

HR-RCNN: Hierarchical Relational Reasoning for Object Detection

Hao Chen, Abhinav Shrivastava
{chenh,abhinav}@cs.umd.edu

University of Maryland
College Park, USA

Abstract

Incorporating relational reasoning in neural networks for object recognition remains an open problem. Although many attempts have been made for relational reasoning, they generally only consider a single type of relationship. For example, pixel relations through self-attention (e.g., non-local networks), scale relations through feature fusion (e.g., feature pyramid networks), or object relations through graph convolutions (e.g., reasoning-RCNN). Little attention has been given to more generalized frameworks that can reason across these relationships. In this paper, we propose a hierarchical relational reasoning framework (HR-RCNN) for object detection, which utilizes a novel graph attention module (GAM). This GAM is a concise module that enables reasoning across heterogeneous nodes by operating on the graph’s edges directly. Leveraging heterogeneous relationships, our HR-RCNN shows great improvement on COCO dataset, for both object detection and instance segmentation.

1 Introduction

Even though convolutional neural networks (ConvNets) [22, 44, 57] have revolutionized the field of object recognition in recent years [23, 42, 43, 51, 55, 56], they are still fairly limited in their ability to *explicitly* model and reason about a myriad of contextual relationships in images [5]. Standard feedforward ConvNets ([34, 57]) rely on a data-driven approach to implicitly learn these relationships, *i.e.*, given enough data, the model should learn whatever is necessary implicitly without explicitly modeling context. This is contrary to human visual pathways, which rely heavily on context and relational reasoning for perception [5, 24, 48, 54]. To address this, many efforts have been made to encode context and relational information in ConvNets, ranging from designing architectures that better learn relationships to explicitly modeling relations by utilizing graph formulations.

We characterize the different relation information into three groups: pixel relations, scale relations, and object relations. For modeling pixel relations, conditional random fields (CRF) [53] model interactions between all (or selected) pairs of pixels in an image and have been successfully integrated into neural networks widely [0, 0, 72, 84]. Similarly, works on non-local relationships [0, 26, 58, 81] utilize a self-attention mechanism to model relationships. Both these approaches rely on assimilating information via their pixel-connectivity to improve feature representations. For scale relations, many efforts have been made on fusing features across scales to alleviate the discrepancy of feature maps from different levels of bottom-up hierarchy and feature scale-space, including top-down information flow [76, 41,

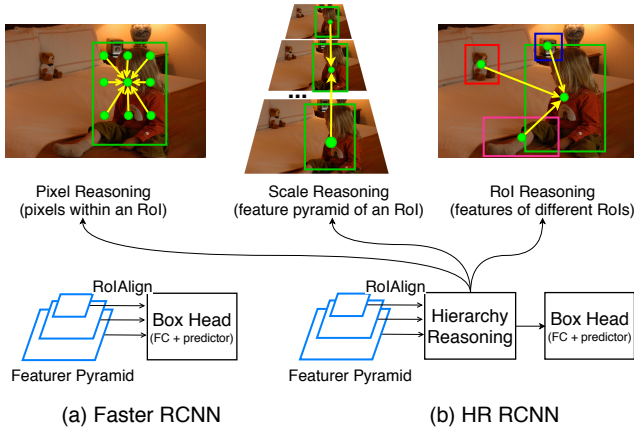


Figure 1: The Hierarchical Relational Reasoning Framework for object detection, where pixel relationships, scale relationships, RoI relationships are incorporated in one network.

[59], an extra bottom-up information path [82, 44, 70], multiple hourglass structures [4, 83], concatenating features from different layers [4, 21, 89, 61] or different tasks [64], gradual multi-stage local information fusions [60, 77], pyramid convolutions [69], *etc.* Even though standard design principles for scale relations are emerging for ConvNet architectures, the problem is far from being solved. For object relations, some early works like [11, 27, 61, 46] used annotated relationships or handcrafted linguistic knowledge to build explicit relationships between object classes; and more recently, several methods [11, 28, 45, 73, 74] learn the relational graph between different regions’ visual features to improve the handcrafted object relation graph further. Although all these approaches attempt to embed relational reasoning one way or another, little attention has been given to a more generic and unified model that can integrate these seamlessly. Moreover, these relations are not independent, but their dependencies are not well established. Therefore, we propose an approach to unify these different contextual relations (*viz.*, pixel, scale, and object relations) in a single model, that also provides a principled way to explore relational hierarchies.

