When Pigs Fly: Contextual Reasoning in Synthetic and Natural Scenes

Philipp Bomatter1,*, Mengmi Zhang2,3,*, Dimitar Karev4, Spandan Madan3,5, Claire Tseng4, and Gabriel Kreiman2,3

* Equal contribution
Address correspondence to gabriel.kreiman@tch.harvard.edu

1ETH Zürich
2Children’s Hospital, Harvard Medical School
3Center for Brains, Minds and Machines
4Harvard College, Harvard University
5School of Engineering and Applied Sciences, Harvard University

Abstract

Context is of fundamental importance to both human and machine vision—an object in the air is more likely to be an airplane, than a pig. The rich notion of context incorporates several aspects including physics rules, statistical co-occurrences, and relative object sizes, among others. While previous works have crowd-sourced out-of-context photographs from the web to study scene context, controlling the nature and extent of contextual violations has been an extremely daunting task. Here we introduce a diverse, synthetic Out-of-Context Dataset (OCD) with fine-grained control over scene context. By leveraging a 3D simulation engine, we systematically control the gravity, object co-occurrences and relative sizes across 36 object categories in a virtual household environment. We then conduct a series of experiments to gain insights into the impact of contextual cues on both human and machine vision using OCD. First, we conduct psycho-physics experiments to establish a human benchmark for out-of-context recognition, and then compare it with state-of-the-art computer vision models to quantify the gap between the two. Finally, we propose a context-aware recognition transformer model, fusing object and contextual information via multi-head attention. Our model captures useful information for contextual reasoning, enabling human-level performance and significantly better robustness in out-of-context conditions compared to baseline models across OCD and other existing out-of-context natural image datasets. All source code and data are publicly available https://github.com/kreimanlab/WhenPigsFlyContext.
1. Introduction

A coffee mug is usually a small object (Fig. 1a). It does not fly on its own (Fig. 1c), and can often be found on a table (Fig. 1a), and not on a chair (Fig. 1d). Such contextual cues have a pronounced impact on the object recognition capabilities of both humans [41], and computer vision models [36, 8]. Neural networks learn co-occurrence statistics between an object’s appearance and its label, but also between the object’s context and its label [12, 32, 2]. Therefore, it is not surprising that recognition models fail to recognize objects in unfamiliar contexts [30].

Despite the fundamental role of context in visual recognition, it remains unclear what and how contextual cues should be integrated with object information. Large-scale, internet-scraped datasets like ImageNet [10] are highly uncontrolled, which makes it hard to quantify how context affects recognition. To address this challenge, here we present a methodology to systematically study the effects of an object’s context on recognition by leveraging a Unity-based 3D simulation engine for image generation [20], and manipulating 3D objects in a virtual home environment [28]. The ability to rigorously control every aspect of the scene enables us to systematically violate contextual rules and assess their impact on recognition.

We focus on three fundamental aspects of context: (1) gravity - objects without physical support, (2) object co-occurrences, and (3) relative size - changes to the size of target objects relative to the background. In contrast to existing out-of-context real-world photographs, our approach provides fine-grained control to alter one aspect of context at a time, and the flexibility to modify various aspects of context including 3D geometric transformations, locations, viewpoints, and materials. We use these generated images to gain insights into how contextual cues impact object recognition in humans and state-of-the-art computer vision models.

Our contributions in this paper are three-fold. Firstly, we introduce a challenging new dataset for in- and out-of-context object recognition that allows fine-grained control over context violations including gravity, object co-occurrences and relative object sizes (out-of-context dataset, OCD). Secondly, we conduct psycho-physics experiments to establish a human benchmark for out-of-context recognition and compare it with state-of-the-art computer vision models, thus quantifying the gap between human and computer vision. Finally, we propose a new context-aware architecture for object recognition which can incorporate object and contextual information to reason in context and also generalize well to out-of-context images. Our Context-aware Recognition Transformer Network (CRTNet) uses two separate streams to process the object and its context independently before integrating them via multi-head attention in transformer decoders. The model then makes a classification decision by weighting these two streams based on a predicted confidence score. Across multiple datasets, we demonstrate that the CRTNet model surpasses other state-of-the-art computational models and classifies objects more robustly despite large contextual variations, much like humans do.

