Learning Visual Context for Group Activity Recognition

Hangjie Yuan,[1] **Dong Ni**[1,2*]

[1] College of Control Science and Engineering, Zhejiang University, Hangzhou, China
[2] State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, China
{hj.yuan, dni}@zju.edu.cn

Abstract

Group activity recognition aims to recognize an overall activity in a multi-person scene. Previous methods strive to reason on individual features. However, they under-explore the person-specific contextual information, which is significant and informative in computer vision tasks. In this paper, we propose a new reasoning paradigm to incorporate global contextual information. Specifically, we propose two modules to bridge the gap between group activity and visual context. The first is Transformer based Context Encoding (TCE) module, which enhances individual representation by encoding global contextual information to individual features and refining the aggregated information. The second is Spatial-Temporal Bilinear Pooling (STBiP) module. It firstly further explores pairwise relationships for the context encoded individual representation, then generates semantic representations via gated message passing on a constructed spatial-temporal graph. On their basis, we further design a two-branch model that integrates the designed modules into a pipeline. Systematic experiments demonstrate each module's effectiveness on either branch. Visualizations indicate that visual contextual cues can be aggregated globally by TCE. Moreover, our method achieves state-of-the-art results on two widely used benchmarks using only RGB images as input and 2D backbones.

Introduction

Group activity recognition (Ibrahim et al. 2016) or collective activity recognition (Choi, Shahid, and Savarese 2009) has attracted more research attention recently due to its significance in video understanding. Group activity recognition is a task aiming to recognize the overall activity of a group of people, which has a promising future for various applications, *e.g.* sports/surveillance video analysis, social activity understanding, video search and retrieval. Given a video clip, the difficulties of this problem lie not only in the recognition of individual actions, but also in the exploration of the scene information and collaborative relation among people.

Recently proposed methods are mostly based on deep learning techniques. We revisit them in the view of a causal graph (Pearl, Glymour, and Jewell 2016) illustrated in Figure 1(a), in which $I \rightarrow C$: generate feature maps from a given image, $C \rightarrow X$: extract individual features from feature maps (usually by RoIAlign (He et al. 2017)), and $X \rightarrow Z$: reason on individual features to generate a scene-level representation and then classify it. However, step $I \rightarrow C$ only crops features aligned with the bounding box, which implies that features are limited to local receptive fields. Therefore global contextual information that benefits recognition is omitted indeliberately. However, previous works (Wang et al. 2018; Cao et al. 2019) have demonstrated the significance of global-range dependency in computer vision tasks, which lacks careful exploration in our problem. For example, when you try to recognize an activity in volleyball games, you need to consider players, referees, line judges and spectators. Like spectators' cheer for the winning team, body orientation of judges, actions and positions of line judges also provide informative cues to the result.

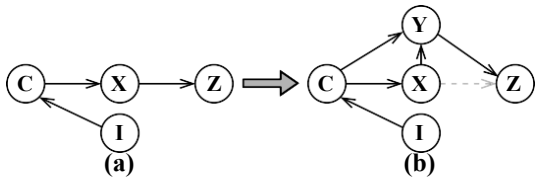


Figure 1: (a) Causal graph of existing models. (b) Causal graph of our proposed model. I: input images. C: global feature map. X: individual features. Y: context encoded individual features. Z: activity prediction.

In an effort to solve the aforementioned drawback, we propose a new method aligned with a new causal graph illustrated in Figure 1(b). The major difference between the two methods is that instead of reasoning on X directly, we firstly generate context encoded features Y by the combination of individual features X and global feature map C, then reason on Y to get a semantic representation Z.

In practice, we resort to the well-known concept of transformer (Vaswani et al. 2017), which adopts the self-attention mechanism to aggregate information from the global input sequence. We design a Transformer based Context Encoding module (TCE) to incorporate the person-specific contextual information. Specifically, we align the individual features X with the global feature map C, aggregate and refine informative contextual features to get context encoded features Y.

*Corresponding author.

Our experiments indicate that TCE emphasizes the key individual features and the informative surrounding features, which both facilitate the following modules with sufficient information to reason on.

To fully explore the interactions between context encoded individual features Y, we propose a Spatial-Temporal Bilinear Pooling module (STBiP) to model the spatial collaborative relationship between different people, and model the temporal dynamics of the same individual. Specifically, we construct a spatial-temporal graph, explore pairwise interactions by bilinear pooling and then perform gated message passing to get a semantic representation. By performing STBiP on context encoded individual features, all pairwise interactions between individuals and contextual information can be exhaustively taken into consideration.

Intuitively, both fine-grained pose information and global features are necessary for group activity recognition since they provide complementary information (Gavrilyuk et al. 2020). Finally, to integrate the above intuitions and designed modules into a pipeline, we design a two-branch model composed of a pose branch and a global branch. Individual features X come from pose features in the pose branch and come from RoIAlign features in the global branch. Two branches can infer alone or be fused by different strategies to benefit from complementarity.

The contributions of our work are summarized as follows:

- We design a new reasoning paradigm to incorporate global visual context for group activity recognition, which aligns with a new causal graph.
- Specifically, we design a TCE module to encode person-specific contextual information suitable for the pose branch and the global branch.
- We design a STBiP module to mine interactions between context encoded features on a constructed spatial-temporal graph, which produces semantic activity representations.
- We integrate above modules into a pipeline by designing a two-branch model. Moreover, our model achieves state-of-the-art results on two widely used datasets while using raw RGB images as input and 2D backbones.

Related Work

Visual Context Encoding There are mainly two kinds of visual context encoding methods. The first kind is implicit encoding (Hu, Shen, and Sun 2018; Woo et al. 2018; Wang et al. 2018; Cao et al. 2019; Girdhar et al. 2019; Huang et al. 2019), which is always instantiated by attention/self-attention mechanism. The second kind is explicit encoding, which is always instance-level and widely used in scene graph generation (Zellers et al. 2018; Tang et al. 2019) and human-object interaction (Yao and Fei-Fei 2010; Chao et al. 2015). Our work follows the first kind and encodes contextual features guided by individual features.

