Detecting Objects with Context-Likelihood Graphs and Graph Refinement

Aritra Bhowmik[1]   Yu Wang[2]   Nora Baka[2]   Martin R. Oswald[1]   Cees G. M. Snoek[1]

Atlas Lab - [1]University of Amsterdam   [2]TomTom

{a.bhowmik, m.r.oswald, cgmsnoek}@uva.nl   {nora.baka, yu.wang}@tomtom.com

Abstract

The goal of this paper is to detect objects by exploiting their interrelationships. Contrary to existing methods, which learn objects and relations separately, our key idea is to learn the object-relation distribution jointly. We first propose a novel way of creating a graphical representation of an image from inter-object relation priors and initial class predictions, we call a context-likelihood graph. We then learn the joint distribution with an energy-based modeling technique which allows to sample and refine the context-likelihood graph iteratively for a given image. Our formulation of jointly learning the distribution enables us to generate a more accurate graph representation of an image which leads to a better object detection performance. We demonstrate the benefits of our context-likelihood graph formulation and the energy-based graph refinement via experiments on the Visual Genome and MS-COCO datasets where we achieve a consistent improvement over object detectors like DETR and Faster-RCNN, as well as alternative methods modeling object interrelationships separately. Our method is detector agnostic, end-to-end trainable, and especially beneficial for rare object classes.

1. Introduction

Object detection is a classical task in computer vision [2, 29], where methods in the last decade have evolved from deep convolutional architectures [38, 37, 26, 4] to self-attention based transformer networks [5, 49, 35, 42]. Although these two approaches vary widely in architecture, they both focus exclusively on the image space for feature representation, rather than incorporating contextual semantics about object categories. Once fed with sufficient data, these architectures might implicitly learn some object relations, but prior works [21, 45, 46, 44] have shown that explicitly modeling object relations enhances object detection performance by a substantial margin.

Amongst existing work on modeling object relations, Jiang *et al.* [21] hypothesize that in case a dataset has little information about certain classes, incorporating prior

Faster-RCNN [27]   **Ground truth detections**

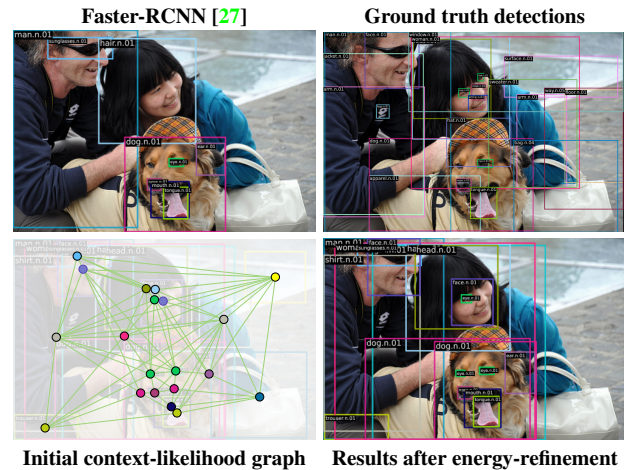**Initial context-likelihood graph**   **Results after energy-refinement**

Figure 1. We generate a context-likelihood graph from the initial predictions of a base detector, here Faster-RCNN, by incorporating inter-object relation priors. Having learnt the joint object-relation distribution, we iteratively sample a refined graph by energy optimization. This enables us to predict objects undetected by Faster-RCNN, due to their relations or co-occurrence patterns with already detected objects. For instance, the detection of a face makes the occurrence of eyes or a nose much more likely and vice versa. Likewise for the trousers in the presence of a man, which are not even present in the ground truth.

knowledge in the training of a detector should improve detection. This prior knowledge could be from another source, or it could even be distilled from the original dataset. The authors establish a novel way of explicitly incorporating semantic prior knowledge into a base object detection framework by predicting object-relations from features, creating a graphical representation of the image. Xu *et al.* [45] propose to predict object relations solely from object features and their spatial orientation, without any extra prior information, by creating a sparse graphical representation. Both these methods are dependent on the correctness of the predicted relations since they refine the object features for classification. Instead of predicting an explicit graph representation, Xu *et al.* [46] enhance object detection by evolving high-level semantic representations globally, rather than relying solely on visual features. This is accomplished

through a network that generates a global semantic pool and adapts each object's features by attending to different semantic contexts in the pool, automatically discovering the most relevant categories for feature evolution. Although all these works utilize prior knowledge about objects in their problem formulation, they do so consecutively making them critically dependant on the object features. Instead, we propose to model the objects and their relations jointly from the start, in order to have a better understanding of the scene and infer better class predictions.

Our main contributions are thereby as follows:

(1) We propose a novel way of leveraging inter-object relation priors for object detectors during training by creating a context-likelihood graph with relation edges based on initial class predictions. (Sec. 3.1).

(2) We demonstrate the potential of our context-likelihood graph with an empirical evaluation showing that incorporating inter-object relations from the start leads to substantially better object detection rates when using a graph built with ground truth class predictions (Sec. 3.2).

(3) We introduce an energy-based method for learning object-relation joint distribution, which allows to iteratively refine our context-likelihood graph to further benefit us in our end task (Sec. 3.3).

Experiments on the Visual Genome [23] and MS-COCO [28] datasets demonstrate our method is detector agnostic, end-to-end trainable, and especially beneficial for rare object classes. What is more, we establish a consistent improvement over object detectors like DETR [5] and Faster-RCNN [38], as well as alternative methods modeling object interrelationships. Before introducing our method, we first provide more background on related works.