2. Related Works

Out-of-context object recognition: Context is of incredible importance to machine vision models for object recognition [26, 23]. Deep networks trained on natural image datasets like e.g. ImageNet [21] rely implicitly but strongly on context [15, 4, 31]. Indeed, these algorithms often fail when objects are placed in an incongruent context. Most work in the literature have represented context as a monolithic property in the form of the target object’s background. This includes testing the generalization to new backgrounds [2], incongruent backgrounds [41], exploring impact of foreground-background relationships on data augmentation [14], and replacing image sub-regions by another sub-image i.e. object transplanting [30]. To the best of our knowledge, there is no existing work exploring aspects of object context (e.g. gravity) in a quantitatively controlled, systematic manner as done in this paper.

3D simulation engines and computer vision: Recent works have demonstrated the success of using 3D virtual environments for tasks such as object recognition with simple and uniform backgrounds [3], routine program synthesis [28], 3D animal pose estimation [25], and for studying the generalization capabilities of CNNs [24, 17]. However, to the best of our knowledge, none of these studies have tackled the challenging problem of contextual modulation. Compared with other works on contextual effects on a set of limited number of natural images [27, 30], 3D simulation engines allow us to easily synthesize as many pictures as possible and violate contextual rules, which is impractical to achieve with real-world photographs. Moreover, these simulation engines also enable us to control contextual parameters precisely such that we can vary one at a time in a systematic and quantifiable manner.

Models for context-aware object recognition: To tackle the problem of context-aware object recognition, researchers have proposed classical approaches, e.g. Conditional Random Field (CRF) [16, 40, 22, 7], and graph-based methods [34, 39, 35, 8]. Recent works have extended this line of work to deep graph neural networks [18, 9, 11, 1]. Breaking away from these previous works where graph optimization is performed globally for contextual reasoning in object recognition, our model has a two-stream architecture which separately processes visual information on both target objects and context, and then integrates them with multi-head attention in stacks of transformer decoding layers. Contrasting other vision
We demonstrate the advantage of non-shared parameters in not enforcing sharing between the parameters $\theta$ features in $I$ the DenseNet architecture [19] with weights pre-trained on $\theta$ are parameterized by class label probabilities $y$ object and context information via hierarchical reasoning. The two streams are encoded through two separate 2D-CNNs. After the encoding stage, CRTNet tokenizes the feature maps of $t$ and $c$ via average pooling:

$$T_{i} = \frac{1}{L} \sum_{i=1,\ldots,L} a_{i}^{t} \quad (1)$$

To encode the spatial relations between the target token and the context tokens, as well as between different context tokens, we learn a positional embedding of size $D$ for each location $i$ and add it to the corresponding context token $a_{i}^{c}$. For the target token $T_{i}$, we use the positional embedding corresponding to the location, within which the bounding box midpoint is contained. The positionally-encoded context and target tokens are denoted by $z_{c}$ and $z_{t}$ respectively.

### 3.3. Tokenization and Positional Encoding

We tokenize the context feature map $a_{c}$ by splitting it into patches based on locations, following [13]. Each context token corresponds to a feature vector $a_{c}^{i}$ of dimension $D$ at location $i$ where $i \in \{1, \ldots, L = H \times W\}$. To compute target token $T_{t}$, CRTNet aggregates the target feature map $a_{t}$ via average pooling:

$$T_{t} = \frac{1}{L} \sum_{i=1,\ldots,L} a_{i}^{t} \quad (1)$$

A model that always relies on context can make mistakes under unusual context. To increase robustness, CRTNet makes a second prediction $y_{t}$, based on target object information alone, estimates the confidence $p$ of this prediction, and computes a confidence-weighted average of $y_{t}$ and $y_{c}$ to get the final prediction $y_{p}$. If the model makes a confident prediction with the object only, it overrules the context reasoning stage.