Bilinear Pooling Bilinear pooling was firstly proposed to provide a sufficiently expressive representations of two-factor interactions (Tenenbaum and Freeman 2000). Now it has developed into an extensively used feature fusion technique and has been successfully applied in various visual-related tasks like fine-grained recognition (Lin, RoyChowdhury, and Maji 2015; Gao et al. 2016; Kong and Fowlkes 2017; Wei et al. 2018; Yu et al. 2018a; Zhang et al. 2019) and visual question answering (Fukui et al. 2016; Kim et al. 2016; Yu et al. 2018b). We incorporate bilinear pooling into our reasoning scheme to obtain semantic representations given context encoded features.

Group Activity Recognition Initially, many works resolving group activity recognition were based on handcrafted features and probabilistic graphical models (Choi, Shahid, and Savarese 2011; Lan et al. 2011; Lan, Sigal, and Mori 2012; Amer, Lei, and Todorovic 2014). They mainly designed structured models to aggregate significant information in spatial-temporal domain.

Recent works are mostly based on deep learning methods, following the graph in Figure 1(a). Their differences mainly lie in reasoning methods (*i.e.* $X \rightarrow Z$). The core idea is to emphasize relevant actors and suppress irrelevant ones. The most frequently adopted reasoning method stems from RNN and its derivative models (Ibrahim et al. 2016; Shu, Todorovic, and Zhu 2017; Bagautdinov et al. 2017; Yan et al. 2018; Tang et al. 2018; Ibrahim and Mori 2018; Qi et al. 2018), owing to RNN's strong capacity of sequence modeling. Graph convolution network (Wu et al. 2019), transformer (Gavrilyuk et al. 2020) and deep reinforcement learning (Hu et al. 2020) also demonstrate their effectiveness of reasoning. Note that previous methods like (Wang, Ni, and Yang 2017; Tang et al. 2018) also mention 'context' but defines it upon X. To the best of our knowledge, we are the first to explore global visual context in C.

Our Approach

The overall framework of our proposed method is illustrated in Figure 2. In general, our method composes of two branches (one branch also works), namely the pose branch and the global branch, which share similar pipeline and can be fused optionally. For each branch, it contains a selected base network, TCE modules and STBiP modules. The detailed model descriptions are stated in subsections below.

Base Network Architecture

To recognize a group activity, we solve it by recognizing individual actions first. For individual action recognition, we need to observe not only how a particular person behaves with his/her body joints but also the whole contexts he/she is in. The former is treated as pose features, the latter global individual features (*i.e.* RoIAlign features). In order to capture the above information, distinct backbones are required. Specifically, we adopt HRNet-w32 as pose feature extractor following (Gavrilyuk et al. 2020) and Inception-v3/VGG16 as global feature extractor following (Shu, Todorovic, and Zhu 2017; Bagautdinov et al. 2017).

To extract features from a T-frame video clip, we divide the clip into K temporal segments first. Afterwards, K frames are uniformly sampled from K segment. For the pose branch, persons are cropped from the sampled images

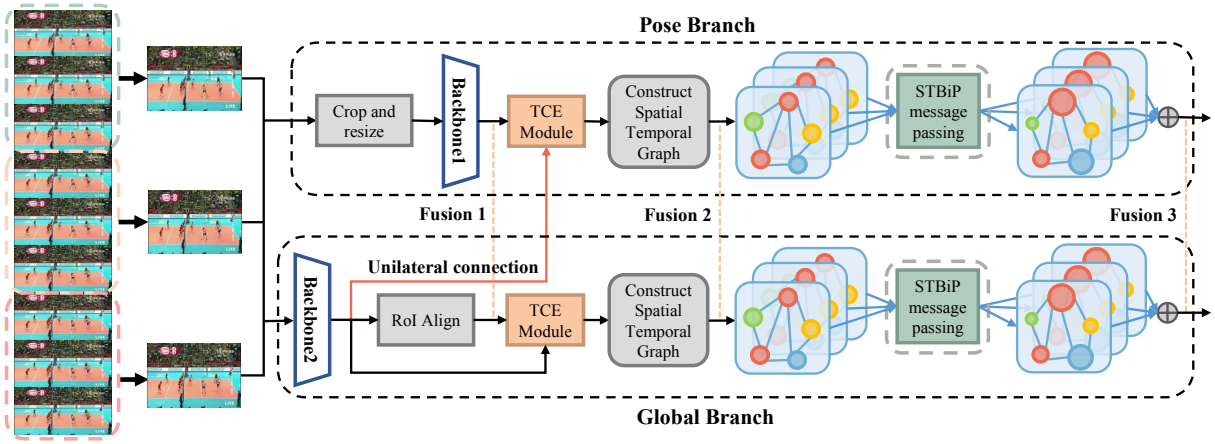


Figure 2: The overall framework of our proposed method, which incorporates TCE modules and STBiP modules to the pose branch and the global branch respectively. The dash lines which link two branches denote different fusion positions. $\bigoplus$ at the end of a branch stands for global attribute pooling.

and resized to a fixed size. Then, they are fed to HRNet-w32 to get individual features of d_e-dimension. For the global branch, frames pass through Inception-v3/VGG16 and RoIAlign to get individual features of d_e-dimension. Every frame has a total of N person bounding boxes.

Transformer Based Context Encoding

In this subsection, we introduce our TCE module. This module takes in a group of individual features $\boldsymbol{X} \in \mathbb{R}^{N \times d_e}$ and a global feature map $\boldsymbol{C} \in \mathbb{R}^{h \times w \times d_g}$. The outputs of this module is context encoded individual features denoted as $\boldsymbol{Y} \in \mathbb{R}^{N \times d_y}$. For clarity, we will detail the generation of ith context encoded feature $\boldsymbol{y}_i \in \mathbb{R}^{d_y}$ given ith individual feature $\boldsymbol{x}_i \in \mathbb{R}^{d_e}$ and global feature map $\boldsymbol{C}$ in following equations and illustrations.

