

Learning Human-Object Interactions by Graph Parsing Neural Networks

Siyuan Qi^{*1,2}, Wenguan Wang^{*1,3}, Baoxiong Jia^{1,4}, Jianbing Shen^{†3,5}, and Song-Chun Zhu^{1,2}

¹ University of California, Los Angeles

² International Center for AI and Robot Autonomy (CARA)

³ Beijing Institute of Technology

⁴ Peking University

⁵ Inception Institute of Artificial Intelligence

syqi@cs.ucla.edu wenguanwang.ai@gmail.com baoxiongjia@ucla.edu

shenjianbing@bit.edu.cn sczhu@stat.ucla.edu

Abstract. This paper addresses the task of detecting and recognizing human-object interactions (HOI) in images and videos. We introduce the Graph Parsing Neural Network (GPNN), a framework that incorporates structural knowledge while being differentiable end-to-end. For a given scene, GPNN infers a parse graph that includes i) the HOI graph structure represented by an adjacency matrix, and ii) the node labels. Within a message passing inference framework, GPNN iteratively computes the adjacency matrices and node labels. We extensively evaluate our model on three HOI detection benchmarks on images and videos: HICO-DET, V-COCO, and CAD-120 datasets. Our approach significantly outperforms state-of-art methods, verifying that GPNN is scalable to large datasets and applies to spatial-temporal settings.

Keywords: Human-Object Interaction · Message Passing · Graph Parsing · Neural Networks

1 Introduction

The task of human-object interaction (HOI) understanding aims to infer the relationships between human and objects, such as “riding a bike” or “washing a bike”. Beyond traditional visual recognition of individual instances, *e.g.*, human pose estimation, action recognition, and object detection, recognizing HOIs requires a deeper semantic understanding of image contents. Recently, deep neural networks (DNNs) have shown impressive progress on above individual tasks of instance recognition, while relatively few methods [1, 2, 14, 38] were proposed for HOI recognition. This is mainly because it requires *reasoning* beyond *perception*, by integrating information from human, objects, and their complex relationships.

* Equal contribution. † Corresponding author.

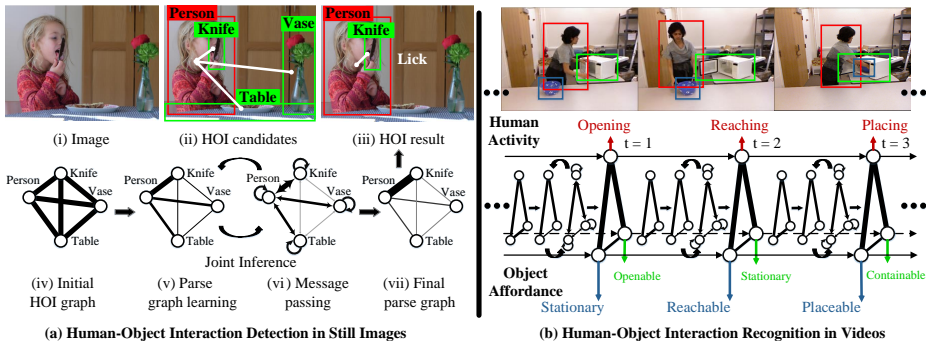


Fig. 1. Illustration of the proposed GPNN for learning HOI. GPNN offers a generic HOI representation that applies to (a) HOI detection in images and (b) HOI recognition in videos. With the integration of graphical model and neural network, GPNN can iteratively learn/infer the graph structures (a.v) and message passing (a.vi). The final parse graph explains a given scene with the graph structure (e.g., the link between the person and the knife) and the node labels (e.g., lick). A thicker edge corresponds to stronger information flow between nodes in the graph.

In this paper, we propose a novel model, Graph Parsing Neural Network (GPNN), for HOI recognition. The proposed GPNN offers a general framework that explicitly represents HOI structures with graphs and automatically parses the optimal graph structures in an end-to-end manner. In principle, it is a generalization of Message Passing Neural Network (MPNN) [12]. An overview of GPNN is shown in Fig. 1. The following two aspects motivate our design.

First, we seek a unified framework that utilizes the learning capability of neural networks and the power of graphical representations. Recent deep learning based HOI models showed promising results, but few touched how to interpret well and explicitly leverage spatial and temporal dependencies and human-object relations in such structured task. Aiming for this, we introduce GPNN. It inherits the complementary strengths of neural networks and graphical models, for forming a coherent HOI representation with strong learning ability. Specifically, with the structured representation of an HOI graph, the rich relations are explicitly utilized, and the information from individual elements can be efficiently integrated and broadcasted over the structures. The whole model and message passing operations are well-defined and fully differentiable. Thus it can be efficiently learned from data in an end-to-end manner.

Second, based on our efficient HOI representation and learning power, GPNN applies to diverse HOI tasks in both static and dynamic scenes. Previous studies for HOI achieved good performance in their specific domains (spatial [1, 14] or temporal [20, 34, 35]). However, none of them addresses a generic framework for representing and learning HOI in both images and videos. The key difficulty lies in the diverse relations between components. Given a set of human and objects candidates, there may exist an uncertain number of human-object interaction pairs (see Fig. 1 (a.ii) as an example). The relations become more complex after

taking temporal factors into consideration. Thus pre-fixed graph structures, as adopted by most previous graphical or structured DNN models [11, 20, 22, 43], are not an optimal choice. Seeking a better generalization ability, GPNN incorporates an essential *link function* for addressing the problem of graph structure learning. It learns to infer the adjacency matrix in an end-to-end manner and thus can infer a parse graph that explicitly explains the HOI relations. With such learnable graph structure, GPNN could also limit the information flow from irrelevant nodes while encouraging message to propagate between related nodes, thus improving graph parsing.

We extensively evaluate the proposed GPNN on three HOI datasets, namely HICO-DET [1], V-COCO [17] and CAD-120 [22], for HOI detection from images (HICO-DET, V-COCO) and HOI recognition and anticipation in spatial-temporal settings (CAD-120). The experimental results verify the generality and scalability of our GPNN based HOI representation and show substantial improvements over state-of-the-art approaches, including pure graphical models and pure neural networks. We also demonstrate GPNN outperforms its variants and other graph neural networks with pre-fixed structures.

