and kernel sizes of temporal convolution layers. Note that we do not utilize zero-padding in this experiment. In most cases, MCA and MPCA increase as 1D convolutional layers are stacked. The performance is also affected by the kernel size of 1D convolutional layers, and it increases as the receptive field of temporal convolution layers gets wider.

# 3. More experimental results

In this section, we provide more visualizations and qualitative results that are omitted in the main paper due to the space limit.

Fig. 1 shows the confusion matrix on NBA and Volleyball datasets. For the NBA dataset (Fig. 1a), the most confusing cases are *2p-layup-fail.-off.* versus *2p-layup-fail.-def.* which only differs who rebound a ball after shooting a layup. For the 8 class classification (Fig. 1b), the most confusing cases are *r set* versus *r pass* and *l set* versus *l pass*. This is challenging because our model does not utilize individual action label in training which gives clues to classify *set* and *pass* class. For the merged 6 class classification (Fig. 1c) which merges *pass* and *set* class into *pass-set* class, our model achieves satiable accuracies on *right pass-set* and *left pass-set* class. Nevertheless, our model struggles to classify *spike* and *pass-set* due to a class imbalance problem.

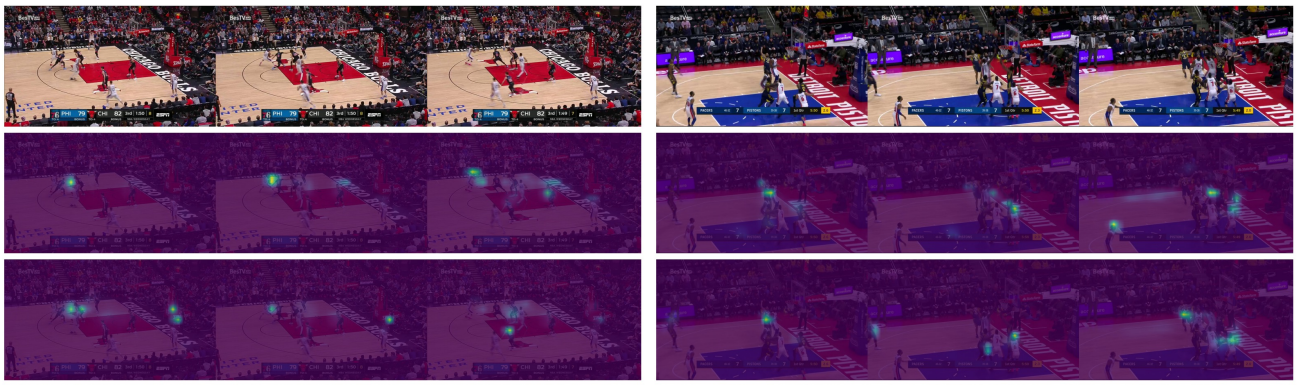
Fig. 2 and 3 show more visualizations on NBA and Volleyball dataset, respectively.

| Layers | ResNet-18 | Feature map size |
|---|---|---|
| $conv_1$ | $7 \times 7$, 64, stride (2, 2) | $T \times 360 \times 640$ |
| $pool_1$ | $3 \times 3$, stride (2, 2) | $T \times 180 \times 320$ |
| $res_2$ | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$ | $T \times 180 \times 320$ |
| $res_3$ | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$ | $T \times 90 \times 160$ |
| $res_4$ | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$ | $T \times 45 \times 80$ |
| $res_5$ | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$ | $T \times 23 \times 40$ |

Table 1. ResNet-18 backbone details. $[k \times k, c] \times n$ denotes $n$ convolutional layers with kernel size of $k$ and $c$ channels.

| Model | # params | MCA | MPCA |
|---|---|---|---|
| $[3 \times 1, 256] \times 3$ | 16.91M | 72.7 | 68.0 |
| $[5 \times 1, 256] \times 1$ | 16.65M | 70.2 | 65.5 |
| $[5 \times 1, 256] \times 2$ | 16.98M | 71.1 | 65.2 |
| Ours ($[5 \times 1, 256] \times 3$) | 17.31M | **73.6** | **69.0** |

