Detector-Free Weakly Supervised Group Activity Recognition

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/

Abstract

Group activity recognition is the task of understanding the activity conducted by a group of people as a whole in a multi-person video. Existing models for this task are often impractical in that they demand ground-truth bounding box labels of actors even in testing or rely on off-the-shelf object detectors. Motivated by this, we propose a novel model for group activity recognition that depends neither on bounding box labels nor on object detector. Our model based on Transformer localizes and encodes partial contexts of a group activity by leveraging the attention mechanism, and represents a video clip as a set of partial context embeddings. The embedding vectors are then aggregated to form a single group representation that reflects the entire context of an activity while capturing temporal evolution of each partial context. Our method achieves outstanding performance on two benchmarks, Volleyball and NBA datasets, surpassing not only the state of the art trained with the same level of supervision, but also some of existing models relying on stronger supervision.

1. Introduction

Group activity recognition (GAR) is the task of classifying the activity that a group of people are doing as a whole in a given video clip. It has attracted increasing attention due to a variety of its applications including sports video analysis, video surveillance, and social scene understanding. Unlike the conventional action recognition that focuses on understanding individual actions [9, 16, 18, 26, 31, 35, 44, 46, 50, 52], GAR demands comprehensive and precise understanding of interactions between multiple actors, which introduces inherent challenges such as localization of actors and modeling their spatio-temporal relations.

Due to the difficulty of the task, most of existing methods for GAR [15,17,21,23,29,36,53,55,57] require ground-truth bounding boxes of individual actors for both training and testing, and their action class labels for training. In particular, the bounding box labels are used to extract features of individual actors (*e.g.*, *RoIPool* [39] and *RoIAlign* [19])
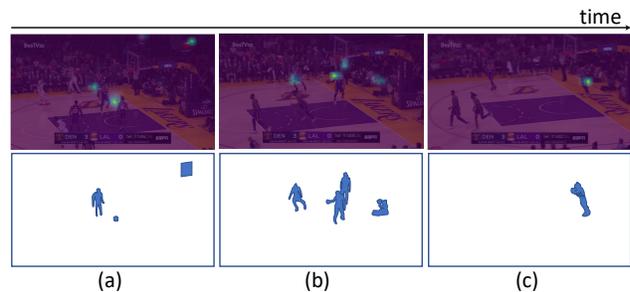

Figure 1. Visualization of partial contexts captured by a token across time. The token in this example focuses on how players behave after they conceded a goal. (a) Right after the event *3p-succ*, the timer has been reset to 24 secs and a defender stares at the ball. (b) Players prepare for the next attack, while a referee and a cameraman point at who takes the ball. (c) A player initiates the next attack. With such tokens each representing different pieces of the whole group activity, our model acquires the encapsulated semantics of the target activity.

and discover their spatio-temporal relations precisely; such actor features are aggregated while considering the relations between actors to form a group-level video representation, which is in turn fed to a group activity classifier. Though these methods have demonstrated impressive performance on the challenging task, their dependence on the heavy annotations, especially bounding boxes at inference, is impractical and in consequence restricts their applicability significantly.

One way to resolve this issue is to jointly learn group activity recognition and person detection using bounding box labels [6, 60] to estimate bounding boxes of actors at inference. This approach however still requires ground-truth bounding boxes of individual actors for training videos. To further reduce the annotation cost, Yan *et al.* [56] introduced weakly supervised GAR (WSGAR) that does not demand actor-level labels at both training and inference. They address the lack of bounding box labels by generating actor box proposals through a detector pretrained on an external dataset and learning to prune irrelevant proposals.

The detector-based WSGAR however has several drawbacks as follows. First of all, a detector often suffers

from occlusion and background clutter, and thus frequently causes missing and false detections that degrade GAR accuracy. Second, the detector-based approach loses contextual information that are useful for GAR since it concentrates only on people; in sports video analysis, for example, entities other than people, such as a ball and a scoreboard, may provide crucial information for the task. Third, object detection is costly to itself and imposes additional overheads in both computation and memory.

In this paper, we propose a detector-free model for WS-GAR that depends neither on ground-truth bounding boxes nor on object detector. It bypasses explicit object detection by drawing attention on entities involved in a group activity through a Transformer encoder [48] placed on top of a convolutional neural network (CNN) backbone. Specifically, we define learnable tokens as input to the encoder so that each of them learns to localize partial contexts of a group activity through the attention mechanism of the encoder; the tokens capture not only key actors but also other useful clues as shown in Fig. 1. Since a set of learnable tokens are shared for all frames, a predefined number of token embeddings are computed by the encoder for every frame. A video clip is then represented as a bag of token embeddings, which are aggregated into a group representation in two steps: Those computed from the same token at different frames are first aggregated to capture the temporal evolution of each token, then the results are fused to form a single feature vector for group activity classification.

In addition, for further performance improvement, the backbone of our model is designed to compute motion-augmented features. Unlike previous work on GAR [4, 17, 29, 36], it does not rely on off-the-shelf optical flow that is prohibitively expensive and thus has been a computational bottleneck. Instead, inspired by recent video representation architectures [16, 26, 27, 35, 49], it learns to capture motion information in feature levels by embedding local correlation between the feature maps of two adjacent frames.

We evaluate the proposed framework on two datasets, Volleyball [23] and NBA [56]. Our framework achieves the state-of-the-art performance on the two benchmarks in the weakly supervised learning setting, and is as competitive as existing methods relying on stronger supervision such as ground-truth bounding boxes and individual action class labels. The contribution of this paper is three-fold:

- We present the first detector-free method dedicated to WSGAR, which demands neither ground-truth bounding box labels nor object detector.