### 3.2. Convolutional Feature Extraction

CRTNet takes $I_{c}$ and $I_{t}$ as inputs and uses two 2D-CNNs, $E_{c}(\cdot)$ and $E_{t}(\cdot)$, to extract context and target feature maps $a_{c}$ and $a_{t}$, respectively, where $E_{c}(\cdot)$ and $E_{t}(\cdot)$ are parameterized by $\theta_{E_{c}}$ and $\theta_{E_{t}}$. Concretely, we use the DenseNet architecture [19] with weights pre-trained on ImageNet [10] and fine-tune it. Assuming that different features in $I_{c}$ and $I_{t}$ are useful for recognition, we do not enforce sharing between the parameters $\theta_{E_{c}}$ and $\theta_{E_{t}}$. We demonstrate the advantage of non-shared parameters in the ablation study (Sec. 5.5). To allow CRTNet to focus on specific parts of the image and select features at those locations, we preserve the spatial organization of features and define $a_{c}$ and $a_{t}$ as the output feature maps from the last convolution layer of DenseNet. Both $a_{c}$ and $a_{t}$ are of size $D \times W \times H = 1,664 \times 7 \times 7$, where $D$, $W$ and $H$ denote the number of channels, width and height of the feature maps respectively.

### 3.1. Overview

We propose the Context-aware Recognition Transformer Network (CRTNet, Figure 2) and introduce three novel designs in CRTNet: First, CRTNet involves transformer decoder modules for integrating object and contextual information to reason about context via stacks of multi-headed encoder-decoder attention. Second, we introduce a confidence-weighting mechanism that improves the model’s robustness and gives it the flexibility to select what information to rely on for recognition. Third, we introduce a confidence-weighting mechanism that improves the model’s robustness and gives it the flexibility to select what information to rely on for recognition. Third, we introduce a confidence-weighting mechanism that improves the model’s robustness and gives it the flexibility to select what information to rely on for recognition.

### 3.4. Transformer Decoder

We follow the original transformer decoder [38], taking $z_{c}$ to compute keys and values, and $z_{t}$ to generate the queries in the transformer encoder-decoder multi-head attention layer. Since we only have a single target token, we omit the self-attention layer. In the experiments, we also tested CRTNet with self-attention enabled and we did not observe performance improvements. Our decoder layer consists of alternating layers of encoder-decoder attention (EDA) and multi-layer perceptron (MLP) blocks. Layernorm (LN) is applied after each residual connection. Dropout (DRO) is applied within each residual connection and MLP block. The MLP contains two layers with a ReLU non-linearity and DROP.

$$z_{t,c} = LN(DROP(EDA(z_{t}, z_{c})) + z_{t}) \quad (2)$$

$$z'_{t,c} = LN(DROP(MLP(z_{t,c})) + z_{t,c}) \quad (3)$$

Our transformer decoder has a stack of $X = 6$ layers, indexed by $x$. We repeat the operations in Eqs 2 and 3 for each transformer decoder layer by recursively assigning $z'_{t,c}$ back to $z_{t}$ as input to the next transformer decoding layer. Each EDA layer integrates useful information between the context and the target object with 8-headed selective attention. Based on accumulated information from all the
Figure 2: Architecture overview of the Context-aware Recognition Transformer Network (CRTNet). The diagram depicts the modular steps carried out by CRTNet in the context-aware object recognition task. CRTNet consists of 3 main modules: feature extraction, integration of context and target information, and confidence-modulated classification. CRTNet takes the cropped target object \( I_t \) and the entire context image \( I_c \) as inputs and extracts their respective features. These feature maps are then tokenized and the information of the two streams is integrated over multiple transformer decoding layers. CRTNet also estimates a confidence score of recognizing the target object based on object features alone, which is used to modulate the contributions of \( y_t \) and \( y_{t,c} \) to the final prediction \( y_p \). The dashed lines in backward direction denote gradient flows during backpropagation. The two black crosses denote where the gradient updates stop. See Sec. 3 for details.

3.5. Confidence-modulated Recognition

The context classifier \( G_z(\cdot) \) with parameters \( \theta_{G_z} \) consists of a fully-connected layer and a softmax layer. It takes the feature embedding \( z'_{t,c} \) from the last transformer decoder layer and outputs the predicted class distribution vector: \( y_{t,c} = G_z(z'_{t,c}) \). Similarly, the target classifier \( G_t(\cdot) \), takes the feature maps \( a_t \) as input and outputs the predicted class distribution vector: \( y_t = G_t(a_t) \).