A neat architecture of TCE is illustrated in Figure 3(a), which splits the module into three stages: input alignment, attention aggregation and Feed-Forward Network (FFN) refining. Following (Vaswani et al. 2017), the original transformer encoder finds a query, a set of keys and values, and implements self-attention on them to explore the alignment between the query and the keys, and then get a weighted sum of the values. Specifically in our architecture, we firstly perform input alignment, which composes of two steps: (1) encoding position information respectively for the global feature map and individual features, (2) using a pointwise convolution (Howard et al. 2017) on the global feature map to get $\tilde{\boldsymbol{C}} \in \mathbb{R}^{h \times w \times d_c}$ and using a linear projection layer on individual features to get $\tilde{\boldsymbol{x}}_i \in \mathbb{R}^{d_c}$. The spatial dimension of the global feature map is then collapsed to $\tilde{\boldsymbol{C}} \in \mathbb{R}^{hw \times d_c}$ for the encoder requires a sequence as input.

Second, we perform attention aggregation. In our problem, the query is the reduced individual features. Keys and values are the reduced global feature map. To get the primary encoded representations, we adopt the scaled dot-product attention, followed by a softmax function, a weighted summation and a residual connection. It can be expressed as

$$\alpha_{i,j} = \frac{e^{\frac{1}{\sqrt{d_c}} \tilde{\boldsymbol{x}}_i^T \tilde{\boldsymbol{c}}_j}}{\sum_{j=1}^{hw} e^{\frac{1}{\sqrt{d_c}} \tilde{\boldsymbol{x}}_i^T \tilde{\boldsymbol{c}}_j}}; \quad \boldsymbol{a}_i = \sum_{j=1}^{hw} \alpha_{i,j} \tilde{\boldsymbol{c}}_j \tag{1}$$

$$\tilde{\boldsymbol{x}}_i^{'} = \mathrm{LN}\left(\tilde{\boldsymbol{x}}_i + \mathrm{Dropout}(\boldsymbol{a}_i)\right) \tag{2}$$

where LN stands for LayerNorm (Ba, Kiros, and Hinton 2016) layer; $\tilde{\boldsymbol{c}}_j \in \mathbb{R}^{d_c}$ is the jth feature of $\tilde{\boldsymbol{C}}$; $\boldsymbol{a}_i \in \mathbb{R}^{d_c}$ is the aggregated feature by self-attention mechanism; $\tilde{\boldsymbol{x}}_i^{'} \in \mathbb{R}^{d_c}$ is the primary contextual feature for $\tilde{\boldsymbol{x}}_i$.

Finally, we perform an FFN refining to further refine the encoded representation. This step contains a LN layer on the feed-forwarded features and a residual connection. The above refining architecture can be written as

$$\mathrm{FFN}(\tilde{\boldsymbol{x}}_i^{'}) = \mathrm{Linear}\left(\mathrm{Dropout}\left(\mathrm{ReLU}\left(\mathrm{Linear}\left(\tilde{\boldsymbol{x}}_i^{'}\right)\right)\right)\right) \tag{3}$$

$$\tilde{\boldsymbol{x}}_i^{''} = \mathrm{LN}\left(\tilde{\boldsymbol{x}}_i^{'} + \mathrm{Dropout}\left(\mathrm{FFN}\left(\tilde{\boldsymbol{x}}_i^{'}\right)\right)\right) \tag{4}$$

where $\tilde{\boldsymbol{x}}_i^{''} \in \mathbb{R}^{d_c}$ is the FFN refined contextual feature.

This module can be easily extended to a multi-head manner. To enhance its representation power, we introduce residual connection and use concatenation as a fusion method. In summary, the output of TCE module can be formulated as

$$\boldsymbol{y}_i = \boldsymbol{x}_i \| \{ \|_{n=1}^{n_t} \mathrm{tce}(\boldsymbol{x}_i, \boldsymbol{C})_n \} \tag{5}$$

where $\|$ denotes concatenation on the channel axis, $\mathrm{tce}(\boldsymbol{x}_i, \boldsymbol{C})$ denotes one head of TCE module and n_t denotes the number of TCE heads. Each head has independent parameters.

Unilateral Connection For the base network of the pose branch, it takes in the cropped and resized person images, which produces very fine-grained pose embedding as individual features but drops the global contextual information. To plug TCE module into the pose branch as well, we establish a unilateral connection to create inter-branch feature

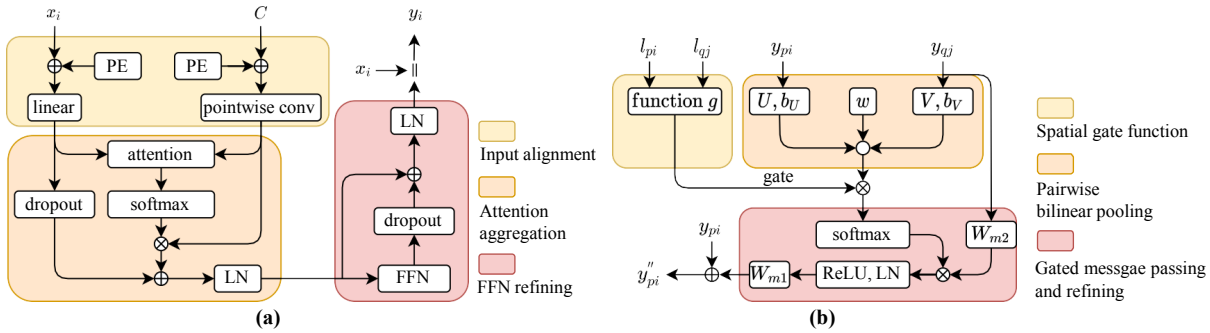


Figure 3: (a) A *single-head architecture* of TCE module which encodes C to x_i. (b) A *single-head architecture* of STBiP which mines the relationship of all y_{qj} to the given y_{pi}. The softmax function is used across all y_{qj} in (b). For both, $\oplus$ denotes element-wise summation, $\otimes$ denotes matrix multiplication, $\|$ denotes concatenation on the channel axis and $\circ$ denotes Hadamard product.

flow. In practice, the global branch feeds the pose branch with the feature map $\boldsymbol{C}$ generated from its base network, as shown in Figure 2. Combined with the pose branch's individual features can the input requirements of the TCE module be met.

