

# Detecting out-of-context objects using contextual cues

**Manoj Acharya<sup>1</sup>, Anirban Roy<sup>2</sup>, Kaushik Koneripalli<sup>2</sup>, Susmit Jha<sup>2</sup>, Christopher Kanan<sup>1,3,4</sup> and Ajay Divakaran<sup>2</sup>**

<sup>1</sup>Rochester Institute of Technology, Rochester NY 14623, USA

<sup>2</sup>SRI International, Princeton NJ 08540, USA

<sup>3</sup>Paige, New York NY 10036, USA, <sup>4</sup>Cornell Tech, New York NY 10044, USA

{manoj,kanan}@rit.edu, {anirban.roy, kaushik.koneripalli, susmit.jha, ajay.divakaran}@sri.com

## Abstract

This paper presents an approach to detect out-of-context (OOC) objects in an image. Given an image with a set of objects, our goal is to determine if an object is inconsistent with the scene context and detect the OOC object with a bounding box. In this work, we consider commonly explored contextual relations such as co-occurrence relations, the relative size of an object with respect to other objects, and the position of the object in the scene. We posit that contextual cues are useful to determine object labels for in-context objects and inconsistent context cues are detrimental to determining object labels for out-of-context objects. To realize this hypothesis, we propose a graph contextual reasoning network (GCRN) to detect OOC objects. GCRN consists of two separate graphs to predict object labels based on the contextual cues in the image: 1) a representation graph to learn object features based on the neighboring objects and 2) a context graph to explicitly capture contextual cues from the neighboring objects. GCRN explicitly captures the contextual cues to improve the detection of in-context objects and identify objects that violate contextual relations. In order to evaluate our approach, we create a large-scale dataset by adding OOC object instances to the COCO images. We also evaluate on recent OCD benchmark. Our results show that GCRN outperforms competitive baselines in detecting OOC objects and correctly detecting in-context objects.

## 1 Introduction

Our goal is to detect objects that appear out-of-context (OOC) in an image. Given an image with a set of objects, the task we tackle is to detect objects that are OOC and also correctly identify the in-context objects. Typically, objects in natural images appear in a suitable context and it is useful to consider context while detecting objects [Choi *et al.*, 2010; Oliva and Torralba, 2007; Gould *et al.*, 2008; Divvala *et al.*, 2009; Koller and Friedman, 2009; Beery *et al.*, 2018; Zhang and Chen, 2012; Sun and Jacobs, 2017; Bomatter *et al.*, 2021]. While appropriate contextual cues are shown to be

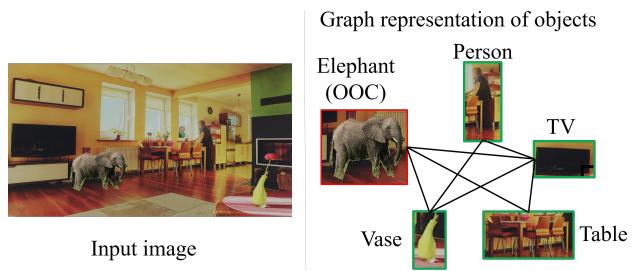


Figure 1: We aim to detect OOC object instances in an image. Given the ‘elephant in the room’ image, we build a graph based on the objects to share context cues between the objects. In-context objects are highlighted with green frames and the OOC object is highlighted with a red frame. Here, the ‘elephant’ is OOC as it is not consistent with the co-occurrence relations with other objects in the image.

useful for object detection, incorrect contextual cues can negatively impact the performance of object detection for both humans [Zhang *et al.*, 2020; Bomatter *et al.*, 2021] and machine learning approaches [Torralba, 2003; Choi *et al.*, 2010; Rosenfeld *et al.*, 2018; Madras and Zemel, 2021]. Thus, in order to develop reliable object detection systems, it is crucial to detect objects that appear in unusual contexts where the predictions may not be reliable.

While detecting in-context objects is extensively explored, detecting OOC objects is much less well studied; however, recent work has shown that the presence of OOC objects can severely affect an object detectors’ ability to detect in-context objects in the image [Rosenfeld *et al.*, 2018]. Detecting OOC objects is difficult for humans as well [Bomatter *et al.*, 2021]. Motivated by these observations, we develop an approach for detecting OOC objects by capturing contextual cues and checking inconsistent object-context relations.