2. Related Work

Object detection. Modern object detection models can be broadly classified into two groups: those based on convolution architectures and those based on transformers. In the convolution family, methods like Faster-RCNN [38] and R-FCN [8] are well established. They consists of a region proposal network to generate regions of interest and a classification and bounding box regression module to predict a set of objects. YOLO [37] and SSD [30] predict objects directly from convolutional feature maps. ObjectBox [48] treats objects as center-points in a shape and size agnostic fashion and allows learning at all scales. All these methods typically utilize a pretrained backbone like ResNet [18] for feature extraction. Recently, there has been a rise in methods which use an attention mechanism in a transformer network. For instance, ViT [42], SwinTransFormer [32] and MViTv2 [25]. Carion *et al.* [5] introduced a detection transformer (DETR) which uses an attention mech-

anism for object detection in an end-to-end manner. To speed up convergence and reduce the computational cost for the self-attention on image features, Zhu *et al.* [49] introduced multi-head deformable attention replacing the dot-product in the attention computation with two linear projections for sampling points and computing their attention weights. Nguyen *et al.* [35] introduced BoxeR which samples boxes instead of points for attention computation, and demonstrates utility for object detection, instance segmentation and 3D detection. Although these object detectors use spatial context information in the image space, they do not explicitly model the relation between (detected) objects. Using Faster-RCNN and DETR as representative examples of the convolution and transformer approach to object detection, we demonstrate that creating a graphical representation of an image from inter-object relation priors and initial class predictions improves object detection.

Object graph representation. Aside from classic object detection, many works have aimed to represent the interactions between objects and scenes. Early works [14, 15, 34] use object relations as a post-processing step on simpler datasets like PASCAL VOC [13]. Recent works model objects and relations together in a graph structure, *e.g.*, [6, 7, 22, 33] where Xu *et al.* [47] use a unified conditional random field to model the joint distribution of all the objects and their relations in a scene graph. However, this requires ground truth labels for both the objects and their relations for each image during training. To avoid explicit relation annotations, others, *e.g.*, [21, 45, 46, 44] exploit priors like co-occurrence statistics or attributes of object classes to obtain a graph representation of an image. Jiang *et al.* [21] predict the edge values representing the relations between object pairs, whereas Xu *et al.* [45] propose a form of self-attention to obtain these edges. In a follow-up work, Xu *et al.* [46] further introduce a global semantic pool, created using the weights of individual object classifiers, as a relationship prior instead of generating an explicit graph representation. To aggregate information from different domains, they also introduce graph transfer learning [44]. All these works enhance the object feature representation by propagating the relation information via a graph structure. These works utilize the relation information as is and do not consider the joint modeling of the relation with the objects in the scene. In our work, we learn the joint distribution of objects and their relations by sampling meaningful relations for object pairs before the graph propagation.

Energy-based models. Energy-based models (EBMs) bring forth simplicity and generality in likelihood modeling [1, 19, 24, 20]. Various tasks like image generation [11, 36], out-of-distribution detection [31], memory modeling [3] and anomaly detection [9] have been addressed with the help of EBMs. Recently, there has been an increasing interest in applying EBMs to the task of image generation.
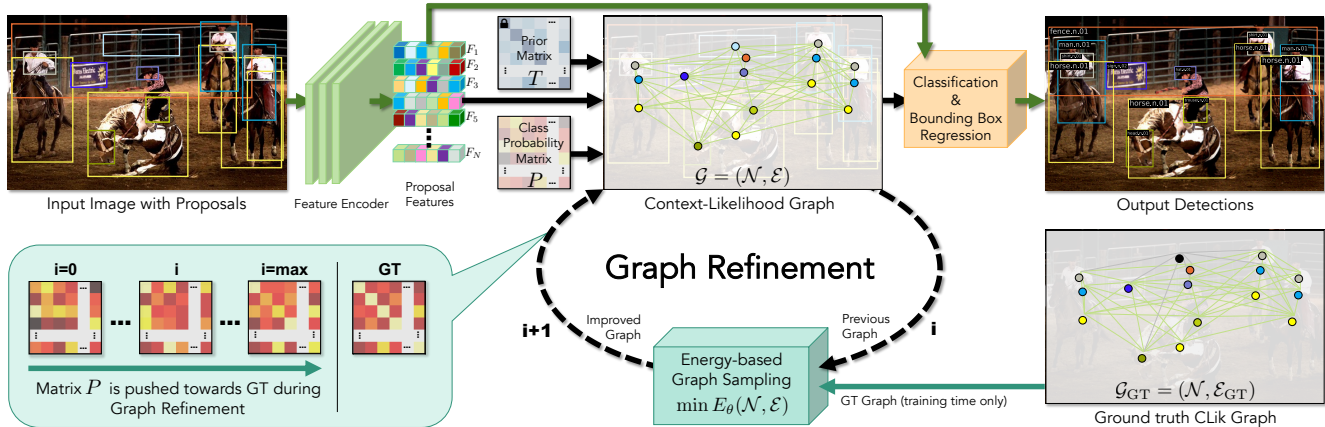
Figure 2. **Method overview.** Our end-to-end trainable architecture extends object detectors by inserting a context-likelihood (CL) graph between the first stage (object proposal generation) and the second stage (proposal classification). The default detector pipeline is visualized by the green arrows. With the use of our context-likelihood graph before the classification stage, the proposals can be better assessed since the presence or absence of other objects can be explicitly taken into account which for example can enforce proposals with low visual evidence. The context-likelihood graph is used in two ways: (1) During training, to learn the energy landscape and train the classifier. (2) At test time, to obtain the refined graph via energy-based optimization.

Both Du *et al.* [12] and Grathwohl *et al.* [17] learn the underlying data distribution and use the learnt distribution to generate new data. Moving forward from only modeling the data distribution, [17, 43] model the joint distribution of data and labels to benefit classification tasks by a joint energy-based model. Recently, Suhail *et al.* [40] propose an energy-based learning framework for scene graph generation to learn the joint distribution of objects and their relations together. To the best of our knowledge, our work is the first to utilize energy-based modeling to learn a graph representation of an image from just object annotations.

3. Method

Given an input image and object proposals, we aim to detect all objects in the image in the form of bounding boxes and their corresponding class labels. An overview of our method is provided in Fig. 2. The green arrows in the figure show the regular object detection pipeline. We add two key components to it which are described in the following subsections: First, we present context-likelihood graphs, a new way of building a graph representations, based on initial class predictions in Sec. 3.1. We demonstrate the potential of context-likelihood graphs with a simple oracle experiment in Sec. 3.2. Second, we introduce an iterative method for obtaining refined graph via energy-based optimization in Sec. 3.3.