Positional Encoding The positional information is important because it exposes positional priors to the encoder and the followed reasoning method. Thus we add positional encoding (Vaswani et al. 2017) to both branches in TCE modules. In practice, for a given individual feature, we apply $\sin$ and $\cos$ function to encode the center coordinates (l_w, l_h) of its original bounding box. Half of the dimension is encoded with l_w and the second half with l_h. It can be formulated as

$$\begin{aligned} \mathrm{PE}_{(l,2k)} &= \sin\left(\frac{l}{10000^{2k/d_e}}\right) \\ \mathrm{PE}_{(l,2k+1)} &= \cos\left(\frac{l}{10000^{2k/d_e}}\right) \end{aligned} \tag{6}$$

where l denotes coordinate l_w or l_h. If $l = l_w$, then dimension $k \in \{0, 1, ..., \frac{d_e}{4} - 1\}$. If $l = l_h$, then dimension $k \in \{\frac{d_e}{4}, \frac{d_e}{4} + 1, ..., \frac{d_e}{2} - 1\}$.

For the global feature map C, we encode the coordinates on the feature map multiplied by the output stride using the same method. The multiplication of output stride is to match the order of magnitude with the positional encoding of individual features.

Spatial-Temporal Bilinear Pooling

In this subsection, we introduce the construction of our spatial-temporal graph, the STBiP module and the final global attribute pooling. A neat illustration of STBiP module is shown in Figure 3(b). STBiP contains three parts: spatial gate function, pairwise bilinear pooling, gated message passing and refining.

To model the relationship, a spatial-temporal graph $G = (\boldsymbol{Y}_K, \boldsymbol{R})$ is firstly constructed from context encoded features at every vertex $\boldsymbol{Y}_K = \{\boldsymbol{y}_{ti} | t = 1, ..., K; i = 1, ..., N\}$ and their pairwise relationship $\boldsymbol{R} = \{r_{pi,qj} | p, q = 1, ..., K; i, j = 1, ..., N\}$. $\boldsymbol{Y}_K$ is the temporal extended $\boldsymbol{Y}$, which makes the graph include all the people from the sampled K frames.

In order to mine the relationship between individuals, we ought to consider the spatial relationship and visual embedding relationship. In this work, we model them separately and fusion their results by multiplication. If we denote the center coordinates of the bounding box for $\boldsymbol{y}_{pi}$ and $\boldsymbol{y}_{qj}$ to be $\boldsymbol{l}_{pi}$ and $\boldsymbol{l}_{qj}$, their pairwise relationship can be neatly denoted as

$$r_{pi,qj} = F(\boldsymbol{l}_{pi}, \boldsymbol{l}_{qj}, \boldsymbol{y}_{pi}, \boldsymbol{y}_{qj}) = g(\boldsymbol{l}_{pi}, \boldsymbol{l}_{qj}) f(\boldsymbol{y}_{pi}, \boldsymbol{y}_{qj}) \tag{7}$$

Spatial Gate Function For the modeling of function $g(\boldsymbol{l}_{pi}, \boldsymbol{l}_{qj})$, we consider pairwise positions as a gate for message passing. It facilitates message passing operation to aggregate spatially adjacent features. In practice, we formulate our gate function as

$$g(\boldsymbol{l}_{pi}, \boldsymbol{l}_{qj}) = \begin{cases} 1, & \text{if } \mathrm{dist}(\boldsymbol{l}_{pi}, \boldsymbol{l}_{qj}) < \theta \\ -\infty, & \text{otherwise} \end{cases} \tag{8}$$

where $\mathrm{dist}(\boldsymbol{l}_{pi}, \boldsymbol{l}_{qj})$ stands for the Euclidean distance between $\boldsymbol{l}_{pi}$ and $\boldsymbol{l}_{qj}$; θ is the preset threshold.

Pairwise Bilinear Pooling To exhaustively explore the visual embedding relationship via function $f(\boldsymbol{y}_{pi}, \boldsymbol{y}_{qj})$, we resort to bilinear pooling (Kim et al. 2016) which consider every feature's pairwise interaction by quadratic expansion. $\boldsymbol{y}_{pi}$ and $\boldsymbol{y}_{qj}$ that output from the TCE module contain contextual information and original individual appearance information. Thus, interpersonal appearance-appearance, context-context and appearance-context relation can be thoroughly explored via bilinear pooling.

In practice, we consider pairwise interaction by following function

$$\begin{aligned} f(\boldsymbol{y}_{pi}, \boldsymbol{y}_{qj}) &= {\boldsymbol{y}_{pi}}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{y}_{qj} = {\boldsymbol{y}_{pi}}^{\mathrm{T}} \boldsymbol{U} \boldsymbol{V}^{\mathrm{T}} \boldsymbol{y}_{qj} \\ &= \mathbf{1}^{\mathrm{T}} (\boldsymbol{U}^{\mathrm{T}} \boldsymbol{y}_{pi} \circ \boldsymbol{V}^{\mathrm{T}} \boldsymbol{y}_{qj}) \end{aligned} \tag{9}$$

where $\boldsymbol{y}_{pi}, \boldsymbol{y}_{qj} \in \mathbb{R}^{d_y}$; $\boldsymbol{W} \in \mathbb{R}^{d_y \times d_y}$ is the learned weight matrix; $\mathbf{1}$ is column vector of ones; $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{d_y \times d_w}$ are learned linear projections and $\circ$ stands for Hadamard product. Different from (Kim et al. 2016), we don't expect the

model to be low-rank because the number of pairwise relationship ($K^2 \times N^2$ in one graph) is a small quantity in our problem setting. Instead, we use high rank pooling to discover abundant relation. We then extend sum aggregation to weighted aggregation and follow (Kim et al. 2016) to add bias term to linear projections.

