

Structure Inference Machines: Recurrent Neural Networks for Analyzing Relations in Group Activity Recognition

Zhiwei Deng Arash Vahdat Hexiang Hu Greg Mori
School of Computer Science, Simon Fraser University, Canada
{zhiweid, avahdat, hexiangh}@sfu.ca, mori@cs.sfu.ca

Abstract

Rich semantic relations are important in a variety of visual recognition problems. As a concrete example, group activity recognition involves the interactions and relative spatial relations of a set of people in a scene. State of the art recognition methods center on deep learning approaches for training highly effective, complex classifiers for interpreting images. However, bridging the relatively low-level concepts output by these methods to interpret higher-level compositional scenes remains a challenge. Graphical models are a standard tool for this task. In this paper, we propose a method to integrate graphical models and deep neural networks into a joint framework. Instead of using a traditional inference method, we use a sequential inference modeled by a recurrent neural network. Beyond this, the appropriate structure for inference can be learned by imposing gates on edges between nodes. Empirical results on group activity recognition demonstrate the potential of this model to handle highly structured learning tasks.

1. Introduction

Relations between image entities are an important facet of higher-level visual understanding. Building relationships, such as the spatial distance between people in a scene, their relative motions, or concurrent actions can be used to drive recognition of higher-level activities. Models for interpreting such scenes require the need to accurately interpret image cues, determine relevant relations between entities, and infer the properties of these relations.

In this paper we present a general-purpose method for this task, illustrated in Fig. 1. The method builds upon deep networks for image analysis, endowing these networks with the ability to reason over structures and relationships. This is accomplished by building higher-level recurrent networks that equip the model with the ability to perform inference over lower-level network outputs, including learning structures that are effective for higher-level tasks.

We ground the work by developing specific models for

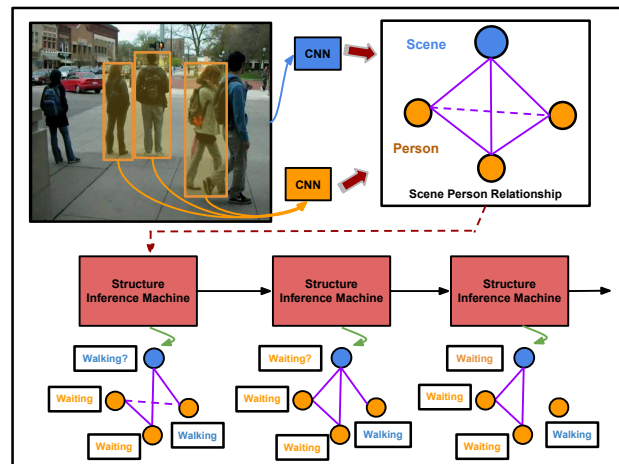


Figure 1. Structure learning in a deep network. Analyzing group activity requires reasoning about relations between the actions of individual people. Our structure inference machine iteratively reasons about which people in a scene are interacting and which are involved in a group activity.

group activity analysis. Group activity analysis involves reasoning over individual people in a scene and considering their relations. Multiple people in a scene could either be performing the same action at the same time, or have varied actions and interactions that compose a collective activity. Effective models need to jointly consider the rich relations between components of visual appearance.

Standard approaches to this problem utilize graphical models to encode spatial relations and interactions. Recent work in this vein includes Choi et al. [8], who discover sub-groups of interacting people. Lan et al. [26] proposed a hierarchical graphical model that considers the interactions on the social role level. Hajimirsadeghi and Mori [16] proposed a gradient boosting training method. Amer et al. [1] adopted a HiRF model to perform recognition and detection simultaneously.

On another track, deep learning has proven successful in many computer vision applications, such as image clas-

sification, object recognition, and action recognition. On the image side, seminal work by Krizhevsky et al. [24] demonstrated the effectiveness of deep networks for object recognition; recent state of the art methods include GoogLeNet [34]. On the video side, Simonyan and Zisserman [33] proposed a two-stream convnet pipeline to apply deep learning to video analysis. Karpathy et al. [20] adopted various fusion techniques in convolutional neural networks to consider temporal information in video sequences. These methods have demonstrated the power of deep networks for classification tasks.

However, these models are trained to produce a flat classification output, categorizing an image/video according to the existence of a set of object/action labels. For highly compositional tasks such as group activity recognition, models reasoning over structures can bring benefits, allowing the classification of higher-level concepts built from recognition over lower-level entities.

The main contribution of this paper centers on developing a model that bridges from low-level classifications to higher-level compositions. We contribute an end-to-end trainable deep network that (1) classifies low-level image inputs according to their content, (2) refines these classifications by passing messages between outputs, (3) performs structure learning via gating functions that determine which outputs to connect, and (4) results in effective classification of high-level concepts.

2. Previous Work

This paper contributes a general-purpose deep learning inference machine and demonstrates its effectiveness for group activity recognition. In this section we review relevant work on modeling structures in deep learning and specific models deployed for group activity recognition.

Deep Learning with Structures: Recently, there have been several interesting approaches to address the problem of combining graphical models and deep neural networks, primarily in the context of semantic image segmentation. Chen et al. [6] proposed DeepLab, that feeds coarse responses at the final layer of a deep neural network to the CRF model proposed by Krähenbühl and Koltun [23]. Zheng et al. [38] proposed a CRF-RNN that trains the same model in an end-to-end fashion by transforming the approximate inference method [23] into a Recurrent Neural Network (RNN). Schwing and Urtasun [31] proposed an iterative procedure for end-to-end training of the CRF model. We expand on this line of work by relaxing the assumptions that the underlying graphical model i) is fully connected and/or ii) has Gaussian kernels in the pairwise potential functions. Similarly, Chen et al. [7] extended to general MRFs using approximate inference. We use different inference techniques and enable structure learning.