- We propose a novel Transformer-based model that captures key actors and objects involved in a group activity through the attention mechanism. Moreover, our model is carefully designed to capture their temporal dynamics to produce a rich group-level video feature.

- On the two benchmarks, the proposed method largely

outperforms existing WSGAR models. Also, it even beats early GAR models that depend on stronger supervision than ours.

2. Related work

2.1. Group activity recognition

On account of its various applications in the real world, group activity recognition (GAR) has been studied extensively. Earlier attempts employ hand-crafted features with probabilistic graphical models [1, 10, 11, 28, 40] or AND-OR graphs [2, 3, 43] based on inherent relationship between individual actions and group activities. RNN-based methods [6, 12, 22, 23, 30, 38, 42, 51, 54] have shown the effectiveness of hierarchical temporal modeling. LSTM architecture is often structured in hierarchical ways [23, 42, 51] to model individual action dynamics and aggregate individual features to infer group activity. Moreover, graphically constructed RNN models [12, 38] are proposed to utilize relationship among the individual features.

Recent approaches show more tendency towards adopting relational modeling [4, 15, 21, 53, 55, 58]. Graph-based approaches have been widely used to model spatio-temporal relationship between actors [15, 21, 53, 55, 58]; they first extract features from bounding boxes, and then place features as nodes and defining their relations as edges. These approaches subsequently employ their own way of evolving relation graphs, such as graph convolutional networks (GCN) [53] or graph attention networks (GAT) [15]. More complicated ways to develop relation graphs, for instance, constructing cross inference module to embed spatio-temporal features [55] or utilizing dynamic relation and dynamic walk offsets to build person-specific interaction graph [58], have also been introduced. On the other hand, Azar *et al.* [4] introduce the notion of activity map to encode spatial relations between individuals.

Transformer-based methods [17, 29, 36, 37, 57] model the relationship between features of group activities, and show significant improvements in GAR. They place a Transformer on top of the actor features to embed spatio-temporal relational contexts with conditional random fields (CRF) [36] or joint spatio-temporal contexts regarding intra- and inter-group relations [29]. The most related study by Yuan *et al.* [57] encodes person-specific scene context per individual feature. However, it still relies on person detector and only captures person-specific context. On the other hand, our method further considers multiple people and does not depend on any off-the-shelf detector, and thus enables the model to be trained effectively with less supervision. Moreover, ours utilizes learnable tokens to form partial context shared by different group activities.

Weakly supervised group activity recognition. GAR has many obstacles to overcome in order to be applicable in
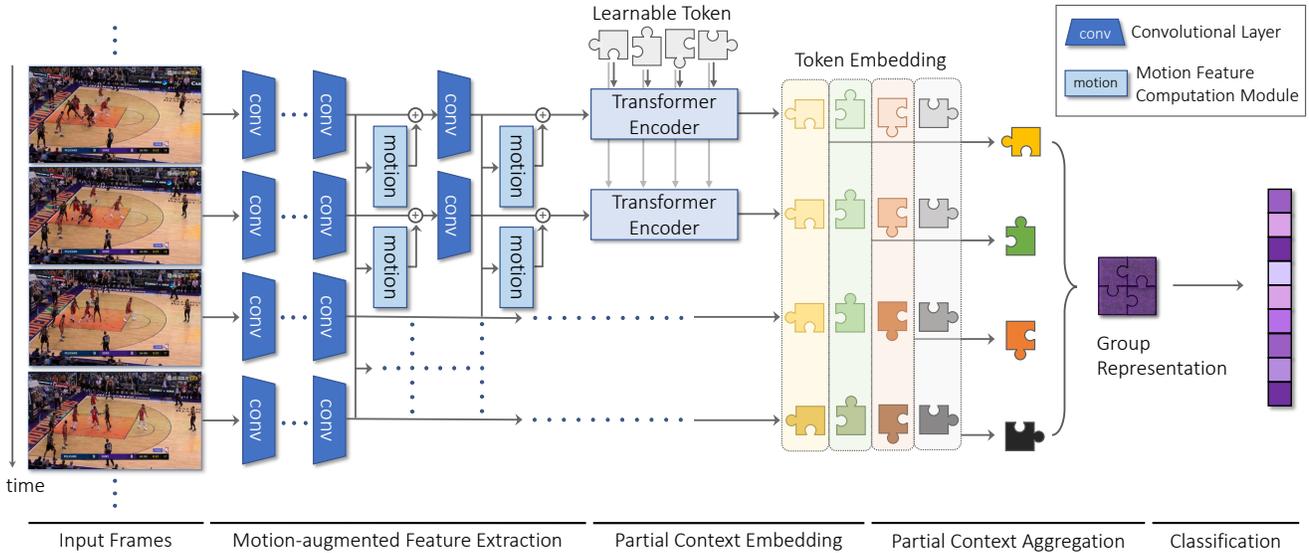
Figure 2. Overall architecture of our model. A CNN incorporating motion feature computation modules extracts a motion-augmented feature map per frame. At each frame, a set of learnable tokens (unpainted pieces of Jigsaw puzzles) are embedded to localize clues useful for group activity recognition through the attention mechanism of the Transformer encoder. The token embeddings (painted pieces of Jigsaw puzzles) are then fused to form a group representation in two steps: First aggregate embeddings of the same token (pieces with the same shape) across time, and then aggregate the results of different tokens (pieces with different shapes and colors). Finally, the group representation is fed into the classifier which predicts group activity class scores.

real life. In particular, heavy annotations such as bounding boxes and individual actions are rarely provided. Thus, several methods have addressed GAR with weaker supervision such as utilizing bounding boxes only to train their built-in detector [6, 60] or activity map [4]. Accordingly, Yan *et al.* [56] propose WSGAR, the task where bounding box annotations are not used in both training and inference. They address the absence of the bounding boxes by placing an off-the-shelf object detector inside the model. To prune off noisy outputs of the object detector, a relation graph is constructed regarding the relatedness of detected bounding boxes. Zhang *et al.* [61] propose a method using activity-specific features for multi-label activity recognition, which also shows an improvement in WSGAR. However, it is not designed for GAR and the performance gap from baseline is subtle in the WSGAR setting. Unlike the previous work [56, 61], we propose a detector-free method dedicated to WSGAR, which is not only free from actor-level annotation but also from object detector.

