Detecting Out-Of-Context Objects Using Graph Context Reasoning Network

Manoj Acharya[1,2] , **Anirban Roy**[1] , **Kaushik Koneripalli**[1] , **Susmit Jha**[1] ,
Christopher Kanan[2] and **Ajay Divakaran**[1]
[1]SRI International, Menlo Park CA 94025, USA
[2]Rochester Institute of Technology, Rochester NY 14623, USA
{manoj.acharya, anirban.roy, kaushik.koneripalli, susmit.jha, ajay.divakaran}@sri.com,
{ma7583, kanan}@rit.edu

Abstract

This paper presents an approach for detecting out-of-context (OOC) objects in images. Given an image with a set of objects, our goal is to determine if an object is inconsistent with the contextual relations and detect the OOC object with a bounding box. In this work, we consider common 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. Code and data: https://nusci.csl.sri.com/project/trinity-ooc

1 Introduction

We address the problem of detecting out-of-context (OOC) objects in an image. Given an image with a set of objects, the goal 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 considering contextual cues are useful for object detection [Oliva and Torralba, 2007; 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 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 [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.

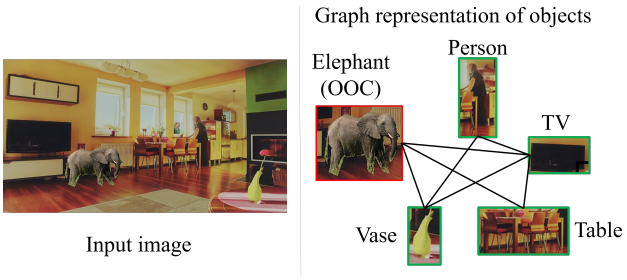

Figure 1: Given the 'elephant in the room' image, we build a graph based on the objects to share contextual cues between the objects. In-context objects are highlighted with green frames and the OOC object is highlighted with a red frame. Here, the 'elephant' is an OOC object as it is inconsistent with the co-occurrence relations with respect to other objects in the image.

While detecting in-context objects is extensively explored, detecting OOC objects remains less studied. Recent work has shown that the presence of OOC objects can severely affect an object detector's ability to detect in-context objects in the image [Rosenfeld *et al.*, 2018]. Detecting OOC objects can become difficult for humans as well [Bomatter *et al.*, 2021]. Motivated by these observations, we develop an approach to explicitly capture the contextual cues and detect OOC objects by checking inconsistent object-context relations.

Detecting OOC objects is challenging as these objects appear normal in isolation and thus relying on the object's appearance alone may not be sufficient for detection. We also assume that OOC object classes are present in the training data and thus these objects are not novel object instances. A few early approaches consider hand-crafted contextual relationships (e.g., co-occurrence, the constraint on object size) for OOC detection [Choi *et al.*, 2012]. Some recent approaches consider neural network-based models to learn con-

textual relations in a data-driven manner [Bomatter *et al.*, 2021]. Many of such approaches consider context as generic background and do not exploit informative cues such as label dependencies and relative object properties in the image [Beery *et al.*, 2018; Zhang *et al.*, 2020]. To address these challenges, we propose an approach to explicitly capture contextual cues for object detection and exploit contextual inconsistencies to detect OOC objects. 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 more likely to match, and for out-of-context objects, predictions with and without context are less likely to match due to the inconsistent contextual cues. In other words, 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 an in-context object such as 'person' can benefit from the contextual cues from other indoor objects. However, in the case of the OOC 'elephant' object, inconsistent contextual cues can be confusing as elephants usually do not appear indoor scenes and do not co-occur with indoor objects.

We propose a graph contextual reasoning network (GCRN) to capture contextual cues for predicting object class labels. We define a graph where nodes represent objects in an image and edges represent object-to-object relations. Specifically, our GCRN consists of two graphical models: 1) representation graph (repG) to learn useful visual representations, and 2) context graph (conG) to capture contextual cues from the scene in order to predict object labels. We consider three contextual relations for OOC detection: co-occurrence of objects, location, and shape similarity of object boxes. These contextual cues are shown to effective for object detection [Koller and Friedman, 2009; Zhang and Chen, 2012]. Each graphical 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 and contextual relations 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 [Zhang and Chen, 2012], GCRN learns these cues in a data-driven manner. Compared to the standard GCN models [Dai *et al.*, 2016; Kipf and Welling, 2017; Hamilton *et al.*, 2017], conG in GRCN explicitly captures the contextual dependencies among the objects while predicting node labels. Our experiments show explicitly capturing these contextual cues is crucial for detecting OOC objects.