In this paper, we present a Hierarchical Relational framework for object detection (HR-RCNN), which is illustrated in Fig. 1. We build on a Faster R-CNN (Fig. 1(a)) detection model, where a backbone network extracts feature pyramid and generates region proposals for an image, the per-region features are extracted from a specific level of the feature pyramid, and these features are input to a box head and processed separately. In contrast to this paradigm, when given region proposals and a feature pyramid, our HR-RCNN (Fig. 1(b)) inserts a hierarchical relational reasoning (HR) component between the feature pyramid and box head. Specifically, we embed three relational reasoning components in the HR component: a pixel graph, a scale graph, and a region-of-interest (RoI) graph. To simplify and unify the model architecture, we design a novel and concise graph attention module (GAM), illustrated in Fig. 2 which can assimilate information from heterogeneous graphs. In GAM, every node encodes the semantic and spatial distance from its neighbors into edges, and outputs attention weights by directly operating on the edges. Then, following standard graph neural network (GNN) approaches [3, 19, 20, 29, 61, 63], we enhance the node feature by assimilating messages from its neighbors weighted by their respective relations. By reasoning through heterogeneous nodes, our GAM aggregates information from the whole graph

and leads to a refined representation. Since we have multiple relation graphs, we build a hierarchy between these relations. and it leads to our final HR-RCNN architecture. We conduct extensive experiments on the COCO object detection and instance segmentation dataset to demonstrate the efficiency of our approach and provide a comprehensive analysis.

We summarize our contributions into three parts. **First**, we propose a hierarchical relational reasoning framework which integrates pixel relations, scale relations, and RoI relations into a single model. **Second**, we design a concise graph attention module for visual reasoning and explore different ways of build hierarchies of relation graphs. **Finally**, we demonstrate consistent improvements with HR-RCNN on COCO dataset.

2 Related Works

Pixel relations. Encoding context information through pixel relation has a long history in computer vision. The standard paradigm before deep learning was the algorithm by [65] that attempted to model pixel relation of all pixel pairs through conditional random fields (CRF), under mild assumptions. And while [11, 12, 72, 84] utilize similar CRF as a separate module in their neural networks, the process remains cumbersome and computationally expensive. Therefore, learning-based approaches have recently drawn more attention. In that space, deformable convolution [13, 86] learns the offsets with respect to a predefined grid to generate content-adaptive inputs. Self-attention methods [2, 76, 68, 81], on the other hand, model pairwise relationships and generate attention weights through scaled-dot-product. While most pixel relation modules are adopted in the backbone for scene understanding of the whole image, our pixel graph is built only from the pixel features within an RoI.

Scale relations. The literature to learn scale invariance and relationships across scales can be divided into two groups: image pyramids and feature pyramids. In recent years, image pyramid approaches, like SNIP [68] and SNIPER [69], introduce scale normalization to improve the performance efficiently. Approaches that model feature pyramids attempt to fuse information from low-level features (rich in details) and high-level features (rich in semantic information). Towards this, TDM [66] and Feature Pyramid Networks (FPN) [47] introduce a top-down and lateral connections to integrate information of multiple levels and thus passing on semantic information into low-level features. Based on FPN, many structure enhancement are proposed by PANet [44], Bi-FPN [62], NAS-FPN [17], SEPC [69], and DetectorRS [49]. For HR-RCNN, we incorporate the scale relations in a scale graph, where nodes are different levels of corresponding features in the feature pyramid.

Object relations. Visual reasoning from the interaction or dependency information between objects has been widely studied across a wide range of vision tasks, such as image classification [46], scene parsing [78], scene graph generation [18, 58] and large-scale object detection [11, 73, 74]. Object relation attempts to model contextual relationships between objects for visual understanding. Recent works utilize such relationships by an explicit hand-crafted knowledge graph [11, 77, 81, 46], or an implicit learning graph [11, 45]. For object detection, the literature has explored region-region relationships, class-region relationships, and class-class relationships. For example, relation network [75] incorporates region-region relationships using self-attention, SGRN [73] reasons based on a spatial-aware region-region graph including both appearance and spatial dependencies, Reasoning-RCNN [74] builds a class-class graph with global image-wise information, and Chen, Xinlei *et al.* [11] iteratively stacks multiple graphs in a reasoning framework. Compared to these, our HR-RCNN

encodes three levels of relationships (pixel, scale, and RoI) in a unified framework.

Graph attention network. Built upon graph convolution networks (GCN), graph attention networks (GAT) [63] proposes a self-attention framework for any type of structure data. In GAT, every node is assigned an attention weight, which is used in the following feature aggregation. Due to its generalization and effectiveness, GAT has been utilized in many fields, such as point cloud instance segmentation [67], visual question answering [66], trajectory forecasting [63], and referring expression comprehension [75]. Similar to GAT [63], our graph attention module (GAM) collects information from heterogeneous nodes by a corresponding attention weight. In our GAM, we concatenate the semantic and spatial relationships into the edges and operate on the edge directly to output the attention weight.

Other attention detectors. RelationNet++ [12] combines heterogeneous visual representations (*e.g.*, representations for anchor box, region proposal, corner/center points) with an efficient bridging visual representations (BVR), via key sampling and shared location embedding. HoughNet [52], as a voting-based bottom-up object detector, integrates local and long-range context information for object localization. Dynamic Head [14] also integrates multiple attention modules, but their attention implementation are quite different for three different parts: linear function for scale-aware attention, deformable convolution for spatial location sampling, and a gating subnetwork for task-aware attention in channel dimension. Also, They stack attention blocks multiple times to boost the feature representation. We provide comparison with DyHead in the supplementary material.