Detecting OOC objects is challenging as these objects appear normal in isolation and it is only the context that makes them unusual. We also assume that OOC object classes are present during training and thus relying on object appearance alone may not be sufficient for OOC detection. A few early works aim to detect OOC objects [Choi *et al.*, 2010] where hand-crafted contextual relationships (e.g., co-occurrence, the constraint on object size) are exploited for OOC detection. Some recent approaches consider neural network-based

models to learn contextual relations in a data-driven manner [Bomatter *et al.*, 2021; Dvornik *et al.*, 2018]. Many of such approaches model context as generic background and do not exploit informative cues such as label dependencies and objects properties in the image [Beery *et al.*, 2018; Zhang *et al.*, 2020]. To address these challenges, we propose an approach to explicitly consider contextual cues for object detection in order to discover the inconsistency for OOC detection. We posit a simple yet effective hypothesis to detect OOC objects in an image - for in-context objects, label predictions with and without contextual cues are expected to match, and for out-of-context objects, predictions with and without context are *not* expected to match as the objects are not consistent with the context. Contextual cues are only useful to detect objects that are in-context and adversely affect the detection of out-of-context objects. For example, as shown in Fig. 1, detecting in-context objects such as ‘person’ can benefit from the context cues in terms of other indoor objects. However, in the case of the OOC ‘elephant’ object, inconsistent contextual cues can be confusing as elephants usually do not appear in indoor scenes.

We propose a graph contextual reasoning network (GCRN) to capture contextual cues for predicting object class prediction. We model context by defining a graph over the objects in an image where nodes represent objects and edges represent object-to-object relations. Specifically, our GCRN model consists of two graph models: 1) representation graph (repG) to learn useful node representations to predict object labels, and 2) context graph (conG) to capture contextual cues to predict object labels. We consider three contextual cues for OOC detection: co-occurrence of objects, location, and shape similarity of object boxes. These contextual cues are shown to effective for in-context object detection [Choi *et al.*, 2010; Gould *et al.*, 2008; Divvala *et al.*, 2009; Koller and Friedman, 2009; Zhang and Chen, 2012]. Each graph model is realized by a Graph Convolutional Network (GCN) to ensure efficient learning and inference [Dai *et al.*, 2016; Kipf and Welling, 2017; Hamilton *et al.*, 2017; Velickovic *et al.*, 2019; Qu *et al.*, 2019]. Both the models are trained together to ensure GCRN learns informative node representation as well as context-dependency among the objects. GCRN has several advantages over existing approaches. Compared to the graph-based models that define contextual relations using hand-crafted features, such as conditional random fields [Gould *et al.*, 2008; Gould *et al.*, 2009; Zhang and Chen, 2012], GCRN learns these cues in a data-driven manner. Compared to the standard GCN models that do not consider the dependency among the node predictions [Dai *et al.*, 2016; Kipf and Welling, 2017; Hamilton *et al.*, 2017], conG in GCRN explicitly captures the context while predicting node labels. Our experiments show GCRN significantly outperforms standard GCNs in OOC detection.

Our main contributions include:

- We propose an approach to simultaneously detect OOC objects and in-context objects in an image.
- We propose a graph contextual reasoning network (GCRN) to detect OOC objects by explicitly modeling contextual cues to predict object labels. GCRN allows a

flexible framework to model and learn contextual cues in a data-driven manner. Unlike prior works, GCRN does not require manual specification of the contextual constraints.

- We create a large-scale OOC dataset to evaluate our approach. The dataset consists of roughly 100K OOC images where objects occur in various OOC scenarios.

## 2 Related Works

**Context for object detection.** Contextual cues are important for object detection and segmentation. Graph-based models provide a flexible way to represent context where nodes represent objects and edges represent pair-wise relations among the objects [Choi *et al.*, 2010; Gould *et al.*, 2008; Zhang and Chen, 2012]. Among graph-based models, conditional random fields (CRF) are explored extensively where contextual cues are represented by edge potentials. Common contextual cues include co-occurrence, spatial distance, geometric and appearance similarity [Choi *et al.*, 2010; Gould *et al.*, 2008; Divvala *et al.*, 2009; Koller and Friedman, 2009; Zhang and Chen, 2012]. More recently, graphical models are combined with neural networks to exploit data-driven feature learning. Graph convolutional networks (GCN) [Dai *et al.*, 2016; Kipf and Welling, 2017; Hamilton *et al.*, 2017] provide a convolutional implementation of the graphical models combining the power of representation learning of neural networks with the structured representation of graphs. However, standard GCNs do not explicitly capture the contextual relations that are crucial to detect OOC objects [Qu *et al.*, 2019]. Our GCRN learns two GCNs, one for learning the feature representation and another for capturing context cues, to effectively detect OOC objects.