Since neural networks are often fooled by incongruent context \([41]\), we propose a confidence-modulated recognition mechanism balancing the predictions from \( G_t(\cdot) \) and \( G_z(\cdot) \). The confidence estimator \( U(\cdot) \) with parameters \( \theta_U \) takes the target feature map \( a_t \) as input and outputs a value \( p \) indicating how confident CRTNet is about the prediction \( y_t \). \( U(\cdot) \) is a feed-forward multi-layer perceptron network with a sigmoid function to normalize the confidence score to \([0, 1]\).

\[
p = \frac{1}{1 + e^{-U(a_t)}} \quad (4)
\]

We then use \( p \) to compute a confidence-weighted average of \( y_{t,c} \) and \( y_t \) for the final predicted class distribution \( y_p \): \( y_p = py_t + (1 - p)y_{t,c} \). The higher the confidence \( p \), the more CRTNet relies on the target object itself rather than the integrated contextual information for classification. We demonstrate the advantage of using \( y_p \) rather than \( y_{t,c} \) or \( y_t \) as a final prediction in the ablation study (Sec. 5.5).

3.6. Training

CRTNet is trained end-to-end with three cross-entropy losses introduced below: (i) to train the confidence estimator \( U(\cdot) \), we use a cross-entropy loss with respect to the confidence-weighted prediction \( y_p \). Intuitively, this allows \( U(\cdot) \) to learn to increase the confidence value \( p \) for samples where the prediction \( y_t \), based on target object information alone, tends to be correct. (ii) To train \( G_t(\cdot) \), we use a cross-entropy loss with respect to \( y_t \). (iii) For the rest of components in CRTNet including the transformer decoder and classifier \( G_z(\cdot) \), we use a cross-entropy loss with respect to \( y_{t,c} \). Instead of training everything based on \( y_p \), the three cross-entropy losses altogether maintain strong learning signals for all parts in the architecture irrespective of the confidence value \( p \).

To facilitate learning for specific components in CRTNet, we also introduced gradient detachments during backpropogations (Fig. 2). Gradients flowing through both \( U(\cdot) \) and \( G_t(\cdot) \) are detached from \( E_t(\cdot) \) to prevent them from driving the target encoder to learn more discriminative features, which could impact the efficacy of the transformer modules and \( G_z(\cdot) \). We demonstrate the benefit of these design decisions in ablation studies (Sec. 5.5).
4. Experimental Details

4.1. Baselines

We compared CRTNet against several baselines:

- CATNet [41] is a context-aware two-stream object recognition model. It processes the visual features of cropped target object and context in parallel, dynamically incorporates object and contextual information by constantly updating its attention over image locations, and sequentially reasons about the class label for the target object via a recurrent neural network.

- Faster R-CNN [29] is an object detection algorithm. We adapted it to the context-aware object recognition task by replacing the region proposal network with the ground truth bounding box indicating the location of the target object.

- DenseNet [19] is a 2D-CNN with dense connections that takes the cropped target object patch $I_t$ as input.

4.2. Datasets

4.2.1 In- and Out-of-context Dataset (OCD)

Our out-of-context dataset (OCD) contains 36 object classes, with 15,773 testing images in 6 contextual conditions. We leveraged the VirtualHome environment [28] developed in the Unity simulation engine to synthesize these images in indoor home environments within 7 apartments and 5 rooms per apartment. These rooms include furnished bedrooms, kitchens, study rooms, living rooms and bathrooms [28] (see Fig. 1 for examples). We extended VirtualHome with additional functionalities to manipulate object properties, such as materials and scales, and to place objects in out-of-context locations. The target object is always centered in the camera view. Collision checking and camera ray casting are enabled to prevent object collisions and occlusions.

To our best knowledge, our OCD is the first large dataset tackling the problem of contextual modulation in object recognition. Different from large-scale image classification datasets, such as ImageNet [10] where one single large object is typically centered on an image, our OCD involves highly complex and rich scenes with multiple objects in an image. Our OCD enables us to study context-aware recognition on both humans and models in a systematic and quantifiable manner. The far-from-perfect results (Sec. 5) demonstrate that our challenging OCD provides great venues to improve our current recognition models.