3 Our Approach

Preliminaries. Firstly, we briefly revisit the architecture of a region-based object detector (illustrated in Fig. 1(a)). The detection network can be divided into three parts: a backbone network for feature extraction, a region proposal network (RPN) for proposal generation, and a box head for final classification and localization. To enable hierarchical reasoning, we plug three visual reasoning modules between the backbone and box head, using a concise graph attention module (GAM). In GAM, all relationships are embedded in the edge attribute, and we operate directly on the edges to output the attention weights for feature enhancement. Finally, we combine heterogeneous visual reasoning modules into our HR-RCNN.

Graph construction Firstly, we construct graphs for visual reasoning. For pixel graph, we use a single-pixel within an RoI as a node, and a fully-connected graph is the feature representation for the RoI. For scale graph, we use an RoI-pooled [51] feature from a particular feature pyramid level as a node, and the graph representing the RoI connects these feature nodes across different levels. Finally, for the RoI graph, we use an RoI’s feature as a node, and the graph is constructed by connecting nodes from different RoIs in an image. To fully leverage the relationships, all graphs are fully-connected and adaptively assign attention weights to their neighbors.

Edge attribute, which models the relationship between nodes, can be divided into two parts: semantic distance measuring node distance in the feature space and spatial distance measuring node distance in the spatial space (Fig. 2(a)). For semantic distance $s_{i,j}$, following [10, 27, 68], we first divide the node attribute into g groups and then compute their semantic distance by groups, $s_{ij} = \text{SemanticDistance}(f_i, f_j)$, where i is the query node and j is the key node. The semantic distance can be implemented in many ways, including dot product, cosine similarity, or euclidean distance.

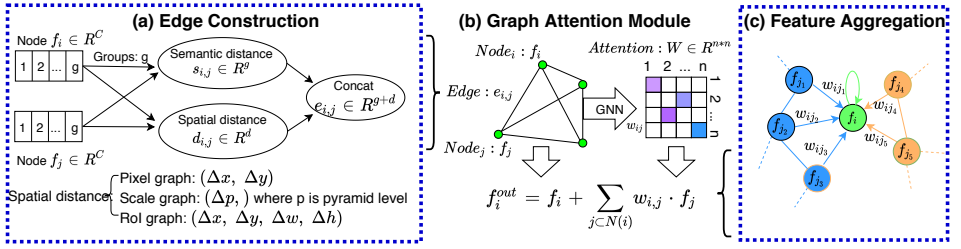


Figure 2: Graph attention module (GAM).

For spatial distance $d_{i,j}$, different relation graphs have their own definitions. Pixel graph uses normalized (x,y) distance in the pixel space: $d_{i,j}^P = (\Delta x, \Delta y) = \left(\frac{x_i - x_j}{w}, \frac{y_i - y_j}{h} \right)$, where w, h is the RoI width and height. Scale graph uses the normalized level distance in the feature pyramid space: $d_{i,j}^S = (\Delta p) = \left(\frac{p_i - p_j}{P} \right)$, where P is the number of pyramid levels. The RoI graph uses normalized $(x_{\text{center}}, y_{\text{center}})$, width, and height distance: $d_{i,j}^R = (\Delta x, \Delta y, \Delta w, \Delta h) = \left(\frac{x_i - x_j}{w_i}, \frac{y_i - y_j}{h_i}, \frac{w_j}{w_i}, \frac{h_j}{h_i} \right)$. Finally, we concatenate the semantic distance and spatial distance as the edge attribute. An illustration of edge construction is shown in Fig 2 (a).

Graph Attention Module Once we have the relation graph constructed, a message-passing strategy needs to be designed for relational reasoning. Similar to graph attention network (GAT) [15], our Graph Attention Module (GAM, illustrated in Fig 2(b)) generates attention weights for all neighbor nodes, and a weighted sum operation is utilized for feature aggregation. Since we already collect all the relationships in the edge attributes, our GAM operates directly on edges to produce the attention weights. Specifically, we introduce a multi-layer perceptron (mlp) to generate the attention weight α_{ij} : $\alpha_{ij} = \text{mlp}(e_{ij})$, which is followed by a global reasoning strategy, where we normalize the attention weights for all its neighbors. For simplicity, we implement this step as a softmax function,

$$w_{ij} = \text{softmax}_j(\alpha_{ij}/T) = \frac{\exp(\alpha_{ij}/T)}{\sum_{k \in N(i)} \exp(\alpha_{ik}/T)}, \quad (1)$$

where w_{ij} is the normalized attention weight, T is the softmax temperature (set as 2), $N(i)$ is the neighbor nodes for query node i . Finally, we refine the node features by aggregating information from the graph structure and all neighbors. Specifically, we enhance the node feature using residual connections, where the enhancement comes from the weighted sum of neighbor nodes (illustrated in Fig 2(c)).

$$f_i^{\text{out}} = f_i + \sum_{j \in N(i)} w_{ij} f_j. \quad (2)$$

After we obtain the updated node features, an additional FC layer is added for further fusion.