$$f(\boldsymbol{y}_{pi}, \boldsymbol{y}_{qj}) = \boldsymbol{w}^{\mathrm{T}}((\boldsymbol{U}^{\mathrm{T}}\boldsymbol{y}_{pi} + \boldsymbol{b}_U) \circ (\boldsymbol{V}^{\mathrm{T}}\boldsymbol{y}_{qj} + \boldsymbol{b}_V)) \quad (10)$$

where $\boldsymbol{b}_U, \boldsymbol{b}_V \in \mathbb{R}^{d_w}$ are the learned bias and $\boldsymbol{w} \in \mathbb{R}^{d_w}$ is the learned weight vector.

Gated Message Passing and Refining As graph convolution network demonstrates its effectiveness in reasoning problem, we perform similar message passing operations on our constructed graph. In detail, for a given $\boldsymbol{y}_{pi}$, we aggregate information from all vertices in the graph according to their pairwise relationship $r_{pi,qj}$. After message passing, we employ a refining operation containing a ReLU activation, an LN layer and a fully-connected layer. It refines the aggregated information in case it contains redundant information (Wang et al. 2018; Lin et al. 2020). Based on this, the updated feature of $\boldsymbol{y}_{pi}$ can be written as

$$m_{pi,qj} = \frac{e^{g(\boldsymbol{l}_{pi}, \boldsymbol{l}_{qj}) f(\boldsymbol{y}_{pi}, \boldsymbol{y}_{qj})}}{\sum_{q=1}^{K}\sum_{j=1}^{N} e^{g(\boldsymbol{l}_{pi}, \boldsymbol{l}_{qj}) f(\boldsymbol{y}_{pi}, \boldsymbol{y}_{qj})}}$$
$$\boldsymbol{y}_{pi}^{'} = \boldsymbol{W}_{m1}\mathrm{ReLU}\left(\mathrm{LN}\left(\sum_{q=1}^{K}\sum_{j=1}^{N} m_{pi,qj}\boldsymbol{W}_{m2}\boldsymbol{y}_{qj}\right)\right) \quad (11)$$

where $\boldsymbol{y}_{pi}^{'} \in \mathbb{R}^{d_y}$ is the refined feature; $\boldsymbol{W}_{m1}, \boldsymbol{W}_{m2} \in \mathbb{R}^{d_y \times d_y}$ are learned parameters.

We can extend the gated message passing result to a multi-head manner. In addition, residual connection is introduced to enhance its representational ability. Then the multi-head results and residual feature will be fused by summation. The final individual representation of the given $\boldsymbol{y}_{pi}$ can be denoted as

$$\boldsymbol{y}_{pi}^{''} = \boldsymbol{y}_{pi} + \sum_{n=1}^{n_s} \mathrm{sp}(\boldsymbol{y}_{pi}, \boldsymbol{Y}_K)_n \quad (12)$$

where $\boldsymbol{y}_{pi}^{''} \in \mathbb{R}^{d_y}$ is the output of STBiP module; $\mathrm{sp}(\boldsymbol{y}_{pi}, \boldsymbol{Y}_K)$ denotes one head of STBiP modules; n_s denotes the number of STBiP heads. Each head has independent parameters.

Finally, we perform a global attribute pooling to obtain the final activity representation. Specifically, features belong to the same individual are average pooled on the temporal axis and then max pooled among different individuals to obtain the activity representation $\boldsymbol{z} \in \mathbb{R}^{d_y}$.

Branch Fusion and Training Loss

The above statement solves the problem of inferring the group activity on one branch. As mentioned, two branches hold complementary information and previous work has proved it effective (Simonyan and Zisserman 2014; Azar et al. 2019; Gavrilyuk et al. 2020) to fuse the independently inferred feature for the same objective. In our problem, We mainly study three kinds of fusion strategy: (1) The first is fusion before the TCE module. Concatenation is used because element-wise summation does not perform well on fusing features from different backbones (Gavrilyuk et al. 2020). (2) The second is fusion before the STBiP module and concatenation is employed. (3) The third is to train two branches separately and fuse the softmax scores of two branches by weighted summation when testing. The fusion position is marked in Figure 2. Later fusion allows two branches to reason on their own features more independently. Their comparisons will be shown in the experiment.

To train the model in an end-to-end manner, we apply two standard cross-entropy loss

$$\mathcal{L} = \mathcal{L}_g(z_G, \hat{z}_G) + \lambda\mathcal{L}_a(z_I, \hat{z}_I) \quad (13)$$

where $\mathcal{L}_g$ and $\mathcal{L}_a$ are cross-entropy losses for group activity recognition and individuals action recognition respectively; z_G and z_I are ground truth labels for group activity and individual action; $\hat{z}_G$ and $\hat{z}_I$ are model predictions. λ serves as a hyper-parameter to balance two losses.

Experiments

Datasets and Implementation Details

Datasets There are two frequently adopted datasets named the Volleyball dataset (VD) and the Collective Activity dataset (CAD). The Volleyball dataset (Ibrahim et al. 2016) gathers from 55 video recordings of volleyball games, which are clipped and split into 3493 training clips and 1337 testing clips. The center frame of each clip is annotated with bounding box coordinates for all individual players and their action labels (*i.e.* blocking, digging, falling, jumping, moving, setting, spiking, standing and waiting). Each clip is annotated with one group activity label out of eight labels (*i.e.* right set, right spike, right pass, right winpoint, left set, left spike, left pass and left winpoint). For the unannotated frames, we use the tracklets generated by (Bagautdinov et al. 2017). Two metrics are employed to evaluate on this dataset, which are Multi-class Classification Accuracy (MCA, %) and Mean Per Class Accuracy (MPCA, %) following (Shu, Todorovic, and Zhu 2017).