2.2. Transformer

Transformer [48] is originally devised to solve sequence-to-sequence task such as machine translation. It introduces self-attention mechanism with the aim of capturing global dependencies of the input elements. Recently, Transformer is widely adopted in many vision tasks either with a CNN feature extractor [8, 18, 24, 33, 62] or as a pure transformer architecture [13]. Detection Transformer (DETR) [8], a

Transformer model for object detection, is presented and utilized in detection-based tasks such as human-object interaction detection [24, 62] and object tracking [33]. Moreover, several attempts apply self-attention mechanism to process videos [7, 18, 41, 52]. Girdhar *et al.* [18] utilize spatio-temporal context around the person to localize and recognize human actions simultaneously. Ryoo *et al.* [41] propose TokenLearner, which efficiently learns to convey meaningful features of an input.

3. Proposed method

Our goal is to recognize a group activity in a multi-person video without using ground-truth bounding boxes or object detector. We achieve this goal by leveraging the attention mechanism to localize and encode partial contexts of a group activity, and then aggregating them into a group-level video representation while capturing their temporal dynamics. Our model is divided into three parts: motion-augmented feature extraction, partial context embedding, and partial context aggregation. Its overall architecture is illustrated in Fig. 2, and the remainder of this section elaborates on each of the three parts.

3.1. Motion-augmented feature extraction

Given a video clip of T frames $\mathbf{X}_{video} \in \mathbb{R}^{T \times H^0 \times W^0 \times 3}$ as input, an ImageNet pretrained ResNet [20] backbone extracts features $\mathbf{F}_{video} \in \mathbb{R}^{T \times H \times W \times C}$ in a frame-wise manner. To incorporate motion information into the features

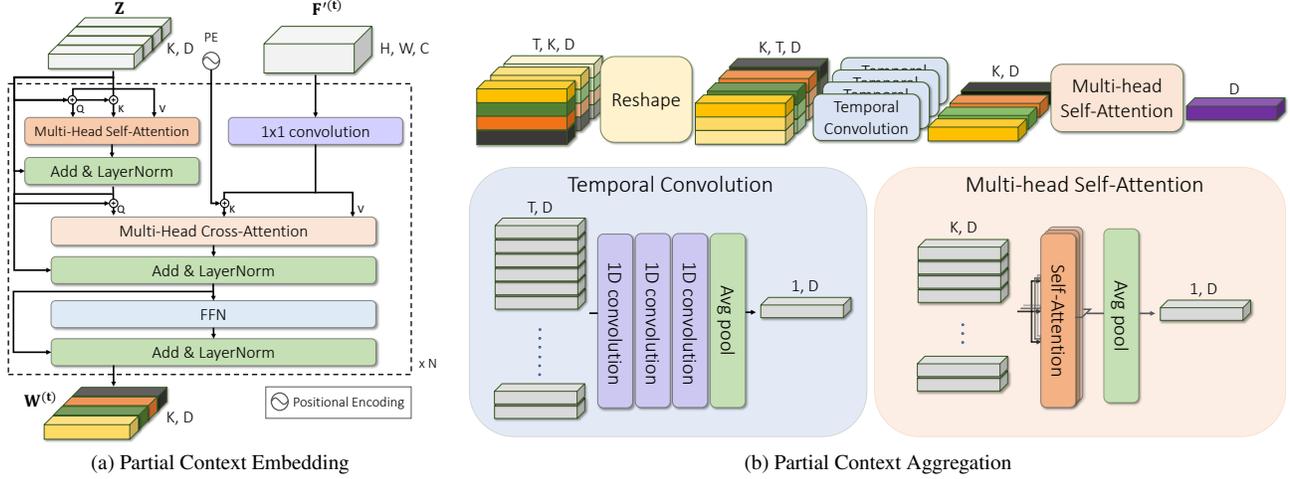(a) Partial Context Embedding   (b) Partial Context Aggregation

Figure 3. Detailed architectures of the partial context embedding and partial context aggregation modules.

without using computationally heavy 3D CNNs [9, 46] or optical flows [45, 59], our model computes local correlation between two adjacent intermediate feature maps, $\mathbf{F}^{(t)}$ and $\mathbf{F}^{(t+1)}$, and encodes the correlations into frame-wise motion features, similar to recent motion feature learning methods [26, 49].