Hierarchical Relation Reasoning We propose our hierarchical reasoning framework based on a hierarchy of different relationships (Figure 3). Given the context it captures, we build two reasoning modules using these three graphs: an intra-level and inter-level context reasoning module, where level is a feature pyramid level. Both these modules use the RoI graph jointly with pixel and scale graphs respectively. The first stage embeds pixel graph and RoI

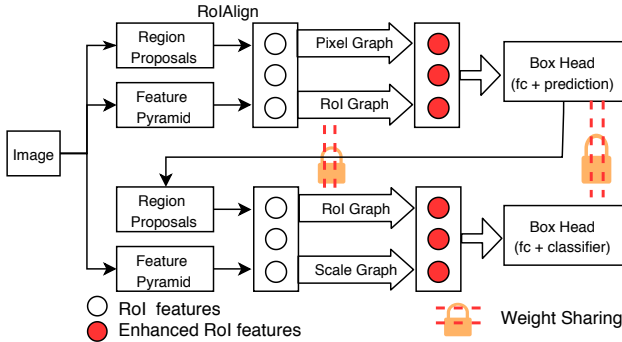


Figure 3: The proposed HR-RCNN structure.

graph jointly, while the second stage embeds scale graph and RoI graph jointly. Finally, to better utilize the learned knowledge from the first reasoning stage, we also utilize first stage’s predictions as the region proposals for the second stage, *i.e.*, iterative bounding-box regression [4, 5]). This leads to our HR-RCNN framework (illustrated in Fig. 3), where hierarchical reasoning happens in two stages. We train all reasoning branches jointly. This particular design is partly inspired by empirical findings on the complementary nature of different relation graphs, presented in Tab. 6 and supplementary material.

4 Experiment

Datasets and implementation details. We conduct all experiments on the COCO 2017 dataset [40], with train split ($\sim 118k$ images) for training, val ($\sim 5k$ images) split and test-dev split ($\sim 20k$ images, annotations withheld) for evaluation. All experiments are implemented using Detectron2 [41]. The input images are resized to have a shorter size of 800 pixels while the longer side no more than 1333 pixels. By default, we train the models with a total of 16 images per minibatch on 4 GPUs. Unless otherwise specified, all models are trained for 90k iterations (denoted as $1 \times \text{lr_scheduler}$) with an initial learning rate of 0.02, decreasing by a ratio of 0.1 at 60k and 80k respectively. We utilize ResNet-50 and ResNet-101 with feature pyramid network as backbones, and the batch normalization layers are fixed during training. All other hyper-parameters in this paper follow the settings in Detectron2. For HR-RCNN, we set group size for semantic distance to 2. *More results, graph combinations, other backbone main results, temperature/group size ablations, attention weights and detection results visualisation etc., can be found in the supplementary material.*

4.1 Main Results

Generalization across backbones. In this section, we evaluate HR-RCNN on COCO val2017 set with backbone different architectures, including FPN [42] with ResNet-50 [43], ResNet-101 [44], Deformable ResNet-50 [45]. For fair comparisons, we report our re-implemented results as the baseline; *i.e.*, our Faster R-CNN implementation with these backbones (which are generally better than originally reported) and then HR-CNN applied on that model. Since HR-RCNN has one more stage than the baseline, we also include cascade RCNN here. These

Table 1: **Main Results** on COCO validation set. The impact of using HR-RCNN with different backbones. All methods are based on Faster RCNN with feature pyramid network.

Methods	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
ResNet50 [12]	38.0	58.6	41.4	22.1	41.8	48.8
R50_cascade	40.4	60.2	43.8	23.8	44.1	52.2
HR-RCNN	41.6	61.8	45.2	25	45.1	54.2
ResNet101 [12]	40.2	61.2	43.8	24.1	43.8	52.1
R101_cascade	42.1	62.3	45.4	24.7	45.5	54.2
HR-RCNN	42.8	63.1	46.3	25.5	46.4	55.8
DCN-V2 [66]	40.8	62.0	44.5	24.2	44.0	54.0
R50_DCN_cascade	42.3	62.5	45.7	25.8	45.1	56.2
HR-RCNN	42.9	63.2	46.6	26.2	45.9	57.1

Table 2: **Hierarchical reasoning vs. single-level reasoning.** Methods with * means using segmentation annotations, with † means trained for 2× epochs.

Methods	Backbone	Pixel	Scale	RoI	AP	AP ₅₀	AP ₇₅
GCNet [9]*	Res50	✓			38.7	61.1	41.7
Non-local [65]*	Res50	✓			39.0	61.1	41.9
DCN-V2 [66]*	Res50	✓			39.9	-	-
DGMN [61]*	Res50	✓			40.2	62.0	43.4
AugFPN [18]	Res50		✓		38.8	61.5	42.0
SEPC [65]	Res50		✓		38.5	59.9	41.4
NAS-FPN [10]	Res50		✓		39.7	57	41.8
RelationNet [15]	Res50			✓	38.8	60.3	42.9
HR-RCNN (Ours)	Res50	✓	✓	✓	41.6	61.8	45.2
SGRN [14] †	Res101			✓	41.7	62.3	45.5
Reasoning-RCNN [14] †	Res101			✓	42.9	-	-
HR-RCNN† (Ours)	Res101	✓	✓	✓	44.6	64.7	48.1

results are shown in Tab. 1, where HR-RCNN consistently improves the performance for ResNet in all metrics. Note that all our backbones utilize the FPN structure, which already considers scale relation to address the scale invariance. This demonstrates that our approach is complementary to the feature pyramid architectures.