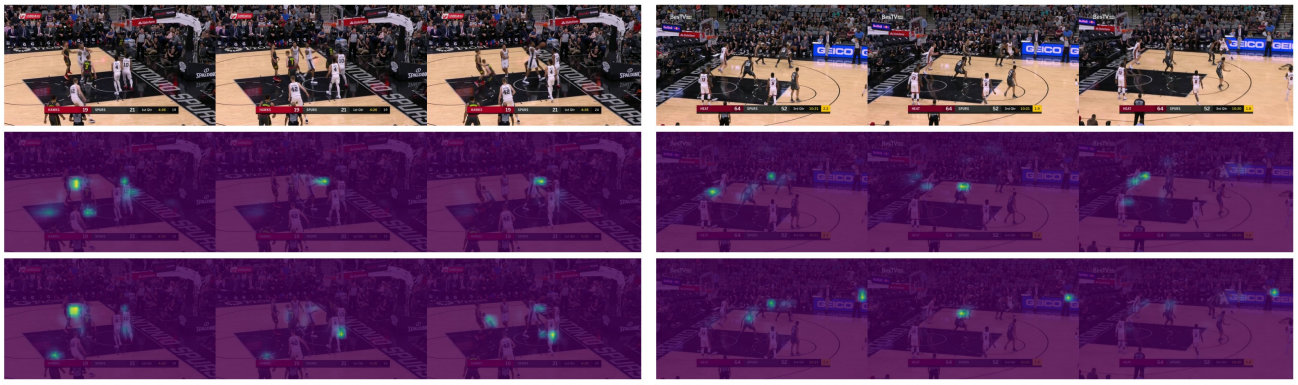
Table 2. Ablation on the different forms of temporal convolution layers. $[t \times 1, D] \times n$ denotes $n$ 1D convolutional layers with kernel size of $t$ and $D$ channels.



(a) 2p-fail-def



(b) 2p-fail-off



(c) 2p-success



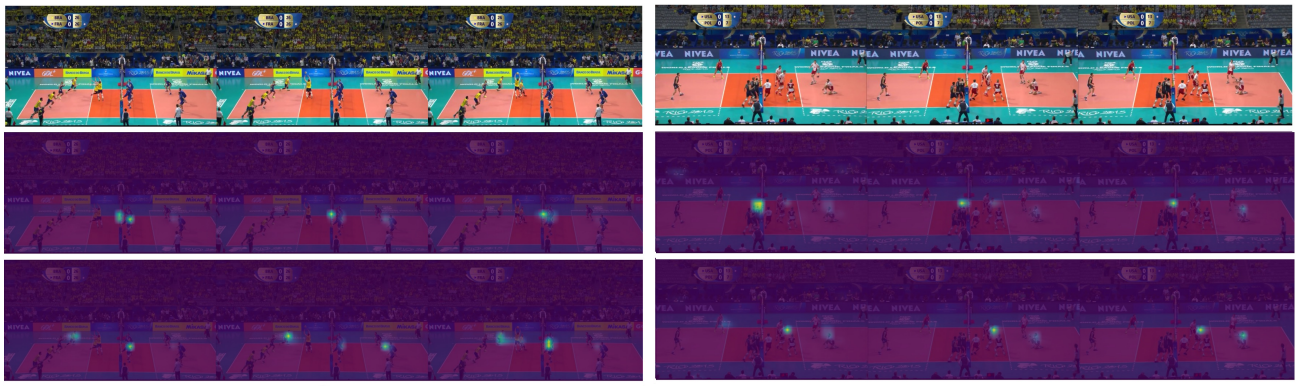
(d) 3p-success

(e) 3p-fail-def



(f) 3p-fail-off

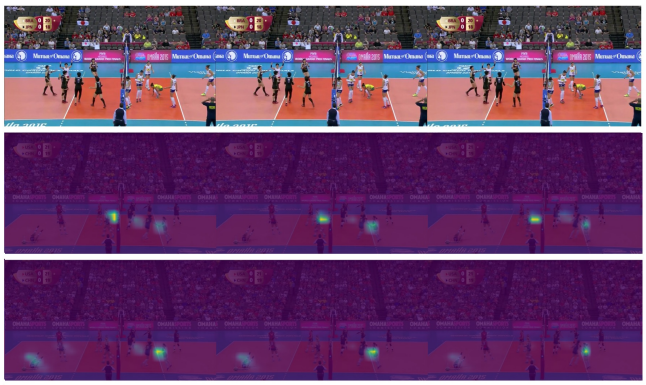
Figure 2. Visualizations of the cross-attention maps on NBA dataset. Attention maps for 2 tokens among 12 tokens are displayed. Tokens tend to capture different part of each group activity: in this example, the first token focuses more on activities happening among players and the ball, while the second token weighs more on peripheral clues besides the main scene.