Our main contributions include:

- We propose a graph contextual reasoning network to detect OOC objects by explicitly capturing contextual cues in a data-driven manner. GCRN does not require manual specification of the contextual relationships.
- We create a large-scale OOC dataset based on the COCO dataset [Lin *et al.*, 2014] where objects appear in various OOC scenarios based on common contextual relations: co-occurrence, location, and shape of the objects.

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 [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 [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 [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. 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.

Predictive Coding inspired Robust Learning. In recent work, predictive coding [Friston, 2018] - a theory of mind, has inspired an effective approach for robust learning [Jha *et al.*, 2020]. The central idea [Roy *et al.*, 2022] is to build a predictive context model and rather than use the output of deep learning models directly, the input is first validated and fused with the context model to detect whether the inputs

present a surprise to the model. This is an example of "analysis by synthesis" approaches [Yuille and Kersten, 2006; Jha *et al.*, 2019], where hypotheses are formulated in the form of candidate models and those whose predictions match the input data are preferred. The presented out of context detection approach in this paper can be viewed as an implementation of such a top-down predictive coding based approach, where the predictive context is learned as a graph contextual reasoning network.

3 Proposed Approach

We hypothesize that, for OOC objects, predictions using contextual cues may not be reliable due to the inconsistent contextual cues. On the other hand, contextual cues are expected to help predict the labels for in-context objects. 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 dependencies among the labels [Qu *et al.*, 2019]. 2) To complement RepG, we propose a context graph (ConG) that learns context dependencies at each node by sharing the contextual 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 implementations 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 $X = \{\boldsymbol{x}_i\}$ and $Y = \{y_i\}$ as the feature representation and label of ith node, respectively. Given this definition, the goal is to predict object labels for each node

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

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) at the final layer H^L.

Context-informed prediction. We formulate context-dependent 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 [Koller and Friedman, 2009]. Lets denote θ 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].

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

where $N(i)$ is the set of neighbors of i. Intuitively, with this approximation, we only consider the contextual 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 [Kipf and Welling, 2017; Koller and Friedman, 2009]. Note that in Equ. 4, the label of a node y_i is conditioned on both the neighboring context ($y_{j\in N(i)}$) and the feature representation X. To avoid this dependency, we consider an iterative optimization where we optimize for the representations in one phase and for the context dependency in another phase [Qu *et al.*, 2019].

Representation graph (RepG). In the first phase, we consider a mean-field approximation to remove context dependency and learn only the visual representation for nodes. The node-wise label predictions are made as

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

where θ_R is the parameters for representation graph. Our representation graph is implemented by a GCN where object features are used to initialize node representations and parameters are learned to predict node labels.

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

where X_R denote the initial node representations and W_R denotes the GCN parameters. We consider visual features from the bounding box as the node representations. Note that neighboring nodes share feature representations by message passing (6) but context dependencies are ignored due to the mean field approximation (5).

Context graph (ConG). In the second phase, we aim to make context-dependent predictions. Similar to RepG, ConG is also implemented using a GCN but instead of visual features, we consider context features as node representations

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

where X_C denotes the contextual features as node representations and W_C denotes the GCN parameters. Specifically, X_C includes softmax distribution over the labels from RepG along with spatial features such as position and size of the objects. The label distribution helps capturing co-occurrence cues and the relative spatial cues are captured from position and size features. ConG learns the context dependencies between objects while making the final predictions.

