Group-DINOmics: Incorporating People Dynamics into DINO for Self-supervised Group Activity Feature Learning (Supplementary Material)

Ryuki Tezuka Chihiro Nakatani Norimichi Ukita
Toyota Technological Institute
{sd24437,sd23501,ukita}@toyota-ti.ac.jp

6. Implementation Details

This section presents implementation details not covered in the main paper.

6.1. Pretext tasks

Figure 11 shows the details of our pretext tasks (i.e., person flow estimation and group-relevant object localization). In person flow estimation, the optical flow image is obtained from two adjacent frames in a video (e.g., τ and $\tau + 1$ in Fig. 11) with a pre-trained optical flow estimator (i.e., RAFT [44] in our implementation). The pseudo labels of flow values for each person are obtained from the optical flow image by extracting the flow values of the center of the person's bounding box. For group-relevant object localization, annotated and detected ball coordinates are used to compute $\mathcal{L}_{\mathrm{O}}$ for the Volleyball dataset (VBD) and NBA dataset (NBA), respectively. In VBD and NBA, the xy values of the center point of the bounding box are calculated from the bounding box coordinates as the pseudo labels (i.e., $\hat{\boldsymbol{O}}$) for $\mathcal{L}_{\mathrm{O}}$.

6.2. Experimental Conditions

We trained our models on a single NVIDIA RTX A6000 GPU with a batch size of 8. For VBD, we use video clips consisting of 10 frames (i.e., $T = 10$) as with GAFL [28]. For NBA, we use 12 frames (i.e., $T = 12$). On each dataset, the same number of frames is used for all compared methods to ensure a fair comparison. For training the $\mathcal{L}_{\mathrm{F}}$, we use $N = 12$ for VBD and $N = 10$ for NBA, since 12 players are observed in most images in VBD and 10 in NBA. For inpainting, given the xy coordinates of a ball, we define the mask as a circle centered at the corresponding location, where the radius is manually fixed for all experiments. In our experiments with YOLOX [12] to detect person bounding boxes, we use the publicly released Deep-EIoU [65] weights. The code is available at https://github.com/hsiangwei0903/Deep-EIoU. For the comparison of group activity retrieval, we use HRN [16] and GAFL [28]. These codes are available at https://github.com/chihina/GAFL-CVPR2024.

Table 6. Comparison with state-of-the-art self-supervised GAF learning methods trained with supervision of person action classes on VBD. All other results are from [28].

	VBD		
Method	Hit@1	Hit@2	Hit@3
HiGCIN [57]	50.0	66.3	74.5
DIN [58]	57.0	73.1	81.1
Dual-AI [13]	64.4	76.5	82.0
GAFL [28]	**84.8**	89.6	91.8
Ours	82.7	**90.0**	**93.0**

7. Additional Experiments

This section presents additional experiments that could not be included in the main paper.

7.1. Evaluation in Group Activity Retrieval

7.1.1. Comparative experiments

While our work proposes two pretext tasks (i.e., person flow estimation and group-relevant object location estimation), another pretext task estimates person action classes, as proposed in [28]. All of these three pretext tasks can be achieved with pretrained estimators (i.e., flow estimators for person flow estimation, object detectors for group-relevant object location estimation, and action recognizers for person action recognition). Therefore, to demonstrate the effectiveness of our two pretext tasks compared with the one with person action classification, our method is compared with four networks that are trained with person action classification.

Table 6 shows the results on VBD. While all methods except "Ours" train $\boldsymbol{G}$ with person-action supervision from manual annotations, "Ours" utilizes pseudo labels estimated by person flow and group-relevant object location estima-