**Out-of-context object detection.** Existing studies have argued the importance of OOC object detection as these affect the performance of object detection for both humans and machines [Choi *et al.*, 2010; Rosenfeld *et al.*, 2018; Zhang *et al.*, 2020; Bomatter *et al.*, 2021]. Choi et al. [Choi *et al.*, 2012] define OOC objects that violate common contextual rules (e.g., flying cars) in terms of unusual background, unusual size, etc. They consider a graph-based model to capture such relations among the objects. Some of the approaches consider context as the entire background with respect to an object and detect OOC objects that are inconsistent with the background. Dvornik et al. [Dvornik *et al.*, 2018] usual pair of objects as OOC instances. Unlike these approaches, we define the context for a target object as its relation (object classes, relative size, relative location) with other objects and the scene.

**Graph Convolutional Networks.** GCNs provide an end-to-end neural network-based realization of graph models and are shown to be successful in object detection [Dai *et al.*, 2016; Kipf and Welling, 2017; Hamilton *et al.*, 2017; Qu *et al.*, 2019]. In GCNs, a robust node representation is learned by sharing the representations with neighboring nodes. This node-to-node exchange is implemented via a convolutional operation that facilitates efficient learning and inference in GCNs. However, GCNs typically avoid modeling the label dependency among the nodes [Qu *et al.*, 2019].

Thus, common GCNs frameworks are not effective to capture global contextual cues. Recently, graph Markov neural networks (GMNN) [Qu *et al.*, 2019] are proposed to capture label dependency by simultaneously learning two GCNs - one for learning node representations and another for learning label dependency. We consider a similar setup where one network learns the node representation and another network captures context for predicting mode labels.

### 3 Proposed approach

In order to detect out-of-context (OOC) objects in an image, we hypothesize that, for OOC objects, accurate labels cannot be predicted from the context as these objects are inconsistent with the context. On the other hand, contextual cues are expected to help predict the labels for in-context objects [Choi *et al.*, 2010; Oliva and Torralba, 2007; Gould *et al.*, 2008; Gould *et al.*, 2009; Divvala *et al.*, 2009; Koller and Friedman, 2009; Beery *et al.*, 2018; Zhang and Chen, 2012]. To realize this hypothesis, we propose a graph contextual reasoning network (GCRN) that consists of two graph models: 1) Representation graph (RepG) that learns precise object representation at each node by sharing the representations with its neighbors. RepG relies on the shared representations to predict object labels ignoring context dependency among the labels [Qu *et al.*, 2019]. 2) To complement RepG, we propose a context graph (ConG) that learns context dependency at each node by sharing the context cues with its neighbors. We consider graph convolutional networks (GCN) [Kipf and Welling, 2017; Hamilton *et al.*, 2017] to instantiate both RepG and ConG. We first introduce the GCN framework and then discuss the implementation details of RepG and ConG.

**Graph convolutional network.** Given an image, we build a graph over the objects. Consider a graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of edges. We define  $X = \{x_i\}$  and  $Y = \{y_i\}$  as the feature representation and label of  $i$ th node, respectively. Given this definition, the goal is to predict object labels from the features.

$$H^{l+1} = f^l(W^l, H^l, E), p(y_i|X, E) = \text{SM}(H^L), \quad (1)$$

where  $f^l(\cdot)$  is the convolutional function corresponding to a layer  $l$ .  $f^l(\cdot)$  iteratively updates node representations to  $H^{l+1}$  from the current representation  $H^l$  and using the edges between nodes  $E$ .  $W^l$  represents the parameter for layer  $l$ . Note that  $H^0 = X$ , the initial node representations. The prediction is made by applying a softmax operation (SM) on the final representation  $H^L$ .

**Context-informed prediction.** We formulate context-informed label prediction in a conditional random field framework [Lafferty *et al.*, 2001; Koller and Friedman, 2009] where the conditional distribution over the labels is given by

$$p(Y|X, E) = \frac{1}{Z(X, E)} \prod_{(i,j) \in E} \phi_{i,i}(y_i, y_j, X), \quad (2)$$

where  $Z(X, E)$  is the partition function over the graph and  $\phi_{i,i}(y_i, y_j, X)$  is the potential function over a pair of nodes  $i, j$  [Gould *et al.*, 2008; Gould *et al.*, 2009; Koller and Friedman, 2009]. Lets denote  $\theta$  as the graph parameters. Then,

learning can be done by maximizing the following conditional log-likelihood:

$$\ell_{Y|X}(\theta) = \log p_\theta(Y|X, \theta). \quad (3)$$

However, directly maximizing this likelihood function is intractable due to the combinatorial nature of the partition function [Koller and Friedman, 2009]. Thus, we consider optimizing the approximate likelihood that is shown to successful in learning similar graphical models [Richardson and Domingos, 2006].