This paper makes three major contributions. **First**, we propose the GPNN that incorporates structural knowledge and DNNs for learning and inference. **Second**, with a set of well defined modular functions, GPNN addresses the HOI problem by jointly performing graph structure inference and message passing. **Third**, we empirically show that GPNN offers a scalable and generic HOI representation that applies to both static and dynamic settings.

2 Related Work

Human-Object Interaction. Reasoning human actions with objects (like “playing baseball”, “playing guitar”), rather than recognizing individual actions (“playing”) or object instances (“baseball”, “guitar”), is essential for a more comprehensive understanding of what is happening in the scene. Early work in HOI understanding studied Bayesian model [15, 16], utilized contextual relationship between human and objects [47–49], learned structured representations with spatial interaction and context [8], exploited compositional models [9], or referred to a set of HOI exemplars [19]. They were mainly based on handcrafted features (*e.g.*, color, HOG, and SIFT) with object and human detectors. More recently, inspired by the notable success of deep learning and the availability of large-scale HOI datasets [1, 2], several deep learning based HOI models were then proposed. Specifically, Mallya *et al.* [29] modified Fast RCNN model [13] for HOI recognition, with the assistance of Visual Question Answering (VQA). In [38], zero-shot learning was applied for addressing the long-tail problem in HOI recognition. In [1], the human proposals, object regions, and their combinations were fed into a multi-stream network for tackling the HOI detection problem. Gkioxari *et al.* [14] estimated an action-type specific density map for identifying the interacted object locations, with a modified Faster RCNN architecture [36].

Although promising results were achieved by above deep HOI models, we still observe two unsolved issues. First, they lack a powerful tool to represent the structures in HOI tasks explicitly and encodes them into modern network architectures efficiently. Second, despite the successes in specific tasks, a complete and generic HOI representation is missing. These approaches can not be easily extended to HOI recognition from videos. Aiming to address those issues, we introduce GPNN for imposing high-level relations into DNN, leading to a powerful HOI representation that is applicable in both static and dynamic settings.

Neural Networks with Graphs/Graphical Models. In the literature, some approaches were proposed to combine graphical models and neural networks. The most intuitive approach is to build graphical models upon DNN, where the network that generates features is trained first, and its output is used to compute potential functions for the graphical predictor. Typical methods were used in human pose estimation [42], human part parsing [33, 45], and semantic image segmentation [3, 4]. These methods lack a deep integration in the sense that the computation process of graphical models cannot be learned end-to-end. Some attempts [7, 21, 31, 32, 37, 40, 44, 51] were made to generalize neural network operations (*e.g.*, convolutions) directly from regular grids (*e.g.*, images) to graphs. For the HOI problem, however, a structured representation is needed to capture the high-level spatial-temporal relations between humans and objects. Some other work integrated network architectures with graphical models [12, 20] and gained promising results on applications such as scene understanding [24, 30, 46], object detection and parsing [27, 50], and VQA [41]. However, these methods only apply to problems that have pre-fixed graph structures. Liang *et al.* [26] merged graph nodes using Long Short-Term Memory (LSTM) for human parsing problem, under the assumption that the nodes are mergeable.

Those methods achieved promising results in their specific tasks and well demonstrated the benefit in completing deep architectures with domain-specific structures. However, most of them are based on pre-fixed graph structures, and they have not yet been studied in HOI recognition. In this work, we extend previous graphical neural networks with learnable graph structures, which well addresses the rich and high-level relations in HOI problems. The proposed GPNN can automatically infer the graph structure and utilize that structure for enhancing information propagation and further inference. It offers a generic HOI representation for both spatial and spatial-temporal settings. To the best of our knowledge, this is a first attempt to integrate graph models with neural networks in a unified framework to achieve state-of-art results in HOI recognition.

3 Graph Parsing Neural Network for HOI

3.1 Formulation

For HOI understanding, human and objects are represented by nodes, and their relations are defined as edges. Given a complete HOI graph that includes all the possible relationships among human and objects, we want to automatically infer a parse graph by keeping the meaningful edges and labeling the nodes.

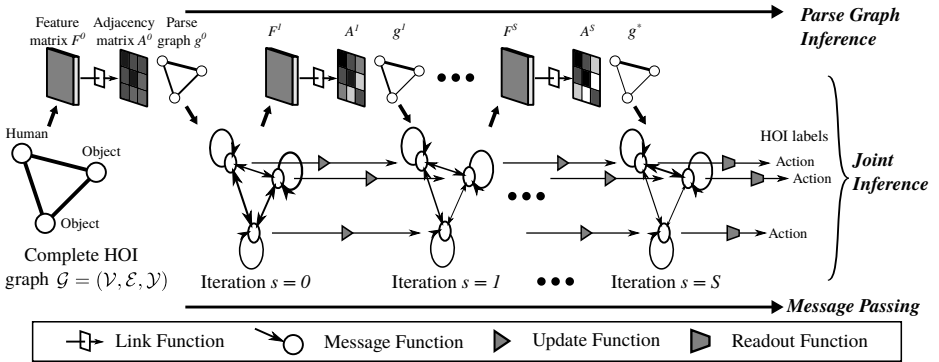


Fig. 2. Illustration of the forward pass of GPNN. GPNN takes node and edge features as input, and outputs a parse graph in a message passing fashion. The structure of the parse graph is given by a soft adjacency matrix. It is computed by the *link function* based on the features (or hidden node states). The darker the color in the adjacency matrix, the stronger the connectivity is. Then *message functions* compute incoming messages for each node as a weighted sum of the messages from other nodes. Thicker edges indicate larger information flows. The *update functions* update the hidden internal states of each node. Above process is repeated for several steps, iteratively and jointly learning the computation of graph structures and message passing. Finally, for each node, the *readout functions* output HOI action or object labels from the hidden node states. See § 3 for more details.

Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{Y})$ denote the complete HOI graph. Nodes $v \in \mathcal{V}$ take unique values from $\{1, \dots, |\mathcal{V}|\}$. Edges $e \in \mathcal{E}$ are two-tuples $e = (v, w) \in \mathcal{V} \times \mathcal{V}$. Each node v has an output state $y_v \in \mathcal{Y}$ that takes a value from a set of labels $\{1, \dots, Y_v\}$ (e.g., actions). A parse graph $g = (\mathcal{V}_g, \mathcal{E}_g, \mathcal{Y}_g)$ is a sub-graph of \mathcal{G} , where $\mathcal{V}_g \subseteq \mathcal{V}$ and $\mathcal{E}_g \subseteq \mathcal{E}$. Given the node features $\Gamma^{\mathcal{V}}$ and edge features $\Gamma^{\mathcal{E}}$, we want to infer the optimal parse graph g^* that best explains the data according to a probability distribution p :

$$\begin{aligned} g^* &= \operatorname{argmax}_g p(g|\Gamma, \mathcal{G}) = \operatorname{argmax}_g p(\mathcal{V}_g, \mathcal{E}_g, \mathcal{Y}_g|\Gamma, \mathcal{G}) \\ &= \operatorname{argmax}_g p(\mathcal{Y}_g|\mathcal{V}_g, \mathcal{E}_g, \Gamma)p(\mathcal{V}_g, \mathcal{E}_g|\Gamma, \mathcal{G}) \end{aligned} \quad (1)$$

where $\Gamma = \{\Gamma^{\mathcal{V}}, \Gamma^{\mathcal{E}}\}$. Here $p(\mathcal{V}_g, \mathcal{E}_g|\Gamma, \mathcal{G})$ evaluates the graph structure, and $p(\mathcal{Y}_g|\mathcal{V}_g, \mathcal{E}_g, \Gamma)$ is the labeling probability for the nodes in the parse graph.

This formulation provides us a principled guideline for designing the GPNN. We design the network to approximate the computations of $\operatorname{argmax}_g p(\mathcal{V}_g, \mathcal{E}_g|\Gamma, \mathcal{G})$ and $\operatorname{argmax}_g p(\mathcal{Y}_g|\mathcal{V}_g, \mathcal{E}_g, \Gamma)$. We introduce four types of functions as individual modules in the forward pass of a GPNN: *link functions*, *message functions*, *update functions*, and *readout functions* (as illustrated in Fig. 2). The link functions $L(\cdot)$ estimate the graph structure, giving an approximation of $p(\mathcal{V}_g, \mathcal{E}_g|\Gamma, \mathcal{G})$. The message, update and readout functions together resemble the belief propagation process and approximate $\operatorname{argmax}_{\mathcal{Y}_g} p(\mathcal{Y}_g|\mathcal{V}_g, \mathcal{E}_g, \Gamma)$.

Specifically, the link function (\dashv) takes edge features (\blacksquare) as input and infers the connectivities between nodes. The soft adjacency matrix (\blacksquare) is thus constructed and used as weights for messages passing through edges between nodes. The incoming messages for a node are summarized by the message function (\succ), then the hidden embedding state of the node is updated based on the messages by an update function (\blacktriangleright). Finally, readout functions (\blacktriangleright) compute the target outputs for each nodes. Those four types of functions are defined as follows:

Link Function. We first infer an adjacency matrix that represents connectivities (*i.e.*, the graph structure) between nodes by a link function. A link function $L(\cdot)$ takes the node features $\Gamma^{\mathcal{V}}$, and edge features $\Gamma^{\mathcal{E}}$ as input and outputs an adjacency matrix $A \in [0, 1]^{|\mathcal{V}| \times |\mathcal{V}|}$:

$$A_{vw} = L(\Gamma_v, \Gamma_w, \Gamma_{vw}) \quad (2)$$

where A_{vw} denotes the (v, w) -th entry of the matrix A . Here we overload the notation and let Γ_v denote node features and Γ_{vw} denote edge features. In this way, the structure of a parse graph g can be approximated by the adjacency matrix. Then we start to propagate messages over the parse graph, where the soft adjacency matrix controls the information to be passed through edges.

Message and Update Functions. Based on the learned graph structure, the message passing algorithm is adopted for inference of node labels. During belief propagation, the hidden states of the nodes are iteratively updated by communicating with other nodes. Specially, message functions $M(\cdot)$ summarize messages to nodes coming from other nodes, and update functions $U(\cdot)$ update the hidden node states according to the incoming messages. At each iteration step s , the two functions computes:

$$m_v^s = \sum_w A_{vw} M(h_v^{s-1}, h_w^{s-1}, \Gamma_{vw}) \quad (3)$$

$$h_v^s = U(h_v^{s-1}, m_v^s) \quad (4)$$

where m_v^s is the summarized incoming message for node v at s -th iteration and h_v^s is the hidden state for node v . The node connectivity A encourages the information flow between nodes in the parse graph. The message passing phase runs for S steps towards convergence. At the first step, the node hidden states h_v^0 are initialized by node features Γ_v .

Readout Function. Finally, for each node, hidden state is fed into a readout function to output a label:

$$y_v = R(h_v^S). \quad (5)$$

Here the readout function $R(\cdot)$ computes output y_v for node v by activating its hidden state h_v^S (node embeddings).

Iterative Parsing. Based on the above four functions, the messages are passed along the graph and weighted by the learned adjacency matrix A . We further extend above process into a joint learning framework that iteratively infers the graph structure and propagates the information to infer node labels. In particular, instead of learning A only at the beginning, we iteratively infer A with the

updated node information and edge features at each step s :

$$A_{vw}^s = L(h_v^{s-1}, h_w^{s-1}, m_{vw}^{s-1}). \quad (6)$$


Then the messages in Eq. 3 are redefined as:

$$m_v^s = \sum_w A_{vw}^s M(h_v^{s-1}, h_w^{s-1}, \Gamma_{vw}). \quad (7)$$

In this way, both the graph structure and the message update can be jointly and iteratively learned in a unified framework. In practice, we find such a strategy would bring better performance (detailed in § 4.3).