3.1. Context-Likelihood Graphs

Components. For an input image I, visual features $\{F_i\}_1^N$ of dimension D are extracted via a base object detector for N object proposals. These feature vectors along with contextual prior knowledge are used in pairs [21, 45] to obtain

pair-wise edge relationships, thus creating an edge connectivity matrix $\mathcal{E}$. A popular source of prior knowledge about objects is the large scale Visual Genome dataset [23] which consists of pairwise relationships between object classes such as "subject-verb-object" relationships (*e.g.*, walking, swimming) and spatial relationships (*e.g.*, at, far, in). By considering the most frequent relationship between subject and object boxes in the dataset and counting the co-occurrence among class pairs, a $C \times C$ frequent statistics or prior matrix T is generated. Each entry in the matrix is a floating point value between 0 and 1, indicating the strength of the relation between the object pairs in the dataset.

Definition. We obtain a graphical representation of the input image based on the feature vectors and the edge values defined as

$$\mathcal{G} = (\mathcal{N}, \mathcal{E}). \tag{1}$$

where $\mathcal{N}$ are the set of nodes and $\mathcal{E}$ is the edge connectivity matrix. We call this our context-likelihood graph. This graph is propagated via some message passing algorithm to calculate refined features, from which we obtain the final classification and bounding box regression results. To create this graph, we model object relations as a function of the initial class predictions, instead of directly predicting them from feature vector pairs [21]. Corresponding to the feature vectors $\{F_i\}_1^N$, we obtain their classification probabilities P from the base detector where $P \in \mathbb{R}^{N \times C}$, C being the number of classes in the dataset. Given the class prediction matrix P and the prior matrix T, we calculate the edge connectivity matrix as:

$$\mathcal{E} = P \times T \times P^T . \tag{2}$$

Each term e_{ij} in the matrix $\mathcal{E}$ is a result of all possible prob-

abilistic combinations of classes for nodes i and j along with their corresponding relation prior values. By modeling the edge matrix in this manner, we penalize the class predictions jointly with the edge values, the former guiding the correctness of the latter. Thus, optimizing for the final task of classification would ensure refined edge values, which in turn guides the end task. Our joint task loss function is given by:

$$\mathcal{L}_{Task} = \mathcal{L}_{CE}\{P^{'}, P^*\} + \mathcal{L}_{CE}\{P, P^*\} \ , \qquad (3)$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss, $P^{'}$ is the final classification probability matrix output from our model, P is the initial class prediction matrix of the base detector and P^* is the ground truth object labels matrix. Optimality is obtained when initial predictions equal the ground truth labels.

3.2. Potential of Context-Likelihood Graphs

To test the optimal condition of our context-likelihood graph, we perform an oracle experiment where we replace the set of edges $\mathcal{E}$, with ground truth relation set $\mathcal{E}_{GT}$. To calculate $\mathcal{E}_{GT}$, P is replaced by P^* in Eq. (2) where each row of P^* is a one-hot encoded class label corresponding to each object proposal, obtained by IoU-matching. To perform IoU-matching, we check the overlap of the regions of interests of each proposal with each ground truth bounding box, and assign the class label of the highest overlapping ground truth bounding box to the proposal. The ground truth context-likelihood graph $\mathcal{G}_{GT} = (\mathcal{N}, \mathcal{E}_{GT})$, optimally captures the underlying object-pair relationships. Propagating this graph generates the refined object features which are used for the final classification results. As ground truth we rely on the 1000 (VG_{1000}) and 3000 (VG_{3000}) most frequent object classes from Visual Genome [23].

Analysis. We use two proposal-based object detectors as our base network, the transformer-based DETR [5] and convolution-based Faster-RCNN [38]. Both these methods first identify rectangular regions of interest in an image, extract their corresponding feature encodings and create classification predictions for each such region. From Fig. 3, we see that knowing the relations between objects in a scene improves the detection results of the base network by a large margin. For the VG_{1000} dataset we achieve an 119% relative improvement in mean average precision results for DETR, while for Faster-RCNN, it is around 110%. Note that we do not embed the ground truth information in any way other than to use it for looking up the true relation prior between proposed object-pairs.

This experiment shows the potential of our context-likelihood graph, irrespective of base object detectors, in the optimal setting. However, we will not have access to ground truth during inference, which prevents us from reaching optimality in the above described fashion. The edge matrix $\mathcal{E}$ in Eq. (2) is a function of the class probability matrix,
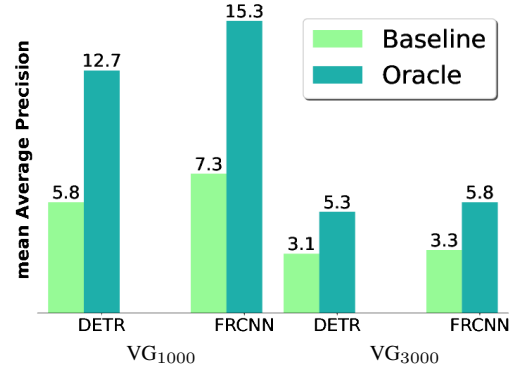

Figure 3. **Potential of context-likelihood graph.** An oracle experiment with the ground truth context-likelihood graph, created by replacing the initial predictions by ground truth class labels in Eq. (2), yields substantial improvements on the VG_{1000} and VG_{3000} partitions of visual genome [23], for both a DETR [5] and F-RCNN [38] object detector.

P which in turn is generated from feature vector set $\mathcal{N}$. The optimal solution $\mathcal{E}_{GT}$ lies in the joint distribution space of $(\mathcal{N}, \mathcal{E})$. In order to sample a refined context-likelihood graph, we need to learn the joint distribution, for which we propose an energy-based method in the following section.