Motion feature computation. Given two feature maps of adjacent frames, we first reduce their channel dimension to C' by 1×1 convolution. Then, the local correlation function s is defined as follows:

$$s : (\mathbf{F}^{(t)}, \mathbf{F}^{(t+1)}) \longmapsto \mathbf{S}^{(t)} \in \mathbb{R}^{H \times W \times P \times P}$$

where

$$(\mathbf{S}^{(t)})_{(\mathbf{x},\mathbf{p})} = \langle (\mathbf{F}^{(t)})_{(\mathbf{x})}, (\mathbf{F}^{(t+1)})_{(\mathbf{x}+\mathbf{p})} \rangle, \quad (1)$$

$\mathbf{x} \in [0, H-1] \times [0, W-1]$, and $\mathbf{p} \in [-l, l]^2$. An element $(\mathbf{S}^{(t)})_{(\mathbf{x},\mathbf{p})}$ of the local correlation tensor $\mathbf{S}^{(t)}$ is calculated by the dot product similarity of displaced vectors between adjacent frames $\mathbf{F}^{(t)}$ and $\mathbf{F}^{(t+1)}$. By restricting the maximum displacement to l, the correlation scores of spatial position $\mathbf{x}$ are computed only in its local neighborhood of size $P = 2l + 1$. Note that each feature map is zero-padded with the size of l. Hence, $\mathbf{S}^{(t)}$ reveals the motion at each location of $\mathbf{F}^{(t)}$ in the form of $P \times P$ local correlation map. In the case of the T-th frame, $\mathbf{S}^{(T)}$ is computed by the self-correlation $s(\mathbf{F}^{(T)}, \mathbf{F}^{(T)})$. To integrate the local correlation tensor into the backbone, 1×1 convolution transforms the local correlation tensor $\mathbf{S}^{(t)} \in \mathbb{R}^{H \times W \times P^2}$ into motion features $\mathbf{M}^{(t)} \in \mathbb{R}^{H \times W \times C}$. Then the motion features are inserted into the backbone using a residual connection, *i.e.*, $\mathbf{F}'^{(t)} = \mathbf{F}^{(t)} + \mathbf{M}^{(t)}$; this operation endows the output features with the sense of motion. In our model, two motion feature computation modules are inserted after the last two residual blocks of ResNet. Following [26], we adopt the local correlation computation implemented in FlowNet [14].

3.2. Partial context embedding (PCE)

Given the motion-augmented feature $\mathbf{F}' \in \mathbb{R}^{T \times H \times W \times C}$ for T frames, a set of K learnable tokens $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{K}$, where $\mathbf{z}_i \in \mathbb{R}^D$ for all i, is trained to encode partial contexts of a group activity through the Transformer encoder in a frame-wise manner. For each frame, K tokens are transformed into token embeddings $\mathbf{W}^{(t)}$ of the same size using the mechanism of the encoder, that is, the set of learnable tokens $\mathbf{Z}$ is shared by every frame to capture temporal dynamics of each token embeddings across time. To this end, we adopt a Transformer architecture [48] composed of multi-head cross-attention layer, multi-head self-attention layer, and feed forward network (FFN). The tokens are embedded while considering the relations with other tokens through the self-attention, and capture the partial contexts from the motion-augmented features through the cross-attention. We call this process *partial context embedding*, and its implementation follows the decoder of DETR [8]. Note that the weights of the Transformer encoder are shared across T frames. For clarity, we describe the detailed process of partial context embedding for a frame t given motion-augmented feature $\mathbf{F}'^{(t)} \in \mathbb{R}^{H \times W \times C}$ and a set of learnable tokens $\mathbf{Z}$ (Fig. 3a).

First, a point-wise convolution operation is applied to the feature map to reduce the channel dimension C to D. Then the spatial dimension of the feature map is flattened to convert its overall shape to $HW \times D$. For the multi-head cross-attention layer, the *Query* is the sum of two elements: (1) the learnable tokens $\mathbf{Z}$, and (2) the output of the multi-head self-attention applied to the learnable token $\mathbf{Z}$ and the output token embeddings of the previous encoder layer. The *Key* and *Value* come from the flattened feature map, and the spatial positional encoding is added to the *Key*. To be specific, the spatial coordinates of the input feature map $\mathbf{F}'^{(t)}$ are transformed into the spatial positional encod-

ing using the sinusoidal function [48]; each half of D channels are encoded with width and height coordinates, respectively. Through the attention mechanism, each token learns to localize and encode partial contexts including key actors and entities from the given feature maps. For T frames, the output token embeddings $\mathbf{W}^{(t)} = \{\mathbf{w}_i^{(t)}\}_{i=1}^K$ are stacked to form $\mathbf{W} = [\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \ldots, \mathbf{W}^{(T)}] \in \mathbb{R}^{T \times K \times D}$, a bag of $T \times K$ token embeddings of dimension D. Note that $\mathbf{w}_i^{(t)}$ denotes the i-th token embedding for a frame t.

3.3. Partial context aggregation (PCA)

The partial context aggregation module takes the output token embeddings $\mathbf{W} \in \mathbb{R}^{T \times K \times D}$ and aggregates them to the final group representation $\mathbf{g} \in \mathbb{R}^D$. This process is divided into two steps. The first step aggregates embeddings of the same token across time and the second step assembles the results from the first step to build a single video representation. The overall process is illustrated in Fig. 3b, and discussed in detail below.

In the first step, the output token embeddings $\mathbf{W}_i = \{\mathbf{w}_i^{(t)}\}_{t=1}^T \in \mathbb{R}^{T \times D}$ from the i-th token at different frames are fused to form an i-th aggregated token feature $\tilde{\mathbf{w}}_i \in \mathbb{R}^D$ for all i. After reshaping the output token embedding $\mathbf{W}$ to $K \times T \times D$ tensor, a stack of 1D convolutional layers, each followed by ReLU [34], is applied along T dimension. The output is then fed to $AvgPool$ operation. This temporal convolution block $f(\cdot)$ gradually aggregates the token embedding along the time dimension, *i.e.*, $f : \mathbb{R}^{K \times T \times D} \to \mathbb{R}^{K \times D}$. It achieves robustness to a temporal shift of activity in the video via parameter sharing in time while effectively capturing the temporal dynamics of token embeddings via stacked layers. In the second step, K aggregated token features $\tilde{\mathbf{W}} = \{\tilde{\mathbf{w}}_i\}_{i=1}^K \in \mathbb{R}^{K \times D}$ are fused to form the group representation $\mathbf{g}$. Specifically, layer normalization [5] is first applied to $\tilde{\mathbf{W}}$, and then a single layer multi-head self-attention is adopted to capture the dynamic relations between the K aggregated token embeddings. Finally, the group representation $\mathbf{g}$ is obtained by applying $AvgPool$ operation over K dimension.