The Collective Activity dataset (Choi, Shahid, and Savarese 2009) composes of 44 clips containing frames ranging from 194 to 1814. The train set and test set split follows (Qi et al. 2018). The center frame of every ten frames is annotated with bounding box coordinates of all individuals and their action labels (*i.e.* NA, crossing, waiting, queueing, walking and talking). Every ten frames are given one group activity label out of five (*i.e.* crossing, waiting, queueing, walking and talking). We merge the classes "walking" and "crossing" as "moving" and report the MPCA (%) to evaluate the performance following (Wang, Ni, and Yang 2017).

Implementation Details As other methods have done, we resize images from the VD to $H \times W = 720 \times 1280$ and images from the CAD to $H \times W = 480 \times 720$. For the pose branch, we crop the person bounding boxes from the sampled images and resize them to a fixed size of 256×192. For fair comparison, we use $T = 10$ frames for every clip, with 5 frames before the center frame and 4 frames after. We divide the clips into $K = 3$ temporal segments for two

TCE for PB		MCA	TCE for GB		MCA
# heads	PE		# heads	PE	
-	-	89.7	-	-	89.8
1	✓	91.4	2	✓	89.9
2	✓	91.8	3	✓	91.6
3	✓	**92.0**	4	✓	**91.9**
3		90.7	4		91.6
4	✓	91.7	5	✓	90.8

Table 1: Ablation study on the TCE module. The first row *without* any TCE is the *base model*. 'PE' is short for Positional Encoding, 'PB' for Pose Branch and 'GB' for Global Branch.

STBiP for PB		MCA	STBiP for GB		MCA
# heads	w/ TCE		# heads	w/ TCE	
-	✓	92.0	-	✓	91.9
2	✓	92.5	2	✓	92.8
4	✓	**92.9**	4	✓	92.9
4		90.9	8	✓	**93.3**
8	✓	92.5	8		91.3
16	✓	92.0	16	✓	92.8

Table 2: Ablation study on the STBiP module. Note that the first row of results are *TCE modules with best reported MCA* in Table 1, without any STBiP. The following STBiP modules are appended to the chosen TCE module if w/ TCE is marked with ✓. Otherwise, they are appended to the base network.

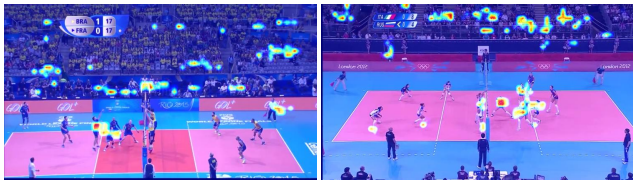
branches. We set dimension of individual feature to $d_e = 1024$ for both HRNet-w32 and Inception-v3/VGG16 base networks. For TCE module, we set the encoding dimension to $d_c = 128$ and dropout ratio to 0.1. For STBiP module, we set the threshold for the gate function to $\theta = 0.3H$ and set $d_w = d_y$. For the training loss, we use $\lambda = 1$ for all experiments. For the training of VD, we adopt the Adam optimizer (Kingma and Ba 2014) with its hyper-parameter fixed to $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We use a mini-batch size of 6 and train the network in 160 epochs with an initial learning rate 10^{-4}, which decreases by a factor of 2 every 40 epochs. For the training of CAD, we use Adam with same hyper-parameters and a fixed learning rate of 10^{-4}.

Ablation Study

In this subsection, we perform a series of ablation studies on the VD for two branches respectively, aiming to examine the effectiveness and respective contributions of proposed modules. We specifically adopt HRNet-w32 for the pose branch and Inception-v3 for the global branch.

TCE Module We start our experiments by appending our TCE module to the base network. We mainly experiment with the number of layers and effectiveness of positional encoding. The results are shown in Tabel 1.

First, we append the TCE module (default with positional encoding) with an increasing number of heads directly to the base model. We can conclude that (1) the TCE module with too few heads fails to encode sufficient global contextual information to individual features, while too many heads cause redundancy. Then we remove the positional encoding of the best performing model and report their results. Two conclusions can be drawn that (2) positional encoding helps encode person-specific context and achieves improvement for both branches; (3) positional encoding helps encode more informative position priors for the pose branch than the global branch. We speculate it's due to the underlying implication for positional information contained in RoIAlign features.



Left spike Right spike

Figure 4: Visualization of contextual attention map. Better view in digital version.

To observe what contextual information we aggregate using TCE, we visualize the summation of N people's attention weight aggregated from one sampled image by a 4-head TCE on the global branch. The obtained contextual attention map is shown in Figure 4. The attention map reveals its focus on two elements which contribute to a better recognition: (1) The person who performs the significant action, like the person who perform the 'spiking' action in both images. (Only key players have high attention weights because we use residual connection in TCE, which makes players who need to be reemphasized have high weights.) (2) The spectator's reaction and line judges which are contextual information that paid little attention in previous methods. Their connections to any player on the field are expected to be caught by TCE. These two elements combined result in a good performance.

STBiP Module To study the effectiveness of STBiP modules, we firstly append STBiP modules with an increasing number of heads to the best TCE model (3-head TCE for the pose branch and 4-head TCE for the global branch). The result is shown in Table 2. We can conclude that (1) with the addition of STBiP modules to TCE modules, both branches can achieve higher MCA scores, which demonstrate the ability of STBiP to mine the underlying relationship for context encoded features; (2) with too few heads added, the reasoning capacity is insufficient to mine pairwise relationship and too many cause redundancy. Then we remove TCE modules of the best performing model (*i.e.* append STBiP to the base network) and report their results. We can conclude that (3) STBiP modules facilitate to explore relationships between original individual features as well.