Another line of work aims at modeling class relations in

a graphical model that could be trained with a deep neural network. Deng et al. [12] proposed Hierarchy and Exclusion (HEX) graphs for modeling inclusion and exclusion relations between object classes and showed how these graphs could be used for computing HEX-based marginalized distributions of labels on top of a deep neural network. Ding et al. [14] extended HEX graphs to probabilistic HEX graphs modeled by Ising models and showed how standard Loopy Belief Propagation (LBP) can be used in the inference. Both works rely on a special-purpose graphical model that represents particular relations between binary variables with a pre-designed structure.

In the context of other structured problems, Bottou et al. [5] proposed Graph Transformer Networks to jointly optimize subtasks. In this work, it was assumed that exact inference can be performed during a forward-backward pass. Ross et al. [29] phrased structured prediction as a series of message passing steps. Tompson et al. [35] proposed a feed-forward neural network that mimics a single iteration of the message passing algorithm for a markov random field for the task of human body pose recognition. Zhang et al. [37] incorporate structured prediction as a loss layer in a neural network. Deng et al. [13] conduct message passing to do inference over a fixed structure for group activity recognition. In contrast to these approaches, we infer structure via a gated network, allowing the model to determine the appropriate connections to use for inference.

Group Activity Recognition: Group activity recognition is typically modeled as a structured prediction problem that considers both individual actions and interactions with other people in a scene. Many previous work have used various forms of graphical models to address this problem: hierarchical graphical models [2, 26, 28, 30, 9], AND-OR graphs [3, 15] and dynamic Bayesian networks [39] are among the popular models. Lan et al. [28] and Amer et al. [2] have shown the effectiveness of adaptive structures to the group activity recognition problem. Modeled by latent structure [28] or grouping nodes [2], an adaptive structure can adjust its structure to the most discriminative interactions in a scene. Khamis et al. [22, 21] utilize track-level and person-level features to determine group activity. Shu et al. [32] reason about groups, roles, and events on top of noisy tracklets with a spatio-temporal AND-OR graph model. Kwak et al. [25] reason over temporal logic primitives in a quadratic programming formulation. These models are trained in a sophisticated framework using shallow features and cannot easily be adopted in a deep learning framework. In this work, we propose a general framework for integrating graphical models into a deep neural network that is capable of adapting their structure in an instance-based approach.

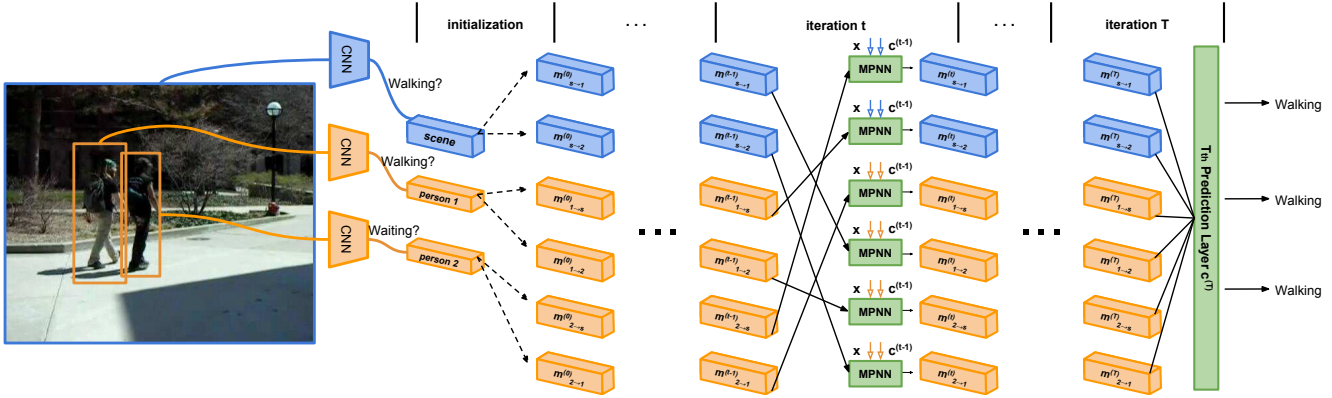


Figure 2. The pipeline of inference in an RNN. We first use the unary scores to initialize the messages. In every iteration, new messages are computed using related message units, unary scores (x), and output predictions from the previous timestep ($c^{(t-1)}$). Note that for each timestep, a prediction layer outputs predictions (only illustrated in last layer), and in training receives loss as in a standard RNN.

3. Structure Inference Machine

Group activity recognition requires reasoning about structures. Interpreting an image of a scene of people involves determining what each individual person is doing and reasoning about their relations. These tasks are both challenging due to ambiguity in image features and uncertainty in determining relations between people. The ability to infer structures over the people in a scene is helpful for suppressing noise in the form of inaccurate human detections, mistaken low-level action recognition results, and spurious people not involved in a particular group activity.

Beyond group activity recognition, many other visual recognition tasks benefit from similar lines of reasoning. Detecting and classifying individual component elements can be improved by considering structured relations among them.