- Normal Context and No Context: There are 2,309 images for normal context (Fig. 1b), and 2,309 images for no-context condition (Fig. 1g). For the normal context condition, each target object is placed in its “typical” location, defined by the default settings of VirtualHome. Then we generate a corresponding no context image for every normal context image by replacing all the pixels surrounding the target object with either uniform grey pixels or salt and pepper noise.

- Gravity: We generated 2,934 images where we move the target object along the vertical direction such that it is no longer supported (Fig. 1c). To avoid cases where objects are lifted so high that their surroundings change completely, we set the lifting offset to 0.25 meters.

- Object Co-occurrences: To examine the importance of the statistics of object co-occurrences, four human subjects were asked to indicate the most likely room and location for the target objects. We use the output of these responses to generate 1,453 images where we place the target objects on surfaces with lower co-occurrence probability, e.g. a microwave in the bathroom and Fig. 1d.

- Object Co-occurrences + Gravity: We generated 910 images where the objects are both lifted and placed in unlikely locations. We chose walls, windows, and doorways of rooms where the target object is typically absent (Fig. 1e). We place target objects at half of the apartment’s height.

- Size: We created 5,858 images where we change the target object size to 2, 3, or 4 times its original size while keeping the remaining objects in the scene intact (Fig. 1f).

4.2.2 Real-world Out-of-context Datasets

The Cut-and-paste Dataset [41] contains 2,259 out-of-context images spanning 55 object classes. These images are grouped into 16 conditions obtained through the combinations of 4 object sizes and 4 context conditions (normal, minimal, congruent, and incongruent) (Fig. 3b).

The UnRel [27] Dataset contains more than 1,000 images with unusual relations among objects spanning 100 object classes. The dataset was collected from the web based on triplet queries, such as “dog rides bike” (Fig. 3c).

4.3. Performance Evaluation

Evaluation of Computational Models: We trained the models on natural images from COCO-Stuff [5] using the annotations for object classes overlapping with those in the respective test set (16 overlapping classes between VirtualHome and COCO-Stuff, 55 overlapping classes between Cut-and-paste and COCO-Stuff and 33 overlapping classes between UnRel and COCO-Stuff). These models were then tested on all six VirtualHome conditions, the Cut-and-paste Dataset, UnRel, and on a COCO-Stuff test split.

Behavioral Experiments: As a benchmark, we evaluated human recognition in the 6 contextual conditions described above (Sec. 4.2.1, Fig. 1), as schematically illustrated in Fig. 3d, on Amazon Mechanical Turk (MTurk) [37]. We recruited 400 subjects per experiment, yielding $\approx 67,000$ trials. To avoid biases and potential memory
Figure 3: **Three out-of-context datasets and human benchmark experiment.** (a-c) Two example images from each out-of-context dataset. The red box shows the target location. (a), two contextual modifications (gravity and size) are shown and the target object is always in the center. (b) Cut-and-paste dataset created from [41]. The same target object is cut and pasted in either incongruent or congruent conditions. The target object location is always the same across both conditions. (c). UnRel dataset [27]. There are no controlled conditions for comparison as (a) and (b). See Sec. 4 for further details. (d) Subjects were presented with a fixation cross (500 ms), followed by a bounding box indicating the target object location (1000 ms). The image was shown for 200 ms. After image offset, subjects typed one word to identify the target object. The correct answer (here, "mug") is not shown in the actual experiment).

Computer vision and most psychophysics experiments enforce N-way categorization (e.g., [33]). Here we used a more unbiased probing mechanism whereby subjects could use any word to describe the target object. We independently collected ground truth answers for each object in a separate MTurk experiment with infinite viewing time and normal context conditions. These Mturk subjects did not participate in the main experiments. Answers in the main experiments were correct if they matched any of the ground truth responses [41]. Although computational models are evaluated using N-way categorization, we find it instructive to report model results alongside human behavior for comparison purposes. We also show human-model correlations to describe their relative trends across all conditions.

5. Results

5.1. Recognition in our OCD dataset

Figure 4 (left) reports recognition accuracy for humans over the 6 context conditions (Sec. 4.2.1, Fig. 1) and 2 target object sizes (total of 12 conditions). Comparing the no-context condition (white) versus normal context (black), it is evident that contextual cues lead to improvement in recognition, especially for smaller objects, consistent with previous work [41].