3.4. Training objective

After obtaining the group representation, a classifier is applied to predict group activity class scores. Our model is trained with the standard cross-entropy loss in an end-to-end manner.

4. Experiments

We evaluate the proposed detector-free model for WS-GAR on two datasets, Volleyball [23] and NBA [56], where our model is compared with the state-of-the-art WSGAR and GAR methods. We also validate the effectiveness of the model by extensive ablation studies and qualitative analysis.

4.1. Datasets

Volleyball dataset. This dataset consists of 55 videos, which are further divided into 4830 clips. 3494 clips among them are used for training and 1337 clips are kept for testing. The center frame of each clip is labeled with (i) one of 8 group activity labels, (ii) one of 9 action labels per player, and (iii) a bounding box per player. Bounding box tracklets of players along the 10 frames before and after the center frame are also provided by Bagautdinov *et al.* [6], and serve as the ground-truth bounding box labels of these frames. However, in the WSGAR setting, models including ours utilize the group activity labels only and disuse the stronger and fine-grained annotations. We adopt Multi-class Classification Accuracy (MCA) and Merged MCA for evaluation throughout our experiments. In particular, for computing Merged MCA, we merge the classes *right set* and *right pass* into *right pass-set*, and *left set* and *left pass* into *left pass-set* as in SAM [56] for a fair comparison.

NBA dataset. This dataset consists of 7624 clips for training and 1548 clips for testing. Currently, it is the only dataset proposed for WSGAR, which only provides one of 9 group activity labels for each clip. Thanks to its low annotation cost, it is currently the largest group activity recognition dataset. Since each video clip is 6-second long and usually exhibits a nontrivial temporal structure, the dataset requires a model that captures long-term temporal dynamics compared with other GAR benchmarks. Also, it is a challenging benchmark due to fast movement, camera view change, and a varying number of people in each frame. For evaluation, we adopt Multi-class Classification Accuracy (MCA) and Mean Per Class Accuracy (MPCA) metrics; MPCA is adopted due to the class imbalance issue of the dataset.

4.2. Implementation details

Sampling strategy. For both datasets, T frames are sampled using the segment-based sampling [50] and each frame is resized to 720×1280. Note that $T = 18$ for the NBA dataset and $T = 5$ for the Volleyball dataset.

Hyperparameters. We adopt an ImageNet pretrained ResNet-18 [20] as the backbone. For motion-augmented feature extraction, a 1×1 convolution operation reduces the channel dimension to $C' = 64$ and local neighborhood size is set to $P = 11$. We stack 6 Transformer encoder layers with 4 attention heads and 256 channels for the NBA dataset, and 2 Transformer encoder layers with 2 attention heads and 256 channels for the Volleyball dataset. We test different numbers of learnable tokens including $K = 1, 2, 4, 8, 12, 16$, and use 12 for both datasets. For partial context aggregation module, three 1D convolutional layers with kernel size of 5 without padding are used for NBA and two 1D convolutional layers with kernel size of 3 with zero-padding are used for Volleyball. In the multi-head self-attention (MHSA) aggregation, a single layer MHSA

with 256 channels is used for both datasets and the number of heads is 4 for NBA and 2 for Volleyball.

Training. On both datasets, our model is optimized by ADAM [25] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-8$ for 30 epochs. Weight decay is set to $1e-4$ for the NBA dataset and $1e-3$ for the Volleyball dataset. Learning rate is initially set to $1e-6$ with linear warmup to $1e-4$ for 5 epochs, and linearly decayed after the 6th epoch. We use a mini-batch of size 4 on NBA and 8 on Volleyball.

4.3. Comparison with the state-of-the-art methods

NBA dataset. For the NBA dataset, we compare our method with the state of the art in GAR and WSGAR, which use bounding box proposals provided by SAM [56], and also with recent video backbones in the weakly supervised learning setting. For a fair comparison, we set their backbones to ResNet-18 except for VideoSwin [32] and use only RGB frames as input for all the methods. Table 1 summarizes the results. Note that the scores of reproduced SAM [56] are higher than those reported in its original paper. The proposed method beats all the GAR and WSGAR methods by a large margin: 14.2%p of MCA and 14.4%p of MPCA. Regarding complexity, our method demands less parameters and slightly more FLOPs than other GAR methods although we do not count the computational complexity of their object detectors. Our method is also compared with recent video backbones, ResNet-18 TSM [31] and VideoSwin-T [32], used in conventional action recognition. Although these powerful backbones perform well in WS-GAR, ours achieves the best. We also show the result of our method without the motion feature module, which still outperforms all the other methods.

Volleyball dataset. For the Volleyball dataset, we compare our method with the state-of-the-art GAR and WS-GAR methods in two settings: fully supervised setting and weakly supervised setting. The difference of two settings is the use of actor-level labels including ground-truth bounding boxes and individual action class labels, in both training and inference. For a fair comparison, we report the results of previous methods [4, 6, 29, 38, 55, 57] using only RGB inputs, and the reproduced results [17, 36, 53, 54, 58] using the ResNet-18 backbone. Note that the former is brought from the original papers and the latter is the MCA values from [58]. For the weakly supervised setting, we replace the ground-truth bounding boxes with an object detector pretrained on an external dataset and remove the individual action classification head. Table 2 summarizes the results. The first and second section show the results of previous methods in fully supervised setting and weakly supervised setting, respectively. Our method outperforms all the GAR and WSGAR models in weakly supervised setting by a substantial margin: 3.1%p of MCA and 1.3%p of Merged MCA when compared to the models using ResNet-18 backbone.

