Consistency-Aware Graph Network for Human Interaction Understanding

Zhenhua Wang[†], Jiajun Meng[†], Dongyan Guo[†], Jianhua Zhang[♯*], Javen Qinfeng Shi[‡], Shengyong Chen[♯]

[†]Zhejiang University of Technology, [♯]Tianjin University of Technology, [‡]The University of Adelaide

{zhhwang, guodongyan, 1112012021}@zjut.edu.cn, {zjh, sy}@ieee.org, javen.shi@adelaide.edu.au

Abstract

Compared with the progress made on human activity classification, much less success has been achieved on human interaction understanding (HIU). Apart from the latter task is much more challenging, the main cause is that recent approaches learn human interactive relations via shallow graphical models, which is inadequate to model complicated human interactions. In this paper, we propose a consistency-aware graph network, which combines the representative ability of graph network and the consistency-aware reasoning to facilitate HIU. Our network consists of three components, a backbone CNN to extract image features, a factor graph network to learn third-order interactive relations among participants, and a consistency-aware reasoning module to enforce labeling and grouping consistencies. Our key observation is that the consistency-aware-reasoning bias for HIU can be embedded into an energy, minimizing which delivers consistent predictions. An efficient mean-field inference algorithm is proposed, such that all modules of our network could be trained jointly in an end-to-end manner. Experimental results show that our approach achieves leading performance on three benchmarks. Code is available at https://git.io/CAGNet.

1. Introduction

Analyzing human activities in natural scenes is a fundamental task to many potential applications like video surveillance [35], key-event retrieval [12], social behavior interpretation [2] and sports analysis [28]. Abundant techniques have been developed for human activity recognition (HAR, where the goal is to assign an activity label to each image or video) [7, 25, 16, 33, 21, 43, 43, 27], which have gained impressive progress on recognition accuracy. However, the task of human interaction understanding (HIU) is much less successful mainly because current methods learn
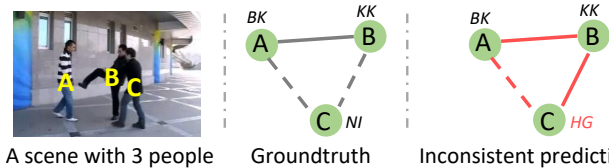

Figure 1. The graphical representation of HIU in a scene with three people. We decompose HIU into two sub-tasks: recognizing person-wise actions (as denoted by the node labels, with KK, BK, HG, NI indicating *kick, be-kicked, hug, no-interaction*, respectively) and predicting if any pair of people are interacting (solid edges) or not (dashed edges). Applying consistency-unaware models to such cases can lead to inconsistent predictions as highlighted by the red edges and labels (see Section 1 for details). We address such issue by presenting a consistency-aware graph network with two types of third-order dependencies incorporated.

human interactive relations via shallow graphical representations [42, 41, 40, 25, 7, 46], which is inadequate to model complicated human interactions, *e.g.* fighting and chasing as two concurrent activities happening in the same scene.

As commonly done in literature [25, 42, 40, 41], we decompose HIU into two sub-tasks illustrated by Figure 1 middle: 1) The individual action prediction task assigning each participant an action label; 2) The pairwise interactive prediction task determining if any pair of participants are interacting or not. Solving the two sub-tasks provides a way to disentangle concurrent human activities with multiple participants, as well as a comprehensive understanding to surveillance scenes. Also note that natural scene may include either/both *physical* interactions (*e.g.*, handshaking, hugging and punching) and *non-physical* interactions (*e.g.*, chasing, talking and queuing). People performing such interactions naturally form a *group*. In this sense, the *action* of a person can be defined as "what is she/he doing under such interactive circumstances?", which is challenging to answer if only considering the local representations. Though HIU performance had been lifted a lot by a conjunctive usage of deep features and rich contextual information, there still exist two main challenges. Since most existing work-

---

*[*]Jianhua Zhang {zjh@ieee.org} is the corresponding author.

s perform piece-wise learning of deep feature representations and contextual models [42, 40], the first challenge is how to learn deep features and contextual relations jointly. The second challenge is how to ensure prediction consistency for the two sub-tasks of HIU. In this paper, we tackle two types of prediction inconsistencies illustrated by Figure 1 right. The first type is called *the labeling inconsistency*, *e.g.* the action label of B (*i.e. kick*) is inconsistent with the action label of C (*i.e. hug*) as they are interacting (denoted by a solid edge). The second type is called *the grouping inconsistency*, under the assumption that interacting people belong to the same group while non-interacting ones belong to separate groups. Consequently, the prediction *(A, C) are not interacting* (denoted by the dashed edge) is inconsistent with the prediction that *(A, B) are interacting* and *(B, C) are interacting as well*. To address the two challenges, we present a consistency-aware graph network (CAGNet), which consists of a backbone CNN to extract image features, a third-order graph network (TOGN) to learn human interactive context, and a consistency-aware reasoning (CAR) module to improve the consistency within action and interaction predictions. All components of CAGNet could be trained jointly and efficiently with GPU acceleration. We empirically validate the effectiveness of these three components on three benchmarks of human interaction understanding.

Our contributions are of three aspects. First, we propose a TOGN for HIU, which is more powerful than the widely adopted pairwise graph networks in terms of representing the interactive relations among people. Second, we present an efficient CAR module to resolve the labeling and grouping inconsistencies within HIU predictions. Third, our proposed CAGNet, which takes the TOGN and CAR modules as its building-blocks, outperforms the state-of-the-art results by salient margins on three evaluated benchmarks.

2. Related Work

Relevant Tasks HIU is closely related to HAR. Numerous works on HAR have been proposed [16, 33, 38, 17, 39, 5, 23, 44] in order to extract powerful feature representations of human motions. These approaches are also applicable to the recognition of collective activities wherein a number of participants perform a group activity. Nevertheless, an increasing number of works justify the importance of modeling the spatio-temporal correlations among action variables of different people [7, 22, 6, 8, 2, 31, 15, 28, 43, 27]. Early works in this vein explore conditional random fields (CRFs) [7, 22, 6], while recent efforts contribute most on the joint learning of image features and human relations with RNN [8, 2, 31, 28, 32] or deep graphical models [15, 43, 27]. These approaches are designed to predict each input an activity category, leaving the HIU task rather unsolved. Another relevant task is the recognition of human-object interactions [45] [29], which shares similar problem structure with HIU while it focuses on reasoning the interactive relations of people and objects instead.