$$\ell_{Y|X}(\theta) = \sum_i \log p_\theta(y_i|y_{j \in B(i)}, X, \theta), \quad (4)$$

where  $B(i)$  is the set of neighbors of  $i$ . Intuitively, with this approximation, we only consider the context dependencies based on the neighboring nodes. However, as we consider GCN to capture this dependencies, a few iterations of GCN allows to capture log-range dependencies by iterative message passing [Koller and Friedman, 2009]. Note that even in Equ. 4, the label of a node  $y_i$  is conditioned on both the neighboring context ( $y_{j \in B(i)}$ ) and the feature representation  $X$ . To avoid this dependency, we propose an iterative optimization where in one phase, we optimize to learn only representations and the context dependency in the following phase. This allows each iteration to be optimized efficiently in a pair of GCN framework.

**Representation graph.** In the first phase, we consider a mean-field approximation to remove context dependency and learn only the representation. Thus, the label distribution is defined as

$$p_{\theta_R}(Y|X, E) = \prod_i p_{\theta_R}(y_i|X, E), \quad (5)$$

where  $\theta_R$  is the parameters for representation graph. Our representation graph is implemented by a GCN where object features are used as node representations.

$$H_R^{l+1} = f_R^l(W_R^l, H_R^l, E), p(y_i|X_R, E) = \text{SM}(H_R^L), \quad (6)$$

where  $X_R$  denote the node representations and  $W_R$  denotes the GCN parameters. We consider bounding box features as the node representations. It is evident from Equ. 1 and Equ. 5 that neighboring nodes share feature representations by message passing but context related to object labels are not shared.

**Context graph.** In the second phase, we aim to predict labels from the context. This is also realized by a context GCN as follows.

$$H_C^{l+1} = f_C^l(W_C^l, H_C^l, E), p(y_i|X_C, E) = \text{SM}(H_C^L), \quad (7)$$

where  $X_C$  denote the context-related node representations and  $W_C$  denotes the GCN parameters. We consider context features as the node representations including class labels, position, size of neighboring object boxes.

**Learning.** We perform an iterative learning in an expectation-maximizing (EM) framework [Richardson and Domingos, 2006; Qu *et al.*, 2019]. In the M-step, we learn the representation graph based on the predicted labels of the context graph and in the E-step, keeping the representation fixed,

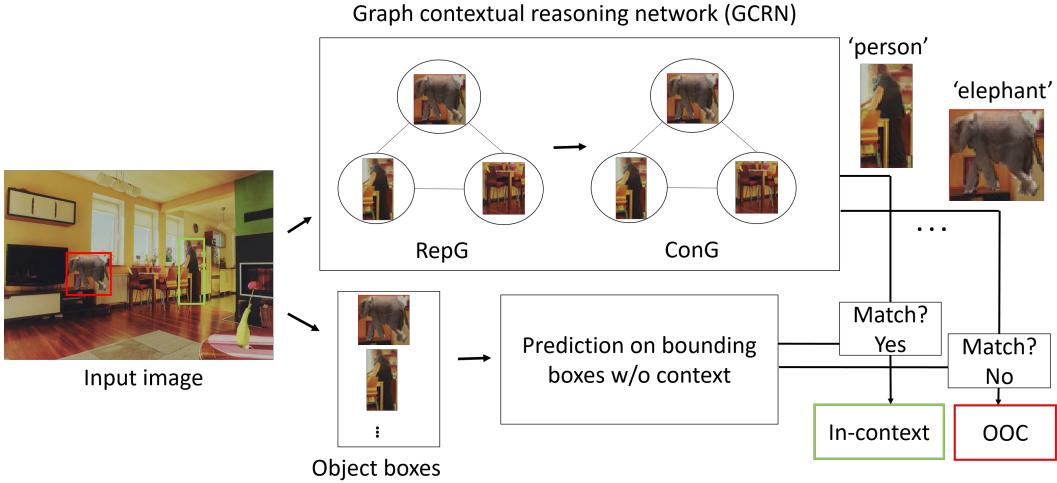


Figure 2: GCRN framework takes in an input image with bounding boxes to construct an image graph  $G = (V, E)$  with vertices  $V$  and edges  $E$ . The graph  $G$  is fed into a RepG network that learns object dependent representation and also ConG network that learns the contextual dependency of objects. Finally, for any vertex  $v$  from  $V$  in relation to graph  $G$ , we compare with the learned context to determine if it is In-Context or Out-Of-Context (OOC).

we update the context graph based on the ground truth labels. Specifically, in E-step, we start with a pre-trained RepG to predict node labels. Then the labels are updated by a fixed ConG by aggregating context from the neighboring nodes. Finally, the RepG is updated to match ConG’s predictions. In M-step, we update ConG based on the ground-truth labels. We continue these iterations until convergence, i.e., the difference of the predictions between RepG and ConG becomes zero or below a threshold.