In next section, we show that by implementing each function by neural networks, the entire system is differentiable end-to-end. Hence all the parameters can be learned using gradient-based optimization.

3.2 Network Architecture

Link Function. Given the complete HOI graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{Y})$, we use d_V and d_E to denote the dimension of the node features and the edge features, respectively. In a message passing step s , we first concatenate all the node features (hidden states) $\{h_v^s \in \mathbb{R}^{d_V}\}_v$ and all the edge features (messages) $\{m_{vw}^s \in \mathbb{R}^{d_E}\}_{v,w}$ to form a feature matrix $F^s \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}| \times (2d_V + d_E)}$ (see  in Fig. 2). The link function is defined as a small neural network with one or several convolutional layer(s) (with $1 \times 1 \times (2d_V + d_E)$ kernels) and a *sigmoid* activation. Then the adjacency matrix $A^s \in [0, 1]^{|\mathcal{V}| \times |\mathcal{V}|}$ can be computed as:

$$A^s = \sigma(\mathbf{W}^L * F^s), \quad (8)$$

where \mathbf{W}^L is the learnable parameters of the link function network $L(\cdot)$ and $*$ denotes conv operation. The *sigmoid* operation $\sigma(\cdot)$ is for normalizing the values of the elements of A^s into $[0, 1]$. The essential effect of multiple convolutional layers with 1×1 kernels is similar to fully connected layers applied to each individual edge features, except that the filter weights are shared by all the edges. In practice, we find such operation generates good enough results and leads to a high computation efficiency.

For spatial-temporal problems where the adjacency matrices should account for the previous states, we use convolutional LSTMs [39] for modeling $L(\cdot)$ in temporal domain. At time t , the link function takes $F^{s,t}$ as input features and the previous adjacency matrix $A^{s,t-1}$ as hidden state: $A^{s,t} = \text{convLSTM}(F^{s,t}, A^{s,t-1})$. Again, the kernel size for the conv layer in convLSTM is $1 \times 1 \times (2d_V + d_E)$.

Message Function. In our implementation, the message function $M(\cdot)$ in Eq. 3 is computed by:

$$M(h_v, h_w, \Gamma_{vw}) = [\mathbf{W}_V^M h_v, \mathbf{W}_V^M h_w, \mathbf{W}_E^M \Gamma_{vw}], \quad (9)$$

where $[\cdot, \cdot]$ denotes concatenation. It concatenates the outputs of linear transforms (*i.e.*, fully connected layers parametrized by \mathbf{W}_V^M and \mathbf{W}_E^M) that takes node hidden states h_v or edge features Γ_{vw} as input.

Update Function. Recurrent neural networks [10, 18] are natural choices for simulating the iterative update process, as done by previous works [12]. Here we apply Gated Recurrent Unit (GRU) [5] as the update function, because of its recurrent nature and smaller amount of parameters. Thus the update function in Eq. 4 is implemented as:

$$h_v^s = U(h_v^{s-1}, m_v^s) = GRU(h_v^{s-1}, m_v^s), \quad (10)$$

where h_v^s is the hidden state and m_v^s is used as input features. As demonstrated in [25], the GRU is more effective than vanilla recurrent neural networks.

Readout Function. A typical implementation of readout functions is combining several fully connected layers (parameterized by \mathbf{W}^R) followed by an activation function:

$$y_v = R(h_v^S) = \varphi(\mathbf{W}^R h_v^S). \quad (11)$$

Here the activation function $\varphi(\cdot)$ can be used as *softmax* (one-class outputs) or *sigmoid* (multi-class outputs) according to different HOI tasks.

In this way, the entire GPNN is implemented to be fully differentiable and end-to-end trainable. The loss for specific HOI task can be computed for the outputs of readout functions, and the error can propagate back according to chain rule. In next section, we will offer more details for implementing GPNN for HOI tasks on spatial and spatial-temporal settings and present qualitative as well as quantitative results.

4 Experiments

To verify the effectiveness and generic applicability of GPNN, we perform experiments on two HOI problems: i) HOI detection in images [1, 17], and ii) HOI recognition and anticipation from videos [22]. The first experiment is performed on HICO-DET [1] and V-COCO [17] datasets, showing that our approach is scalable to large datasets (about 60K images in total) and achieves a good detection accuracy over a large number of classes (more than 600 classes of HOIs). The second experiment is reported on CAD-120 dataset [22], showing that our method is well applicable to spatial-temporal domains.

4.1 Human-Object Interaction Detection in Images

For HOI detection in an image, the goal is to detect pairs of a human and an object bounding box with an interaction class label connecting them.

Datasets. We use HICO-DET [1] and V-COCO [17] datasets for benchmarking our GPNN model. HICO-DET provides more than 150K annotated instances of human-object pairs in 47,051 images (37,536 training and 9,515 testing). It has the same 80 object categories as MS-COCO [28] and 117 action categories. V-COCO is a subset of MS-COCO [28]. It consists of a total of 10,346 images with 16,199 people instances, where ~ 2.5 K images in the train set, ~ 2.8 K images for validation and ~ 4.9 K images for testing. Each annotated person has binary

Table 1. HOI detection results (mAP) on HICO-DET dataset [1]. Higher values are better. The best scores are marked in **bold**.

| Methods | Full (mAP %) \uparrow | Rare (mAP %) \uparrow | Non-rare (mAP %) \uparrow |
|-----------------------------|-------------------------|-------------------------|-----------------------------|
| Random | 1.35×10^{-3} | 5.72×10^{-4} | 1.62×10^{-3} |
| Fast-RCNN(union) [13] | 1.75 | 0.58 | 2.10 |
| Fast-RCNN(score) [13] | 2.85 | 1.55 | 3.23 |
| HO-RCNN [1] | 5.73 | 3.21 | 6.48 |
| HO-RCNN+IP [1] | 7.30 | 4.68 | 8.08 |
| HO-RCNN+IP+S [1] | 7.81 | 5.37 | 8.54 |
| Gupta <i>et al.</i> [17] | 9.09 | 7.02 | 9.71 |
| Shen <i>et al.</i> [38] | 6.46 | 4.24 | 7.12 |
| InteractNet [14] | 9.94 | 7.16 | 10.77 |
| GPNN | 13.11 | 9.34 | 14.23 |
| <i>Performance Gain</i> (%) | 31.89 | 30.45 | 32.13 |