3.3. Energy-Based Graph Refinement

Before detailing our graph refinement approach, we first provide a brief background on energy-based models.

Energy model background. For two variables x and y, energy-based models [24] represent their joint probability distribution as $p_\theta(x,y) = \frac{\exp(-E_\theta(x,y))}{Z(\theta)}$, where $\exp(-E_\theta(x,y))$ is known as the energy function that maps each input pair to a scalar value and $Z(\theta) = \int \exp(-E_\theta(x,y))$ is referred to as the normalization constant or partition function. Computing $Z(\theta)$ typically becomes intractable since it computes the integration over the entire joint input space of (x,y). Thus, we cannot learn the parameter θ of the energy-based model by using a direct log likelihood approach. In order to understand the problems with the partition function, we investigate the derivatives [12, 39] of the log likelihood function given as:

$$\frac{\partial \log p_\theta(x,y)}{\partial \theta} = \qquad (4)$$

$$\mathbb{E}_{p_d(x,y)}\left[-\frac{\partial E_\theta(x,y)}{\partial \theta} \right] + \mathbb{E}_{p_\theta(x,y)}\left[\frac{\partial E_\theta(x,y)}{\partial \theta} \right] \ .$$

The first term is an expectation over the data distribution and the second term is over the model distribution. Maximizing the log-likelihood $\log p_\theta(x,y)$ is equivalent to minimizing the KL-divergence $D_{KL}(p_d(x,y)||p_\theta(x,y))$ which is achieved by minimizing the contrastive divergence [20]:

$$\mathcal{L} = D_{KL}(p_d(x,y)||p_\theta(x,y)) - D_{KL}(q_\theta(x,y)||p_\theta(x,y)) \ , \qquad (5)$$

where $q_\theta(x,y) = \prod_\theta^t p_d(x,y)$ represents t sequential Markov chain Monte Carlo transitions [20], used to approximate the model distribution $p_\theta(x,y)$. An effective example being Stochastic gradient Langevin Dynamics (SGLD) [41]:

$$x^{i+1} = x^i - \frac{\lambda}{2}\frac{\partial E_\theta(x^i)}{\partial x^i} + \epsilon, \quad s.t. \quad \epsilon \sim \mathbb{N}(0,\lambda) \ . \quad (6)$$

Eq. (5) can be rewritten in expectation terms as follows:

$$\mathcal{L} = \mathbb{E}_{p_d(x,y)}[E_\theta(x,y)] - \mathbb{E}_{\blacksquare(q_\theta(x,y))}[E_\theta(x,y)] \quad (7)$$
$$\mathbb{E}_{q_\theta(x,y)}[E_{\blacksquare(\theta)}(x,y)] + \mathbb{E}_{q_\theta(x,y)}[\log q_\theta(x,y)] \ .$$

Here, the lock symbol $\blacksquare(\cdot)$ denotes that the variables in the argument list are not optimized, *i.e.*, their gradient flow is stopped. With this knowledge of training an energy model to learn a joint probability distribution, we formulate our context-likelihood energy-refined graph.

Energy model for object detection. Langevin dynamics [41] simulate samples of the distribution expressed by the energy-based model through gradient-based updates, with no restriction on the sample initialization if the sampling steps are sufficient. We leverage this property to sample a refined graph of the image by setting the context-likelihood graph as the initialization of the Langevin process and then updating it iteratively to obtain a more accurate context-likelihood graph.

Our problem represents the image as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes in the graph represented by a matrix of dimension $R^{n \times D}$, n being the number of objects in the image and D being the object feature dimension, and $\mathcal{E} \in R^{n \times n}$ is the set of edges represented by the relations between the objects in the image. In our case, we represent $\mathcal{E}$ as a connectivity matrix of dimension $R^{n \times n}$ where each entry is a value between 0 and 1, representing the strength of the relation between the object pairs. Although estimating the underlying object relationship of the image is important to us, our main goal is to correctly detect the objects in the image, which is a discriminative task. Therefore we want to learn the conditional distribution $p(z|\mathcal{N}, \mathcal{E})$, where $p(z)$ is the classification probability for the n object proposals, along with the data distribution $p(\mathcal{N}, \mathcal{E})$. Thus we model the joint distribution for our tasks as follows:

$$p_{\theta,\phi}(\mathcal{N},\mathcal{E},z) = p_\phi(z|\mathcal{N},\mathcal{E})\frac{\exp(-E_\theta(\mathcal{N},\mathcal{E}))}{Z_{\theta,\phi}}, \quad (8)$$

where $p_\phi(z|\mathcal{N},\mathcal{E})$ represents the classification model and $E_\theta(\mathcal{N},\mathcal{E})$ is the energy function, which is modeled by a neural network parameterized by θ. An image I is initially passed through the base detector producing initial predictions which are used to create the context-likelihood graph $\mathcal{G}_0 = (\mathcal{N}_0, \mathcal{E}_0)$. We consider $\mathcal{G}_0$ to be the initial state of our Langevin dynamics sampling process which is updated

through the energy function $E_\theta(\mathcal{N},\mathcal{E})$ to produce the refined graph, $\mathcal{G}_t$ after t sampling iterations and then passed through the message passing and the classification model $p_\phi(z|\mathcal{N},\mathcal{E})$ to get the final predictions.