Hierarchical vs. Single-level Reasoning. To show the advantage of hierarchical reasoning over single-level reasoning, we compare with many single-level reasoning methods in Tab. 2. A clear AP improvement can be seen from hierarchical visual reasoning as compared to single-level reasoning. Therefore, by iteratively extracting heterogeneous relationships, our HR-RCNN framework can gradually enhance the feature representation of stacked ConvNets and greatly improve the final performance.

Instance Segmentation To further evaluate the hierarchical reasoning framework, we extend it to the instance segmentation task. We take the Mask RCNN [23] with ResNet50-FPN backbone as the instance segmentation baseline. To incorporate hierarchical visual reasoning, we keep the box head same as HR-RCNN and put a single P+R reasoning component in the mask head. As shown in Tab. 3, we improve the segmentation AP by 1.9 points, which shows the potential of hierarchical visual reasoning to other tasks.

Table 3: Instance segmentation results

	Box			Segmentation		
	AP	AP ₅₀	AP ₇₅	AP	AP ₅₀	AP ₇₅
Mask RCNN	38.6	59.5	42.1	35.2	56.3	37.5
Cascade Mask RCNN	41.3	60.1	45.1	36.1	57.3	38.8
HR-Mask RCNN	41.8	61.7	45.4	37.1	58.6	39.7

Table 4: Ablations results for individual relations

	Semantic	Spatial	Δ params	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
Baseline	-	-	-	38.0	58.6	41.4	22.1	41.8	48.8
Pixel graph	✓		64	38.3	59.1	41.6	22.2	41.6	49.8
	✓	✓	608	38.5	59.3	41.9	22.5	42.1	49.9
Scale graph	✓		96	38.2	58.9	41.4	22.2	41.5	49.2
	✓	✓	192	38.5	59.3	41.7	22.4	42	49.5
RoI graph	✓		96	38.5	59.5	42.1	22.5	42	49.5
	✓	✓	1.2k	38.9	60.3	42.2	23.2	42.4	49.8

Table 5: Combination ablation. P: pixel relation, S: scale relation, R: RoI relation

	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
P+S	38.7	59.5	41.9	23.3	42.0	50.1
S+R	39.4	60.7	42.7	23.3	42.4	51.8
P+R	39.5	60.5	43.1	23.7	42.9	51.1
P+S+R	39.4	60.3	43.1	23.4	42.6	51.5
Cascade RCNN	40.4	60.2	43.8	23.8	44.1	52.2
SR + PR	41.5	61.9	45	25.5	45	54.1
PR + SR	41.6	61.8	45.2	25.0	45.1	54.2

Table 6: HR-RCNN component ablation for sharing box heads (denoted as 'Share') and hierarchical reasoning (denoted as 'HR')

	Share	HR	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
HR-RCNN w/o HR	✓	✓	41.6	61.8	45.2	25	45.1	54.2
w/o sharing head	✓		40.2	60.1	43.5	23.5	43.7	52.8
Cascade RCNN		✓	41.1	61.1	44.5	24.9	44.4	53.5
			40.4	60.2	43.8	23.8	44.1	52.2

4.2 Ablation Study

Firstly, we show the results of each single relational reasoning component in Tab. 4. RoI relational reasoning brings the highest +0.9 improvement for AP, while pixel and scale reasoning improves by +0.5 each. Note that the extra parameters are quite marginal.

Then, we demonstrate the results of different combinations for different relation modules. As can be seen in Tab. 5, combinations of any two reasoning components are complementary and consistently improve the performance, where +1.4/+1.5 gain come from scale-RoI reasoning and pixel-RoI reasoning respectively. But using all three (pixel-scale-RoI relation) jointly doesn't improve the performance any further. Finally, our HR-RCNN framework is able to successfully utilize these reasoning components hierarchically and performs the best.

Finally, we present an ablation study on the importance of weight sharing and hierarchical reasoning in Tab. 6. Effectively, our approach without hierarchical reasoning is a two-stage Cascade RCNN with shared box heads. Note that this drops the AP by 1.4 points. Next, similar to Cascade RCNN, we remove weight sharing and train disjoint the parameters for all branches. This not only increases the model parameters and inference time, but also hurts the AP by 0.5 points; further attesting to the importance of multitask hierarchical reasoning formulation. Finally, compared with the original Cascade RCNN, HR-RCNN performs better by 1.2 points. Note that though weight sharing helps in HR-RCNN, it hurts Cascade RCNN by 0.2 points. This supports our claim that hierarchical relationships can be fused implicitly through the shared box head.