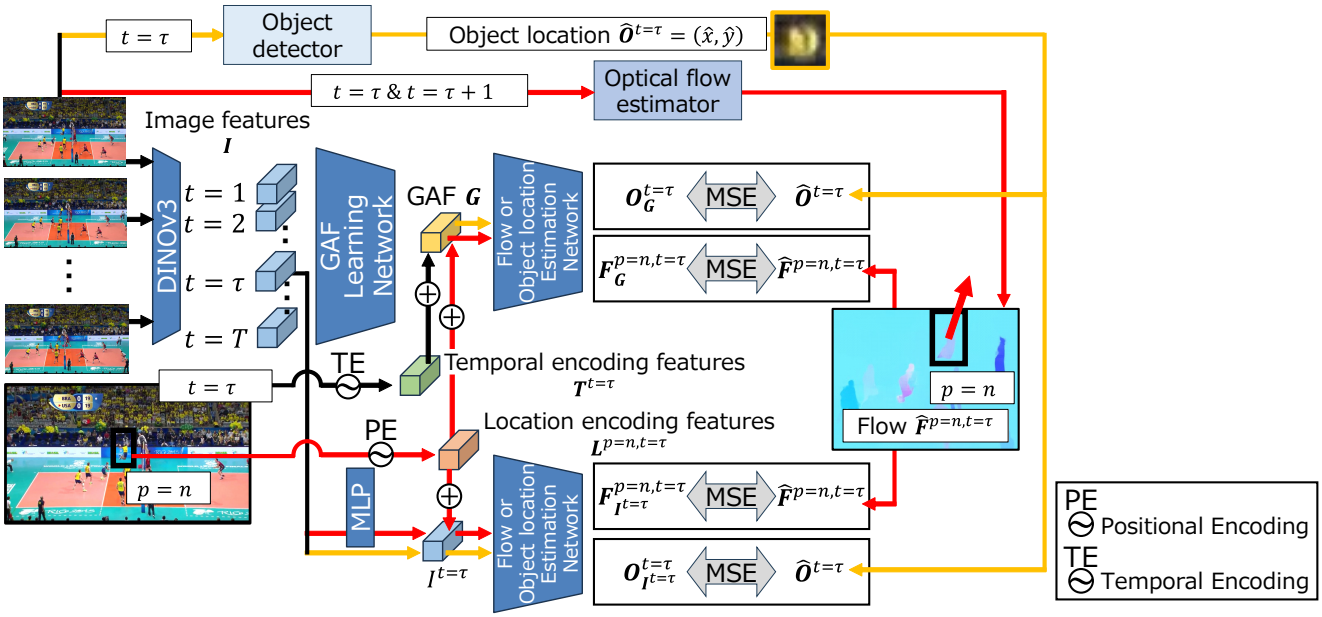


Figure 11. Detail of our pretext tasks. Our pretext tasks comprise person-flow estimation and group-relevant object location estimation. Person flow estimation utilizes a pre-trained optical flow estimator. Group-relevant object location estimation utilizes a pre-trained object detector. Since both pretext tasks use only these pre-trained models, training can be conducted in a self-supervised manner with pseudo labels. The black paths are shared by both pretext tasks, while the **red** paths are used only for person flow estimation, and the **yellow** paths are used only for group-relevant object location estimation.

Table 7. Detailed analysis of inpainting on VBD and NBA. Unlike Table 3 of the main paper, results when only $\mathcal{L}_{\mathrm{O}}$ is used for training are shown for further analysis.

Input image	VBD	NBA
	Hit@1	Hit@1
Original	72.9	30.5
Masking	45.0	18.5
Inpainting (Ours)	**74.0**	**40.3**

Table 8. Effects of inpainting during inference in our GAF learning on VBD and NBA. ✓ indicates that the inpainting is applied during inference.

Inpainting during inference	VBD	NBA
	Hit@1	Hit@1
✓	81.5	43.4
- (Ours)	**82.7**	**43.9**

tion methods. The results show that our method is better than the other three methods (i.e., HiGCIN [57], DIN [58], and Dual-AI [13]) and achieves comparable performance to GAFL. Even though these four previous methods are trained with the ground truth labels of person actions, our method achieves good performance only with estimated person flows and object locations, thereby eliminating the need for the manual annotations of action class labels.

7.1.2. Ablation studies on Inpainting

Effectiveness of Inpainting with only $\mathcal{L}_{\mathrm{O}}$. Table 7 shows the effectiveness of group-relevant object inpainting during training. Different from the results obtained with $\mathcal{L}_{\mathrm{F}}$ and $\mathcal{L}_{\mathrm{O}}$ shown in Table 3 of the main paper, Table 7 shows results with only $\mathcal{L}_{\mathrm{O}}$ on VBD and NBA for further analysis because this inpainting may affect only $\mathcal{L}_{\mathrm{O}}$. As with Table 3 of the main paper, in "Original," original whole images are directly fed into the network to extract a GAF without inpainting of a ball. In "Masking," the region of a ball in the whole image is filled with black pixels to remove the appearance of the ball. In "Inpainting (Ours)," the region of a ball is masked via inpainting.