Optimization. We optimize the model parameters jointly by minimizing the joint loss function stated below

$$\mathcal{L}_{\text{total}} = -\mathbb{E}_{p_d(\mathcal{N},z)}[\log p_\phi(z|\mathcal{N},\mathcal{E})] + \mathbb{E}_{p_d(\mathcal{N},z)}[E_\theta(\mathcal{N},\mathcal{E})]$$
$$- \mathbb{E}_{\blacksquare(q_{\theta,\phi})}[E_\theta(\mathcal{N},\mathcal{E})] \quad (9)$$
$$+ \mathbb{E}_{q_{\theta,\phi}(\mathcal{N},\mathcal{E})}\left[E_{\blacksquare(\theta)}(\mathcal{N},\mathcal{E}) - \log p_{\blacksquare(\phi)}(z|\mathcal{N},\mathcal{E})\right] \ ,$$

where $p_d(\mathcal{N},z)$ and $q_{\theta,\phi}(\mathcal{N},\mathcal{E})$ denote the data distribution and the t sequential MCMC sampling from the joint energy distribution. The first term, referred to as $\mathcal{L}_{CE}\{P,P^*\}$ in Eq. (3) maximizes the data log likelihood for ground truth samples by training the classification model; the second and third terms are the ground truth and sampled energy values which trains the energy model; the fourth term corresponds to the KL divergence loss which trains the sampling process, and the final term corresponds to the log likelihood of the sampled data. Training the whole network jointly forces the classification model to adapt to ground truth samples, minimizes the ground truth energy value, and forces the sampling process to sample a graph representation whose energy will be close to the ground truth energy value, as well as a correct classification for the object detection task.

Inference. During testing, we pass an image through the base network, gather the object proposals and initial classification probabilities, P. We calculate $\mathcal{E}_0 = P \times T \times P^T$ and $\mathcal{N}_0$ with the feature vectors creating the initial graph state $\mathcal{G}_0$. Then through the SGLD sampling, we refine this graph t times to produce the final graph state $\mathcal{G}_t$, which is then fed through the message passing and classification model to obtain the final classification and bounding box results.

4. Experiments

4.1. Datasets, Evaluation and Implementation

We conduct our experiments on two established benchmark datasets: Visual Genome [23] and MS-COCO 2017 [28]. The task is to localize an object by its bounding box and classify it into one of the preset object categories.

Visual Genome. For the Visual Genome dataset [23], we use the latest release (v1.4). Following standard practice, we consider two sets of target classes: the 1000 most frequent categories and the 3000 most frequent categories, resulting in two settings: $\textit{VG}_{1000}$ and $\textit{VG}_{3000}$. We split the 92.9K images of the dataset with objects on these class sets into 87.9K for training and 5K for testing. Instead of the raw names of the categories, we use the synsets as class labels due to inconsistent label annotations, following previous works [21, 46]. The dataset contains an average of 21 objects per image, with a maximum of 145.

MS-COCO. We evaluate on the 80 object categories of MS-COCO 2017 [28] with 118k images for training and 5k for evaluation. There are a maximum of 93 object instances per image in this dataset with an average of 7 objects.

Evaluation criteria. We follow the common evaluation conventions from the literature [21, 46]. Thus, we adopt the full set of metrics from the COCO detection evaluation criteria comprising the mean Average Precision (mAP) across IoU thresholds ranging from 0.5 to 0.95 with an interval of 0.05 as well as mAP at IoU thresholds of 0.5 and 0.75, multiple scales (small, medium and large), Average Recall (AR) with different number of given detections per image (1, 10, 100) and different scales (small, medium, big).

Two base detectors. We implement our method on top of two base detectors: the convolutional Faster-RCNN [38] and transformer DETR [5], using feature pyramid networks [27] along with a ResNet-101 [18] pretrained on ImageNet [10] as their backbone. For training Faster-RCNN, we use an image size of 1333×800 and augment with random horizontal flips. We choose $N{=}512$ region proposals, represented by features of dimension 1024. We use stochastic gradient descent as the optimizer and train on 8 GPUs with a mini-batch size of 2; an initial learning rate of 0.02, reducing it twice by 10; a weight decay of 10^{-4} is used along with a momentum of 0.9. We train the model for 20 epochs with learning rate reductions at the 8th and 11th epoch. For DETR, we use 6 encoder and decoder layers, 200 output queries, a learning rate of 10^{-4}, and train it for 100 epochs on 8 GPUs with a mini-batch size of 2.

Context-likelihood graph propagation. Edges in our graph represent the relationship between two objects in a scene. We treat it as an attention weight between the two nodes. Taking inspiration from [16], we use a graph attention based network to propagate the node and edge features jointly, additional details of which are provided in the supplementary material. For the network, we use two layers with an input dimension of 1024 and a final output dimension of 256. We use skip connections in between the layers to propagate feature information. Sigmoid activation is used in the graph message passing computations. The final output features of dimension 256 are concatenated with the original features of dimension 1024, and fed into the classification and bounding box regression layers.

Energy model. The graph representation is pooled into a vector via attention pooling. We use two linear layers after pooling which converts the features into a scalar energy value. The noise added during sampling has a variance of 0.0001; the SGLD learning rate is fixed at 10, and we use 5 steps of sampling to get the final output graph.

4.2. Ablations

Context-likelihood graph and graph refinement ablation In Tab. 1, we report the contributions of the components

	AP↑	AP$_{50}$↑	AP$_{75}$↑	AP$_S$↑	AP$_M$↑	AP$_L$↑
F-RCNN [27]	7.3	12.7	7.6	4.2	7.8	11.0
F-RCNN w CL	8.2	13.3	8.6	4.5	8.7	12.8
F-RCNN w GR	**8.5**	**13.7**	**8.8**	**4.8**	**8.8**	**13.4**
F-RCNN w GT	15.3	20.1	16.5	12.5	15.9	17.5
DETR [5]	5.8	9.9	5.7	2.1	5.6	10.6
DETR w CL	6.6	10.7	6.6	2.6	6.8	11.6
DETR w GR	**6.8**	**11.2**	**6.8**	**2.7**	**7.0**	**12.1**
DETR w GT	12.7	21.0	12.9	5.2	11.8	19.7

Table 1. **Context-likelihood graph and graph refinement ablation** on VG$_{1000}$. We compare the baseline detector with context-likelihood graph (w **CL**) and the graph refinement (w **GR**). For reference we show the result obtained with ground truth relation set (w **GT**) in gray. Best results are highlighted as first , second , and third . Using context-likelihood, we achieve consistent improvement while graph refinement improves results further.