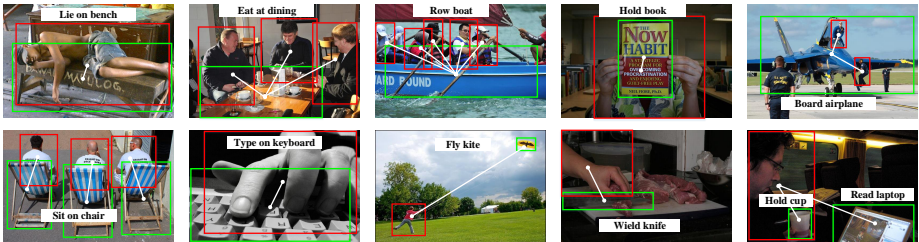


Fig. 3. HOI detection results on HICO-DET [1] test images. Human and objects are shown in red and green rectangles, respectively. Best viewed in color.

labels for 26 different action classes. Note that three actions (*i.e.*, *cut*, *eat*, and *hit*) are annotated with two types of targets: *instrument* and *direct object*.

Implementation Details. Humans and objects are represented by nodes in the graph, while human-object interactions are represented by edges. In this experiment, we use a pre-trained deformable convolutional network [6] for object detection and features extraction. Based on the detected bounding boxes, we extract node features ($7 \times 7 \times 80$) from the position-sensitive region of interest (PS RoI) pooling layer from the deformable ConvNet. We extract the edge feature from a combined bounding box, *i.e.*, the smallest bounding box that contains both two nodes' bounding boxes. The functions of GPNN are implemented as follows. We use a convolutional network (128-128-1)-Sigmoid(\cdot) with 1×1 kernels for the link function. The message functions are composed of a fully connected layer, concatenation, and summation. For a node v , the neighboring node feature Γ_w and edge feature Γ_{vw} are passed through a fully connected layer and concatenated. The final incoming message is a weighted sum of messages from all neighboring nodes. Specifically, the message for node v coming from node w through edge $e = (v, w)$ is the concatenation of output from $\text{FC}(d_V - d_V)$ and $\text{FC}(d_E - d_E)$. A $\text{GRU}(d_V)$ is used for the update function. The propagation step number S

Table 2. HOI detection results (mAP) on V-COCO [17] dataset. Legend: *Set 1* indicates 18 HOI actions with one object, and *Set 2* corresponds to 3 HOI actions (i.e., *cut*, *eat*, *hit*) with two objects (*instrument* and *object*).

| Method | Set 1 (mAP %) \uparrow | Set 2 (mAP %) \uparrow | Ave. (mAP %) \uparrow |
|-----------------------------|--------------------------|--------------------------|-------------------------|
| Gupta <i>et al.</i> [17] | 33.5 | 26.7 | 31.8 |
| InteractNet [14] | 42.2 | 33.2 | 40.0 |
| GPNN | 44.5 | 42.8 | 44.0 |
| <i>Performance Gain</i> (%) | 5.5 | 28.9 | 10.0 |

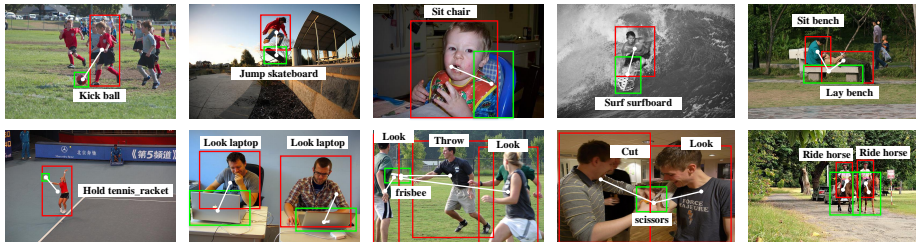


Fig. 4. HOI detection results on V-COCO [17] test images. Human and objects are shown in red and green rectangles, respectively. Best viewed in color.

is set to be 3. For the readout function, we use a $FC(d_V-117)$ -Sigmoid(\cdot) and $FC(d_V-26)$ -Sigmoid(\cdot) for HICO-DET and V-COCO, respectively.

The probability of an HOI label of a human-object pair is given by the product of the final output probabilities from the human node and the object node. We employ an L1 loss for the adjacency matrix. For the node outputs, we use a weighted multi-class multi-label hinge loss. The reasons are two-folds: the training examples are not balanced, and it is essentially a multi-label problem for each node (there might not even exist a meaningful human-object interaction for detected humans and objects).

Our model is implemented using PyTorch and trained with a machine with a single Nvidia Titan Xp GPU. We start with a learning rate of 1e-3, and the rate decays every 5 epochs by 0.8. The training process takes about 20 epochs (~ 15 hours) to roughly converge with a batch size of 32.

Comparative Methods. We compare our method with eight baselines: (1) Fast-RCNN (union) [13]: for each human-object proposal from detection results, their attention windows are used as the region proposal for Fast-RCNN. (2) Fast-RCNN (score) [13]: given human-object proposals, HOI is predicted by linearly combining the human and object detection scores. (3) HO-RCNN [1]: a multi-stream architecture with a ConvNet to classify human, object and human-object proposals, respectively. The final output is computed by combining the scores from all the three streams. (4) HO-RCNN+IP [1] and (5) HO-RCNN+IP+S [1]: HO-RCNN with additional components. Interaction Patterns (IP) acts as a attention filter to images. S is an extra path with a single neuron that uses the

raw object detection score to produce an offset for the final detection. More detailed descriptions of above five baselines can be found in [1]. (6) Gupta *et al.* [17]: trained based on Fast-RCNN [13]. We use the scores reported in [14]. (7) Shen *et al.* [38]: final predictions are from two Faster RCNN [36] based networks which are trained for predicting verb and object classes, respectively. (8) InteractNet [14]: a modified Faster RCNN [36] with an additional human-centric branch that estimates an action-specific density map for locating objects.