Fusion We study the performance of three strategies. The results are reported in Table 4. For fusion 1, two branches share 4-head TCE and 8-head STBiP. For fusion 2, two branches possess independent 4-head TCE and shared 8-

Method	Backbone	MCA-V	MPCA-V	MPCA-C
CRM (Azar et al. 2019)	I3D	92.1 (93.0‡)	-	- (94.2‡)
Actor Transformer (Gavrilyuk et al. 2020)	I3D	91.4 (93.0‡)	-	-
	HRNet	92.3	-	-
	I3D + HRNet	**93.5** (94.4‡)	- (94.2‡)	-
SBGAR (Li and Choo Chuah 2017)	Inception-v3	38.7 (66.9‡)	- (67.6‡)	- (89.9‡)
SSU (Bagautdinov et al. 2017)	Inception-v3	89.9	-	-
ARG (Wu et al. 2019)	Inception-v3	92.5	-	-
	VGG16	91.9	-	-
CERN-2 (Shu, Todorovic, and Zhu 2017)	VGG16	83.3	83.6	88.3
SPA+KD (Tang et al. 2018)	VGG16	89.3 (90.7‡)	89.0 (90.0‡)	92.5 (95.7‡)
stagNet (Qi et al. 2018)	VGG16	89.3	-	89.1
PRL(Hu et al. 2020)	VGG16	91.4	**91.8**	**93.8**
Ours-TCE+STBiP	Inception-v3	93.3	93.4	95.1
	HRNet*	92.9	92.9	95.0
	Inception-v3 + HRNet*	94.1	94.3	95.4
	VGG16	94.1	94.4	95.4
	HRNet†	92.9	93.2	94.9
	VGG16 + HRNet†	**94.7**	**95.0**	**96.4**

Table 3: Comparisons with the state-of-the-art methods. *Suffix '-V'* denotes results for the VD and *suffix '-C'* for the CAD. Superscript * denotes unilateral connection from Inception-v3. Superscript † denotes unilateral connection from VGG16. Superscript ‡ denotes results with additional optical flow inputs.

Fusion Position	Fusion 1	Fusion 2	Fusion 3
MCA	91.8	92.3	**93.8**

Table 4: Branch fusion results. The exact fusion position can be found in Figure 2.

head STBiP. For fusion 3, two branches possess independent 4-head TCE and independent 8-head STBiP (Note that the pose branch with 4-head TCE and 8-head STBiP achieves 92.5%, while the global branch with the same structure achieves 93.3%). For fusion 3, we set the weight for the pose branch to be $\frac{1}{3}$ and the global branch $\frac{2}{3}$ following (Simonyan and Zisserman 2014). It can be inferred that early fusion results in over-fitting. Also, too early fusion causes the model to be worse than any one-branch model. To reason more independently produces better results.

Comparison with the State-of-the-Art

We compare our methods with other state-of-the-art methods on the VD and the CAD in Table 3. We list results of global branch models (Inception-v3/VGG16), pose branch models (HRNet-w32) and two-branch models using the third fusion strategy. On the CAD, both branches are equipped with 1-head TCE and 8-head STBiP. On the VD, the pose branch is equipped with 3-head TCE and 4-head STBiP, while the global branch is equipped with 4-head TCE and 8-head STBiP.

For VD, our method on the global branch outperforms other methods with the same backbone by at least 0.8% MCA for Inception-v3 and 2.2% MCA for VGG16. By adding global features, our method on pose branch outperforms (Gavrilyuk et al. 2020) with the same HRNet by 0.6%. Our two-branch model even surpasses methods using additional optical flow inputs, which demonstrates the global-dependent context is informative. The confusion matrix of VGG16+HRNet model illustrated in the Appendices indicates our model distinguish well between activities performed by the right team and the left team due to the modeling of the spatial information. TCE can also help to distinguish 'winpoint' well because they have relatively distinct spatial layouts from other activities. Most failed cases mistake an activity performed by one team for another activity performed by the same team. For example, some 'right set' ('left set') activities are mistaken for 'right pass' ('left pass'). It's because they share similar spatial-temporal information and contextual information.

For CAD, our method outperforms other methods using the same VGG16 by at least 1.6%. Our two-branch method even surpass other methods using additional optical flow inputs by 0.7%. Comparing methods adopting 3D convolution like I3D, our model shows its superiority and outperforms it by 2.2%. The confusion matrix of VGG16+HRNet model illustrated in the Appendices indicates that the most failed cases mistake 'waiting' to 'moving'. It may due to the fact that both classes have similar visual context which is hard to distinguish, and the temporal dynamics of video clips isn't enough to catch two classes' differences.

Conclusion

In this work, we manage to bridge the gap between group activity recognition and visual context. In practice, we present Transformer based Context Encoding (TCE) to explore person-specific visual context and Spatial-Temporal Bilinear Pooling (STBiP) to mine abundant pairwise relationship. Experiments have demonstrated the effectiveness and outstanding performance of incorporating visual context to group activity recognition.

Acknowledgements

We would like to thank the anonymous reviewers and Jiayang Ren for their valuable feedback. This work was supported by the National Science Foundation China grant No. U1609213.

Appendices

Figure 5: Confusion matrices for VGG16+HRNet model on Volleyball dataset.

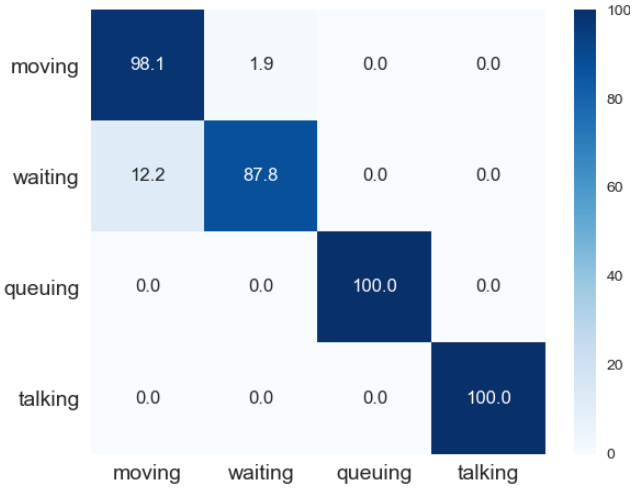


Figure 6: Confusion matrices for VGG16+HRNet model on Collective Activity dataset.