Gravity violations led to a reduction in accuracy. For small object, the gravity condition was even slightly worse than the no context condition; the unusual context can be misleading for humans. The effects were similar for the changes in object co-occurrences and relative object size. Objects were enlarged by a factor of 2, 3, or 4 in the relative size condition. Since the target object gets larger, and because of the improvement in recognition with object size, we would expect a higher accuracy in the size condition compared to normal context. However, increasing the size of the target object while keeping all other objects intact, violates the basic statistics of expected relative sizes (e.g., we expect a chair to be larger than an apple). Thus, the drop in performance in the size condition is particularly remarkable and shows that violation of contextual cues can even override basic object recognition cues.

Combining changes in gravity and in the statistics of object co-occurrences led to a pronounced drop in accuracy. Especially for the small target objects, violation of gravity and statistical co-occurrences led to performance well below that in the no context condition.

These results show that context can play a facilitatory role (compare normal versus no context), but context can also impair performance (compare gravity+co-occurrence versus no context). In other words, unorthodox contextual information hurts recognition.

Figure 4 (right) reports accuracy for CRTNet. Adding normal contextual information led to an improvement of 4% (normal context vs no context) in performance for both small and large target objects. However, in contrast with humans (20% improvement in small objects versus 10% improvement in large objects), we did not observe the difference between the effect of contextual modulation for small versus large target objects for CRTNet. Remarkably, the CRTNet model qualitatively captured the effects of
contextual violations. Overall, the performance of the model was below humans, particularly for small objects. However, the basic trends associated with the role of contextual information in humans can also be appreciated in the CRTNet results. Gravity, object co-occurrences, and relative object size changes led to a decrease in performance. As in the behavioral measurements, these effects were more pronounced for the small objects. For CRTNet, all conditions led to worse performance than the no context condition for small objects.

5.2. Recognition in Cut-and-paste

Synthetic images offer the possibility to systematically control every aspect of the scene, but these images still do not follow all the statistics of the natural world. Therefore, we evaluated whether CRTNet can generalize to naturalistic settings on the Cut-and-paste dataset [41]. For comparison, we reproduced the the human psychophysics results in Table 2. The CRTNet model yielded results that were consistent, and in many conditions better than, human performance. As observed in the human data, performance increases with object size. In addition, the effect of context was more pronounced for smaller objects (compare normal context (NC) versus minimal context (MC) conditions).

Consistent with previous work [41], compared to the minimal context condition, congruent contextual information (CG) typically enhanced recognition whereas incongruent context (IC) impaired performance. Although the congruent context typically shares similar correlations between objects and scene properties, pasting the object in a congruent context led to weaker enhancement than the normal context. This lower contextual facilitation may be due to erroneous relative sizes between objects, unnatural boundaries created by pasting, or contextual cues specific to each image. CRTNet was relatively oblivious to these effects and performance in the congruent condition was closer to that in the normal context condition.

5.3. Recognition in Natural Images

The Cut-and-paste dataset introduces artifacts (such as unnatural boundaries and erroneous relative sizes) due to the cut-and-paste process. Therefore, we next evaluated CRTNet on the UnRel dataset [27]. We use the performance on the COCO-Stuff [5] test split as reference for normal context in natural images. CRTNet showed a slightly lower recognition accuracy in the out-of-context setting (Fig. 5).

5.4. Comparison with baseline models

Performance Evaluation: Although Faster R-CNN and CATNet leverage global context information, CRTNet outperformed both models, especially on small objects (OCD: Table 1 and Supp. Fig. S7-S8; Cut-and-Paste: Table 2; UnRel: Fig. 5). CRTNet led CATNet by 7% and
Faster R-CNN by 20% for normal context and [0.5,1] dva size in the Cut-and-paste dataset (Table 2; similar results for other experiments in Supp. Fig. S7-S8). CRTNet is also more similar to human performance by a large margin in OCD and the Cut-and-Paste datasets (Table I and 2).

Architectural Differences: While all baseline models can rely on an intrinsic notion of spatial relations, CRTNet learns about spatial relations between target and context tokens through a positional embedding. A visualization of the learned positional embeddings (Supp. Fig. S1) shows that CRTNet learns image topology by encoding distance within the image in the similarity of position embeddings.