Method	# Params	FLOPs	MCA	MPCA
Video backbone				
TSM [31]	11.2M	303G	66.6	60.3
VideoSwin [32]	27.9M	478G	64.3	60.6
GAR model				
ARG [53]	49.5M	307G	59.0	56.8
AT [17]	29.6M	305G	47.1	41.5
SACRF [36]	53.7M	339G	56.3	52.8
DIN [58]	26.0M	304G	61.6	56.0
[†]SAM [56]	-	-	49.1	47.5
SAM [56]	25.5M	304G	54.3	51.5
Ours w/o motion	17.3M	311G	<u>73.6</u>	<u>69.0</u>
Ours	17.5M	313G	**75.8**	**71.2**

Table 1. Comparison with the state-of-the-art GAR models and video backbones on the NBA dataset. All models except VideoSwin adopt ResNet-18 backbone. Numbers in **bold** indicate the best performance and <u>underlined</u> ones are the second best. '[†]' indicates that the result is copied directly from SAM [56]. All the other results are reproduced by us.

Method	Backbone	MCA	Merged MCA
Fully supervised			
SSU [6]	Inception-v3	89.9	-
PCTDM [54]	ResNet-18	90.3	94.3
StagNet [38]	VGG-16	89.3	-
ARG [53]	ResNet-18	91.1	<u>95.1</u>
CRM [4]	I3D	92.1	-
HiGCIN [55]	ResNet-18	91.4	-
AT [17]	ResNet-18	90.0	94.0
SACRF [36]	ResNet-18	90.7	92.7
DIN [58]	ResNet-18	<u>93.1</u>	**95.6**
TCE+STBiP [57]	VGG-16	**94.1**	-
GroupFormer [29]	Inception-v3	**94.1**	-
Weakly supervised			
PCTDM [54]	ResNet-18	80.5	90.0
ARG [53]	ResNet-18	87.4	92.9
AT [17]	ResBet-18	84.3	89.6
SACRF [36]	ResNet-18	83.3	86.1
DIN [58]	ResNet-18	86.5	93.1
SAM [56]	ResNet-18	86.3	93.1
[†]SAM [56]	Inception-v3	-	94.0
Ours w/o motion	ResNet-18	<u>88.1</u>	94.0
Ours	ResNet-18	**90.5**	**94.4**

Table 2. Comparison with the state-of-the-art methods on the Volleyball dataset. '-' indicates that the result is not provided, and '[†]' indicates that the result is copied directly from SAM [56].

It also beats the current state of the art based on Inception-v3. We state the results without motion feature module to

show its competitiveness. Compared to the GAR methods in the fully supervised settings, our method surpasses the recent GAR methods [6,17,36,38,54], using stronger actor-level supervision.

4.4. Ablation studies

We also demonstrate the effectiveness of our method by ablation studies and analysis on the NBA dataset.

Effects of the proposed modules. Table 3 summarizes the effects of each module. Base model consists of a backbone, a global average pooling layer, and a group activity classifier. Without the partial context aggregation (PCA), all token embeddings are averaged to form a group representation. Unless the motion feature module is inserted, the original backbone is utilized. All training setups except the model architecture are same. From the result, all three components consistently enhance the model in terms of two metrics. Partial context embedding (PCE) module improves MCA from 58.4% to 64.1% and MPCA from 51.7% to 58.5%. Regarding that the base model predicts a group activity using a global scene feature, it emphasizes the importance of capturing partial contexts for recognizing a group activity. PCA module further increases MCA from 64.1% to 73.6% and MPCA from 58.5% to 69.0%. This also shows the effectiveness of our aggregation method versus naive average pooling aggregation. The detailed analysis of the token aggregation methods will be discussed later (Table 4). Motion feature module gives an additional gain regardless of other components, which implies that the sense of motion helps to understand group activities.

Effects of the token aggregation methods. For this ablation, we do not adopt motion feature module, and plain ResNet-18 backbone is used as a feature extractor. Table 4 shows the performance of various token aggregation methods. The first and second section of the table contains the results of one-stage aggregation and two-stage aggregation method, respectively. Given a bag of token embeddings, the two-stage aggregation method separates the aggregation across different frames and the aggregation across different tokens, whereas the one-stage aggregation methods fuse all tokens at once. MLP aggregation concatenates token embeddings and project them using a linear layer, and it is regarded as a baseline for aggregating vectors along certain axis. Multi-head self-attention (MHSA) aggregation is composed of a single layer MHSA and an average pooling layer. From the result, the two-stage aggregation methods exceed the one-stage methods in most cases. Compared to the two-stage MLP aggregation, our method, 1D convolution across frames followed by MHSA across tokens, performs better. This shows the effectiveness of temporal convolution and MHSA aggregation, which are robust to a temporal shift in the video and able to capture dynamic relation between partial contexts. In addition, the order of aggre-

Model	MCA	MPCA
Base model	58.4	51.7
Base model + Motion	62.7	55.1
PCE	64.1	58.5
PCE + Motion	65.1	59.1
PCE + PCA	73.6	69.0
PCE + PCA + Motion	**75.8**	**71.2**

Table 3. Contributions of the proposed modules. PCE and PCA denote partial context embedding and partial context aggregation modules, respectively.