Human Interaction Understanding To understand human interactions, abundant conditional random field (CRF)-based models have been proposed [46, 20, 21, 26, 25, 41, 40, 42] to model the interactive relations in both spatial and temporal domains. The main drawback is that these CRFs are of shallow graphical-representations, which is neither effective in terms of learning complicated human interactions nor efficient in solving the associated maximum a posteriori inference [42]. Moreover, they perform deep feature learning and relational reasoning separately, which typically results in sub-optimal solutions. Our CAGNet addresses these issues by presenting a deep graph network, which synthesizes the feature-learning ability of CNNs and the contextual-modeling power of graphical representation.

Graph Networks have become popular choices to many tasks involving modeling and reasoning relations among components within a system [4, 18, 48, 47, 11, 49]. They share the computational efficiency of deep architectures while are more powerful and flexible in terms of modeling relations in non-grid structures, for instance, the correspondences between two sets of points in a matching problem [48], the correlations between query and support pixels in one-shot semantic segmentation [47], human gaze communication [11], and the inter-person relations for collective activity classification [43]. As these networks operate on a graph structure, they are only able to capture pairwise relations. Very recently, work [49] proposes a factor graph neural network (FGNN) that enables the incorporation of high-order dependencies. Inspired by this, we propose the TOGN which shares the same feature-updating mechanism (detailed in Section 3) with FGNN but uses customized third-order factor graphs to model the interactive relations in human activities.

Deep Logical Reasoning As a way to higher-level intelligence, logical reasoning has seen a renaissance in very recent years [9, 10]. Since traditional logical reasoning has relied on methods and tools which are very different from deep learning models, such as Prolog language, SMT solvers and discrete algorithms, a key problem is how to bridge logic and deep models effectively and efficiently. Recent works viewed graph networks as a general tool to make such a connection. For example, [3, 4] take graph networks to incorporate explicitly logic reasoning bias, [24] builds a neuro-symbolic reasoning module to connect scene representation and symbolic programs, and work [1] introduces a differentiable first-order logic formalism for visual question answering. Like [3, 4], our proposed CAR module explicitly incorporates the consistency-aware-reasoning bias of HIU as well, but accomplishes the reasoning differently via solving a particular energy minimization task.
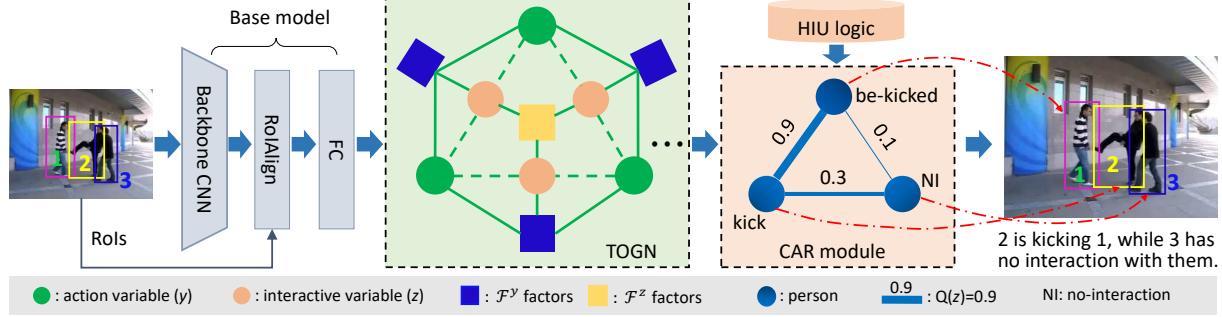
Figure 2. An overview of the proposed CAGNet, which includes a base-model, a TOGN and a CAR module. The TOGN is designed to incorporate two types of factors to learn human-interaction-context, as indicated by yellow and blue nodes. Leveraging the consistency-aware-reasoning bias of HIU, our CAR block fixes possible inconsistent predictions and improves the interpretability of HIU. Here "KK", "BKK" and "NI" represent "kick", "be-kicked" and "no-interaction". All model parameters could be trained in an end-to-end manner.

3. Preliminary

As our TOGN shares the identical feature-updating mechanism with FGNN [49], we first review this technique concisely. FGNN operates on a bipartite factor graph $\mathcal{G} = (\mathcal{V}, \mathcal{C}, \mathcal{E})$, where $\mathcal{V}, \mathcal{C}, \mathcal{E}$ denote the node set, the factor node set and the edge set respectively. Each $i \in \mathcal{V}$ is associated with a discrete variable $x_i \in \mathcal{X}_i$. Each edge $(c, i) \in \mathcal{E}$ connects a factor node $c \in \mathcal{C}$ and a node $i \in \mathcal{V}$. The factor graph defines a factorization of some function f with n variables. Specifically, $f(x_1, \ldots, x_n) = \prod_{c \in C} f_c(\mathbf{x}_c)$, where $\mathbf{x}_c$ denotes the variables associated with the nodes which have edge-connections with c. In practice, the functions f_c could be parameterized with deep networks.

Given $\mathcal{G}$, let $[\mathbf{f}_i^l]_{i \in \mathcal{V}}$ be a group of input node features, and let $[\mathbf{g}_c^l]_{c \in \mathcal{C}}$ be a group of input factor features, for the l-th layer of FGNN. Let $[\mathbf{t}_e]_{e \in \mathcal{E}}$ be a group of edge features shared by all FGNN layers. Here $\mathbf{f}^l \in \mathbb{R}^{D_l}$, $\mathbf{g}^l \in \mathbb{R}^{D_l}$ and $\mathbf{t} \in \mathbb{R}^H$. FGNN updates factor and node features separately via implementing two modules:

$$\mathbf{g}_c^{l+1} = \max_{i:(c,i) \in \mathcal{E}} \mathcal{Q}(\mathbf{t}_{ci}|\Phi_{VF}^l)\mathcal{M}([\mathbf{g}_c^l, \mathbf{f}_i^l]|\Theta_{VF}^l), \quad (1)$$

$$\mathbf{f}_i^{l+1} = \max_{c:(c,i) \in \mathcal{E}} \mathcal{Q}(\mathbf{t}_{ci}|\Phi_{FV}^l)\mathcal{M}([\mathbf{g}_c^l, \mathbf{f}_i^l]|\Theta_{FV}^l), \quad (2)$$

where $[\cdot, \cdot]$ denotes vector concatenation. The first equation is a factor-to-variable (VF) module and the second equation is a variable-to-factor (FV) module. $\mathcal{M}$ is a MLP (parameterized by Θ which is shared by all edges) maps the concatenation of factor and node features to a new feature vector of length D_l, and $\mathcal{Q}$ is another MLP (parameterized by Φ, which is also shared by all edges) maps its input edge feature vector to a $D_{l+1} \times D_l$ weight matrix. Here D_{l+1} denotes the length of the updated features (*i.e.* the length of the input node features of the next layer), and the operator $\max$ actually performs max-pooling.