**Detecting OOC instances.** Given an image, we localize all objects by bounding boxes and consider them as the nodes of the graph. GCRN predicts softmax distribution over class labels for each box by considering context cues from other boxes. We also train an object classifier to predict object labels only from the bounding box ignoring the context cues. Finally, we compare the KL divergence (KLD) between the softmax distribution from both predictions as a measure of OOC. In in-context objects, KLD is expected to be low and for OOC objects, the KLD is expected to be high. Thus we can detect OOC instances by choosing an appropriate threshold. Our prediction framework is shown in figure 2.

**Implementation Details.** We implement the GCRN framework using the DGL toolbox [Wang *et al.*, 2019]. For both RepG and ConG, we consider GCN with four graph convolution layers with residual connections between the layers. The numbers of neurons at these layers are 256, 128, 64, and 64 respectively. Models are trained using an AdamW optimizer with a learning rate of 0.001 without decay. For our GCRN framework, we first train RepG for five epochs in the first phase and alternate between RepG and ConG until convergence. In our experiments, convergence is reached within ten iterations. Residual connections between the layers were crucial for efficient learning.

To create a graph for an image, we detect objects in images and then extract features for the objects to seed

the graph nodes. We consider a MaskRCNN [He *et al.*, 2017] model pre-trained on the COCO dataset to detect objects. We train a ResNet50 [He *et al.*, 2016] network for feature extraction. For context cues, we consider additional geometric features from the bounding box as  $(xmin, ymin, xmax, ymax)$  and 7D spatial feature vector  $[\frac{w}{W}, \frac{h}{H}, \frac{a}{A}, \frac{xmin}{W}, \frac{ymin}{H}, \frac{xmax}{W}, \frac{ymax}{H}]$  where  $w, h, a$  represent the width, height and area of the bounding box and  $W, H, A$  represent the same for the image. These spatial features are important to capture usual size and location of objects training images.

## 4 Experiments

In the following, we introduce the dataset, describe experimental setup, metrics, and present results.

### Datasets.

**The COCO-OOC Dataset.** We create a large-scale OOC dataset to evaluate OOC detection due to the absence of such datasets. As we consider objects and their properties as context, we require object annotations for all the objects along with the OOC object. We consider COCO 2014 [Lin *et al.*, 2014] dataset, consisting 80 indoor and outdoor objects classes, to create the COCO-OOC dataset. Following the common strategy [Blum *et al.*, 2021; Zhang *et al.*, 2020], we place objects in images that violate the contextual relations. We leverage available object mask to transplant objects in images to create OOC scenarios. COCO-OOC consists of 106,036 images with three types of OOC violating co-occurrence relation and size constraints [Blum *et al.*, 2021; Zhang *et al.*, 2020; Bomatter *et al.*, 2021]. Additional details about OOC creation and example images are provided in the supplemental material.

**The OCD Dataset.** In addition to our own dataset, we also use the recent OCD benchmark that has synthetically

Table 1: Comparison with the baselines on the COCO-OOC dataset.

Approach	AUC score
Softmax confidence	0.043
GCRN (w/o ConG)	0.589
GCRN	0.980

generated OOC indoor scenes [Bomatter *et al.*, 2021]. The OOC images are generated from the VirutalHome [Puig *et al.*, 2018] environment. OCD has 11,155 OOC images where objects violate co-occurrence, size, and gravity constraints.

**Experimental setup and metrics.** We train the GCRN model on the COCO train images where objects appear in-context to capture the usual context for object. We test on OOC images aiming to detect the OOC object instance. Note that we assume all OOC objects are available in training and are not novel during the test. Thus relying on appearance cues is not sufficient to detect the OOC instances. As selecting a threshold is often crucial to separating OOC from in-context objects, we consider the AUC score as the metric to robustly evaluate our performance.

**Baselines.** We propose the following baselines based on the state-of-the-art approaches to evaluate various aspects of our GCRN .