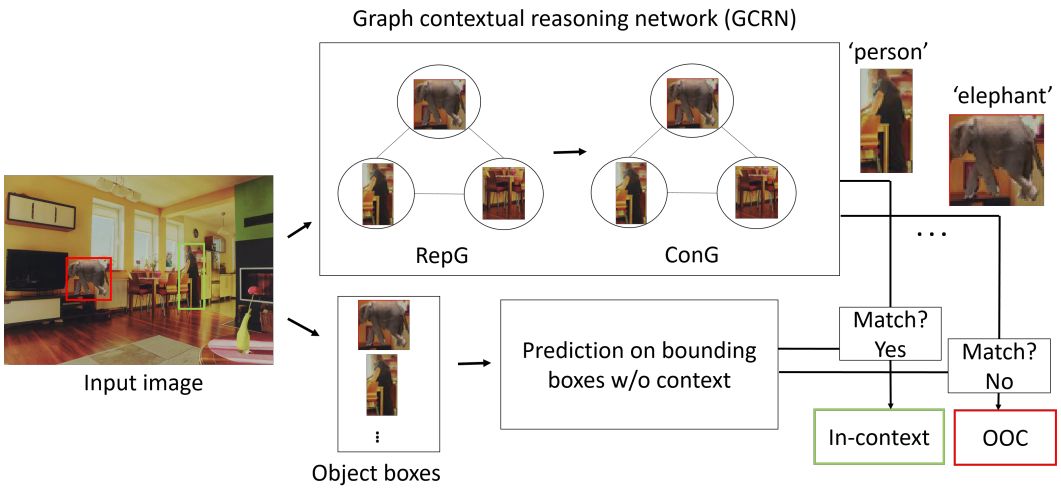

Figure 2: GCRN considers an image with bounding boxes as input to construct a graph over the objects. This graph is fed to RepG which learns visual representations of objects and ConG which learns contextual relations between objects. Finally, the GRCN predictions are compared with the object classifier's predictions to detect OOC objects. Two predictions are likely to match for in-context objects and differ for OOC objects.

Learning. We perform an iterative learning in an expectation-maximizing (EM) framework [Richardson and Domingos, 2006; Qu *et al.*, 2019]. At the E-step, we learn the parameters of RepG considering a fixed ConG and at the M-step, keeping RepG fixed, we update the parameters of ConG. Specifically, at the E-step, we make intermediate predictions using RepG. Then, considering these intermediate predictions and context dependencies between objects, ConG makes final predictions. The node-wise prediction loss is used to update RepG. At the M-step, we keep RepG fixed and update ConG based on node-wise prediction loss. We continue these iterations until convergence, i.e., the difference in the predictions between RepG and ConG is zero or below a threshold.

Detecting OOC instances. Given an image, we localize the objects by bounding boxes and consider them as the nodes of the graph. GCRN predicts softmax distribution over class labels for each node by considering context cues from other nodes. We also train an object classifier to predict object labels only from the bounding boxes ignoring the context cues. Finally, we compare the KL divergence (KLD) between the softmax distributions from both the predictions as a measure of OOC. For in-context objects, KLD is expected to be low and for OOC objects, the KLD is expected to be high. Thus, OOD detection can be performed by applying a threshold to the KLD. 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 a 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. GCNs 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 are crucial for efficient learning and convergence.

To create a graph for an image, we first detect objects in images and then extract features for the objects to initialize the graph representation. We consider the MaskRCNN [He *et al.*, 2017] pre-trained on the COCO dataset to detect objects. We train a ResNet50 [He *et al.*, 2016] network for feature extraction. For contextual cues, we use the geometrical features for each objects with co-ordinates (xmin, ymin, xmax,ymax) and calculate a 7D spatial feature vector $\left[\frac{w}{W}, \frac{h}{H}, \frac{a}{A}, \frac{\text{xmin}}{W}, \frac{\text{ymin}}{H}, \frac{\text{xmax}}{W}, \frac{\text{ymax}}{H}\right]$ 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 in images.

4 Experiments

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

The COCO-OOC Dataset. To evaluate the performance of OOC detection, we create a large-scale OOC dataset based on the COCO object detection benchmark [Lin *et al.*, 2014]. We aim to localize OOC objects and leverage contextual cues from other objects in an image. This requires bounding box annotations for all the objects along with the OOC object. We consider COCO 2014 [Lin *et al.*, 2014] dataset consisting of 80 indoor and outdoor objects classes. Following the common strategy [Blum *et al.*, 2021; Zhang *et al.*, 2020], we place objects in images that violate the contextual relations. We

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

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

leverage available object segmentation masks to transplant objects in images to create OOC scenarios. COCO-OOC consists of 106,036 images with three types of OOC violating co-occurrence, location and size relations [Blum *et al.*, 2021; Zhang *et al.*, 2020; Bomatter *et al.*, 2021].