Equations (1) and Equation (2) just comprise one layer of FGNN. To obtain a more powerful representation, one can stack a number of such layers, in which the output of the current layer is taken as the input to the subsequent layer. We refer readers to [49] for more details of FGNN.

4. Our Approach

Task Description and Notations Given an input image $\mathbf{I}$ and the bounding boxes (RoIs) of n detected human bodies, the HIU task is decomposed into two sub-tasks: 1) predicting the action category $\mathbf{y} = (y_i)_{i=1}^n$ for every individual where $y \in \mathcal{Y}$ ($\mathcal{Y}$ takes all action categories), and 2) predicting all pairwise interactive relations $\mathbf{z} = (z_{j,k})_{j=1,\ldots,n;k=1,\ldots,n}$ for each pair of people, where $z_{j,k} \in \{0, 1\}$ represents if the j-th and the k-th participants are interacting ($z_{s,t} = 1$) or not ($z_{s,t} = 0$). All vectors in this paper will be column vectors unless otherwise stated.

4.1. Model Overview

Figure 2 gives an overview of the proposed CAGNet, which consists of three components including a base-model, a TOGN and a CAR module. Given an input image and the detected human bodies as RoIs, the base-model takes a backbone CNN to extract features from the input, which are then processed by a RoIAlign module [13] to generate local features for each individual. Afterwards the local features are processed by one FC layer to generate *base features* as inputs to TOGN. Our TOGN graph (Section 4.2) includes two types of variable nodes (circles): one type is the y node to represent the action category of the associated person, the other type is the z node to represent the existence of interactive relation between a pair of people. The graph also includes a series of factor nodes (squares) in order to capture two types of third-order dependencies, respectively encoded by the $(y_i, y_j, z_{i,j})$ triplets (blue factor nodes) and the $(z_{u,v}, z_{v,w}, z_{u,w})$ triplets (the yellow factor node). We take the base features to initialize TOGN, and perform feature updating by passing messages between factor nodes and variable nodes such that rich contextual informa-

tion could be embedded. Though TOGN is able to learn rich contextual representations to facilitate the HIU task, the labeling and grouping consistencies among variables are not explicitly modeled. To alleviate this, we introduce a CAR module, which essentially conducts a deductive reasoning leveraging the oracles presented in Section 4.3. In practice the reasoning is implemented via solving a surrogate mean-field inference with differentiable high-order energy functions, which allows end-to-end learning of all modules within our CAGNet with GPU acceleration (Section 4.4).

4.2. Third-Order Graph Network for HIU

We now elaborate our TOGN for HIU in order capture two categories of third-order dependencies among action and interactive-relation variables.

Third-Order Factor Graph Formally, we define the factor graph as $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$, where $\mathcal{V}$ is the set of variable nodes, $\mathcal{F}$ is the set of factor nodes, and $\mathcal{E}$ is the set of edges. The node set is split into two disjoint subsets: $\mathcal{V} = \mathcal{V}^y \cup \mathcal{V}^z, \mathcal{V}^y \cap \mathcal{V}^z = \emptyset$. Specifically, $\mathcal{V}^y = \{1, \cdots, n\}$, and $\mathcal{V}^z = \{n+1, \cdots, n + \binom{n}{2}\}$. For each node $i \in \mathcal{V}^y$, a variable $y_i \in \mathcal{Y}$ is associated with it to represent the action category of the i-th individual. Let $g(u, v)$ be a function:

$$g(u, v) : \mathcal{V}^y \times \mathcal{V}^y \mapsto \mathcal{V}^z, \forall u, v \in \mathcal{V}^y, u < v. \quad (3)$$

For each node $k \in \mathcal{V}^z$, a variable $z_{u,v} \in \{0, 1\}$ is associated with it to represent if the pair of people (u, v) are interacting ($z_{u,v} = 1$) or not ($z_{u,v} = 0$), where $k = g(u, v)$.

To encode different relations, we create two groups of factor nodes. The first group is

$$\mathcal{F}^y = \{(i, j, g(i, j)) \mid \forall i, j \in \mathcal{V}^y, i < j\}, \quad (4)$$

which is taken to implicitly model the correlations among y_i, y_j and $z_{i,j}$ based on their base features. Intuitively, action labels (y_i, y_j) are highly correlated when the associated people are interacting (taking the *kicking* interaction in Figure 2 as an example), while this correlation vanishes if they are not interacting (*e.g.*, Person 2 and Person 3 in Figure 2).

The second group of factors is defined as

$$\mathcal{F}^z = \{(g(r,s), g(s,t), g(r,t)) \mid \forall r, s, t \in \mathcal{V}^y, r < s < t\}, \quad (5)$$

which is leveraged to implicitly model the correlations among $z_{r,s}$, $z_{s,t}$ and $z_{r,t}$ for each triplet of people (r, s, t). With such factors, we encourage the model to learn representations for the prediction of consistent interactive relations for each triplet. Fortunately, higher-order consistencies can be guaranteed if all third-order consistencies are satisfied (detailed in Section 4.3).

In summary, the factor node set is $\mathcal{F} = \mathcal{F}^y \cup \mathcal{F}^z$. Given $\mathcal{F}$ and $\mathcal{V}$, the edge set $\mathcal{E}$ is set up by connecting variable

nodes with factor nodes. Specifically, for each factor node $c = (i, j, k) \in \mathcal{F}$, we put three edges (c, i), (c, j) and (c, k) into $\mathcal{E}$, which finalizes the construction of the TOGN graph.