Experiment Results. Following the standard settings in HICO-DET and V-COCO benchmarks, we evaluate HOI detection using mean average precision (mAP). An HOI detection is considered as a true positive when the human detection, the object detection, and the interaction class are all correct. The human and object bounding boxes are considered as true positives if they overlap with a ground truth bounding boxes of the same class with an intersection over union (IoU) greater than 0.5. For HICO-DET dataset, we report the mAP over three different HOI category sets: i) all 600 HOI categories in HICO (Full); ii) 138 HOI categories with less than 10 training instances (Rare); and iii) 462 HOI categories with 10 or more training instances (Non-Rare). For V-COCO dataset, since we concentrate on HOI detection, we report the mAP on three groups: i) 18 HOI action classes with one target object; ii) 3 HOI categories with two types of objects; iii) all 24 ($=18 + 3 \times 2$) HOI classes. Results are evaluated on the test sets and reported in Table 1 and Table 2.

As shown in Table 1, the proposed GPNN substantially outperforms the comparative methods, achieving **31.89%**, **30.45%**, and **32.13%** improvement over the second best methods on the three HOI category sets on the HICO-DET dataset. The results on V-COCO dataset (in Table 2) also consistently demonstrate the superior performance of the proposed GPNN. Two important **conclusions** can be drawn from the results: **i)** our method is scalable to large datasets; **ii)** and our method performs better than pure neural network. Some visual results can be found in Fig. 3 and Fig. 4.

4.2 Human-Object Interaction Recognition in Videos

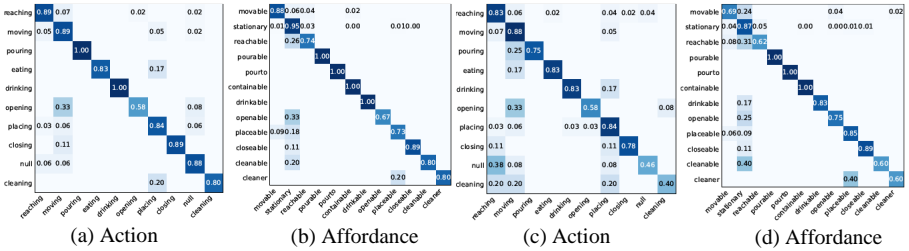
The goal of this experiment is to detect and predict the human sub-activity labels and object affordance labels as the human-object interaction progresses in videos. The problem is challenging since it involves complex interactions that humans make with multiple objects, and objects also interact with each other.

CAD-120 dataset [22]. It has 120 RGB-D videos of 4 subjects performing 10 activities, each of which is a sequence of sub-activities involving 10 actions (*e.g.*, reaching, opening), and 12 object affordances (*e.g.*, reachable, openable) in total.

Implementation Details. The link function is implemented as: convLSTM(1024-1024-1024-1)-Sigmoid(\cdot) (*i.e.*, a four-layer convLSTM). We use the same architecture as the previous experiment for message functions and update functions: $[\text{FC}(d_V-d_V), \text{FC}(d_E-d_E)]$ for message function and $\text{GRU}(d_V)$ for update function. The propagation step number S is set to be 3. We use a $\text{FC}(d_V-10)$ -Softmax(\cdot) and a $\text{FC}(d_V-12)$ -Softmax(\cdot) for readout functions of sub-activity and object affordance detection/anticipation, respectively. We employ an L1 loss for

Table 3. Human activity detection and future anticipation results on CAD-120 [22] dataset, measured via F1-score.

| Method | Detection (F1-score) \uparrow | | Anticipation (F1-score) \uparrow | |
|----------------------------|---------------------------------|----------------------|------------------------------------|----------------------|
| | Sub-activity(%) | Object Affordance(%) | Sub-activity(%) | Object Affordance(%) |
| ATCRF [22] | 80.4 | 81.5 | 37.9 | 36.7 |
| S-RNN [20] | 83.2 | 88.7 | 62.3 | 80.7 |
| S-RNN (multi-task) [20] | 82.4 | 91.1 | 65.6 | 80.9 |
| GPNN | 88.9 | 88.8 | 75.6 | 81.9 |
| <i>Performance Gain(%)</i> | 8.1 | - | 15.2 | 1.2 |

**Fig. 5. Confusion matrices of HOI detection (a)(b) and anticipation (c)(d) results on CAD-120 [22] dataset. Zoom in for more details.**

the adjacency matrix and a cross entropy loss for the node outputs. We use the publicly available node and edge features from [23].

Comparative Methods. We compare our method with two baselines: anticipatory temporal CRF (ATCRF) [22] and structural RNN (S-RNN) [20]. ATCRF is a top-performing graphical model approach for this problem, while S-RNN is the state-of-art method using structured neural networks. ATCRF models the human activities through a spatial-temporal conditional random field. S-RNN casts a pre-defined spatial-temporal graph as an RNN mixture by representing nodes and edges as LSTMs.