Compared with "Original," our method (i.e., "Inpainting (Ours)") is 1.1% (1.1%=74.0%-72.9%) better on VBD and 9.8% (9.8%=40.3%-30.5%) better on NBA. Additionally, we observe a significant performance improvement of our method compared to "Masking" on VBD and NBA. This is because the network can easily localize the ball from the appearance cues of the masked region in "Masking," and this localization cannot contribute to our GAF learning. These results clearly verify that ball inpainting is effective for our GAF learning with $\mathcal{L}_{\mathrm{O}}$.

Table 9. Effects of the location-guidance in our GAF learning on VBD and NBA. Results obtained by the two settings (i.e., $\mathcal{L}_{\mathrm{F}}$ is only used or both $\mathcal{L}_{\mathrm{F}}$ and $\mathcal{L}_{\mathrm{O}}$ are used) are separated by the line.

	Loss		VBD	NBA
Method	$\mathcal{L}_{\mathrm{F}}$	$\mathcal{L}_{\mathrm{O}}$	Hit@1	Hit@1
Ours w/o $\boldsymbol{L}^{p,t}$	✓	-	63.4	29.2
Ours	✓	-	**75.4**	**34.1**
Ours w/o $\boldsymbol{L}^{p,t}$	✓	✓	79.0	37.7
Ours	✓	✓	**82.7**	**43.9**

Table 10. Additionl ablation studies on VBD and NBA.

	$\mathcal{L}_{\mathrm{F}}$		$\mathcal{L}_{\mathrm{O}}$		VBD	NBA
	$\mathcal{L}_{\mathrm{F},\boldsymbol{G}}$	$\mathcal{L}_{\mathrm{F},\boldsymbol{I}}$	$\mathcal{L}_{\mathrm{O},\boldsymbol{G}}$	$\mathcal{L}_{\mathrm{O},\boldsymbol{I}}$	Hit@1	Hit@1
(a)	-		-		43.0	22.6
(j)	-	✓	-		71.3	29.5
(k)	-		-	✓	72.5	37.0

Effectiveness of Inpainting during Inference. Table 8 shows the results when we also apply group-relevant object inpainting at inference time. In our method, we do not apply inpainting during inference due to the additional computational cost incurred by the object detector and the inpainting model. However, since our method applies inpainting during training, it may introduce a domain gap when it is not applied during inference.

Despite eliminating this domain gap, the results show that inpainting at inference degrades retrieval performance. This suggests that, in inference, the potential domain gap is less harmful than the errors introduced by imperfect ball localization and inpainting artifacts.

7.1.3. Ablation studies on location-guidance

Table 9 shows the ablation results on the location-guidance (i.e., $\boldsymbol{L}^{p,t}$ used in Eq. 1, 2) in our method. Under both training settings, namely using only $\mathcal{L}_{\mathrm{F}}$ and jointly using $\mathcal{L}_{\mathrm{F}}$ and $\mathcal{L}_{\mathrm{O}}$, "Ours" outperforms "Ours w/o $\boldsymbol{L}^{p,t}$" on VBD and NBA. The significant performance gain suggests that "Ours w/o $\boldsymbol{L}^{p,t}$" only captures coarse motion over the whole image but fails to focus on fine-grained motions of each person in an image. We can interpret that location-guidance in our method explicitly encourages $\boldsymbol{G}$ to learn the fine-grained motions of each person.

7.1.4. Effectiveness of Auxiliary Losses and Transformer

The Transformer encoder (shown in (b) of Fig. 2 of the main paper) is trained from scratch but is not optimized with the auxiliary losses. It is therefore impossible to train our network solely with them. Instead, this Transformer is replaced by max pooling in (j) and (k) of Table 10. Even the auxiliary losses alone contribute to performance improvements, while the Transformer further improves performance, as shown in (b) and (c) of Table 2 of the main paper.

Table 11. Comparison of GAFL with different image feature extractors. We replace VGG19, the original image feature extractor in GAFL, with DINOv3

Method	Backbone	VBD	NBA
		Hit@1	Hit@1
GAFL [28]	VGG19	61.1	24.7
	DINOv3	38.1	25.3
Ours	DINOv3	**82.7**	**43.9**

Table 12. Effects of architecture size for GAF learning on VBD and NBA. Results obtained by GAFL [28] and our method are separated by a double line. In our method, results obtained by different ViTs in DINOv3 are shown for further discussion.