The question we address in this paper is how to model such structured relations. We take an approach building upon neural networks. Deep learning-based methods have benefits in highly effective low-level action recognition, and we want to utilize this effectiveness within an end-to-end trainable model for higher-level reasoning.

Two components are used to cast this problem as a neural network formulation.

1. Recurrent neural networks for message passing.

Consider an individual person within an image of a scene. Ambiguity in inferring the action of the individual person is a fundamental problem. As per standard arguments around context [36, 18, 10, 28], using the actions of other people in the scene can help to disambiguate the action of this individual. We accomplish this by a recurrent neural network that aggregates cues about the actions of other people in a scene by repeatedly passing messages that refine estimates of an individual person’s action.

2. Gating functions to learn structures.

Deciding who is interacting with whom in a group activity is an important part of inference. Sub-groups of people can be engaged in different activities [8]; individuals can be outliers compared to the group activity [28]. Reasoning over structures that determine connections between people in a scene can bring many benefits: which people are relevant to detecting the presence of an overarching group activity, which people provide context for which others. Within a neural network structure, we accomplish this by introducing trainable gating functions that can turn on and off connections between individual people in the scene.

Fig. 2 summarizes our structure inference machine. The following sections present its details. First, we present the use of recurrent neural networks as a tool for inference in a group activity model in Sec. 4. The use of gating functions to learn structure is presented in Sec. 5. By un-tying weights in these networks, we can relax assumptions regarding the message passing, leading to a general structure inference machine presented in Sec. 5.1.

4. Group Activity Recognition with an RNN

We build our model on top of a set of person detections in an image. The model includes the actions for these individual people as well as the group activity for the whole image. Given a set of M detected persons in a scene, a classifier (using a CNN) is used to provide visual classification $\{x_i\}_{i=1}^M$ of each person’s action based on an image window cropped at the person detection. Each x_i is a probability distribution over individual action for person i . In addition, a classifier that operates on the entire image can be used to directly estimate the group activity in the scene. We denote by x_s the group activity classification obtained from this whole-image classifier; x_s is a distribution over possible scene-level group activities.

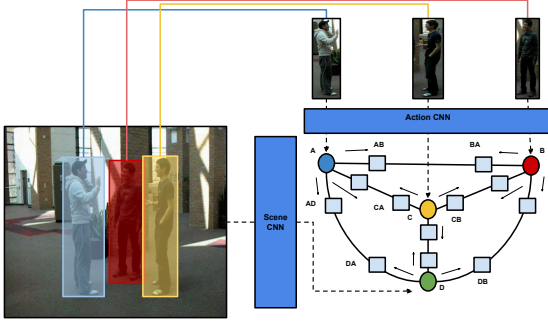


Figure 3. A group activity scenario represented as a graphical model. Estimates of individual person actions and a group activity are refined via message passing. The squares are messages. These message units carry information from the source node and are propagated to the target node.

A graphical model is built over these individual actions and the group activity, as shown in Fig. 3. We use this graphical model to motivate our framework, and address structure learning in a graphical model with varied potential functions on top of a deep neural network.

4.1. Recurrent Networks for Refining Action/Activity Classification

On top of the individual classifications, we build a neural network structure which considers relations among entities to refine these classifications. This network structure is a recurrent neural network (RNN). The RNN passes information amongst nodes representing individual person classifications and the scene classification in order to facilitate contextual refinement of classification decisions.

Given the graph structure (i.e. probabilistic graphical model in Fig. 3), the RNN structure is built to model the connections between nodes. Distributions over values for each node i can be determined by a combination of classification results x_i of local observations, along with information passed along the graph. We denote by m information coming from related nodes along the graph.

In group activity recognition, this can model contextual relations between the actions of people in the scene. In every iteration of refinement, the values of the contextual information m will be updated, while the inputs corresponding to visual observations of individual actions or scene label remain fixed.

We represent this iterative updating process as a recurrent neural network model. In general, we take as input the local observations for all nodes $x = x_s \cup \{x_i : i = 1, \dots, M\}$. We output a set of refined classification scores¹ $c^{(t)}$ at each timestep t . These refined estimates are based on iteratively computed messages $m^{(t)}$ that pass information

¹These classification scores may in general be over different quantities from, or a subset of, the inputs.

between the nodes:

$$m^{(t)} = f(W_{mm}m^{(t-1)} + W_{xm}x + W_{cm}c^{(t-1)} + b_m) \quad (1)$$

$$c^{(t)} = f(W_{mc}m^{(t)} + W_{xc}x + b_c) \quad (2)$$

In this and all subsequent equations, variables $W_{(\cdot)}$ and $b_{(\cdot)}$ refer to the neural network parameters which are to be learned. Here f is an element-wise activation function for introducing non-linearity. At each time step, x , the scores from all image-based classifiers, is a static input into the recurrent neural network model. The “hidden contextual information” $m^{(t)}$ is updated by considering the previous contextual information $m^{(t-1)}$, the local observation x , and previous classification scores $c^{(t-1)}$. Each step of the RNN involves a single pass of aggregating information from contextual nodes within a graph to refine the scores of each node. Over a series of iterations, RNNs are used to allow finer refinements of these scores $c^{(t)}$.

The formulation above is a general-purpose inference machine for refining estimates and producing classification outputs. In the following section, we describe connections with graphical models and use this to motivate the specific choices of weight matrices and weight sharing we use for our group activity recognition model.