Experiment Results. In Table 3 we show the quantitative comparison of our method with other competitors. It shows the F1-scores averaged over all classes on detection and activity anticipation tasks. GPNN greatly improves over ATCRF and S-RNN, especially on anticipation task. Our method outperforms the other two for the following reasons. i) Comparing to ATCRF limited to the Markov assumption, our method allows arbitrary graph structures with improved representation ability. ii) Our method enjoys the benefit of deep integration of graphical models and neural networks and can be learned in an end-to-end manner. iii) Rather than relying on a pre-fixed graph structure as in S-RNN, we infer the graph structure via learning an adjacency matrix and thus be able to control the information flow between nodes during message passing. Fig. 5 show the confusion matrices for detecting and predicting the sub-activities and object affordances, respectively. From above results we can draw two im-

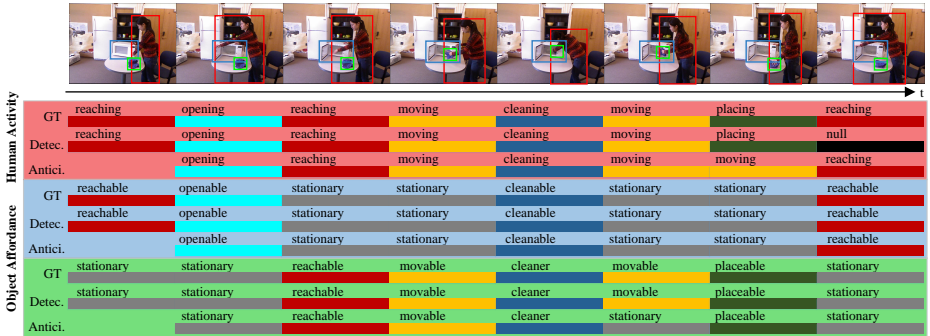


Fig. 6. HOI detection results on a “cleaning objects” activity on CAD-120 [22] dataset. Human are shown in red rectangle. Two objects are shown in green and blue rectangles, respectively. Detection and anticipation results are shown by different bars. For anticipation task, the label of the sub-activity at time t is anticipated at time $t-1$.

portant **conclusions:** **i)** our method is well applicable to the spatio-temporal domain; and **ii)** our method outperforms pure graphical models (*e.g.*, ATCRF) and deep networks with pre-fixed graph structures (*e.g.*, S-RNN). Fig. 6 shows a qualitative visualization of “cleaning objects”. We show one representative frame for each sub-activity as well as the corresponding detections and anticipations.

4.3 Ablation Study

In this section, we analyze the contributions of different model components to the final performance and examine the effectiveness of our main assumptions. Table 4 shows the detailed results on all three datasets.

Integration of DNN with Graphical Model. We first examine the influence of integrating DNN with a graphical model. We directly feed the features, which are originally used for GPNN, into different fully connected networks for predicting HOI action or object classes. From Table 4, we can observe the performance of *w/o graph* is significantly worse than GPNN model over various HOI datasets. This supports our view that modeling high-level structures and leveraging learning capabilities of DNNs together is essential for HOI tasks.

GPNN with Fixed Graph Structures. In § 3, GPNN automatically infers graph structures (*i.e.*, parse graph) via learning a soft adjacency matrix. To assess this strategy, we fix all the entries in the soft adjacency matrices to be constant 1. This way the graph structures are fixed and the information flow between nodes are not weighted. For *constant graph* baseline, we see obvious performance decrease, compared with the full GPNN model. This indicates that inferring graph structures is critical to get reasonable performance.

GPNN without Supervision on Link Functions. We perform experiments by turning off the L1 loss on adjacency matrices (*w/o graph loss* in Table 4). We can observe that the intermediate L1 loss is effective, further verifying our design to learn the graph structure. Another interesting observation is that training

Table 4. Ablation study of GPNN model. Higher values are better.

| Aspect | Method | V-COCO [17] | | | HICO-DET [1] | | | CAD-120 [22] | | | |
|---------------------------|------------------------|------------------------------------|-------------|-------------|------------------------------------|-------------|--------------|--------------------------------------|----------------|---------------------------------------|----------------|
| | | HOI Detection mAP(%) \uparrow | | | HOI Detection mAP(%) \uparrow | | | HOI Detec. F1-score(%) \uparrow | | HOI Antici. F1-score(%) \uparrow | |
| | | Set 1 | Set 2 | Ave. | Full | Rare | Non-rare | Sub-activity | Object Aff.(%) | Sub-activity | Object Aff.(%) |
| | GPNN (3 iterations) | 44.5 | 42.8 | 44.0 | 13.11 | 9.34 | 14.23 | 88.9 | 88.8 | 75.6 | 81.9 |
| <i>graph structure</i> | w/o graph | 27.4 | 30.0 | 28.1 | 7.88 | 2.04 | 9.62 | 50.2 | 20.8 | 32.3 | 19.6 |
| | constant graph | 34.6 | 33.3 | 34.3 | 8.75 | 1.94 | 10.79 | 85.3 | 85.6 | 73.8 | 79.1 |
| | w/o graph loss | 37.7 | 40.5 | 38.4 | 8.15 | 6.24 | 8.72 | 85.2 | 85.8 | 74.7 | 79.2 |
| | w/o joint parsing | 43.6 | 39.4 | 42.5 | 10.17 | 5.81 | 11.47 | 79.3 | 79.2 | 74.7 | 80.3 |
| <i>iterative learning</i> | 1 iteration | 42.0 | 40.7 | 41.7 | 11.38 | 7.27 | 12.61 | 80.5 | 80.7 | 75.2 | 81.1 |
| | 2 iterations | 44.1 | 42.2 | 43.6 | 12.37 | 9.01 | 13.38 | 87.9 | 86.1 | 76.1 | 81.5 |
| | 4 iterations | 43.6 | 40.9 | 42.9 | 12.39 | 8.95 | 13.41 | 87.9 | 85.7 | 75.5 | 80.6 |

the model without this loss has a similar effect to training with constant graph. Hence supervision on the graph is fairly important.

Jointly Learning Parse Graph and Message Passing. We next study the effect of jointly learning graph structures and message passing. By isolating graph parsing from message passing, we obtain *w/o joint parsing*, where the adjacency matrices are directly computed by link functions from edge features at the beginning. We observe a performance decrease in Table 4, showing that learning graph structures and message passing together indeed boost the performance.

Iterative Learning Process. Next we examine the effect of iterative message passing, we report three baselines: *1 iteration*, *2 iterations*, and *4 iterations*, which correspond to the results from different message passing iterations. The baseline *GPNN* (first row in Table 4) are the results after three iterations. From the results we observe that the iterative learning process is able to gradually improve the performance in general. We also observe that when the iteration round is increased to a certain extent, the performance drops slightly.