	#Proposals	AP↑	AP$_{50}$↑	AP$_{75}$↑
This paper w DETR	200	6.6	11.2	6.8
This paper w DETR	300	7.2	11.8	7.4
This paper w F-RCNN	256	8.0	13.0	8.4
This paper w F-RCNN	512	**8.2**	**13.4**	**8.5**

Table 2. **Detectors and proposals.** We test our context-likelihood energy-refined graph on VG_{1000} [23] with DETR [5] and F-RCNN [27] as backbones. Results on DETR are more affected by the number of proposals than F-RCNN.

of our method. We see that only using context-likelihood graph, we improve consistently on both base detectors, with context-likelihood energy-refinement our results improve further. For reference, we also provide the results of our oracle experiment with ground truth relation set $\mathcal{E}_{GT}$.

Detectors and proposals. In Tab. 2, we ablate the influence of selecting the number of object proposals for different object detectors. For DETR, there is a performance gap between 200 and 300 object queries with AP values changing from 6.6 to 7.2. With higher number of queries, the size of attention computation increases exponentially, leading to slower training. For F-RCNN however, the number of proposals during training influences our results only slightly, due to already high number of proposals.

Energy model training. To understand the impact of energy model training in our framework, we plot the classification accuracy as a function of training time in Fig. 4. As expected, training accuracy is highest for the oracle experiment with ground truth context-likelihood graph. The baseline experiment has the lowest accuracy, followed by the one with context-likelihood graph. For context-likelihood energy-refined graph, the training accuracy further improves, moving closer to the oracle experiment.

Benefit for rare objects. In Fig. 5, we show the impact of our method on different object classes. We plot the classes on the x-axis in decreasing order of frequency of occurrence in the VG$_{1000}$ dataset; below it we plot the relative improvement of our method over Faster-RCNN for each class. As
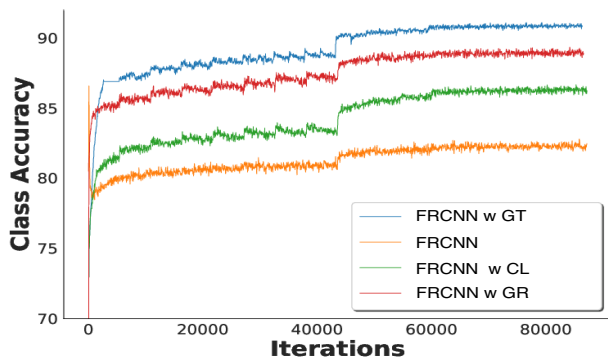
Figure 4. **Energy model training.** As expected, the oracle experiment with ground truth context-likelihood graph (w GT) achieves the highest accuracy. Context-likelihood graph (w CL) reports higher training accuracy over the baseline **F-RCNN**. When context-likelihood energy-refinement (w GR) is introduced, the accuracy of our method improves further.
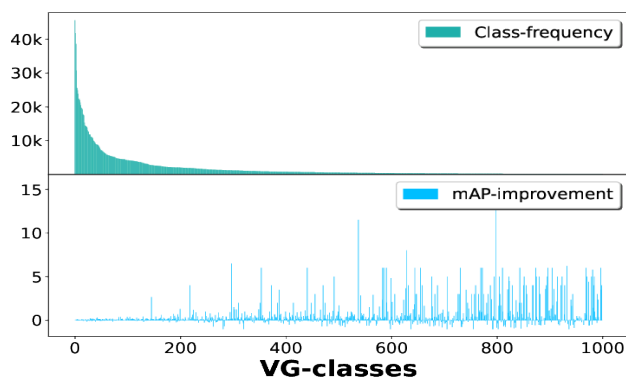

Figure 5. **Benefit for rare objects on VG$_{1000}$ [23].** Object classes plotted on the x-axis are arranged in decreasing order of their frequency in the dataset. As expected, our method has a higher relative improvement for rare object classes.

expected, most of the improvement happens for rare classes of the dataset. We attribute this to the rare objects not having enough examples to learn a meaningful feature representation, so they benefit from inter-object relation statistics more than classes with ample instances.

Inference speed. We compare the inference speed of our method with F-RCNN [27] on VG$_{1000}$ [23]. Our method improves over the baseline from 7.3 to 8.5 AP, at the expense of a drop in fps from 14.6 to 9.4, due to the graph refinement stage. The supplementary material also provides a plot for inference time and AP for various refinement steps.

4.3. Comparison with Others

Comparison on VG$_{1000}$ and VG$_{3000}$ [23]. For the comparative evaluation of our method, we first focus on the large-scale detection benchmarks: VG$_{1000}$ (Visual Genome with 1000 categories) and VG$_{3000}$ (Visual Genome with 3000 categories). For both these settings, we report the re-

sults of our method on top of F-RCNN and DETR and compare with alternative methods modeling object interrelationships which are trained with similar backbone and dataset settings. As can be seen from Tab. 3, we achieve competitive results for both datasets on almost all metrics when our method is applied on F-RCNN, be it that SGRN [45] outperforms us slightly on the AP_L, AR_1 and AR_L metrics for the VG$_{3000}$ dataset. For VG$_{1000}$ we outperform Xu *et al.* [44] when we compare with their intra-domain setting using the same backbone and training data (8.5 AP vs. 7.9 AP). They have an edge for their inter-domain setting (8.8 AP), when training with multiple datasets (COCO, VisualGenome and ADE), suggesting our approach may profit as well from training with more data. We also see a good improvement in our average recall numbers with AR_M improving by 6.5 and 5.1 over baseline F-RCNN for VG$_{1000}$ and VG$_{3000}$ respectively. DETR results are below the results on F-RCNN due to DETR not producing region proposals, making it dependant on the number of object instances per image. Nonetheless, we still observe an improvement over DETR on VG1000 from 5.8 AP to 6.8 AP when our method is applied.

Comparison on MS-COCO [28]. We report results based on F-RCNN on MS-COCO and compare with previously reported results in Tab. 4. We achieve an absolute improvement of 4.2 AP over F-RCNN, and are either better or comparable to alternatives. We do not achieve as much improvements as in the Visual Genome datasets, partly because, the prior statistics have been directly transferred from the Visual Genome dataset based on the common classes between the two, and hence, fail to capture the true prior distribution of the object classes in MS-COCO. For the transformer-based detectors DETR [5] and Deformable DETR [49] we also achieve a consistent improvement.