Method	All Params	Learnable Params	Time[ms]		Hit@1	
			VBD	NBA	VBD	NBA
GAFL [28]	79M	68M	82.4	86.3	61.1	24.7
Ours (ViT-B)	99M	21M	86.6	102.9	73.8	38.8
Ours (ViT-L)	326M	48M	272.3	327.7	**82.7**	**43.9**

7.1.5. Effect of Architectures and Training Strategies

Effect of Backbone Choice. Table 11 shows the results when replacing VGG19, the original image feature extractor in GAFL [28], with DINOv3. As shown in the table, GAFL performs worse with DINOv3. This suggests that merely replacing the feature extractor with DINOv3 is not sufficient to fully exploit its potential, as also observed in [67].

Comparison of DINOv3 architecture. Table 12 shows the architecture sizes of DINOv3, the results when replacing the feature extractor with ViT-B and ViT-L, and the inference-time comparison with GAFL [28]. For ViT-B, we train the last block of its ViT. While ViT-L yields stronger results, we confirm that ViT-B is still sufficient to achieve state-of-the-art performance. Moreover, our design effectively exploits the capability of large-scale pretrained feature extractors, achieving high performance while keeping the number of learnable parameters smaller than GAFL. In addition, Ours (ViT-B) outperforms GAFL [28] with maintaining a similar runtime.

Comparison of training strategies. Table 13 shows the results with various training strategies of DINOv3 in our method. In this table, the results obtained by our method are shown in (i). In (a), (b), (c), and (i), we compared the retrieval performance with different numbers of unfrozen blocks within 24 transformer blocks of DINOv3 (ViT-L). Among these settings, (i) achieves the best performance. These results indicate that fine-tuning DINOv3 by updating only the last two transformer blocks is effective for keeping the pre-trained knowledge.

Table 13. Comparison of training strategies for DINOv3 in our method. Results obtained by our method are shown in (i). In (a), (b), (c), and (i), we fine-tuned different numbers of transformer blocks in the ViT of DINOv3. In (d), DINOv3 is updated by LoRA [64]. In (e), (f), (g), and (h), an additional linear layer or transformer blocks are attached to the frozen DINOv3 for fine-tuning.

	Training setting				VBD	NBA
	Unfrozen Blocks	LoRA	Adapter	Learnable Params	Hit@1	Hit@1
(a)	-	-	-	23M	64.6	30.0
(b)	24	-	-	36M	78.2	40.3
(c)	22, 23, 24	-	-	61M	80.9	36.5
(d)	-	✓	-	24M	76.6	42.3
(e)	-	-	1 linear	24M	65.7	34.4
(f)	-	-	1 trans. block	36M	68.3	38.2
(g)	-	-	2 trans. blocks	48M	70.2	41.0
(h)	-	-	3 trans. blocks	61M	70.3	41.4
(i)	23, 24	-	-	48M	**82.7**	**43.9**

Table 14. Comparison of input features on VBD and NBA. While "Cropped" utilizes cropped-image features as input, our method utilizes whole-image features for GAF learning.

Input features	VBD	NBA
	Hit@1	Hit@1
Cropped	58.8	28.8
Whole (Ours)	**74.0**	**40.3**

We also examine the fine-tuning of DINOv3 with Low Rank Adaptation (LoRA) [64] in (d). We update the query and value projections with rank 16 by LoRA in (d). Compared with (d), (i) is still better than (d). The results demonstrate that fine-tuning only the last two transformer blocks is better than updating a small adapter in (d).

Finally, we conduct fine-tuning with additional adapters while keeping DINOv3 frozen in (e), (f), (g), and (h). As in [66, 67], "1 linear" feeds the CLS token into one linear layer or "n trans. block(s)" passes the CLS token and patch tokens into additional n transformer blocks. Compared with (i), (e) is worse than (i). Furthermore, the performance in (e) is also worse than (d), in which LoRA updates the same number of parameters. In (f), (g), and (h), we added additional transformer blocks on the frozen backbone. The performance is better than (e) thanks to the large number of learnable parameters. However, (i) is still the best.

Based on the discussion, our training strategy (i.e., fine-tuning the last two transformer blocks of ViT in DINOv3) is better than the other strategies. These results may come from the domain gap between the pre-training of DINO and downstream tasks. In [66, 67], DINO is fine-tuned for segmentation and gaze target detection in general scenes. Since such general scenes are used for pre-training of DINOv3, fine-tuning DINOv3 with an additional adapter is enough while freezing DINOv3. Unlike [66, 67], we utilize sports images that are largely underrepresented in the pre-training of DINOv3. This domain gap suggests that directly fine-tuning DINOv3, as in [20], is more effective than the other strategies.