Initial Node Feature For each node $i \in \mathcal{V}^y$, let ϕ_i be the *base feature* extracted from the bounding box region of the i-th person using the base-model. For each $(u, v) \in \mathcal{V}^y \times \mathcal{V}^y$, $u < v$, let $j = g(u, v) \in \mathcal{V}^z$. We concatenate ϕ_u and ϕ_v, and use the concatenation as the *base feature* (denoted by ψ_j) for the variable node j. In order to compute the initial node features, we apply to the *base features* the linear transformations:

$$\mathbf{f}_i^1 = \text{FC}^y(\phi_i), \forall i \in \mathcal{V}^y, \quad (6)$$

$$\mathbf{f}_j^1 = \text{FC}^z(\psi_j), \forall j \in \mathcal{V}^z, \quad (7)$$

which project the original features into $\mathbb{R}^{D_1}$ space:

Initial Factor Feature The factor features are computed based on node features. For each factor node $c = (i, j, g(i, j)) \in \mathcal{F}^y$, the initial factor feature $\mathbf{g}_c^1 \in \mathbb{R}^{D_1}$ is computed with:

$$\mathbf{g}_c^1 = \frac{\mathbf{f}_i^1 + \mathbf{f}_j^1 + \mathbf{f}_{g(i,j)}^1}{3}. \quad (8)$$

For each $d = (g(r, s), g(s, t), g(r, t)) \in \mathcal{F}^z$, the associated factor feature $\mathbf{g}_d^1 \in \mathbb{R}^{D_1}$ is obtained using:

$$\mathbf{g}_d^1 = \frac{\mathbf{f}_{g(r,s)}^1 + \mathbf{f}_{g(s,t)}^1 + \mathbf{f}_{g(r,t)}^1}{3}. \quad (9)$$

Edge Feature For each edge $e = (q, p) \in \mathcal{E}$, the related feature $\mathbf{t}_e \in \mathbb{R}^H$ is given by:

$$\mathbf{t}_e = \text{ReLU}\big(\text{FC}^e([\mathbf{f}_p^1, \mathbf{g}_q^1])\big), \quad (10)$$

where $p \in \mathcal{V}$, $q \in \mathcal{F}$ and FC^e maps the concatenated feature vector to $\mathbb{R}^H$ space.

Taking as inputs the factor graph and the initial features, the first TOGN layer performs feature updating with the method described in Section 3. Afterwards we take the updated features as inputs to the next TOGN layer (which shares the factor graph and the feature updating algorithm with the first TOGN layer, but using different model parameters), and perform feature updating again (see supplementary for exact shapes of the input and output features of each layer). Finally, we compute the classification scores for individual actions and pairwise interactive relations using

$$\rho_i = \text{Softmax}\big(\alpha(\mathbf{f}_i^*)\big) \forall i \in \mathcal{V}^y, \quad (11)$$

$$\theta_j = \text{Softmax}\big(\beta(\mathbf{f}_j^*)\big) \forall j \in \mathcal{V}^z, \quad (12)$$

where $\mathbf{f}_i^*$, $\mathbf{f}_j^*$ are updated node features output by the last TOGN layer, α and β are linear functions which compute classification scores for individual actions ($\rho_i \in \mathbb{R}^{|\mathcal{Y}|}$) and pairwise interactive relations ($\theta_j \in \mathbb{R}^2$), respectively.

4.3. Consistency-Aware Reasoning

To resolve possible inconsistencies (recall Figure 1) incurred by using consistency-unaware models like CNN, PGNN and TOGN, we first present two deductive reasoning bias for human interaction understanding:

- **The compatibility oracle**: For any pair of interacting (denoted by $\leftrightarrow$) people A and B, their action categories must be compatible (denoted by $\odot$). In logical words, this rule is represented by $A \leftrightarrow B \Rightarrow y_A \odot y_B$.

- **The transitivity oracle**: Considering the interactive relations among a triplet of people (A, B, C), we have $(A \leftrightarrow B) \& (B \leftrightarrow C) \Rightarrow (A \leftrightarrow C)$.

Typical compatible examples include (*handshake, handshake*), (*pass, receive*) and (*punch, fall*), and typical incompatible examples are (*handshake, hug*), (*punch, pass*), (*highfive, handshake*). Instead of predesignating such compatibility, which might change across datasets, our CAR module is able to learn them directly from data. Though the *transitivity* oracle only considers triplets of people, it is straightforward to prove that the higher-order transitivity associated with an arbitrary number of people is simply a conclusion of the third-order transitivity. Intuitively, by enforcing the transitivity across all triplets, participants in the scene are split into different groups, such that individuals in the identical group are interacting with each other, while people in different groups have no interaction. One may argue that the transitivity oracle is not necessarily true. For example, if A and B are both passing objects to C, then A and B will be considered as interacting under this assumption. However, we believe it is reasonable to put A, B and C in the identical group as both are passing objects to C.

With such oracles, predictions of the TOGN described in Section 4.2 could be refined by applying the traditional logical reasoning algorithms like resolution. However, embedding such reasoning into deep learning frameworks directly is highly challenging. As a workaround, our reasoning approach first embeds the knowledge into an energy function defined by

$$E(\mathbf{y}, \mathbf{z}; \mathbf{x}) = \sum_{i \in \mathcal{V}^y} -\rho_i(y_i) + \sum_{(j,k,l) \in \mathcal{F}^y} \left[-\theta_{j,k}(z_{j,k}) + K^{\mathcal{C}}(y_j, y_k, z_{j,k}) \right] + \sum_{(r,s,t) \in \mathcal{F}^z} K^{\mathcal{T}}(z_{r,s}, z_{s,t}, z_{r,t}), \quad (13)$$

where $\theta_{j,k}(z_{j,k}) = \theta_{g(j,k)}(z_{j,k})$. The data terms $-\rho_i$ and $-\theta_{j,k}$ (computed by Equations (11) and (12)) are utilized to penalize particular y-label and z-label assignments respectively based on the learned deep representations. The functions $K^{\mathcal{C}}$ and $K^{\mathcal{T}}$ are the so-called P^n-Potts models [19] defined by the equations below:

$$K^{\mathcal{C}}(y_j, y_k, z_{j,k}) = \begin{cases} \lambda^{\mathcal{C}}(y_j, y_k) & \text{if } z_{j,k} = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

$$K^{\mathcal{T}}(z_{r,s}, z_{s,t}, z_{r,t}) = \begin{cases} \lambda^{\mathcal{T}} & \text{if } (z_{r,s}, z_{s,t}, z_{r,t}) \in \Gamma, \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