In CATNet, the attention map iteratively modulates the extracted feature maps from the context image at each time step in a recurrent neural network, whereas CRTNet uses a stack of feedforward transformer decoding layers with multi-headed encoder-decoder attention. These decoding layers hierarchically integrate information via attention maps, modulating the target token features with context information. Transformer architectures also tend to perform better than recurrent neural networks in NLP [38] and computer vision tasks [13, 6].

DenseNet takes cropped targets as input with few surrounding pixels of context and outputs predicted labels. Its performance dramatically decreases for smaller objects, resulting in lower correlation with humans. For example, in the Cut-and-paste dataset, CRTNet outperforms DenseNet by 30% for normal context and small objects (Table 2) and in OCD, DenseNet shows a correlation of 0.66 with human performance (vs. 0.89 for CRTNet, Table 1).

5.5. Ablation Reveals Critical Model Components

We assessed the importance of design choices by training and testing ablated versions of CRTNet on OCD dataset.

Shared Encoder: In our CRTNet model, we trained two separate encoders, \(E_t(\cdot)\) and \(E_c(\cdot)\), to extract features from target objects and the context respectively. Here, we enforced weight-sharing between these two encoders (Ablated-SharedEncoder) to assess whether the features relevant for target object recognition are different from those used in contextual reasoning. The ablation results (Table 1, Supp. Fig. S3) show that CRTNet with a shared encoder achieved a lower recognition accuracy and lower correlation with the psychophysics results.

Recognition Based on Target or Context Alone: During testing, we use the confidence-weighted prediction \(y_p\) as the final prediction. Here, we considered two extreme cases: CRTNet relying only on the target object itself (\(y_{t, \text{Ablated-TargetOnly}}\)) and CRTNet relying only on contextual reasoning (\(y_{t,b, \text{Ablated-Unweighted}}\)). The original model outperforms either of these ablated scenarios (Table 1 and Supp. Figs S4 and S5). This suggests that a confidence-modulated classification mechanism is essential for recognition models to be adaptive and robust given contextual variations.

Joint Training of the Target Encoder: In Sec. 3.6, we make the training of the target encoder \(E_t(\cdot)\) independent of \(G_t(\cdot)\) such that it can not force the target encoder to learn more discriminative features. In this ablated model, we remove this detaching constraint and train the target encoder \(E_t(\cdot)\) jointly (Ablated-NoDetachment, Table 1 and Supp. Fig. S6). The results are inferior to the ones of our original CRTNet, implying that detaching of the target encoder is helpful for recognition.

6. Discussion

We quantitatively studied the role of context in visual recognition in humans and computational models. We introduce a dataset (OCD) consisting of 15,773 images to systematically study out-of-context objects in simulated indoor home environments. We investigated the role of gravity, object co-occurrences, and relative object sizes in object recognition. As an essential benchmark, we tested both humans and models on this dataset. Since these synthetic images can still be easily distinguished from real photos, the domain gap might influence the recognition performance. To further test the generalization of humans.

Table 2: Recognition accuracy of humans, our model (CRTNet), and baselines on the Cut-and-paste Dataset [41]. There are 4 conditions for each size: normal context (NC), congruent context (GC), incongruent context (IC) and minimal context (MC) (Sec. 4.2.2). Bold highlights the best performance. Numbers in brackets denote standard error of the mean.
and models, we consider out-of-context variations in two datasets consisting of real photographs. We show consistent results over all three datasets that contextual cues can enhance visual recognition, but the “wrong” context can also impair visual recognition, both for humans and models.

We proposed a context-aware recognition transformer model that integrates contextual and object cues via multi-head transformer decoding layers. To increase robustness in visual recognition, we introduced a confidence-modulated recognition system which learns to estimate its own confidence. Across a wide range of out-of-context datasets from our synthetic OCD dataset to real-world images, our model demonstrates superior performance over competitive baselines without retraining for each contextual condition, and exhibits human-like behavioral patterns over contextual variations. Despite great model performance, we also noted that there are still significant gaps between models and humans, particularly when recognizing small out-of-context objects.

References