Table 15. Effects of whole image flow estimation on VBD and NBA. While the optical flow image obtained from a whole image is estimated as a pretext task in "Whole image flow," our method proposes to estimate flow values of each person as a pretext task for learning people-relevant features in $\boldsymbol{G}$.

	Loss		VBD	NBA
Method	$\mathcal{L}_{\mathrm{F}}$	$\mathcal{L}_{\mathrm{O}}$	Hit@1	Hit@1
Whole image flow	✓	-	51.9	23.8
Person flow (Ours)	✓	-	**75.4**	**34.1**
Whole image flow	✓	✓	76.7	38.2
Person flow (Ours)	✓	✓	**82.7**	**43.9**

Table 16. Effects of errors in bounding boxes detected by YOLOX [12] on VBD. "GT" indicates that the experiments utilize the ground-truth person bounding boxes. "YOLOX" indicates that the experiments utilize detected tracklets by Deep-EIoU [65].

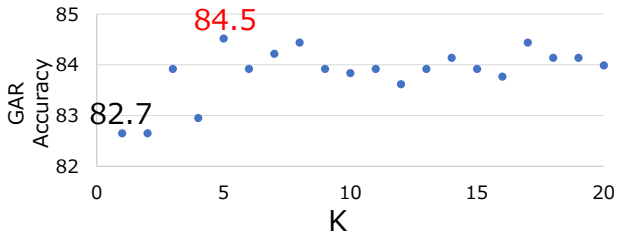
	VBD
Method	Hit@1
GAFL [28] (YOLOX)	51.5
GAFL [28] (GT)	61.1
Ours (YOLOX)	81.2
Ours (GT)	82.7

7.1.6. Detailed analysis

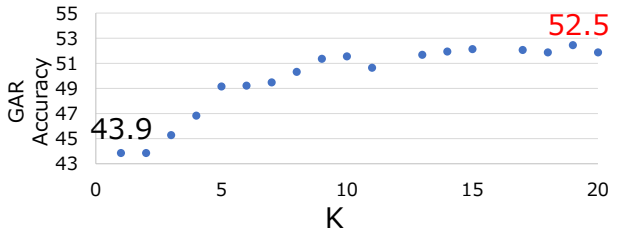
Whole-image features vs. cropped-image features. Unlike previous self-supervised GAF learning methods [16, 28], which take person bounding boxes cropped from an image as input, our method feeds the whole image into DINOv3 to capture global context learned in $\mathcal{L}_{\mathrm{O}}$. To verify the effectiveness of global context learning by $\mathcal{L}_{\mathrm{O}}$, we conduct an experiment where only $\mathcal{L}_{\mathrm{O}}$ is used for GAF learning with cropped person bounding boxes, as shown in Table 14. RoIAlign [63] used in GAFL [28] is not applied to DINOv3 due to the low-resolution patch tokens. Therefore, we average the patch tokens within each person's bounding box to obtain cropped image features as with [62, 68].

In Table 14, our whole-image variant is better than "Cropped" on both VBD and NBA. These results demonstrate that removing the appearance cues of the group-relevant objects by cropping person bounding boxes discards the global context.

Effects of whole image flow estimation. In our method, flow estimation is applied for each person in an image to capture local dynamics. As a variant, our method can easily extend to estimate the optical flow of a whole image. To



(a) Volleyball dataset (VBD)



(b) NBA dataset (NBA)

Figure 12. GAR accuracy curve by the KNN classification on VBD and NBA. K changes from 1 to 20 in our experiment.

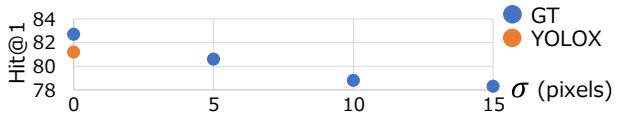


Figure 13. Performance under different noise levels.

estimate optical flow over the whole image, the patch tokens obtained from DINOv3 are processed through a two-layer 1×1 convolutional network.

Table 15 shows the comparison between estimating the optical flow over a whole image and person flow. Our method outperforms "Whole image flow" in all metrics. These results validate that the optical flow estimation over the whole image is not suitable for GAF learning. This is because the flow over the whole image is dominated by background motion, whereas person flows are more important for representing group activities.