4.2. Belief Propagation in an RNN

Recall again the graphical model depicted in Fig. 3. This is a typical structure used in group activity recognition, with all people connected to each other and to the scene node corresponding to group activity [28, 27]. For a probabilistic graphical model of this type, a standard inference algorithm is belief propagation – rounds of passing messages between nodes would be conducted to obtain marginal distributions given observations.

We use the intuition from this probabilistic graphical model viewpoint to construct the specific form of weight matrices used in our structured inference machine.

In particular, in the graphical model formulation, we assume that all people are affecting each other and the scene. The message passing procedure starts by initializing all messages by unary energies from this graphical model. The unary energies are the inverse probability of an input image taking labels across action states, and are generated by a classifier (e.g. CNN). The initial messages are therefore independent probabilities without considering connections or smoothness between nodes.

To model the connections between entities in a group activity, three types of weights are used: (1) the weights to map the relations from individual actions to scene-level group activity, (2) from scene-level group activity to individual actions, and (3) amongst individual actions of different people. This is analogous to typical potential functions which describe pairwise energies between actions and actions or actions and scenes (similar to the shared factors in

[13]). The pairwise connections provide a data-dependent smoothing term analyzing influence between entities and are crucial in understanding a highly structured problem.

4.2.1 Messages in an RNN

In the message passing recurrent neural network, each message unit is a vector composed of a set of neurons. The message passing is conducted between the message units. The configuration of connections in the RNN is determined by the graph structure/connections between nodes. Thus, each message unit's input is its neighbouring connected messages, the static unary inputs, and outputs from the previous time step. As the message units basically correspond to the distribution of the node [29], this process, roughly speaking, can be considered as classifying one entity by other related entities and the local observation of itself. We adopted a recurrent neural network structure which shares the weights of message computation in all time steps.

Consider a message $m_{i \rightarrow j}^{(t)}$. This message coming out of person i on the edge connected to person j corresponds to the distribution of i , and is classified by average pooled scores of neighbouring persons $(\sum_k m_{k \rightarrow i}^{(t-1)})/(|N_i| - 1)$, static unary input x_i , and output $c_i^{(t-1)}$ from last time step. Denote the set of person nodes as V^P , the neighbouring person nodes of i as N_i , $N_i \subseteq V^P$, the scene node as s and the current time step as t . The mathematical form of $m_{i \rightarrow j}^{(t)}$ is:

$$f \left(W_{mm^1} \left[x_i, c_i^{(t-1)}, \frac{\sum_k m_{k \rightarrow i}^{(t-1)}}{|N_i| - 1}, m_{s \rightarrow i}^{(t-1)} \right]^T \right), \quad i \in V^P, k \in N_i \setminus j \quad (3)$$

where W_{mm^1} is the concatenation of a set of weights: $W_{mm^1} = [W_{xm}, W_{cm}, W_{mm}^{(aa)}, W_{mm}^{(sa)}]$. Since message $m_{i \rightarrow j}$ corresponds to the distribution of node i , intuitively the weights $W_{mm}^{(aa)}$ and $W_{mm}^{(sa)}$ can be considered as classifying person i 's action based on other people's action or the scene label respectively. And the unary input x_i and previous iteration's score $c_i^{(t-1)}$ are remapped via a linear transformation using W_{xm} and W_{cm} . The function $f(\cdot)$ denotes the non-linear activation function, in our case simply a softmax function to normalize the message.

Likewise, if i is a person node and s is the scene node, then the message $m_{i \rightarrow s}^{(t)}$ is:

$$f \left(W_{mm^2} \left[x_i, c_i^{(t-1)}, \frac{\sum_k m_{k \rightarrow i}^{(t-1)}}{|N_i|} \right] \right), k \in N_i \quad (4)$$

where W_{mm^2} equals $[W_{xm}, W_{cm}, W_{mm}^{(aa)}]$. Finally, if s is the scene node, and j is a person node, the message $m_{s \rightarrow j}^{(t)}$ is set as:

$$f \left(W_{mm^3} \left[x_s, c_s^{(t-1)}, \frac{\sum_k m_{k \rightarrow s}^{(t-1)}}{|N_s| - 1} \right] \right), k \in N_s \setminus p_j \quad (5)$$

where $W_{mm^3} = [W_{xm}, W_{cm}, W_{mm}^{(as)}]$.

4.2.2 Output Prediction Layer

The message units representing information all over the graph are formulated into the recurrent unit. However, to eventually classify a node in a graphical model, all messages around each node should be collected and used to perform prediction. In our model, we use a prediction layer to collect all messages around each node and infer a scene label for the group activity and action classification for each person. Through this layer, the losses on each time step are imposed on the message unit. More precisely, the prediction function for the scene level node is:

$$c_s^{(t)} = f \left(W_{hc^1} \left[x_s, \frac{\sum_k m_{k \rightarrow s}^{(t)}}{|N_s|} \right] \right), p_k \in N_s \quad (6)$$

where all notation is consistent with the previous message definitions. The activation function we used here is a softmax normalization.

Similarly, for performing action classification on a person-level node:

$$c_i^{(t)} = f \left(W_{hc^2} \left[x_i, \frac{\sum_k m_{k \rightarrow i}^{(t)}}{|N_i|}, m_{s \rightarrow i}^{(t)} \right] \right), k \in N_i \quad (7)$$

As shown above, the hidden contextual information is iteratively refined in a recurrent neural network through message passing. Similar to a standard belief propagation algorithm, the final prediction of each node is performed by collecting all related messages and is done in the prediction layer. The softmax loss imposed on the prediction results with standard mini-batch backpropagation training is used to train the model in an end-to-end framework.