References

Amer, M. R.; Lei, P.; and Todorovic, S. 2014. Hirf: Hierarchical random field for collective activity recognition in videos. In *European Conference on Computer Vision*, 572–585. Springer.

Azar, S. M.; Atigh, M. G.; Nickabadi, A.; and Alahi, A. 2019. Convolutional Relational Machine for Group Activity Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7892–7901.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* .

Bagautdinov, T.; Alahi, A.; Fleuret, F.; Fua, P.; and Savarese, S. 2017. Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4315–4324.

Cao, Y.; Xu, J.; Lin, S.; Wei, F.; and Hu, H. 2019. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0.

Chao, Y.-W.; Wang, Z.; He, Y.; Wang, J.; and Deng, J. 2015. Hico: A benchmark for recognizing human-object interactions in images. In *Proceedings of the IEEE International Conference on Computer Vision*, 1017–1025.

Choi, W.; Shahid, K.; and Savarese, S. 2009. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 1282–1289. IEEE.

Choi, W.; Shahid, K.; and Savarese, S. 2011. Learning context for collective activity recognition. In *CVPR 2011*, 3273–3280. IEEE.

Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847* .

Gao, Y.; Beijbom, O.; Zhang, N.; and Darrell, T. 2016. Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 317–326.

Gavrilyuk, K.; Sanford, R.; Javan, M.; and Snoek, C. G. 2020. Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 839–848.

Girdhar, R.; Carreira, J.; Doersch, C.; and Zisserman, A. 2019. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 244–253.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* .

Hu, G.; Cui, B.; He, Y.; and Yu, S. 2020. Progressive relation learning for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 980–989.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2019. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 603–612.

Ibrahim, M. S.; and Mori, G. 2018. Hierarchical relational networks for group activity recognition and retrieval. In *Proceedings of the European conference on computer vision (ECCV)*, 721–736.

Ibrahim, M. S.; Muralidharan, S.; Deng, Z.; Vahdat, A.; and Mori, G. 2016. A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1971–1980.

Kim, J.-H.; On, K.-W.; Lim, W.; Kim, J.; Ha, J.-W.; and Zhang, B.-T. 2016. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325* .

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kong, S.; and Fowlkes, C. 2017. Low-rank bilinear pooling for fine-grained classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 365–374.

Lan, T.; Sigal, L.; and Mori, G. 2012. Social roles in hierarchical models for human activity recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 1354–1361. IEEE.

Lan, T.; Wang, Y.; Yang, W.; Robinovitch, S. N.; and Mori, G. 2011. Discriminative latent models for recognizing contextual group activities. *IEEE transactions on pattern analysis and machine intelligence* 34(8): 1549–1562.

Li, X.; and Choo Chuah, M. 2017. Sbgar: Semantics based group activity recognition. In *Proceedings of the IEEE international conference on computer vision*, 2876–2885.

Lin, T.-Y.; RoyChowdhury, A.; and Maji, S. 2015. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, 1449–1457.

Lin, X.; Ding, C.; Zeng, J.; and Tao, D. 2020. GPS-Net: Graph Property Sensing Network for Scene Graph Generation. *arXiv preprint arXiv:2003.12962* .

Pearl, J.; Glymour, M.; and Jewell, N. P. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

Qi, M.; Qin, J.; Li, A.; Wang, Y.; Luo, J.; and Van Gool, L. 2018. stagnet: An attentive semantic RNN for group activity recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 101–117.

Shu, T.; Todorovic, S.; and Zhu, S.-C. 2017. CERN: confidence-energy recurrent network for group activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5523–5531.

Simonyan, K.; and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, 568–576.

Tang, K.; Zhang, H.; Wu, B.; Luo, W.; and Liu, W. 2019. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6619–6628.

Tang, Y.; Wang, Z.; Li, P.; Lu, J.; Yang, M.; and Zhou, J. 2018. Mining semantics-preserving attention for group activity recognition. In *Proceedings of the 26th ACM international conference on Multimedia*, 1283–1291.

Tenenbaum, J. B.; and Freeman, W. T. 2000. Separating style and content with bilinear models. *Neural computation* 12(6): 1247–1283.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, M.; Ni, B.; and Yang, X. 2017. Recurrent modeling of interaction context for collective activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3048–3056.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.

Wei, X.; Zhang, Y.; Gong, Y.; Zhang, J.; and Zheng, N. 2018. Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 355–370.

Woo, S.; Park, J.; Lee, J.-Y.; and So Kweon, I. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, 3–19.

Wu, J.; Wang, L.; Wang, L.; Guo, J.; and Wu, G. 2019. Learning actor relation graphs for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9964–9974.

Yan, R.; Tang, J.; Shu, X.; Li, Z.; and Tian, Q. 2018. Participation-contributed temporal dynamic model for group activity recognition. In *Proceedings of the 26th ACM international conference on Multimedia*, 1292–1300.

Yao, B.; and Fei-Fei, L. 2010. Modeling mutual context of object and human pose in human-object interaction activities. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 17–24. IEEE.

Yu, C.; Zhao, X.; Zheng, Q.; Zhang, P.; and You, X. 2018a. Hierarchical bilinear pooling for fine-grained visual recognition. In *Proceedings of the European conference on computer vision (ECCV)*, 574–589.

Yu, Z.; Yu, J.; Xiang, C.; Fan, J.; and Tao, D. 2018b. Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE transactions on neural networks and learning systems* 29(12): 5947–5959.

Zellers, R.; Yatskar, M.; Thomson, S.; and Choi, Y. 2018. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5831–5840.

Zhang, Y.; Tang, S.; Muandet, K.; Jarvers, C.; and Neumann, H. 2019. Local temporal bilinear pooling for fine-grained action parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12005–12015.