**Softmax confidence.** In this baseline, we consider the softmax confidence as a measure of OOC assuming that the confidence would be lower for the OOC objects than usual in-context objects. Softmax confidence is successfully used to detect anomaly [Blum *et al.*, 2021] and novel objects [Liang *et al.*, 2018]. The results in table 1 imply the softmax confidence is not reliable to detect OOCs as, unlike anomalous or novel objects, the OOC objects are observed during training. This baseline implies that relying on objects’ appearance is not sufficient to detect OOC objects.

**Without context graph.** In this baseline, we do not explicitly capture context. Specifically, we omit the context graph and only consider the representation graph. The results in table 1 imply that context graph is important to capture context and representation graph itself is not sufficient to capture context dependencies. This baseline is comparable with the [Bomatter *et al.*, 2021] where object representation is learned through a shared network and context dependencies between the objects (e.g., label dependencies) are not modeled.

**Impact of context on In-context vs. OOC object detection.** We compare the performance of object detection for in-context and OOC objects on COCO-OOC dataset. The results are shown in table 2. As expected, the performance on OOC objects is inferior. Note that ignoring explicit context cues (w/o ConG) degraded the performance for both in-context and OOC objects. This justifies our hypothesis that accurate contextual are helpful for object detection while inconsistent context cues can be confusing.

**Types of OOC.** We evaluate GCRN ’s performance on detecting OOC objects that violate co-occurrence and size constraints in table 3. The performance on co-occurrence is superior as detecting the violation of co-occurrence is relatively easier. For size OOC, we consider the relative size of the ob-

Table 2: Accuracy score across OOC and non-OOC instances in our COCO-OOC dataset. For OOC instances, miclassification refers to correct detection, thus lower accuracy is better while for non-OOC instances higher is better.

Approach	OOC↓	non-OOC↑	Overall
GCRN (w/o ConG)	0.69	0.77	0.76
GCRN	0.30	0.98	0.93

Table 3: AUC results for detecting OOC variants on COCO-OOC.

OOC type	Co-occurrence	Size
AUC score	0.986	0.921

ject boxes to capture these contexts. However, in 2D images, actual size information is partially lost due to the camera projection which makes it harder to capture size context.

**Impact of object detector on OOC detector.** It is important to detect an object to determine whether it is an OOC instance. In this section, we analyze the effect of object detection on OOC detection performance. We consider three modes of operation for our approach: 1) oracle bounding boxes with labels (oracle boxes+labels) where we assume to have access to object locations in the form of bounding boxes and object labels, 2) oracle bounding boxes (oracle boxes, pred labels) where we assume to have to access to only the object locations in the form of bounding boxes but objects labels are to be predicted, 3) predicted bounding boxes (pred boxes) where we do not have access to bounding boxes and an object detector is used to discover the objects. In the oracle boxes+labels setup, errors are completely attributed to OOC detection. In oracle boxes, additional errors may come from misidentifying object class from object classifier. In pred boxes, errors may come from incorrect boxes or wrong labels. The results are shown in table 4. As expected, the performance gradually decreases from oracle boxes+labels to pred boxes. We consider state-of-the-art approaches for both object detection and object classification but these can be further improved to achieve a better performance in OOC detection.

**Results on the OCD dataset.** Following the protocol in [Bomatter *et al.*, 2021], we train on COCO train images and test on OOC images. As in [Bomatter *et al.*, 2021], we consider the overlapping objects as OOC instances. We present the results in table 5. Note that COCO consists of natural images and OCD consists of synthetic images. Due to the domain shift, object detection is relatively more challenging.

Table 4: Impact of the choice of bounding box detector on OOC detection for the COCO-OOC dataset.

Approach	AUC score
GCRN (oracle boxes + labels)	0.980
GCRN (oracle boxes, pred labels)	0.897
GCRN (pred boxes)	0.771