5. Structure Learning for Group Activity Inference

For group activity, the structure of connections between people can greatly influence performance. In general a fully connected graph, in which all people in a scene are connected to all others, and all people are related to the group activity, could model any type of relation. However, including spurious edges relating irrelevant people introduces significant noise. Instead focusing on relevant connections can lead to better models.

Beyond this, in a highly structured group activity, the connections between people or interactions between person and scene may vary according to different situation. Also, the interactions between people could be hard to explicitly capture. For example, in a crossing-the-street scene, both people crossing the street and people waiting for the lights to change are contributing to the group activity, while the people walking behind them may become irrelevant or even bring ambiguity. Hence, connectivity of the model should

Algorithm 1 Structure Inference Machine

Inputs: frame, detected person bounding boxes
Pass image through CNN to get x_s
Pass person bounding boxes through CNN to get $\{x_i\}_{i=1}^M$
Initialize $m_{s \rightarrow i}^{(0)}$ by x_s ; $m_{i \rightarrow s}^{(0)}$, $m_{i \rightarrow j}^{(0)}$ by $\{x_i\}_{i=1}^M$
for each iteration t **do**
 for edge (i, j) **do**
 Compute messages $m_{i \rightarrow j}^{(t)}$ and $m_{j \rightarrow i}^{(t)}$ by Eq. 3-5
 end for
 for edge (i, j) **do**
 Compute gate value $g_{\langle i, j \rangle}^{(t)}$ or $g_{\langle s, i \rangle}^{(t)}$ by Eq. 8-12
 Impose gates on $m_{i \rightarrow j}^{(t)}$ and $m_{j \rightarrow i}^{(t)}$ by Eq. 13
 end for
 for each node i **do**
 Compute node prediction $c_i^{(t)}$ by Eq. 6, 7
 end for
end for
Outputs: Predicted scene label from timestep T , $c_s^{(T)}$

be able to adaptively change and adjust according to the particular input situation.

In the context of a recurrent neural network, gates are widely used as a tool for selecting information on the activation level. Both long short-term memory (LSTM) / gated recurrent units (GRUs) are proven to be successful on many tasks involving iteratively learning and gating information element-wise. In our model, we introduce the concept of “instance level” gates. An instance level gate is used to modify an edge of a graphical model which models the interactions between instances, such as a person to a person, or a person to a scene. Instead of using a vector gate to select information element-wise, we instead learn a scalar value gate function to enforce sparsity on the structure of a graphical model.

Based on the previously introduced message passing RNN model, each node will receive information passed through an edge. Intuitively, the message passed could be noisy and may harm the understanding of actions of the instance. We propose to adopt an architecture similar to a LSTM gate, by taking multiple sources of information to selectively choose connectivity of nodes. To determine whether an edge is useful we compare the messages passed along this edge with other related messages. For example, to measure the gain by including the message $m_{A \rightarrow B}$, the message content of $m_{A \rightarrow B}$ is compared to other messages from other edges, the previous iteration’s classification results, and the unary distribution on node B . The gain of including the edge AB is the average value of gains for $m_{A \rightarrow B}$ and $m_{B \rightarrow A}$. For two person nodes i and j , the mathematical definition of the gating functions for message

$m_{i \rightarrow j}$ is:

$$g_{i \rightarrow j}^{(t)} = \sigma \left(W_{hg^1} \left[x_j, c_j^{(t-1)}, m_{i \rightarrow j}^{(t)}, \frac{\sum_k m_{k \rightarrow j}(t)}{|N_j - 1|} \right] \right), \quad k \in N_j \setminus i \quad (8)$$

where $W_{hg^1} = [W_{xg}, W_{cg}, W_{mm}^{(aa)}, W_{mm}^{(aa)}]$. Here we reuse the weights $W_{mm}^{(aa)}$ which classifies a person’s action by other people’s action labels. The activation function σ squeezes the gate value into $[0, 1]$. In our model, we used a sigmoid activation function.

Similarly, for the scene node s connecting to a person node i , the gate value for message $m_{s \rightarrow i}$ is calculated as:

$$g_{s \rightarrow i}^{(t)} = \sigma \left(W_{hg^2} \left[x_i, c_i^{(t-1)}, m_{s \rightarrow i}^{(t)}, \frac{\sum_k m_{k \rightarrow i}(t)}{|N_i|} \right] \right), \quad k \in N_i \quad (9)$$

where W_{hg^2} is $[W_{xg}, W_{cg}, W_{mm}^{(sa)}, W_{mm}^{(aa)}]$. And the gating function for $m_{i \rightarrow s}$ is:

$$g_{i \rightarrow s}^{(t)} = \sigma \left(W_{hg^3} \left[x_s, c_s^{(t-1)}, m_{i \rightarrow s}^{(t)}, \frac{\sum_k m_{k \rightarrow s}(t)}{|N_s| - 1} \right] \right), \quad k \in N_s \quad (10)$$

where W_{hg^3} equals $[W_{xg}, W_{cg}, W_{mm}^{(as)}, W_{mm}^{(as)}]$. Then the gate values for an edge between a person and the scene $e_{\langle i, s \rangle}$, and between two persons $e_{\langle i, j \rangle}$, at timestep t , are calculated as:

$$g_{\langle i, s \rangle}^{(t)} = (g_{i \rightarrow s}^{(t)} + g_{s \rightarrow i}^{(t)})/2 \quad (11)$$

$$g_{\langle i, j \rangle}^{(t)} = (g_{i \rightarrow j}^{(t)} + g_{j \rightarrow i}^{(t)})/2 \quad (12)$$