Table 7: **Comparison with state-of-the-arts** on COCO test-dev. †: multi-scale testing, +: with soft-NMS, large-batch BN

Method	Train size	Test size	Backbone	Epochs	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
One-stage models:										
FCOS [14]	1333 × 800	1333 × 800	R101-FPN	24	41.5	60.7	45.0	24.4	44.8	51.6
SAPD [15]	1333 × 800	1333 × 800	R101-FPN	24	43.5	63.6	46.5	24.9	46.8	54.6
PAA [16]	1333 × 800	1333 × 800	R101-FPN	24	44.8	63.3	48.7	26.5	48.8	56.3
MAL [17]	1333 × 800	1333 × 800	R101-FPN	24	43.6	62.8	47.1	25.0	46.9	55.8
ATSS [18]	1333 × 800	1333 × 800	R101-FPN	24	43.6	62.1	47.4	26.1	47.0	53.6
SEPC [19]	1333 × 800	1333 × 800	R101-FPN	24	45.5	64.9	49.5	27.0	48.8	56.7
Detr [20]	1333 × 800	1333 × 800	R101-FPN	500	43.5	63.8	46.4	21.9	48.0	61.8
BorderDet [21]	1333 × 800	1333 × 800†	R101-DCN-FPN	24	47.2	66.1	51.0	28.1	50.2	59.9
DDB-Net [22]	1333 × 800	1333 × 800	R101-FPN	24	42.0	61.0	45.1	24.2	45.0	53.3
DyHead [23]	1333 × 800	1333 × 800	R101-FPN	24	46.5	64.5	50.7	28.3	50.3	57.3
VFNet [24]	1333 × 800	1333 × 800	R101-FPN	24	46.0	64.2	50.0	27.5	49.4	56.9
HoughNet[25]	512 × 512	512 × 512†	HG-104	140	46.4	65.1	50.7	29.1	48.5	58.1
RelationNet++ [26]	1333 × 800	1333 × 800†	ResNeXt-64 × 4d-101-DCN	20	52.7	70.4	58.3	35.8	55.3	64.7
Two-stage models:										
DCN-V2 [27]	1333 × 800	1333 × 800	R101-DCN-FPN	12	42.7	63.7	46.8	24.9	46.7	56.8
Cascade RCNN [8]	1312 × 800	1312 × 800	R101-FPN	18	42.8	62.1	46.3	23.7	45.5	55.2
RPDNet [28]	1333 × 800	1333 × 800†	R101-DCN-FPN	12	46.5	67.4	50.9	30.3	49.7	57.1
Cascade RCNN + SABL [29]	1333 × 800	1333 × 800	R101-FPN	12	43.3	60.9	46.2	23.8	46.5	55.7
Dynamic RCNN [30]	1333 × 800	1333 × 800†	R101-DCN-FPN	36	49.2	68.6	54.0	32.5	51.7	60.3
AugFPN [31]	1312 × 800	1312 × 800	R101-augFPN	24	41.5	63.9	45.1	23.8	44.7	52.8
TridentNet+ [32]	1333 × 800	1333 × 800	R101-DCN	36	46.8	67.6	51.5	28	51.2	60.5
HR-RCNN (Ours)	1333 × 800	1333 × 800	R101-FPN	12	43.1	63.3	46.5	25.2	46.0	54.1
HR-RCNN (Ours)	1333 × 800	1333 × 800	R101-FPN	24	44.9	65.1	48.4	26.7	47.6	56.5
HR-RCNN (Ours)	1333 × 800	1333 × 800	R101-DCN-FPN	12	44.8	65.0	48.2	26.1	47.6	57.0
HR-RCNN (Ours)	1333 × 800	1333 × 800	R101-DCN-FPN	24	45.7	65.7	49.3	27.3	48.5	57.6
HR-RCNN (Ours)	1333 × 800	1333 × 800†	R101-DCN-FPN	24	47.7	68.2	51.7	30.8	50.4	59.4

4.3 Comparison with State-of-the-art

Tab. 7 shows comparisons of HR-RCNN with some state-of-the-art methods on the COCO test-dev split. Without bells and whistles, our HR-RCNN with ResNet-101 backbone achieves 44.9 AP using the 2x training scheme. By changing the backbone to deformable ResNet-101, HR-RCNN gets 47.7 mAP with a multi-scale testing. As a two-stage detector, HR-RCNN is comparable with all other two-stage methods under the same setup, *e.g.*, with the same backbone and training epochs. Specifically, for pixel reasoning baseline DCN-V2 [27], our hierarchical reasoning improves its AP by 2.1 points. As a refinement baseline, Cascade RCNN refines the region proposals with three stages of box heads, while our HR-RCNN only have two stages of refinement and a single copy of box head weights. Although we use fewer refine steps, our HR-RCNN still outperforms Cascade RCNN with a considerable margin without adding more box heads. Note that our hierarchical visual reasoning is fairly generic and can be added to many backbones to further boost their performance.