Here Γ is a set $\{(1,1,0), (1,0,1), (0,1,1)\}$ that includes all cases violating the transitivity oracle, $\lambda^{\mathcal{C}}(y_j, y_k)$ and $\lambda^{\mathcal{T}}$ are penalties incurred by predictions which violate the compatibility and transitivity oracles. It is easy to check that when $\lambda^{\mathcal{C}}$ and $\lambda^{\mathcal{T}}$ are sufficiently large, minimizing the energy (13) delivers desirable $\mathbf{y}$ and $\mathbf{z}$ predictions which satisfy the *compatibility* and *transitivity* oracles. In this paper, instead of predesignating suitable $\lambda^{\mathcal{C}}$ and $\lambda^{\mathcal{T}}$ values, we learn them from data in conjunction with other parameters of CAGNet (visualization of some learned $\lambda^{\mathcal{C}}(y_i, y_j) \, \forall y_i \in \mathcal{Y}, \forall y_j \in \mathcal{Y}$ is provided in the supplementary material).

Mean-Field Inference Minimizing (13) is NP-complete. Here we derive an efficient mean-field inference algorithm by first approximating the joint distribution $P(\mathbf{y}, \mathbf{z}|\mathbf{x}) \propto \exp(-E(\mathbf{y}, \mathbf{z}; \mathbf{x}))$ with a product of independent marginal distributions:

$$P(\mathbf{y}, \mathbf{z}|\mathbf{x}) \approx \prod_{i \in \mathcal{V}^y} Q_i(y_i) \prod_{l \in \mathcal{V}^z : g(j,k)=l} Q_{j,k}(z_{j,k}). \quad (16)$$

Then we derive the mean-field updates of all marginal distributions using the techniques described in [37], which gives

$$\tilde{Q}_i^t(y_i) = \sum_{j \in V \setminus \{i\}} \sum_{y_j} \lambda^{\mathcal{C}}(y_i, y_k) Q_j^{t-1}(y_j) Q_{i,j}^{t-1}(z_{i,j}=1), \quad (17)$$

$$Q_i^t(y_i) = \frac{\exp\left(\rho_i(y_i) - \tilde{Q}_i^t(y_i)\right)}{Z_i}, \quad (18)$$

where Z_i is a normalization constant. The marginal distributions on z variables are

$$\tilde{Q}_{k,l}^t(z_{k,l}) = \sum_{y_k, y_l} z_{k,l} \lambda^{\mathcal{C}}(y_k, y_l) Q_k^{t-1}(y_k) Q_l^{t-1}(y_l) + \sum_{m \in V \setminus \{k,l\}} \sum_{z_{k,m}, z_{m,l}} \mathbb{1}\big((z_{k,m}, z_{m,l}, z_{k,l}) \in \Gamma\big) \lambda^{\mathcal{T}} Q_{k,m}^{t-1}(z_{k,m}) Q_{m,l}^{t-1}(z_{m,l}), \quad (19)$$

$$Q_{k,l}^t(z_{k,l}) = \frac{\exp\left(\theta_{k,l}(z_{k,l}) - \tilde{Q}_{k,l}^t(z_{k,l})\right)}{Z_{k,l}}, \quad (20)$$

where $\mathbb{1}(\cdot)$ is an indicator function (gives 1 if the testing condition holds, and 0 otherwise), $t \in \{1, 2, \ldots, T\}$, $Z_{k,l}$ is

Method	UT			BIT			TVHI		
	F1	Accuracy	mean IoU	F1	Accuracy	mean IoU	F1	Accuracy	mean IoU
VGG19 [34]	85.69	91.68	69.03	85.22	89.60	67.03	70.68	76.90	52.30
ResNet50 [14]	90.62	94.64	76.70	87.12	91.20	71.41	81.18	82.61	66.33
Inception V3 [36]	92.20	95.86	80.30	87.84	91.61	72.00	83.00	86.91	71.53
Base model + CAR	92.81	96.26	81.51	88.72	92.23	73.99	83.07	87.23	72.29
TOGN (ours)	93.45	96.50	84.53	91.26	94.84	78.27	90.41	92.51	79.40
TOGN+CAR^C (ours)	94.32	97.03	85.06	92.70	95.34	80.78	91.90	93.44	82.07
TOGN+CAR^T (ours)	93.82	96.71	83.90	92.30	95.20	80.22	90.35	92.63	79.05
TOGN+CAR^{CT} (ours)	**94.55**	**97.06**	**85.50**	**92.79**	**95.41**	**81.32**	**92.83**	**95.29**	**84.02**

Table 1. Ablation study on three benchmarks. All results are in percentage. The proposed TOGN performs much better than the best base model (Inception V3). The proposed CAR module further improves HIU results by clear margins. Bold texts denote best results.

Algorithm 1: The mean-field inference.

Input: The graph $\mathcal{G}$, $\rho_i(y_i)$, $\theta_{k,l}(z_{k,l})$, $\lambda^{\mathcal{C}}$ and $\lambda^{\mathcal{T}}$.
Output: $\check{\rho}_i(y_i)$, $\check{\theta}_{k,l}(z_{k,l})$.
1 **Initialization**: Let $Q_i^0(y_i) = \frac{\exp(\rho_i(y_i))}{Z_i}$, and let $Q_{k,l}^0(z_{k,l}) = \frac{\exp(\theta_{k,l}(z_{k,l}))}{Z_{k,l}}$.
2 **for** $t = 1, 2, \ldots, T$ **do**
3   Compute $\tilde{Q}_i^t(y_i)$, $\tilde{Q}_{k,l}^t(z_{k,l})$, $Q_i^t(y_i)$ and $Q_{k,l}^t(z_{k,l})$ using Equations (17) to (20).
4 **end**
5 $\check{\rho}_i(y_i) \leftarrow \rho_i(y_i) - \tilde{Q}_i^T(y_i)$,
   $\check{\theta}_{k,l}(z_{k,l}) \leftarrow \theta_{k,l}(z_{k,l}) - \tilde{Q}_{k,l}^t(z_{k,l})$.