After imposing the gates on relevant messages, the message units in the previous section are recalculated as:

$$m_{A \rightarrow B}^{(t)} = (g_{\langle A, B \rangle}^{(t)} \odot m_{A \rightarrow B}^{(t)}) \quad (13)$$

Note that the above equation is a general form for gated messages. The symbol \odot represents the product operation between a scalar and a vector. Nodes A and B could either be a person i or a scene node s .

Further more, to enforce the sparsity of connections between nodes and learn the most discriminative structures for each graph, we use the $L1$ regularization on the gate values. Given a particular training data sample (i.e. labeled frame) d , a graphical model is built on top of it as shown in above sections. Let E^d denote the set of edges in the graph. The total loss on gates for the data sample d is:

$$L^d = \lambda \sum_{e=1}^{|E^d|} (|g_e^d(\cdot)|) \quad (14)$$

where λ is the coefficient for this $L1$ regularization term to balance between the sparsity of the graph and prediction loss. $g_e^d(\cdot)$ is the gate value on edge e of the graph for data sample d , where the gate could either be a scene-person or

a person-person edge, or, in general the edge between two nodes. As the whole model is trained by the standard mini-batch method, the loss on each batch B is $\sum_{d \in B} (L^d)$.

This structure selection is performed for each time step of message passing. An overall summary of the structure inference machine is presented in Alg. 1.

5.1. Model Extension: Untying Weights as A Deep Inference Machine

As shown previously, in the recurrent neural network framework, a graphical model described by various potential functions can be represented as weight-shared message predictors. However, by untying the weights of message computation for each step, the model could be further extended to a deep inference machine with structure gates to selectively pass information. A model with high non-linearity could be learned through this inference process.

In summary, this approach provides a general framework for both performing message passing of a RNN built from a graphical model and learning structures of a graphical model. If the weights in the message passing steps are tied over iterations, this has direct analogy to inference in a graphical model. If the assumption of tying of weights is relaxed, instead this process corresponds to a general deep inference machine with structure learning.

6. Experiments

We demonstrate our learning framework on group activity recognition. We provide results on three challenging datasets: (1) Collective Activity Dataset [10]; (2) Collective Activity Extended Dataset [11]; and (3) Nursing Home Dataset [13].

The first two datasets are standard benchmarks widely used for group activity recognition. The Collective Activity Dataset contains 44 videos from 5 group activities (Crossing, Waiting, Queuing, Walking and Talking) and 6 individual actions (NA, Crossing, Waiting, Queuing, Walking and Talking). Collective Activity Extended omits the walking activity, due to ambiguities in its definition, and includes Jogging and Dancing categories. We follow the common protocol in [28] for the Collective Activity Dataset. The scene label for a frame is defined by choosing the activity in which the most people participate.

The Nursing Home Event Dataset contains human activities captured from fixed cameras in various rooms of a nursing home. It contains 80 videos showing 6 actions (walking, standing, bending, squatting, sitting, falling) and two scenes (fall, non-fall). This dataset consists of many challenging frames with highly cluttered scenes and large intra-class variation within actions. We adopted the same protocol used in Deng et al. [13] for evaluation.

Implementation details: Our models are implemented using the Caffe library [19]. To acquire the action scores for

Iterations	1	2	3
CRF + CNN	74.18%		
Struct. SVM + CNN	73.87%		
Tied Weights	73.86%	74.02%	74.02%
Untied Weights	73.86%	74.33%	74.33%
Gated Tied Weights	80.12%	80.90%	81.22%
Gated Untied Weights	80.12%	81.06%	81.22%

Table 1. Results on Collective Activity Dataset. Ablation study including variants of our model.

Method	Accuracy
Learning Latent Constituent [4]	75.1%
Latent SVM with Optimized Graph [28]	79.7%
Deep Struct. Model [13]	80.6%
Unified Tracking And Recognition[9]	80.6%
Cardinality Kernel [17]	83.4%
Our Model	81.2%

Table 2. Comparison with state-of-the-art methods on Collective Activity Dataset.

each person image patch or the whole frame scene score, we fine-tuned the AlexNet architecture [24] pre-trained using the ImageNet data. We assume that an image has been pre-processed by a person detector to get person image patches. The message passing recurrent neural network is trained by adding a softmax loss on top of the output from each timestep. We also found it easier to train the gates by first fixing the weights of learned predictors and then learning structures on it. The number of neurons for the RNN per layer is $\sum_{AB \in \epsilon_G} (|S_A| + |S_B|)$, where ϵ_G is the set of edges of graphical model G , and $|S_A|$ and $|S_B|$ are the numbers of states of nodes A and B respectively.

6.1. Collective Activity Dataset

We compare results of four different methods introduced in our paper with standard baselines. Table 1 provides an ablation study examining the effects of different variants/components of our model. There is a clear benefit by adopting structure gates to adaptively capture connections between nodes. Note that all the CRFs in our experiments are tuned on the validation set. Our model improved the accuracy of nodes over the whole graph: person-level action classification is improved by $\approx 6\%$ after three steps of gated message passing.