4.4 Experimental analysis

To get a better understanding of HR-RCNN, we analysis its performance to number of training samples. We sort categories by the number of training samples, and evaluate how hierarchical reasoning behaves for frequent categories and rare categories. Compared to Faster R-CNN, Fig 4 demonstrates the relationship between AP improvement and number of training samples, and we fit a linear regression line. HR-RCNN improves for all but one classes, with generally larger boost for in-frequent categories. In-frequent classes do not have enough training samples, and thus benefits more from hierarchical visual reasoning.

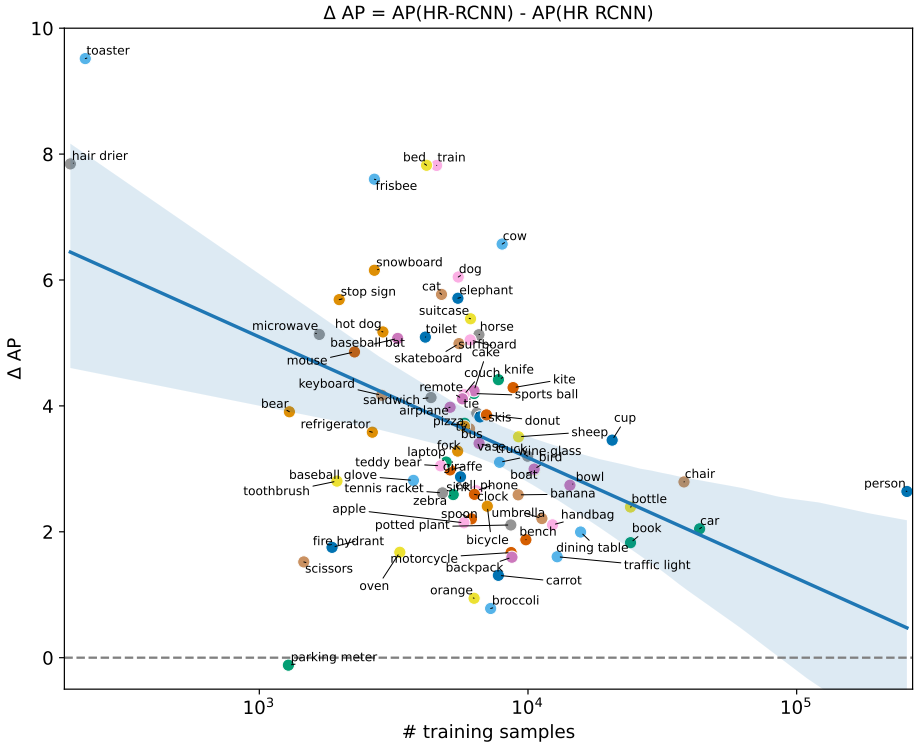


Figure 4: ΔAP vs. training samples. Dots are AP gains by HR-CNN for different classes and we also show a linear regression fit. Nearly all classes can benefit from hierarchical reasoning, especially for in-frequent ones.

5 Conclusion.

In this paper, we propose a hierarchical relational reasoning framework (HR-RCNN) for object detection, which utilizes a concise graph attention module (GAM) to enable visual reasoning across heterogeneous nodes. We also explore different strategies to define the hierarchy between heterogeneous relationships, which leads to our HR-RCNN architecture. Finally, extensive experiments on COCO dataset show its effectiveness.

Acknowledgement. This work was supported by the Air Force STTR award (FA864920C0010), DARPA SemaFor program (HR001120C0124), DARPA SAIL-ON program (W911NF2020009), and Amazon Research Award to AS.

References

- [1] Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip HS Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016.
- [2] Anurag Arnab, Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Måns

- Larsson, Alexander Kirillov, Bogdan Savchynskyy, Carsten Rother, Fredrik Kahl, and Philip HS Torr. Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. *IEEE Signal Processing Magazine*, 2018.
- [3] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *NeurIPS*, 2016.
- [4] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2874–2883, 2016.
- [5] Irving Biederman. On the semantics of a glance at a scene. *Perceptual organization*, 213:253, 1981.
- [6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.
- [7] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv*, 2019.
- [8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. 2020.
- [9] Ran Chen, Yong Liu, Mengdan Zhang, Shu Liu, Bei Yu, and Yu-Wing Tai. Dive deeper into box for object detection. 2020.
- [10] Xinlei Chen and Abhinav Gupta. Spatial memory for context reasoning in object detection. In *ICCV*, 2017.
- [11] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *CVPR*, 2018.
- [12] Cheng Chi, Fangyun Wei, and Han Hu. Relationnet++: Bridging visual representations for object detection via transformer decoder. In *NeurIPS*, 2020.
- [13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.
- [14] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7373–7382, June 2021.
- [15] Bin Fan, Qian Zhang, Shiming Xiang, Chunhong Pan, et al. Augfpn: Improving multi-scale feature learning for object detection. 2020.
- [16] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.
- [17] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *CVPR*, 2019.