Effects of errors in person detection. To reveal the effects of errors in person detection, we utilize the detected person bounding boxes on VBD, instead of their GTs, as shown in Table 16. The results show that our method achieves consistently high performance even with detected person bounding boxes by YOLOX [12]. In "GAFL," the retrieval performance is significantly dropped by the errors in the detected person bounding boxes. From these results, we conclude that our method is more robust to detection errors in person bounding boxes compared with GAFL [28].

Table 17. Additional pretext goal-detection task.

$\mathcal{L}_{\mathrm{F}}$	$\mathcal{L}_{\mathrm{O}}$		NBA
Person Flow	Ball	Goal	Hit@1
-	✓	-	40.3
-	✓	✓	**40.8**
✓	✓	-	43.9
✓	✓	✓	**45.5**

Table 18. Comparison with supervised group activity recognition methods on VBD and NBA. The results of all methods are copied from their papers. The four methods above the middle line take the whole image as input, while the four methods between the middle line and ours use person bounding boxes together with the whole image in both training and inference. For a fair comparison, in all methods, the only manually annotated labels are group activity classes, and only images are used in inference.

Method	Extractor	VBD	NBA
		Merged MCA	MPCA
DFWSGAR [21]	ResNet-18	94.4	71.2
SOGAR [6]	ResNet-18	95.9	73.5
Flaming-Net [29]	Inception-v3	95.2	**76.0**
LiGAR [4]	ResNet-18	76.1	57.1
SAM [56]	ResNet-18	93.1	51.5
Dual-AI [13]	Inception-v3	95.8	50.2
KRGFormer [32]	Inception-v3	95.0	67.1
MP-GCN [26]	YOLOV8x	**96.1**	74.6
Ours	DINOv3	**96.1**	68.2

Robustness to noisy pseudo-labels. Figure 13 shows Hit@1 under different zero-mean Gaussian noise levels added to GT person bounding boxes and GT ball locations. Similar Hit@1 scores under $\sigma = 10$ and $\sigma = 15$ validate the robustness of our method to erroneous bounding boxes. We will also show a sensitivity analysis with respect to other pseudo-labels.

More localization tasks. While our method currently uses only a ball as an activity-related object, we also incorporate a basketball goal system on NBA (Table 17). $\mathcal{L}_{\mathrm{O}}$ with the goal system further improves performance. We also plan to include experiments using a net system on VBD, court lines, and referees in $\mathcal{L}_{\mathrm{O}}$.

7.2. Evaluation in Group Activity Recognition

7.2.1. KNN for Group Activity Recognition

In addition to Group Activity Recognition (GAR) via the 1-nearest neighbor retrieval results shown in Fig. 7 of the main paper, Fig. 12 shows GAR accuracy for different numbers of neighbors K in KNN classification. Following [28], we vary K from 1 to 20.

For VBD, $K > 3$ achieves higher GAR accuracy than $K = 1$, and the best result with $K = 5$ is 1.8% better than $K = 1$. As with the results above, for NBA, $K >$