---

a normalization constant. We initialize the marginal distributions $Q_i^0(y_i)$, $Q_{k,l}^0(z_{k,l})$ by applying the softmax function to the scores output by the graph network. The inference is summarized by Algorithm 1. Note that we can perform the updates of all expectations (Equation (17) and (19)) and marginal probabilities (Equation (18) and (20)) in parallel, which yields very efficient inference.

As mentioned, Algorithm 1 is a surrogate of the consistency-aware reasoning task taking the two oracles as its knowledge-base. This algorithm actually forms the last layer of our CAGNet, which outputs updated action scores $\check{\rho}_i \ \forall i \in V^y$ and interactive scores $\check{\theta}_{j,k}$, $\forall l \in \mathcal{V}^z$ and $g(j, k) = l$. Our experimental results in Section 5 demonstrate that such updated scores are able to deliver more consistent HIU predictions.

4.4. End-to-End Learning

The mean-field inference algorithm allows the backpropagation of the error signals $\frac{\partial Loss}{\partial Q}$ to all parameters of CAGNet (including that of the base-model, the TOGN and the CAR module), which enables the joint training of all parameters from scratch. In practice, we resort to a two-stage training due to the memory limitation of our GPUs. The first stage learns a base-model with the backbone CNN initialized by a model pre-trained on ImageNet. The

second stage trains the TOGN, $\lambda^C(y_j, y_k)$ and λ^T jointly with fixed backbone-parameters. We train all models using the identical cross-entropy losses computed on both **y** and **z** predictions.

Implementation Details Our implementation is based on PyTorch deep learning toolbox and a workstation with three pieces of NVIDIA GeForce GTX 1080 Ti GPU. We test several backbone CNNs including VGG19 [34], ResNet 50 [14] and Inception V3 [36]. We use the official implementation of RoIAlign by PyTorch, which outputs feature maps with a size of $5 \times 5 \times 1056$ (using Inception V3). We add dropout (the ratio is 0.3) followed by a layer-normalization to every FC layer of CAGNet except for the ones computing final classification scores. For the mean-field inference, we set $\lambda^C(y_j, y_k) = 0.5$, $\lambda^T = 0.1$ and $T = 6$ for good compromise between speed and accuracy. We adopt mini-batch SGD with Adam to learn the network parameters, and train all models in 200 epochs. We augment training data with random combinations of scaling, cropping, horizontal flipping and color jittering. For the scaling and flipping operations, the bounding boxes are scaled and flipped as well.

5. Experiment

Dataset We use three benchmarks including UT [30], BIT [46] and TVHI [26]. UT contains 120 short videos of 6 action classes: *handshake*, *hug*, *kick*, *punch*, *push* and *no-action*. As done by [40], we extend original action classes by introducing a passive class for each of the three asymmetrical action classes including kick, punch and push (*be-kicked*, *be-punched* and *be-pushed*). Consequently, we have 9 action classes in total. Following [40], we split samples of UT into 2 subsets for training and testing. BIT covers 9 interaction classes including *box*, *handshake*, *highfive*, *hug*, *kick*, *pat*, *bend*, *push* and *others*, where each class contains 50 short videos. Of each class 34 videos are chosen for training and the rest for testing as recommended by [46]. TVHI contains 300 short videos of television shows, which covers 5 action classes including *handshake*, *highve*, *hug*, *kiss* and *no-action*. As suggested by [26], we split samples

of TVHI into two parts for training and testing. To facilitate element-wise evaluation on y and z predictions, we use annotated bounding boxes for both training and testing.

5.1. Ablation

Evaluation Metric Since the numbers of instances across different classes are significantly imbalanced, we use multiple metrics including *F1-score*, *overall accuracy* and *mean IoU* for evaluation. Specifically, we calculate the *macro-averaged-F1* scores on y and z predictions respectively (using the *f1_score* function in *sklearn* package), and present the mean of the two F1 scores. Likewise, *overall accuracy* calculates the mean of the action-classification accuracy and the interactive-relation-classification accuracy. To obtain *mean IoU*, we first compute IoU value on each class, then average all IoU values. We first analyze the capabilities of different components in the proposed CAGNet, using results provided by Table 1.

Choice of Backbone-CNN. Here we evaluate base models (see Figure 2) taking different backbone CNNs to extract image features. We test three popular backbones: VGG19 [34], Inception V3 [36] and ResNet50 [14], and the results correspond to the first three rows (from top to bottom) in Table 1. Inception V3 performs notably better than other backbones on all benchmarks. The reason might be that Inception V3 is able to learn multi-scale feature representations, which stacks into a feature pyramid to better capture the appearance of human actions. Hence we use Inception V3 as the backbone for all subsequent experiments.

Effect of the TOGN Remember that the proposed TOGN takes the features extracted by Inception V3 as inputs, and learn third-order dependencies among structured variables. Overall, the proposed TOGN outperforms all basemodels on the three benchmarks under all considered metrics. In particular, TOGN results are moderately higher than the best basemodel (*i.e.* Inception V3) on UT, and are significantly better than the basemodel on BIT (91.26 *vs.* 87.84 on F1) and TVHI (90.41 *vs.* 83.0 on F1), thanks to the rich contextual representations learned by our TOGN. Note that the performance gain on UT is much less compared with results on other benchmarks. This is probably because human interactions in UT (each video contains just two individuals, and the background is clear) are simpler than BIT and TVHI, and the performance on this dataset tends to be saturated. Our experiments use TOGN with 10 layers. Empirically, we observed that both deeper and shallower TOGNs give slightly worse results than 10 layers.

Effect of the CAR Module Here we compare four models: 1) *Base model + CAR* that consists of the base-model (Inception V3 as backbone) followed by the proposed CAR module; 2) TOGN+CAR^{C} is the proposed CAGNet without taking the *transitivity oracle* into consideration; 3) TOGN+CAR^{T} is the proposed CAGNet without taking the

Method	F1 (%)	Accuracy (%)	mean IoU (%)
GN [43]	80.57	82.76	66.82
Modified GN [43]	84.18	87.86	71.31
Joint + AS [40]	83.50	87.33	71.64
QP + CCCP [42]	83.42	87.25	71.61
CAGNet (ours)	**92.83**	**95.29**	**84.02**

Table 2. Comparison with recent methods on TVHI. Our CAGNet overshoots competitive models under all evaluation metrics.