Our model is compared to state of the art methods in Table 2. The results are superior to other deep learning and structure learning models. The method of [17] achieves better results, though uses a counting kernel (cardinality kernel) which directly mimics the majority-action scene label definition of Collective Activity Dataset.

6.2. Collective Activity Extended Dataset

We also experiment with the Collective Activity Extended Dataset. As noted in Choi et al. [11], the walking

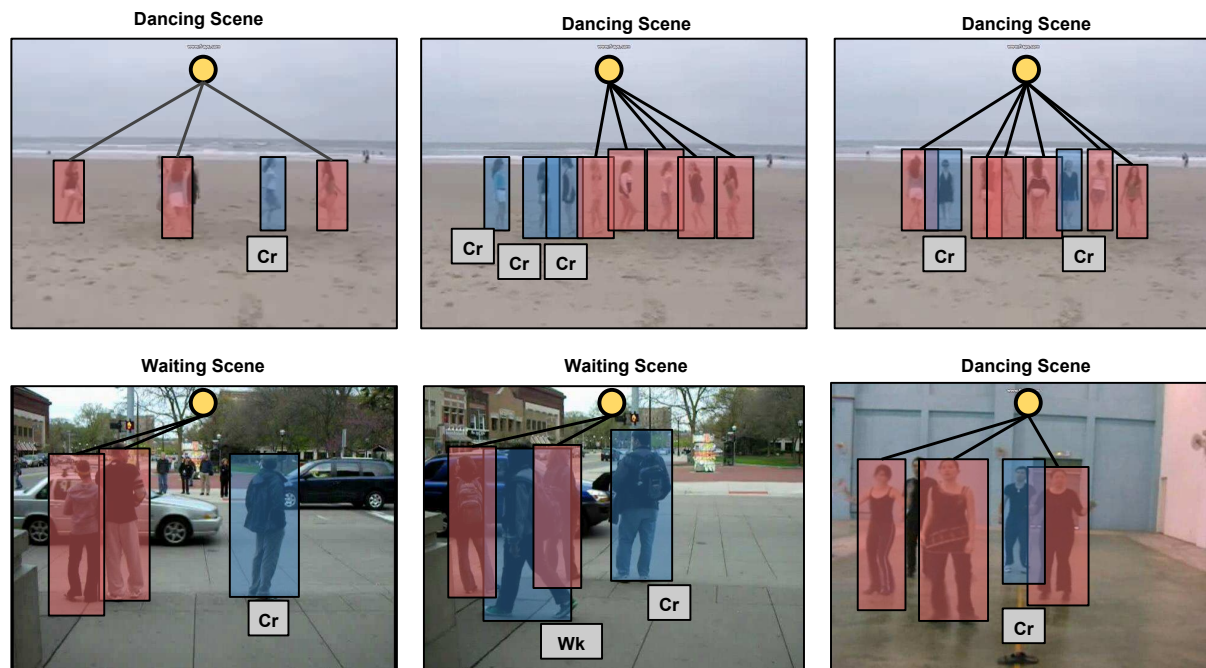


Figure 4. This figure shows visualizations of our experimental results. Note that these images are all misclassified by the fully connected graphical model. We show the scene gates learned in our model after 3 iterations of message passing and structure learning. For visualization, since the gate values are not strictly 0 or 1, we consider < 0.2 as irrelevant/noisy connection versus > 0.7 as useful connections. The red box has the same action class as the scene level node. Labels: “Cr”: Crossing, “Wk”: Walking.

Iterations	1	2	3
CRF + CNN		86.75%	
Struct. SVM + CNN		87.34%	
Tied Weights	84.45%	87.97%	87.97%
Untied Weights	84.45%	88.16%	88.16%
Gated Tied Weights	89.51%	90.14%	90.14%
Gated Untied Weights	89.51%	90.14%	90.23%

Table 3. Results on Collective Activity Extended Dataset.

action is ill-defined, hence we remove it, and include the new actions Jogging and Dancing. The two previous works choose to adopt leave-one-out for testing. However, this is very computationally intensive for a deep learning framework, and further makes hyper-parameter tuning a challenge. We choose to adopt a new train-test split with 2241 frames as training and 1106 as testing.

Results are shown in Table 3. Note that on each person, action classification is improved by $\approx 10\%$ via the structure inference process.

6.3. Nursing Home Dataset

On the Nursing Home Dataset, our person-level action classification accuracy also improved, by $\approx 4\%$ after the second iteration. The accuracy is superior to baselines including Deng et al. [13]. Note that in the Nursing Home Dataset, there is a smaller margin of improvement by adopt-

Iterations	1	2	3
CRF + CNN		83.64%	
Struct. SVM + CNN		82.08%	
Deep Struct. Model [13]		84.7%	
Tied Weights	83.68%	84.91%	84.91%
Untied Weights	83.68%	84.94%	84.94%
Gated Tied Weights	84.46%	85.32%	85.32%
Gated Untied Weights	84.46%	85.50%	85.50%

Table 4. Results on Nursing Home Dataset.

ing gating functions to learn structures. Because in each scene the irrelevant actions, such as sitting, can be identified as non-useful by simply a fully connected graphical model more easily than in the previous two datasets.