The OCD Dataset. In addition to our COCO-OOC dataset, we also use the recent OCD benchmark with synthetically generated OOC indoor scenes [Bomatter *et al.*, 2021]. The OOC images are generated using the VirutalHome [Puig *et al.*, 2018] environment. OCD has 11,155 OOC images where objects violate co-occurrence, size, and gravity relations.

Experimental setup and metrics. To capture usual contextual relations, we train GCRN on the COCO train set where all objects appear in context. We test on OOC images to detect the OOC objects that appear in unusual contexts. Note that we assume all OOC objects are available in training and are not novel during the test. Thus, relying on appearance cues alone is not sufficient for detecting OOC objects. Recall that we compare the KL divergence between predictions from GCRN and object classifier to detect OOC objects. As selecting a threshold is often crucial for separating OOC from in-context objects, we consider the AUC score as the metric that is not sensitive to a specific threshold.

Baselines. We propose the following baselines to evaluate various aspects of 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]. Further, our results in table 1 show that the softmax confidence is not reliable to detect OOC objects as, unlike anomalous or novel objects, the OOC objects are observed during training. This study implies that relying on the appearance alone 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 consider only the representation graph. These 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 representations are learned through a shared network and context dependencies between the objects (e.g., label dependencies) are ignored.

Impact of contextual cues 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 for detecting OOC objects is slightly inferior without ConG. However, considering explicit contextual cues in GCRN improved the performance for in-context objects and degraded the performance for OOC objects. This justifies our hypothesis that accurate contextual cues are helpful for detecting in-contexts while inconsistent contextual cues can be harmful.

Approach	OOC	In-context	Overall
GCRN (w/o ConG)	0.69	0.77	0.76
GCRN	0.30	0.98	0.93

Table 2: Performance for detecting OOC and in-context objects on the COCO-OOC dataset.

OOC type	Co-occurrence	Size
AUC score	0.986	0.921

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

Types of OOC. We evaluate GCRN's performance on detecting OOC objects that violate co-occurrence and size relations in table 3. The performance on co-occurrence is superior as capturing the co-occurrence context is relatively more reliable. To capture the size context, we consider the relative size of the object boxes measured in pixel-space. However, the actual size information is partially lost due to the projection of 3D world to 2D images.

Impact of object detection on OOC detection. We need to accurately detect an object to determine whether it is OOC. In this section, we analyze the effect of object detection on the performance of OOC detection. We consider three setups: 1) oracle bounding boxes and labels (oracle boxes + labels) - here we have access to ground-truth bounding boxes and class labels. 2) oracle bounding boxes (oracle boxes + pred labels) - here we have access to ground-truth bounding boxes but labels are predicted by the object classifier, 3) predicted bounding boxes (pred boxes) - here we do not have access to ground-truth bounding boxes or labels and localize objects with an object detector. In the oracle boxes + labels setup, errors are completely attributed to the OOC detection. In oracle boxes + pred labels, additional errors may come from misclassifying object labels. In pred boxes, errors may come from both incorrect boxes and wrong labels. The results are shown in table 4. As expected, the performance gradually decreases from oracle boxes+labels to pred boxes. This study depicts the importance of accurate object detection for reliable OOC detection.

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

Table 4: Impact of object detector on OOC detection on the COCO-OOC dataset.