Method	F1 (%)	Accuracy (%)	mean IoU (%)
GN [43]	70.52	78.52	65.89
Modified GN [43]	89.95	93.38	76.42
Joint + AS [40]	88.61	91.77	72.12
QP + CCCP [42]	88.80	91.92	72.46
CAGNet (ours)	**92.79**	**95.41**	**81.32**

Table 3. Comparison with recent methods on BIT. Our CAGNet performs much better than other recent approaches.

Method	F1 (%)	Accuracy (%)	mean IoU (%)
GN [43]	89.25	91.78	78.24
Modified GN [43]	93.38	96.39	84.13
Joint + AS [40]	92.20	95.86	80.30
QP + CCCP [42]	89.71	93.23	80.35
CAGNet (ours)	**94.55**	**97.06**	**85.50**

Table 4. Comparison with recent methods on UT. Our CAGNet performs moderately better than other recent approaches.

compatibility oracle into consideration; 4) TOGN+CAR^{CT} actually is our full CAGNet. Here we can draw two conclusions based on the results in Table 1. First, the incorporation of the CAR module (Base model + CAR and TOGN+CAR^{CT}) boosts HIU performance (2.42, 2.78 and 4.62 points better than TOGN on TVHI in terms of F1, Accuracy and mean IoU), which validates the significance of exploiting consistency-aware-reasoning. Second, both oracles are critical to achieve the best results. Though incorporating either the compatibility (TOGN+CAR^{C}) or the transitivity oracle (TOGN+CAR^{T}) already performs better than TOGN, TOGN with both oracles (TOGN+CAR^{CT}) performs notably better than using each of them, which suggests that these two oracles complement each other.

5.2. Comparison with Recent Methods

We consider three recent approaches. ***Joint + AS*** [40] first extracts motion features of individual actions with backbone CNN. Afterwards the deep and contextual features of human interactions are fused by structured SVM. This method is able to predict y and z in a joint manner. ***QP + CCCP*** [42] takes a structured model to represent the correlations between y and z variables as well. It also developed an inference algorithm (namely *QP + CCCP*) to solve the related inference problem. ***GN*** [43] is a recent state-of-the-art for recognizing collective human activities. This model is empowered by both the representative abil-

13375

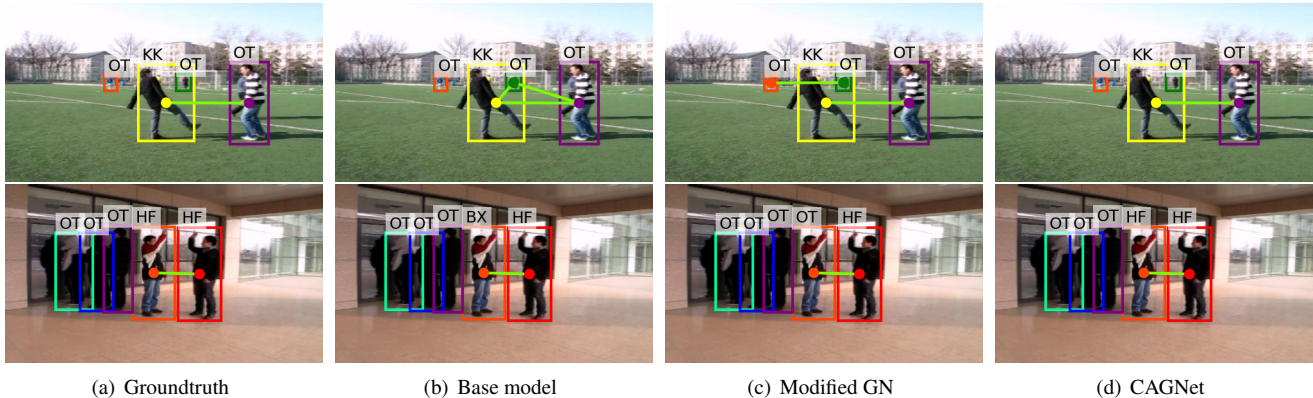|			
(a) Groundtruth	(b) Base model	(c) Modified GN	(d) CAGNet

Figure 3. Visualize HIU results predicted by different models. Figures are best viewed in color. Each row shows an example from BIT. Columns from left to right correspond to results of *groundtruth*, *base-model*, *Modified GN*, and *CAGNet*. Green lines denote predicted interactive pairs ($z = 1$). Texts present predicted individual actions (y variables), where *KK, HF, BX, OT* mean *kick, high-five, box, others* respectively. Note that the predictions of CAGNet (the rightmost column) always obey the two oracles defined in Section 4.3. More visualized results are provided in the supplementary material.

ity of deep CNNs and the attention mechanism of PGNN. Note that *GN* does not yield **z** predictions. We fix this with two solutions. First, we just set $z_{i,j} = 1$ if the learned relation value is greater than 0.5 (see Equation (2) in [43]), otherwise we set $z_{i,j} = 0$ (this solution does not introduce new parameters). Second, we attach a head to the tail of *GN* to make z predictions, and train parameters of this head with cross-entropy loss. We call such a solution the ***Modified GN***. We find that such a straightforward modification offers a boost of performance on HIU compared against the original GN (see Table 2 to Table 4). The reason is that *Modified GN* is trained with additional supervision on interactive relations, which guides the network to learn more useful representations for the prediction of z.

For fair comparison, all methods take Inception V3 as the backbone to extract image features. Results on three datasets are provided in Table 2, Table 3 and Table 4. We can see that CAGNet outperforms *modified GN* and shallow structured models (*Joint + AS* and *QP + CC-CP*) significantly on all evaluated benchmarks. Compared with CAGNet which is able to model higher-order relations, *modified GN* is only able to model pairwise interactive relations, hence it is consistency-unaware. Consequently *GN* and *modified GN* perform much worse than CAGNet. Albeit sharing the same feature extractor (Inception V3) with CAGNet, *Joint + AS* and *QP + CCCP* learn human interactive relations via shallow structured models without incorporating higher-order contextual dependencies and consistency-aware reasoning, hence their performances are much worse than our CAGNet.

To provide a qualitative analysis of different models, we visualize a few predictions in Figure 3. Though the predicted action labels are inconsistent or the predicted interactive relations violate the *transitivity oracle* using either

the *Base-model* or the *modified GN*, thanks to our proposed TOGN and CAR module, CAGNet is able to make perfect predictions, at least on these examples.