7. Conclusion

We presented a method for performing structure learning within a deep learning setting. An inference algorithm for refining estimates of individual nodes and determining connections between nodes is implemented using a recurrent neural network with gating functions. This approach was used to build a model for group activity recognition – connections between individual people in a scene and their relation to the overarching scene-level activity label are learned. This leads to improvements in accuracy over rounds of inference and structure learning via gating functions.

References

- [1] M. R. Amer, P. Lei, and S. Todorovic. Hirf: Hierarchical random field for collective activity recognition in videos. In *European Conference on Computer Vision (ECCV)*, pages 572–585. Springer, 2014. [1](#)
- [2] M. R. Amer, P. Lei, and S. Todorovic. Hirf: Hierarchical random field for collective activity recognition in videos. In *European Conference on Computer Vision (ECCV)*, pages 572–585. Springer, 2014. [2](#)
- [3] M. R. Amer, D. Xie, M. Zhao, S. Todorovic, and S.-C. Zhu. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *European Conference on Computer Vision (ECCV)*, pages 187–200. Springer, 2012. [2](#)
- [4] B. Antic and B. Ommer. Learning latent constituents for recognition of group activities in video. In *European Conference on Computer Vision (ECCV)*, pages 33–47. Springer, 2014. [7](#)
- [5] L. Bottou, Y. Bengio, and Y. Le Cun. Global training of document processing systems using graph transformer networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 489–494. IEEE, 1997. [2](#)
- [6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations (ICLR)*, 2014. [2](#)
- [7] L.-C. Chen, A. G. Schwing, A. L. Yuille, and R. Urtasun. Learning deep structured models. *arXiv preprint arXiv:1407.2538*, 2014. [2](#)
- [8] W. Choi, Y. Chao, C. Pantofaru, and S. Savarese. Discovering groups of people in images. In *ECCV*, 2014. [1](#), [3](#)
- [9] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012. [2](#), [7](#)
- [10] W. Choi, K. Shahid, and S. Savarese. What are they doing? : Collective activity classification using spatio-temporal relationship among people. In *Proc. of 9th International Workshop on Visual Surveillance (VSWS09) in conjunction with ICCV*, 2009. [3](#), [7](#)
- [11] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2011. [7](#)
- [12] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *European Conference on Computer Vision (ECCV)*, pages 48–64. Springer, 2014. [2](#)
- [13] Z. Deng, M. Zhai, L. Chen, Y. Liu, S. Muralidharan, M. Roshtkhari, , and G. Mori. Deep structured models for group activity recognition. In *British Machine Vision Conference (BMVC)*, 2015. [2](#), [5](#), [7](#), [8](#)
- [14] N. Ding, J. Deng, K. Murphy, and H. Neven. Probabilistic label relation graphs with ising models. In *International Conference on Computer Vision (ICCV)*, 2015. [2](#)
- [15] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2012–2019. IEEE, 2009. [2](#)
- [16] H. Hajimirsadeghi and G. Mori. Learning ensembles of potential functions for structured prediction with latent variables. In *International Conference on Computer Vision (ICCV)*, 2015. [1](#)
- [17] H. Hajimirsadeghi, W. Yan, A. Vahdat, and G. Mori. Visual recognition by counting instances: A multi-instance cardinality potential kernel. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. [7](#)
- [18] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006. [3](#)
- [19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. [7](#)
- [20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732. IEEE, 2014. [2](#)
- [21] S. Khamis, V. I. Morariu, and L. S. Davis. Combining per-frame and per-track cues for multi-person action recognition. In *European Conference on Computer Vision (ECCV)*, 2012. [2](#)
- [22] S. Khamis, V. I. Morariu, and L. S. Davis. A flow model for joint action recognition and identity maintenance. In *Computer Vision and Pattern Recognition (CVPR)*, 2012. [2](#)
- [23] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems (NIPS)*, pages 109–117, 2011. [2](#)
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. [2](#), [7](#)
- [25] S. Kwak, B. Han, and J. H. Han. Multi-agent event detection: Localization and role assignment. In *CVPR*, 2013. [2](#)
- [26] T. Lan, L. Sigal, and G. Mori. Social roles in hierarchical models for human activity recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2012. [1](#), [2](#)
- [27] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *Advances in Neural Information Processing Systems (NIPS)*, 2010. [4](#)
- [28] T. Lan, Y. Wang, W. Yang, S. Robinovitch, and G. Mori. Discriminative latent models for recognizing contextual group activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1549–1562, 2012. [2](#), [3](#), [4](#), [7](#)
- [29] S. Ross, D. Munoz, M. Hebert, and J. A. Bagnell. Learning message-passing inference machines for structured prediction. In *CVPR*, 2011. [2](#), [5](#)
- [30] M. Ryoo and J. Aggarwal. Stochastic representation and recognition of high-level group activities. *International Journal of Computer Vision (IJCV)*, 93(2):183–200, 2011. [2](#)
- [31] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015. [2](#)

- [32] T. Shu, D. Xie, B. Rothrock, S. Todorovic, and S.-C. Zhu. Joint inference of groups, events and human roles in aerial videos. In *CVPR*, 2015. 2
- [33] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014. 2
- [34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 2
- [35] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1799–1807, 2014. 2
- [36] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *IEEE Intl. Conference on Computer Vision (ICCV)*, 2003. 3
- [37] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2
- [38] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision (ICCV)*, 2015. 2
- [39] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury. Context-aware modeling and recognition of activities in video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2491–2498. IEEE, 2013. 2