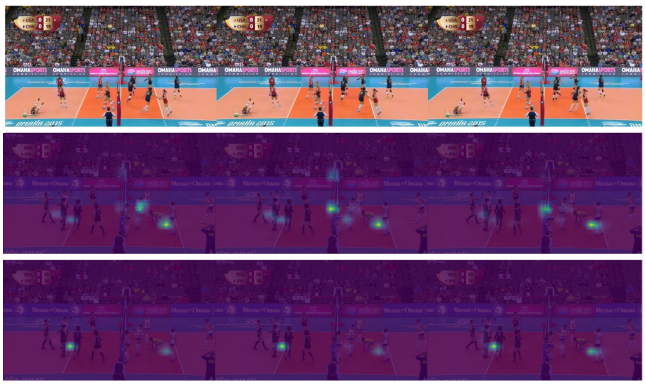
(a) l-pass-set



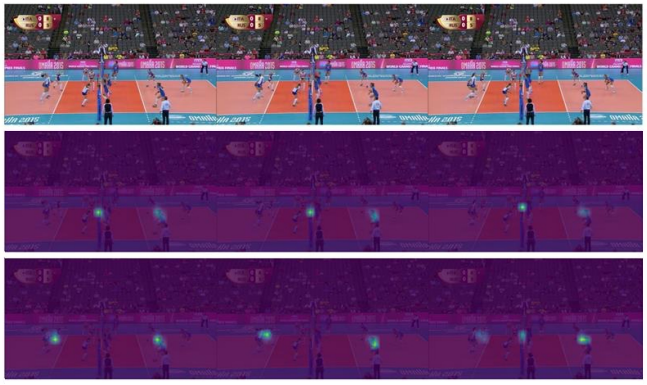
(b) r-pass-set



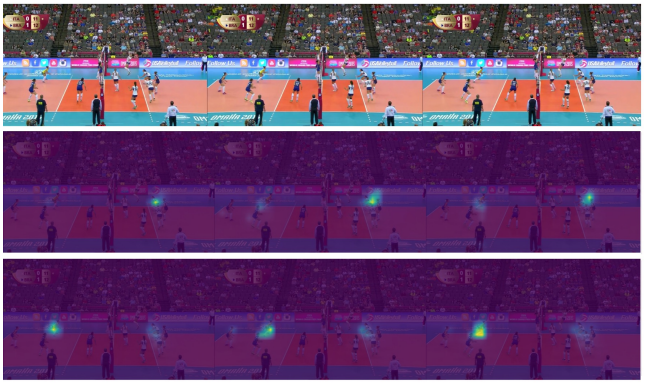
(c) l-winpoint



(d) r-winpoint



(e) l-spike



(f) r-spike

Figure 3. Visualizations of the cross-attention maps on Volleyball dataset. Attention maps for 2 tokens among 12 tokens are displayed. Likewise, tokens understand given group activities in a partial way. The first token watches more on activities happening around the net, while the second token shows more tendency toward capturing activities that involve players and happen farther from the net.

## References


[1] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees GM Snoek. Actor-transformers for group activity recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 839–848, 2020. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[3] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 7083–7093, 2019. 2

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 1

[5] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021. 2

[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 *Proc. European Conference on Computer Vision (ECCV)*, pages 71–90. Springer, 2020. 1

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015. 1

[8] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proc. European Conference on Computer Vision (ECCV)*, pages 20–36. Springer, 2016. 1

[9] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu. Learning actor relation graphs for group activity recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9964–9974, 2019. 1

[10] Rui Yan, Jinhui Tang, Xiangbo Shu, Zechao Li, and Qi Tian. Participation-contributed temporal dynamic model for group activity recognition. In *Proc. ACM Multimedia Conference (ACMMM)*, pages 1292–1300, 2018. 1

[11] Rui Yan, Lingxi Xie, Jinhui Tang, Xiangbo Shu, and Qi Tian. Social adaptive module for weakly-supervised group activity recognition. In *Proc. European Conference on Computer Vision (ECCV)*, pages 208–224. Springer, 2020. 1

[12] Hangjie Yuan, Dong Ni, and Mang Wang. Spatio-temporal dynamic inference network for group activity recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 7476–7485, 2021. 1