6. Conclusion

This work devoted the better understanding of human interactions by presenting the so-called CAGNet, which is able to resolve the labeling and grouping inconsistencies within HIU predictions. Our network relies on a TOGN module and a CAR module. The TOGN module addresses the inconsistency by learning better contextual features with higher-order graph networks, while the proposed CAR module tackles the issue by exploiting the deductive reasoning bias of HIU explicitly. For efficient training and prediction, we have cut the desired deductive reasoning into solving a surrogate energy-minimization, which reduces the chance of obtaining inconsistent HIU predictions and allows the training of all model parameters in an end-to-end way. Note that our CAR module is motivated by the HIU task, instead of proposing a comprehensive system for deep logical reasoning. Ablation study and comparison against the state-of-the-arts on three benchmarks have justified the effectiveness of the proposed approach.

Acknowledgement

References

[1] S. Amizadeh, H. Palangi, O. Polozov, Y. Huang, and K. Koishida. Neuro-symbolic visual reasoning: Disentangling "visual" from "reasoning". In *ICML*, 2020.

[2] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese. Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In *CVPR*, 2017.

[3] P. Barcelo, E. Kostylev, M. Monet, J. Perez, J. Reutter, and J. Silva. The logical expressiveness of graph neural networks. In *ICLR*, 2020.

[4] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[5] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[6] W. Choi and S. Savarese. Understanding collective activitiesof people from videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(6):1242–1257, 2014.

[7] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *CVPR*, 2011.

[8] Z. Deng, A. Vahdat, H. Hu, and G. Mori. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *CVPR*, 2016.

[9] H. Dong, J. Mao, T. Lin, C. Wang, L. Li, and D. Zhou. Neural logic machines. In *ICLR*, 2019.

[10] Y. Du, S. Li, and I. Mordatch. Compositional visual generation with energy based models. In *NeurIPS*, 2020.

[11] L. Fan, W. Wang, S. Huang, X. Tang, and S. Zhu. Understanding human gaze communication by spatio-temporal graph reasoning. In *ICCV*, 2019.

[12] Z. Gao, G. Hua, D. Zhang, N. Jojic, L. Wang, J. Xue, and N. Zheng. Er3: A unified framework for event retrieval, recognition and recounting. In *CVPR*, 2017.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Deep graphical feature learning for the feature matching problem. In *ICCV*, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] M. Ibrahim and Greg Mori. Hierarchical relational networks for group activity recognition and retrieval. In *ECCV*, 2018.

[16] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, 2013.

[17] A. Kar, N. Rai, K. Sikka, and G. Sharma. Adascan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos. In *CVPR*, 2017.

[18] T. Kipf, E. Fetaya, K. Wang, M. Welling, and R. Zemel. Neural relational inference for interacting systems. In *ICML*, 2018.

[19] P. Kohli, P. Kumar, and P. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[20] Y. Kong and Y. Fu. Close human interaction recognition using patch-aware models. *IEEE Trans. Image Proc.*, 25(1):167–178, 2015.

[21] Y. Kong, Y. Jia, and Y. Fu. Interactive phrases: Semantic descriptions for human interaction recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(9):1775–1788, 2014.

[22] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori. Discriminative latent models for recognizing contextual group activities. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(8):1549–1562, 2012.

[23] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[24] J. Mao, C. Gan, P. Kohli, J. Tenenbaum, and J. Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *ICLR*, 2019.

[25] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman. Structured learning of human interactions in tv shows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(12):2441–2453, 2012.

[26] A. Patron-Perez, M. Marszalek, A. Zisserman, and I. Reid. High five: recognising human interactions in tv shows. In *BMVC*, 2010.

[27] R. Pramono, Y. Chen, and W. Fang. Empowering relational network by self-attention augmented conditional random fields for group activity recognition. In *ECCV*, 2020.

[28] M. Qi, J. Qin, A. Li, Y. Wang, J. Luo, and L. Gool. stagnet: An attentive semantic rnn for group activity recognition. In *ECCV*, 2018.

[29] S. Qi, W. Wang, B. Jia, J. Shen, and S. Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, 2018.

[30] M. S. Ryoo and J. K. Aggarwal. UT-interaction dataset, ICPR contest on semantic description of human activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.

[31] T. Shu, S. Todorovic, and S. Zhu. Cern: Confidence-energy recurrent network for group activity recognition. In *CVPR*, 2017.

[32] X. Shu, J. Tang, G. Qi, W. Liu, and J. Yang. Hierarchical long short-term concurrent memory for human interaction recognition (early access). *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[33] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[35] W. Sultani, C. Chen, and M. Shah. Real-world anomaly detection in surveillance videos. In *CVPR*, 2018.

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[37] V. Vineet, J. Warrell, and P. Torr. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. *IJCV*, 110(3):290–307, 2014.

[38] L. Wang, , Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015.

[39] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[40] Z. Wang, J. Jin, T. Liu, S. Liu, J. Zhang, S. Chen, Z. Zhang, D. Guo, and Z. Shao. Understanding human activities in videos: A joint action and interaction learning approach. *Neurocomputing*, 321:216–226, 2018.

[41] Z. Wang, S. Liu, J. Zhang, S. Chen, and Q. Guan. A spatio-temporal crf for human interaction understanding. *IEEE Trans. Circuits Syst. Video Technol.*, 2016.

[42] Z. Wang, T. Liu, Q. Shi, M. Kumar, and J. Zhang. New convex relaxations for MRF inference with unknown graphs. In *ICCV*, 2019.

[43] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu. Learning actor relation graphs for group activity recognition. In *CVPR*, 2019.

[44] A. Yan, Y. Wang, Z. Li, and Y. Qiao. Pa3d: Pose-action 3d machine for video recognition. In *CVPR*, 2019.

[45] D. Yang and Y. Zou. A graph-based interactive reasoning for human-object interaction detection. In *IJCAI*, 2020.

[46] Kong Yu, Yunde Jia, and Fu Yun. Learning human interaction by interactive phrases. In *ECCV*, 2012.

[47] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Yao, and R. Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *ICCV*, 2019.

[48] Z. Zhang and W. S. Lee. Deep graphical feature learning for the feature matching problem. In *ICCV*, 2019.

[49] Z. Zhang, F. Wu, and W. Lee. Factor graph neural network. In *NeurIPS*, 2020.