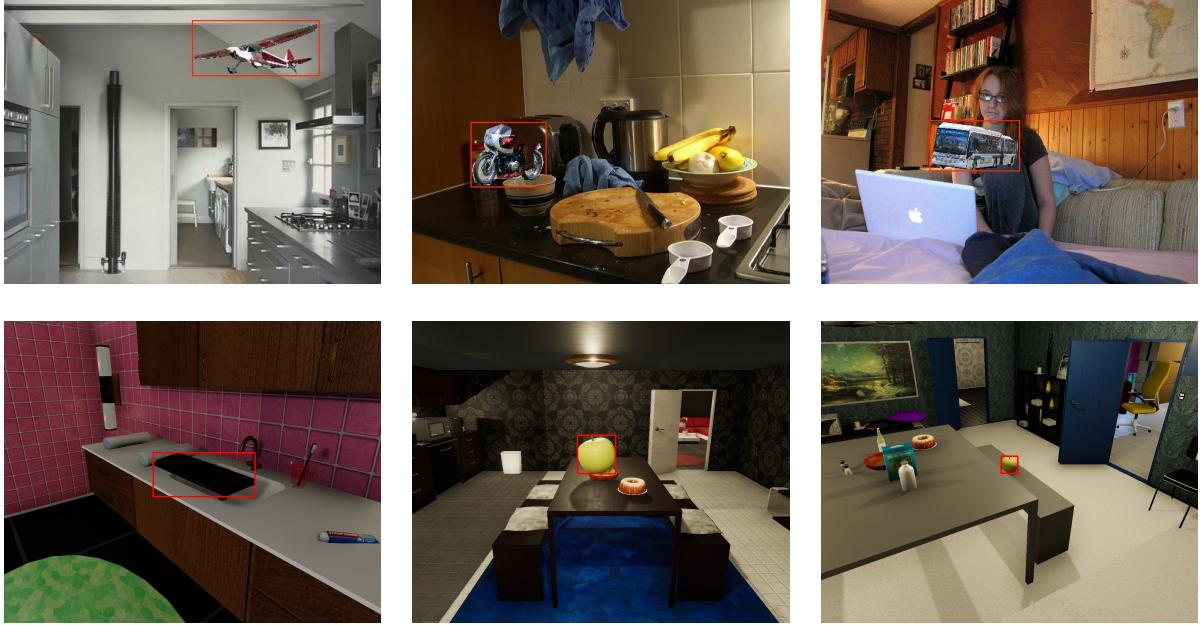


Figure 3: Qualitative examples where our GCRN model correctly identifies Out-Of-Context (OOC) object instances in images. Top row presents images from our COCO-OOC dataset and bottom row presents images from the OCD dataset. The OOC objects are marked by red boxes.

Table 5: Comparison with the baselines on the OCD dataset.

Approach	AUC score
Softmax confidence	0.402
GCRN (oracle boxes, pred labels)	0.587
GCRN (oracle boxes + labels)	0.709

Also context relations, such as co-occurrence and object size, in COCO can be different from those in OCD. Thus, our performance of OOC detection on OCD is lower than that of COCO-OOC.

**Qualitative results.** We present the qualitative results in figure 3 where the top row shows the results on COCO-OOC and the bottom row shows the results on OCD. Our approach successfully detects various types of OOC objects in both natural and synthetic images. Our model is only trained on COCO dataset and can detect OOC instances in the novel OCD images.

**Failure cases.** We present two failure cases in figure 4. In these two cases, ‘laptop’ and ‘backpack’ are not detected as OOC instances. Our model has observed laptops in indoor scenes with persons and indoor furniture as in-context samples. The size of the laptop is slightly larger than usual. Also, a backpack is often observed in outdoor scenes however the size of the backpack is larger than usual. The size features are often partially lost in 2D images due to projection. Having access to additional 3D shape of the objects (e.g., from RGB+Depth) can be helpful to capture context precisely.



Figure 4: Left: laptop is not detected as OOC, Right: backpack is not detected as OOC

## 5 Conclusion

We have presented an approach for detecting OOC objects in images. Detecting OOC objects is crucial to developing a reliable object detection system as detectors perform poorly on OOC objects. The proposed GCRN framework has two components: 1) RepG to learn object representation and 2) ConG to explicitly capture context for object detection. We have created a large-scale OOC dataset to evaluate our performance. We have also evaluated our approach on the recent OCD benchmark. We have considered commonly studied OOC scenarios where objects violate co-occurrence and size constraints. Our evaluation shows context cues are helpful to detect in-context objects while inconsistent context cues can hinder accurate object detection. Thus, explicit modeling of context is crucial for OOC detection, and without this OOC detection is severely affected. We have analyzed the effect of accurate object detection on the performance of OOC detection and as expected accurately localizing and identifying object labels are shown to be important for OOC detection. Finally, we present a few failure cases and propose strategies to mitigate such failures.

## Acknowledgement

This work was supported in part by the U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory under the IoBT REIGN Cooperative Agreement W911NF-17-2-0196, the DARPA Symbiotic Design of Cyber Physical Systems under contract FA8750-20-C-0002, and U.S. National Science Foundation grants #1740079, #1909696, and #2047556. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Army, the United States Department of Defense, or the United States Government.