- [18] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *CVPR*, 2019.
- [19] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. 2017.
- [20] Will Hamilton, Zitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.
- [21] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.
- [24] Howard S Hock, Gregory P Gordon, and Robert Whitehurst. Contextual relations: The influence of familiarity, physical plausibility, and belongingness. *Perception & Psychophysics*, 16(1):4–8, 1974.
- [25] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, 2018.
- [26] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019.
- [27] Chenhan Jiang, Hang Xu, Xiaodan Liang, and Liang Lin. Hybrid knowledge routed modules for large-scale object detection. In *NeurIPS*, 2018.
- [28] Wei Ke, Tianliang Zhang, Zeyi Huang, Qixiang Ye, Jianzhuang Liu, and Dong Huang. Multiple anchor learning for visual object detection. In *CVPR*, 2020.
- [29] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *NeurIPS*, 2017.
- [30] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction for object detection. 2020.
- [31] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [32] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen. Ron: Reverse connection with objectness prior networks for object detection. In *CVPR*, 2017.
- [33] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezaatofghi, and Silvio Savarese. Social-biगत: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *NeurIPS*, 2019.
- [34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

- [35] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NeurIPS*, 2011.
- [36] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *ICCV*, 2019.
- [37] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *ICCV*, 2019.
- [38] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *ECCV*, pages 335–351, 2018.
- [39] Zuoxin Li and Fuqiang Zhou. Fssd: feature fusion single shot multibox detector. *arXiv preprint arXiv:1712.00960*, 2017.
- [40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*. Springer, 2014.
- [41] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.
- [42] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.
- [43] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.
- [44] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018.
- [45] Yong Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Structure inference net: Object detection using scene-level context and instance-level relationships. In *CVPR*, 2018.
- [46] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. *arXiv preprint arXiv:1612.04844*, 2016.
- [47] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.
- [48] tephen E Palmer. The effects of contextual scenes on the identification of objects. *Memory & Cognition*, 3:519–526, 1975.
- [49] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. *arXiv preprint arXiv:2006.02334*, 2020.
- [50] Han Qiu, Yuchen Ma, Zeming Li, Songtao Liu, and Jian Sun. Borderdet: Border feature for dense object detection. 2020.

- [51] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.
- [52] Nermin Samet, Samet Hicsonmez, and Emre Akbas. Houghnet: Integrating near and long-range evidence for bottom-up object detection. In *European Conference on Computer Vision (ECCV)*, 2020.
- [53] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. *arXiv preprint arXiv:1806.01242*, 2018.
- [54] Abhinav Shrivastava and Abhinav Gupta. Contextual Priming and Feedback for Faster R-CNN. In *European Conference on Computer Vision (ECCV)*, 2016.
- [55] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training Region-based Object Detectors with Online Hard Example Mining. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [56] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*, 2016.
- [57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [58] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *CVPR*, 2018.
- [59] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *NeurIPS*, 2018.
- [60] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.
- [61] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.
- [62] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, 2020.
- [63] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.
- [64] Antonio Torralba, Kevin P Murphy, William T Freeman, Mark A Rubin, et al. Context-based vision system for place and object recognition. In *ICCV*, volume 3, pages 273–280, 2003.
- [65] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [66] Jiaqi Wang, Wenwei Zhang, Yuhang Cao, Kai Chen, Jiangmiao Pang, Tao Gong, Jianping Shi, Chen Change Loy, and Dahua Lin. Side-aware boundary localization for more precise object detection. 2020.

- [67] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, 2019.
- [68] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.
- [69] Xinjiang Wang, Shilong Zhang, Zhuoran Yu, Litong Feng, and Wayne Zhang. Scale-equalizing pyramid convolution for object detection. In *CVPR*, 2020.
- [70] Sanghyun Woo, Soonmin Hwang, Ho-Deok Jang, and In So Kweon. Gated bidirectional feature pyramid network for accurate one-shot detection. *Machine Vision and Applications*, 2019.
- [71] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [72] Dan Xu, Wanli Ouyang, Xavier Alameda-Pineda, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In *NeurIPS*, 2017.
- [73] Hang Xu, Chenhan Jiang, Xiaodan Liang, and Zhenguo Li. Spatial-aware graph relation network for large-scale object detection. In *CVPR*, 2019.
- [74] Hang Xu, ChenHan Jiang, Xiaodan Liang, Liang Lin, and Zhenguo Li. Reasoning-rnn: Unifying adaptive global reasoning into large-scale object detection. In *CVPR*, 2019.
- [75] Sibe Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *ICCV*, 2019.
- [76] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *ICCV*, 2019.
- [77] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, 2018.
- [78] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018.
- [79] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sünderhauf. Varifocalnet: An iou-aware dense object detector. In *CVPR*, 2021.
- [80] Hongkai Zhang, Hong Chang, Bingpeng Ma, Naiyan Wang, and Xilin Chen. Dynamic R-CNN: Towards high quality object detection via dynamic training. 2020.
- [81] Li Zhang, D. Xu, A. Arnab, and P. Torr. Dynamic graph message passing networks. 2020.
- [82] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, 2020.

-
- [83] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. In *AAAI*, 2019.
 - [84] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.
 - [85] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. 2020.
 - [86] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019.