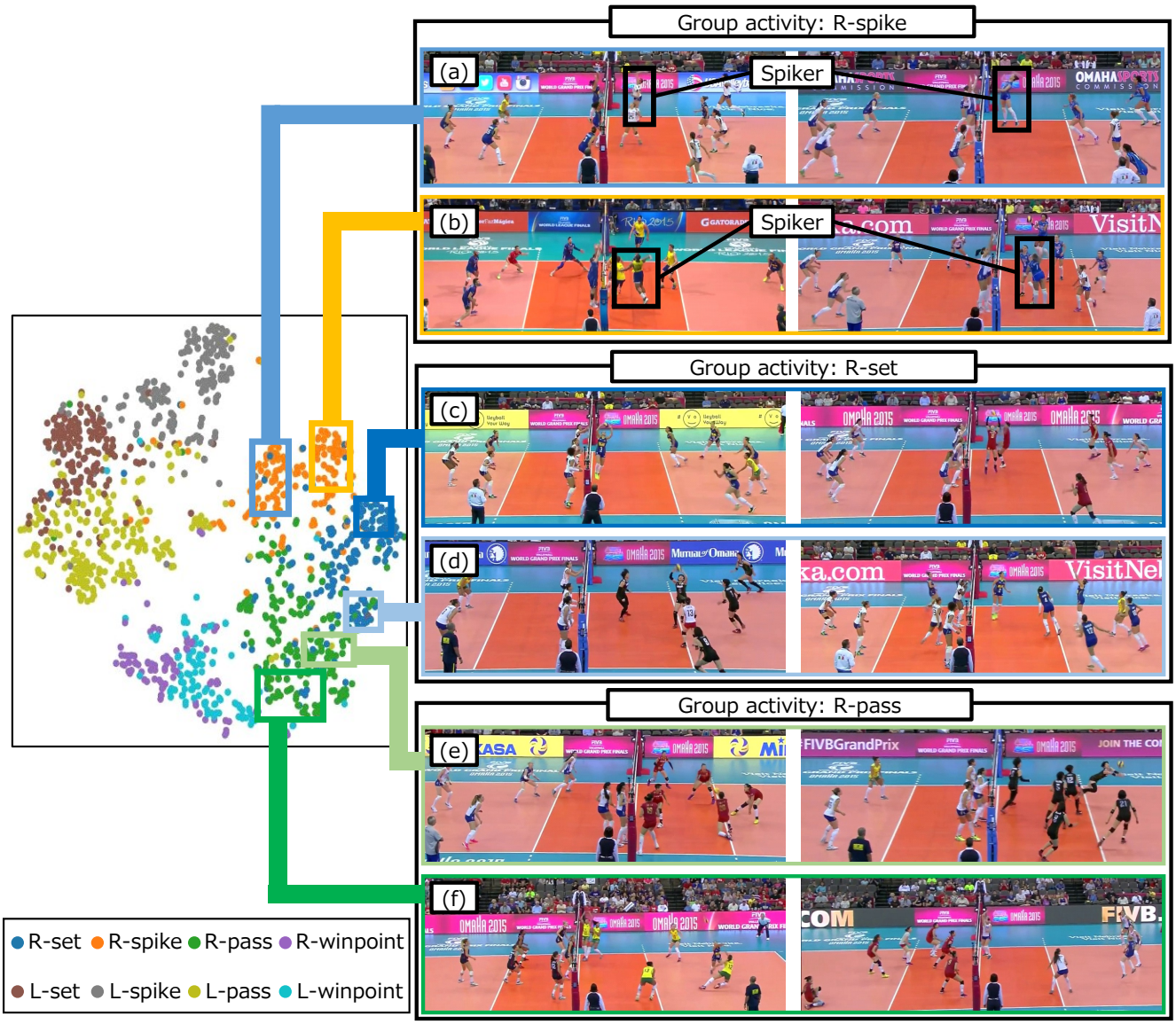


Figure 14. Visualization of the learned GAF space on VBD. Unlike Fig. 10 of the main paper, we also show actual images corresponding to the GAFs. The orange data points (i.e., "R-spike") are divided into (a) and (b) based on the spiker's position. The blue data points (i.e., "R-set") are divided into (c) and (d) based on the setter's position as similar to (a) and (b). In (e) and (f), the green data points (i.e., "R-pass") are split into two clusters that differ in terms of the receiver's position.

3 outperforms $K = 1$, and the best result with $K = 19$ improves accuracy by 8.6%. These results demonstrate that using KNN for GAR is a simple yet effective approach, and that using a larger K is beneficial when the 1-NN accuracy is relatively low, as in NBA with 43.9%, compared with the already high accuracy on VBD with 82.7%.

7.2.2. Fine-tuning for Supervised Recognition

While the results of the supervised recognition on terms of MCA on both VBD and NBA are shown in Table 5 of the main paper, we further show the results of Merged MCA on VBD and MPCA on NBA in Table 18. Merged MCA [56] is an evaluation metric in which the pass and set classes are merged into one class. Although our method achieves

Table 19. Effect of temporal resolutions on NBA. MCA/MPCA are shown. For fair comparison under equivalent conditions, we select [21, 26] whose authors provide publicly available code.

Method	3 frames	6 frames	12 frames	18 frames
DFWSGAR [21]	65.3/60.2	71.8/65.4	**74.4/69.4**	73.9/68.6
MP-GCN [26]	49.4/45.8	63.7/59.3	70.8/66.6	**74.4/70.8**
Ours	**67.4/61.4**	**73.1/67.7**	73.0/68.2	73.2/68.4

state-of-the-art performance on VBD, it is not the best on NBA. One possible reason for the low performance on the NBA is the dataset's characteristics. Compared to VBD, each sequence in NBA is longer, and a group activity label (e.g., 2p-succ, 3p-succ) often spans a long temporal interval. In our method, we process the sequence using a temporal Transformer and then apply max pooling over time to obtain $\boldsymbol{G}$. As a result, features of all frames are compressed into $\boldsymbol{G}$, which prevents us from discriminating between similar group activities. For example, 2p-succ and 3p-succ are similar because of players' behaviors after a successful shot. This may partly explain why our generic architecture is less effective than NBA-specific designs such as SOGAR [6], Flaming-Net [29], and MP-GCN [26], which explicitly incorporate richer temporal modeling specialized for NBA. These results suggest that our GAFs can be improved by more flexible temporal aggregation modules as used in these methods. We leave exploring such NBA-specific temporal modules for future work, while preserving the simplicity of our generic GAF learning network.