## References

- [Beery *et al.*, 2018] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.
- [Blum *et al.*, 2021] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The fishscapes benchmark: Measuring blind spots in semantic segmentation. *International Journal of Computer Vision*, pages 1–17, 2021.
- [Bomatter *et al.*, 2021] Philipp Bomatter, Mengmi Zhang, Dimitar Karel, Spandan Madan, Claire Tseng, and Gabriel Kreiman. When pigs fly: Contextual reasoning in synthetic and natural scenes. *arXiv preprint arXiv:2104.02215*, 2021.
- [Choi *et al.*, 2010] Myung Jin Choi, Joseph J. Lim, Antonio Torralba, and Alan S. Willsky. Exploiting hierarchical context on a large database of object categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [Choi *et al.*, 2012] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 33(7):853–862, 2012.
- [Dai *et al.*, 2016] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning*, pages 2702–2711. PMLR, 2016.
- [Divvala *et al.*, 2009] Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *2009 IEEE Conference on computer vision and Pattern Recognition*, pages 1271–1278. IEEE, 2009.
- [Dvornik *et al.*, 2018] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018.
- [Gould *et al.*, 2008] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *International journal of computer vision*, 80(3):300–316, 2008.
- [Gould *et al.*, 2009] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *2009 IEEE 12th international conference on computer vision*, pages 1–8. IEEE, 2009.
- [Hamilton *et al.*, 2017] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.
- [He *et al.*, 2016] 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.
- [He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *CVPR*, pages 2961–2969, 2017.
- [Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [Lafferty *et al.*, 2001] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.
- [Liang *et al.*, 2018] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.
- [Lin *et al.*, 2014] 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.
- [Madras and Zemel, 2021] David Madras and Richard Zemel. Identifying and benchmarking natural out-of-context prediction problems. In *NeuRIPS*, 2021.
- [Oliva and Torralba, 2007] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007.
- [Puig *et al.*, 2018] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtual-home: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8494–8502, 2018.
- [Qu *et al.*, 2019] Meng Qu, Yoshua Bengio, and Jian Tang. Gmnn: Graph markov neural networks. In *International conference on machine learning*, pages 5241–5250. PMLR, 2019.
- [Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1–2):107–136, 2006.
- [Rosenfeld *et al.*, 2018] Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018.
- [Sun and Jacobs, 2017] Jin Sun and David W Jacobs. Seeing what is not there: Learning context to determine where objects are missing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5716–5724, 2017.
- [Torralba, 2003] Antonio Torralba. Contextual priming for object detection. *International journal of computer vision*, 53(2):169–191, 2003.
- [Velickovic *et al.*, 2019] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *ICLR (Poster)*, 2(3):4, 2019.
- [Wang *et al.*, 2019] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, et al. Deep graph library: Towards efficient and scalable deep learning on graphs. 2019.

[Zhang and Chen, 2012] Yimeng Zhang and Tsuhan Chen. Efficient inference for fully-connected crfs with stationarity. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 582–589. IEEE, 2012.

[Zhang *et al.*, 2020] Mengmi Zhang, Claire Tseng, and Gabriel Kreiman. Putting visual object recognition in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12985–12994, 2020.

## Supplemental Material

### S1 COCO-OOC Dataset Details.

Our COCO-OOC dataset is semi-synthetically generated which allows for more control and ensure sufficient diversity for OOC object detection. We exclusively build using the most common 80 object categories found in the COCO dataset. We hope COCO-OOC will also serve as the robustness evaluation set for the detection models trained using the MSCOCO dataset.

Co-occurrence is one of the common contextual cues that humans and vision models alike use for visual predictions. For example, if we see a grass field then we most likely expect a cow to be grazing on it or conversely if we see a beach then probably seeing a cow is not normal and thus is an out-of-context object for such scene. To generate co-occurrence violation scenarios we use COCO ‘things’ ontology which categorizes COCO objects into outdoor and indoor categories. We use this categorization to determine object co-occurrence violation. In total, we consider six variations:

1. Animals in *indoor* settings: bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe.
2. Outdoor objects in *indoor* settings: traffic light, fire hydrant, street sign, stop sign, parking meter.
3. Vehicles in *indoor* settings: bicycle, car, motorcycle, airplane, bus, train, truck, boat.
4. Appliances in *outdoor* settings: microwave, oven, toaster, sink, refrigerator, blender.
5. Electronic in *outdoor* settings: laptop, mouse, remote, keyboard, cell phone.
6. Food in *outdoor* settings: banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake.

Table S1: OOC categories in the COCO-OOC dataset.

Category	# images
Co-occurrence	72137
Size	33899
Total	106036

Another important contextual cue is the size of the objects. Humans have familiarity about the size of the everyday objects and can use those priors to infer about the normalcy of scenes. To generate size violation OOC scenarios, we consider the average size of objects per categories in the training set and upscale objects randomly within 2 to 5 times the original object size.