5 Conclusion

In this paper, we propose Graph Parsing Neural Network (GPNN) for inferring a parse graph in an end-to-end manner. The network can be decomposed into four distinct functions, namely link functions, message functions, update functions and readout functions, for iterative graph inference and message passing. GPNN provides a generic HOI representation that is applicable in both spatial and spatial-temporal domains. We demonstrate a substantial performance gain on three HOI datasets, showing the effectiveness of the proposed framework.

Acknowledgments. The authors thank Prof. Ying Nian Wu from UCLA Statistics Department for helpful comments on this work. This research is supported by DARPA XAI N66001-17-2-4029, ONR MURI N00014-16-1-2007, ARO W911NF1810296, and N66001-17-2-3602.

References

1. Chao, Y.W., Liu, Y., Liu, X., Zeng, H., Deng, J.: Learning to detect human-object interactions. In: WACV (2018)
2. Chao, Y.W., Wang, Z., He, Y., Wang, J., Deng, J.: HICO: A benchmark for recognizing human-object interactions in images. In: ICCV (2015)
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. PAMI (2016)
4. Chen, L.C., Schwing, A., Yuille, A., Urtasun, R.: Learning deep structured models. In: ICML (2015)
5. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation* p. 103 (2014)
6. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. ICCV (2017)
7. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: NIPS (2016)
8. Delaitre, V., Sivic, J., Laptev, I.: Learning person-object interactions for action recognition in still images. In: NIPS (2011)
9. Desai, C., Ramanan, D.: Detecting actions, poses, and objects with relational phraselets. In: ECCV (2012)
10. Elman, J.L.: Finding structure in time. *Cognitive science* **14**(2), 179–211 (1990)
11. Fang, H.S., Xu, Y., Wang, W., Zhu, S.C.: Learning pose grammar to encode human body configuration for 3d pose estimation. In: AAAI (2018)
12. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. ICML (2017)
13. Girshick, R.: Fast R-CNN. In: ICCV (2015)
14. Gkioxari, G., Girshick, R., Dollár, P., He, K.: Detecting and recognizing human-object interactions. In: CVPR (2018)
15. Gupta, A., Davis, L.S.: Objects in action: An approach for combining action understanding and object perception. In: CVPR (2007)
16. Gupta, A., Kembhavi, A., Davis, L.S.: Observing human-object interactions: Using spatial and functional compatibility for recognition. PAMI **31**(10), 1775–1789 (2009)
17. Gupta, S., Malik, J.: Visual semantic role labeling. arXiv preprint arXiv:1505.04474 (2015)
18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* **9**(8), 1735–1780 (1997)
19. Hu, J.F., Zheng, W.S., Lai, J., Gong, S., Xiang, T.: Recognising human-object interaction via exemplar based modelling. In: ICCV (2013)
20. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-RNN: Deep learning on spatio-temporal graphs. In: CVPR (2016)
21. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
22. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. PAMI (2016)
23. Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from RGB-D videos. *The International Journal of Robotics Research* (2013)

24. Li, R., Tapaswi, M., Liao, R., Jia, J., Urtasun, R., Fidler, S.: Situation recognition with graph neural networks. ICCV (2017)
25. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. ICLR (2016)
26. Liang, X., Lin, L., Shen, X., Feng, J., Yan, S., Xing, E.P.: Interpretable structure-evolving lstm. ICCV (2017)
27. Liang, X., Shen, X., Feng, J., Lin, L., Yan, S.: Semantic object parsing with graph lstm. In: ECCV (2016)
28. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
29. Mallya, A., Lazebnik, S.: Learning models for actions and person-object interactions with transfer to question answering. In: ECCV (2016)
30. Marino, K., Salakhutdinov, R., Gupta, A.: The more you know: Using knowledge graphs for image classification. CVPR (2016)
31. Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. CVPR (2016)
32. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: ICML (2016)
33. Park, S., Nie, X., Zhu, S.C.: Attribute and-or grammar for joint parsing of human pose, parts and attributes. PAMI (2017)
34. Qi, S., Huang, S., Wei, P., Zhu, S.C.: Predicting human activities using stochastic grammar. In: ICCV (2017)
35. Qi, S., Jia, B., Zhu, S.C.: Generalized earley parser: Bridging symbolic grammars and sequence data for future prediction. In: ICML (2018)
36. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
37. Seo, Y., Defferrard, M., Vandergheynst, P., Bresson, X.: Structured sequence modeling with graph convolutional recurrent networks. arXiv preprint arXiv:1612.07659 (2016)
38. Shen, L., Yeung, S., Hoffman, J., Mori, G., Fei-Fei, L.: Scaling human-object interaction recognition through zero-shot learning. In: WACV (2018)
39. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: NIPS (2015)
40. Simonovsky, M., Komodakis, N.: Dynamic edge-conditioned filters in convolutional neural networks on graphs. CVPR (2017)
41. Teney, D., Liu, L., Hengel, A.v.d.: Graph-structured representations for visual question answering. CVPR (2017)
42. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: NIPS (2014)
43. Wang, W., Xu, Y., Shen, J., Zhu, S.C.: Attentive fashion grammar network for fashion landmark detection and clothing category classification. In: CVPR (2018)
44. Wu, Z., Lin, D., Tang, X.: Deep markov random field for image modeling. In: ECCV (2016)
45. Xia, F., Zhu, J., Wang, P., Yuille, A.L.: Pose-guided human parsing by an And/Or graph using pose-context features. In: AAAI (2016)
46. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. ICCV (2017)
47. Yao, B., Fei-Fei, L.: Grouplet: A structured image representation for recognizing human and object interactions. In: CVPR (2010)

48. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: CVPR (2010)
49. Yao, B., Jiang, X., Khosla, A., Lin, A.L., Guibas, L., Fei-Fei, L.: Human action recognition by learning bases of action attributes and parts. In: ICCV. pp. 1331–1338 (2011)
50. Yuan, Y., Liang, X., Wang, X., Yeung, D.Y., Gupta, A.: Temporal dynamic graph LSTM for action-driven video object detection. ICCV (2017)
51. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: ICCV (2015)