This interpretation is further supported by Table 19, which shows results for input videos with different frame counts. While keeping the elapsed time per video unchanged, we vary the number of frames by changing the temporal resolution. Our method shows an advantage in low-frame settings, where the negative impact of pooling is limited because fewer frames are pooled.

7.3. Visualizations

In addition to the visualization of GAFs shown in Fig. 10 of the main paper, we also visualize actual images corresponding to GAFs in Fig. 14.

First, we can see that contextually similar group activities are located in similar regions in Fig. 14. For example, R-set data points (i.e., (c) and (d)) and R-pass data points (i.e., (e) and (f)) are close in the learned GAF space. This is because both group activities include the player interacting with the ball for the future spike. Moreover, we can also see that R-set data points (i.e., (c) and (d)) and R-spike data points (i.e., (a) and (b)) are close in the learned GAF space. These results also verify that our method can learn contextual similarity between R-set and R-spike activities. Based on the discussion above, we can summarize that our GAF learning can capture class-level transitions (e.g., offense progression from Pass to Set to Spike).

Furthermore, Fig. 14 shows that our method can learn fine-grained similarities among various group activities beyond the class-level similarities. First, it can be seen that the orange data points (i.e., R-spike) split into two clusters (a) and (b). While the spiker hits the ball at the top of the image in (a), the spiker hits the ball at the bottom of the image. These differences in spiker position in (a) and (b) demonstrate that our method learns fine-grained information of players who are important for representing group activity (e.g., spikers in these examples).

In (c) and (d), blue data points (i.e., R-set) are split into two clusters. While the players are tossing the ball near the net in (c), the players are tossing the ball far from the net in (d). These examples also validate that our GAF learning preserves the position information of players, as in (a) and (b). In (e) and (f), green data points (i.e., R-pass) are split into two clusters. In (e), the players are receiving the ball near the center of the court. In (f), we can observe that the players are receiving the ball at a distance from the net.

Interestingly, we can also see that (b) and (c) are close in the learned GAF space. The closeness comes from the position similarity of players interacting with the ball. As with these results, (d) and (e) are close in the learned GAF space. This is because both examples contain players interacting with the ball at a similar position. These results demonstrate that our method can learn fine-grained similarities among group activities beyond class-level similarities.

References

[62] Hyungyu Choi, Young Kyun Jang, and Chanho Eom. GOAL: global-local object alignment learning. In *CVPR*, 2025. 4

[63] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, 2017. 4

[64] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022. 4

[65] Hsiang-Wei Huang, Cheng-Yen Yang, Jiacheng Sun, Pyong Kun Kim, Kwang-Ju Kim, Kyoungoh Lee, Chung-I Huang, and Jenq-Neng Hwang. Iterative Scale-Up ExpansionIoU and Deep Features Association for Multi-Object Tracking in Sports. In *WACVW*, 2024. 1, 4

[66] Cijo Jose, Th´eo Moutakanni, Dahyun Kang, Federico Baldassarre, Timoth´ee Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Micha¨el Ramamonjisoa, Maxime Oquab, Oriane Sim´eoni, Huy V. Vo, Patrick Labatut, and Piotr Bojanowski. Dinov2 meets text: A unified framework for image- and pixel-level vision-language alignment. In *CVPR*, 2025. 4

[67] Fiona Ryan, Ajay Bati, Sangmin Lee, Daniel Bolya, Judy Hoffman, and James M. Rehg. Gaze-lle: Gaze target estimation via large-scale learned encoders. In *CVPR*, 2025. 4

[68] Michal Shlapentokh-Rothman, Ansel Blume, Yao Xiao, Yuqun Wu, Sethuraman TV, Heyi Tao, Jae Yong Lee, Wilfredo Torres, Yu-Xiong Wang, and Derek Hoiem. Region-based representations revisited. In *CVPR*, 2024. 4