Limitations. Our approach cannot create a detection if a corresponding region proposal is missing. However, we observe in practice 512 proposals are enough to capture small instances, initially classified incorrectly, and later predicted correctly via our graph refinement. Although our method is optimized for region-based algorithms, adapting it to single-stage detectors like SSD or Yolo requires only minor adjustments in the training process. For example, for Yolo v3, we use the objectness scores to select candidate nodes above a 0.5 threshold, form edges based on initial class predictions, and update features with enriched information from the context-likelihood graph for classification.

5. Conclusion

We demonstrated a new approach to exploit inter-object relations for object detection by generating a context-likelihood graph based on initial object predictions which leads to improved detector performance. Second, we show the potential of this context-likelihood graph via an oracle

	Method	AP↑	AP$_{50}$↑	AP$_{75}$↑	AP$_S$↑	AP$_M$↑	AP$_L$↑	AR$_1$↑	AR$_{10}$↑	AR$_{100}$↑	AR$_S$↑	AR$_M$↑	AR$_L$↑
VG$_{1000}$	Light-head RCNN [26]	6.2	10.9	6.2	2.8	6.5	9.8	14.6	18.0	18.7	7.2	17.1	25.3
	Cascade RCNN [4]	6.5	12.1	6.1	2.4	6.9	11.2	15.3	19.4	19.5	6.1	19.2	27.5
	R-RCNN [46]	8.2	13.3	8.5	4.4	8.9	12.9	16.4	22.2	22.5	12.3	22.1	27.1
	SGRN [45]	8.1	13.6	8.4	4.4	8.2	12.8	19.5	26.0	26.2	12.4	23.9	34.0
	U-RCNN-intra [44]	7.9	13.7	8.3	-	-	-	-	-	-	-	-	-
	U-RCNN-inter [44]	8.8	14.2	9.3	5.0	9.4	13.3	17.5	23.7	24.0	13.4	23.3	28.8
	F-RCNN [27]	7.1	12.7	7.2	3.9	7.6	11.1	14.8	19.7	19.9	10.6	18.8	24.9
	F-RCNN* [27]	7.3	12.7	7.6	4.2	7.8	11.0	15.1	20.1	20.3	11.2	19.4	24.3
	This paper with F-RCNN*	8.5	13.7	8.8	4.8	8.8	13.4	19.8	27.2	27.5	14.1	25.9	34.2
	DETR* [5]	5.8	9.9	5.7	2.1	5.6	10.6	12.1	17.4	18.0	6.6	15.8	26.1
	This paper with DETR*	6.8	11.2	6.8	2.7	7.0	12.1	13.8	20.0	20.6	8.3	18.2	28.9
VG$_{3000}$	Light-head RCNN [26]	3.0	5.1	3.2	1.7	4.0	5.8	7.3	9.0	9.0	4.3	10.3	15.4
	Cascade RCNN [4]	3.8	6.5	3.4	1.9	4.8	4.9	7.1	8.5	8.6	4.2	9.9	13.7
	R-RCNN [46]	4.3	6.9	4.6	3.2	6.0	7.9	8.5	11.1	11.2	8.3	13.7	16.2
	SGRN [45]	4.5	7.4	4.3	2.9	6.0	8.6	10.8	13.7	13.8	8.1	15.1	21.8
	F-RCNN [27]	3.7	6.5	3.7	2.1	4.9	6.8	7.6	9.8	9.9	6.8	11.8	14.6
	F-RCNN* [27]	3.3	5.8	3.4	2.4	4.6	5.9	6.8	8.9	9.0	6.5	11.1	13.0
	This paper with F-RCNN*	4.5	7.4	4.7	3.1	6.1	8.4	10.6	14.0	14.1	8.8	16.2	21.5

Table 3. **Comparison with others on VG$_{1000}$ and VG$_{3000}$ [23].** We report the results of our context-likelihood energy-refined graph on top of F-RCNN and DETR, where we applied context-likelihood energy-refinement graph. We compare with alternative methods modeling object interrelationships which are trained with similar backbone and dataset settings. Methods denoted with an asterisk like F-RCNN* refer to results we obtained by running the original code with default parameters, while the other numbers are the reported values. Rows in gray use more datasets for training. We achieve competitive results for both datasets on almost all metrics. Best results are highlighted as first , second , and third .

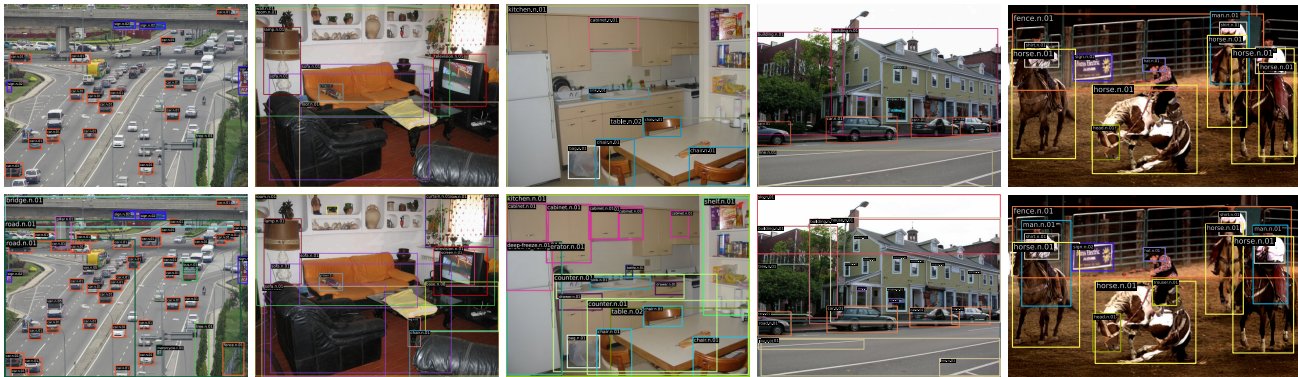Figure 6. **Qualitative results** from the VG$_{1000}$ test set [23] for a F-RCNN* object detector (top) and our results (bottom). We observe positive effects of exploiting contextual-likelihood graphs with energy-based refinement. Especially for repetitive occurrence of objects like the cars, cabinet doors or windows, the information that other such objects exist in the scene helps detection. Baseline performance can be poor, as is evident by failure to detect many cars in the first top-left image, our approach is also beneficial for such challenging cases.