Aggregation method	MCA	MPCA
Average pooling	64.1	58.5
Max pooling	61.3	55.1
MLP (both T, K- dim)	62.8	56.4
MHSA (both T, K- dim)	68.7	64.7
MLP (T- dim) - MLP (K- dim)	68.9	63.1
MHSA (K- dim) - 1D conv (T- dim)	72.4	67.0
Ours (1D conv - MHSA)	**73.6**	**69.0**

Table 4. Ablation on the token aggregation methods. T-dim and K-dim imply aggregation with the same token at different frames and aggregation with different tokens at the same frame, respectively. MHSA stands for multi-head self-attention.

# token	MCA	MPCA
1	72.7	66.6
2	73.1	67.2
4	72.6	67.7
8	74.0	**69.7**
12	73.6	69.0
16	**74.3**	68.8

Model	MCA	MPCA
Base	73.6	69.0
res_2	72.2	68.3
res_3	74.2	70.2
res_4	74.8	69.4
res_5	74.0	70.5
$res_{3,4}$	74.2	69.0
$res_{4,5}$	**75.8**	**71.2**
$res_{3,4,5}$	73.1	68.1

Table 5. Ablation on the number of tokens per frame.

Table 6. Ablation on the position of the motion feature module. Base represents the model without the module.

gation demonstrates notable difference: Assembling with respect to temporal axis first and then to token axis brings a gain of 1.2%p of MCA and 2.0%p of MPCA.

Effects of the number of tokens. Table 5 summarizes the performance of different number of tokens. Note that the ablation is conducted without the motion feature module. When the number of learnable tokens is set to 1, a single token embedding extracts context information as a global scene feature. The performance tends to increase with the number of tokens in general, and the optimal result is gained when the number is in the range of 8 to 16. These results indicate that it is more effective to divide the scene context into several pieces in order to encapsulate enriched group representation.

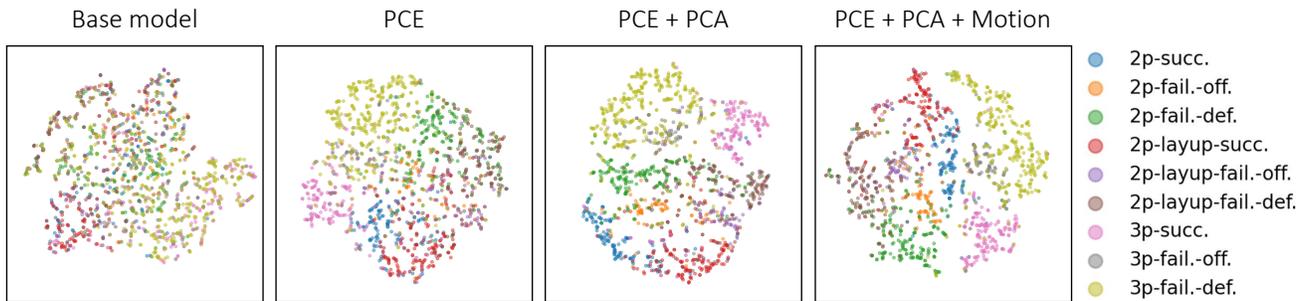Figure 4. Visualization of the Transformer encoder attention maps on the NBA dataset.

| Base model | PCE | PCE + PCA | PCE + PCA + Motion |

- ● 2p-succ.
- ● 2p-fail.-off.
- ● 2p-fail.-def.
- ● 2p-layup-succ.
- ● 2p-layup-fail.-off.
- ● 2p-layup-fail.-def.
- ● 3p-succ.
- ● 3p-fail.-off.
- ● 3p-fail.-def.

Figure 5. t-SNE [47] visualization of feature embedding learned by different model variants on the NBA dataset.

Effects of the position of the motion feature module. We investigate the efficacy of inserting the motion feature module at different positions. The motion feature is placed after a residual block and it is denoted as res_i if the motion feature module is inserted after the i-th residual block. As shown in Table 6, the motion feature module is effective in most cases except the case where it is inserted after the 2nd residual block, which is considered too early to compute the local correlation. Putting multiple motion feature modules is also tested, and inserting motion feature modules after 4th and 5th block performs the best.

4.5. Qualitative analysis

In Fig. 4, we illustrate the attention visualizations obtained from the final Transformer encoder layer on the NBA dataset. The results imply that token embeddings are learned to attend key concepts and follow the activity occurs in a given video clip. Fig. 5 displays the t-SNE [47] visualization results of our model and its variants. Final group representation of each model on NBA is visualized in two-dimensional space. We can find that each proposed modules contribute to a clear separation of each class.

5. Conclusion

We have presented a detector-free method for weakly supervised group activity recognition, which first embeds the partial contexts of an activity through the attention mechanism, then aggregates them while capturing their temporal evolution. We achieve the state of the art on two benchmarks in weakly supervised learning setting, and even outperform several GAR models that rely on stronger supervision. These results suggest that partial contexts captured by our method could be more effective than the human prior given in the form of person bounding boxes. Albeit with such benefits, our model has difficulties generating sufficiently diverse token embeddings due to the absence of direct supervision. Further improvement could be achieved by increasing their diversity without capturing irrelevant contexts.