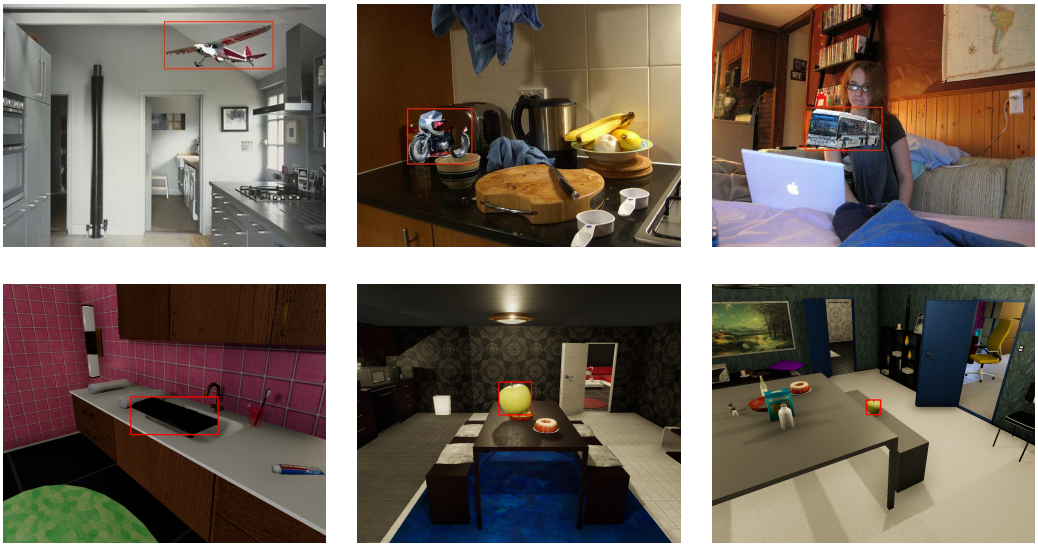

Figure 3: Qualitative examples where the proposed GCRN correctly identifies OOC objects in images. Top row presents images from our COCO-OOC dataset and bottom row presents images from the OCD dataset. The OOC objects are marked by red boxes. GCRN successfully detects OOC objects that violate co-occurrence, location, and size relations.

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

Table 5: Performance of OOC detection on the OCD dataset.

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 between COCO and OCD as OOC objects. 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 more challenging. 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 images and the bottom row shows the results on OCD images. Our GCRN successfully detects various types of OOC objects in both natural and synthetic images. Note that we consider the COCO dataset for training and can detect OOC objects in the synthetic OCD images.

Failure cases. We present two failure cases in figure 4. In these two cases, 'laptop' and 'backpack' are not detected as OOC objects. The size of the laptop is larger than usual making it OOC. Laptops are often observed in indoor scenes with persons and indoor furniture as in-context objects. This may cause the misdetection. In the second case, the size of the OOC backpack is larger than usual. However. backpacks are often observed in outdoor scenes and regarded as an in-context object. Generally, the size features are often partially lost in images due to the 3D to 2D projection. Having access to additional 3D cues of object geometry (e.g., from RGB+Depth) can help capture context more precisely.

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

5 Conclusion

We have presented a GCRN framework to detect OOC objects in images. Detecting OOC objects is crucial for developing a reliable object detection framework as detectors perform poorly on OOC objects. The proposed GCRN framework has two components: RepG to learn object representations and ConG to explicitly capture contextual relation for object detection. We have created a large-scale OOC dataset to evaluate our performance. We have also evaluated GCRN on the recent OCD dataset. We have considered common OOC scenarios where objects violate co-occurrence, location, and size relations. Our evaluation shows that contextual cues are helpful to detect in-context objects while inconsistent contextual cues can hinder accurate object detection. We have showed that explicitly capturing contextual cues is crucial for OOC detection. We have analyzed the effect of object detection on the performance of OOC detection and, as expected, accurately localizing objects and identifying object labels are shown to be important for OOC detection. Finally, we have presented a few failure cases and propose strategies to mitigate such failures.

Acknowledgements

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, 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 fishyscapes 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 Karev, 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.*, 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.

[Friston, 2018] Karl Friston. Does predictive coding have a future? *Nature neuroscience*, 21(8):1019–1021, 2018.

[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 *CVPR*, 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.

[Jha *et al.*, 2019] Susmit Jha, Sunny Raj, Steven Fernandes, Sumit K Jha, Somesh Jha, Brian Jalaian, Gunjan Verma, and Ananthram Swami. Attribution-based confidence metric for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[Jha *et al.*, 2020] Susmit Jha, John Rushby, and Natarajan Shankar. Model-centered assurance for autonomous systems. In *International Conference on Computer Safety, Reliability, and Security*, pages 228–243. Springer, 2020.

[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. Virtualhome: 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.

[Roy *et al.*, 2022] Anirban Roy, Adam Cobb, Nathaniel D. Bastian, Brian Jalaian, and Susmit Jha. Runtime monitoring of deep neural networks using top-down context models inspired by predictive processing and dual process theory. In *AAAI Spring Symposium Designing Artificial Intelligence for Open Worlds*, 2022.

[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.

[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. *arXiv preprint arXiv:1909.01315*, 2019.

[Yuille and Kersten, 2006] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006.

[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.