Method	AP↑	AP$_{50}$↑	AP$_{75}$↑	AR$_1$ ↑	AR$_{10}$↑	AR$_{100}$↑
HKRM [5]	37.8	58.0	41.3	-	-	-
SGRN [46]	41.7	62.3	45.5	-	-	-
R-RCNN [46]	42.9	-	-	-	-	-
F-RCNN [38]	38.3	60.5	42.0	32.2	50.7	53.0
This paper w F-RCNN	42.5	62.1	45.8	34.0	53.4	55.9
DETR [5]	39.9	60.4	41.7	32.2	50.7	53.0
This paper w DETR	40.6	61.2	42.1	34.0	53.4	55.9
Def-DETR [49]	44.3	63.2	48.6	32.2	50.7	53.0
This paper w Def-DETR	44.7	64.0	49.0	34.0	53.4	55.9

Table 4. **Comparisons on MS-COCO [28].** Our context-likelihood energy-refined graph performs comparable to alternative methods, with prior statistics transferred from the Visual Genome dataset, demonstrating the generalizability.

experiment. Finally, we learnt the joint distribution of objects and their relations using an energy-based model, enabling us to sample a refined graph representation of an image. We further demonstrated that refinement of the graph representation at test time yields further improvements, outperforming alternative methods on the challenging VG$_{1000}$ and VG$_{3000}$ datasets.

Acknowledgements

References

[1] David H Ackley, Geoffrey E Hinton, and Terrence J Se-jnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985. 2

[2] Ershat Arkin, Nurbiya Yadikar, Yusnur Muhtar, and Kurban Ubul. A survey of object detection based on cnn and transformer. In *2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML)*, pages 99–108. IEEE, 2021. 1

[3] Sergey Bartunov, Jack W Rae, Simon Osindero, and Timothy P Lillicrap. Meta-learning deep energy-based memory models. *arXiv preprint arXiv:1910.02720*, 2019. 2

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 1, 8

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2, 4, 6, 7, 8

[6] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7239–7248, 2018. 2

[7] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE conference on computer vision and Pattern recognition*, pages 3076–3086, 2017. 2

[8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29, 2016. 2

[9] David Dehaene, Oriel Frigo, Sébastien Combrexelle, and Pierre Eline. Iterative energy-based projection on a normal data manifold for anomaly localization. *arXiv preprint arXiv:2002.03734*, 2020. 2

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[11] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020. 2

[12] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019. 3, 4

[13] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 2

[14] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 2

[15] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location

and appearance. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2

[16] Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9211–9219, 2019. 6

[17] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 6

[19] Geoffrey Hinton, Simon Osindero, Max Welling, and Yee-Whye Teh. Unsupervised discovery of nonlinear structure using contrastive backpropagation. *Cognitive science*, 30(4):725–731, 2006. 2

[20] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002. 2, 4, 5

[21] Chenhan Jiang, Hang Xu, Xiaodan Liang, and Liang Lin. Hybrid knowledge routed modules for large-scale object detection. *Advances in Neural Information Processing Systems*, 31, 2018. 1, 2, 3, 5, 6

[22] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 2, 3, 4, 5, 6, 7, 8

[24] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 2, 4

[25] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 2

[26] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head R-CNN: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017. 1, 8

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 6, 7, 8

[28] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 5, 6, 7, 8

[29] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *IJCV*, 2020. 1

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[31] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020. 2

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[33] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. *arXiv preprint arXiv:1612.04844*, 2016. 2

[34] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014. 2

[35] Duy-Kien Nguyen, Jihong Ju, Olaf Booij, Martin R Oswald, and Cees GM Snoek. Boxer: Box-attention for 2d and 3d transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4773–4782, 2022. 1, 2

[36] Weili Nie, Arash Vahdat, and Anima Anandkumar. Controllable and compositional generation with latent-space energy-based models. *Advances in Neural Information Processing Systems*, 34:13497–13510, 2021. 2

[37] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 2

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 2, 4, 6, 8

[39] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021. 4

[40] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13936–13945, 2021. 3

[41] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011. 5

[42] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020. 1, 2

[43] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models. *arXiv preprint arXiv:2010.00654*, 2020. 3

[44] Hang Xu, Linpu Fang, Xiaodan Liang, Wenxiong Kang, and Zhenguo Li. Universal-RCNN: Universal object detector via transferable graph r-cnn. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12492–12499, 2020. 1, 2, 7, 8

[45] Hang Xu, Chenhan Jiang, Xiaodan Liang, and Zhenguo Li. Spatial-aware graph relation network for large-scale object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9298–9307, 2019. 1, 2, 3, 7, 8

[46] Hang Xu, ChenHan Jiang, Xiaodan Liang, Liang Lin, and Zhenguo Li. Reasoning-RCNN: Unifying adaptive global reasoning into large-scale object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6419–6428, 2019. 1, 2, 5, 6, 8

[47] Minghao Xu, Meng Qu, Bingbing Ni, and Jian Tang. Joint modeling of visual objects and relations for scene graph generation. *Advances in Neural Information Processing Systems*, 34:7689–7702, 2021. 2

[48] Mohsen Zand, Ali Etemad, and Michael Greenspan. Objectbox: From centers to boxes for anchor-free object detection. *arXiv preprint arXiv:2207.06985*, 2022. 2

[49] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1, 2, 7, 8