References

[1] Mohamed Rabie Amer, Peng Lei, and Sinisa Todorovic. Hirf: Hierarchical random field for collective activity recognition in videos. In *Proc. European Conference on Computer Vision (ECCV)*, pages 572–585. Springer, 2014. 2

[2] Mohamed R Amer, Sinisa Todorovic, Alan Fern, and Song-Chun Zhu. Monte carlo tree search for scheduling activity recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1353–1360, 2013. 2

[3] Mohamed R Amer, Dan Xie, Mingtian Zhao, Sinisa Todorovic, and Song-Chun Zhu. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *Proc. European Conference on Computer Vision (ECCV)*, pages 187–200. Springer, 2012. 2

[4] Sina Mokhtarzadeh Azar, Mina Ghadimi Atigh, Ahmad Nickabadi, and Alexandre Alahi. Convolutional relational machine for group activity recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7892–7901, 2019. 2, 3, 6

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[6] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4315–4324, 2017. 1, 2, 3, 5, 6, 7

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proc. International Conference on Machine Learning (ICML)*, 2021. 3

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. European Conference on Computer Vision (ECCV)*, pages 213–229. Springer, 2020. 3, 4

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 4

[10] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *Proc. European Conference on Computer Vision (ECCV)*, pages 215–230. Springer, 2012. 2

[11] Wongun Choi, Khuram Shahid, and Silvio Savarese. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *Proc. IEEE International Conference on Computer Vision (ICCV) Workshops*, pages 1282–1289. IEEE, 2009. 2

[12] Zhiwei Deng, Arash Vahdat, Hexiang Hu, and Greg Mori. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4772–4781, 2016. 2

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. International Conference on Learning Representations (ICLR)*, 2021. 3

[14] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015. 4

[15] Mahsa Ehsanpour, Alireza Abedin, Fatemeh Saleh, Javen Shi, Ian Reid, and Hamid Rezatofighi. Joint learning of social groups, individuals action and sub-group activities in videos. In *Proc. European Conference on Computer Vision (ECCV)*, pages 177–195. Springer, 2020. 1, 2

[16] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[17] 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, 6, 7

[18] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253, 2019. 1, 3

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 1

[20] 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. 3, 5

[21] Guyue Hu, Bo Cui, Yuan He, and Shan Yu. Progressive relation learning for group activity recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 980–989, 2020. 1, 2

[22] Mostafa S Ibrahim and Greg Mori. Hierarchical relational networks for group activity recognition and retrieval. In *Proc. European Conference on Computer Vision (ECCV)*, pages 721–736, 2018. 2

[23] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1971–1980, 2016. 1, 2, 5

[24] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 74–83, 2021. 3

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2015. 6

[26] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 4

[27] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Learning self-similarity in space and time as generalized motion for video action recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 13065–13075, 2021. 2

[28] Tian Lan, Leonid Sigal, and Greg Mori. Social roles in hierarchical models for human activity recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1354–1361. IEEE, 2012. 2

[29] Shuaicheng Li, Qianggang Cao, Lingbo Liu, Kunlin Yang, Shinan Liu, Jun Hou, and Shuai Yi. Groupformer: Group activity recognition with clustered spatial-temporal transformer. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 13668–13677, 2021. 1, 2, 6

[30] Xin Li and Mooi Choo Chuah. Sbgar: Semantics based group activity recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 2876–2885, 2017. 2

[31] 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. 1, 6

[32] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021. 6

[33] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 3

[34] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. International Conference on Machine Learning (ICML)*, 2010. 5

[35] AJ Piergiovanni and Michael S Ryoo. Representation flow for action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9945–9953, 2019. 1, 2

[36] 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, 2, 6, 7

[37] Rizard Renanda Adhi Pramono, Wen-Hsien Fang, and Yie-Tarng Chen. Relational reasoning for group activity recognition via self-attention augmented conditional random field. *IEEE Transactions on Image Processing*, 2021. 2

[38] Mengshi Qi, Jie Qin, Annan Li, Yunhong Wang, Jiebo Luo, and Luc Van Gool. stagnet: An attentive semantic rnn for group activity recognition. In *Proc. European Conference on Computer Vision (ECCV)*, pages 101–117, 2018. 2, 6, 7

[39] 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

[40] MS Ryoo and JK2783696 Aggarwal. Stochastic representation and recognition of high-level group activities. *International Journal of Computer Vision (IJCV)*, 93(2):183–200, 2011. 2

[41] Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. *Proc. Neural Information Processing Systems (NeurIPS)*, 34, 2021. 3

[42] Tianmin Shu, Sinisa Todorovic, and Song-Chun Zhu. Cern: confidence-energy recurrent network for group activity recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5523–5531, 2017. 2

[43] Tianmin Shu, Dan Xie, Brandon Rothrock, Sinisa Todorovic, and Song Chun Zhu. Joint inference of groups, events and human roles in aerial videos. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4576–4584, 2015. 2

[44] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proc. Neural Information Processing Systems (NeurIPS)*, pages 568–576, 2014. 1

[45] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018. 4

[46] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015. 1, 4

[47] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. 2, 3, 4, 5

[49] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4

[50] 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, 5

[51] Minsi Wang, Bingbing Ni, and Xiaokang Yang. Recurrent modeling of interaction context for collective activity recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3048–3056, 2017. 2

[52] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018. 1, 3

[53] 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, 2, 6

[54] 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. 2, 6, 7

[55] 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 (TPAMI)*, 2020. 1, 2, 6

[56] 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, 2, 3, 5, 6

[57] Hangjie Yuan and Dong Ni. Learning visual context for group activity recognition. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 3261–3269, 2021. 1, 2, 6

[58] 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. 2, 6

[59] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007. 4

[60] Peizhen Zhang, Yongyi Tang, Jian-Fang Hu, and Wei-Shi Zheng. Fast collective activity recognition under weak supervision. *IEEE Transactions on Image Processing*, 29:29–43, 2019. 1, 3

[61] Yanyi Zhang, Xinyu Li, and Ivan Marsic. Multi-label activity recognition using activity-specific features and activity correlations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14625–14635, 2021. 3

[62] Cheng Zou, Bohan Wang, Yue Hu, Junqi Liu, Qian Wu, Yu Zhao, Boxun Li, Chenguang Zhang, Chi Zhang, Yichen Wei, et al. End-to-end human object interaction detection with hoi transformer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11825–11834, 2021. 3