Object-Proposal Evaluation Protocol is ‘Gameable’

Neelima Chavali  Harsh Agrawal  Aroma Mahendru  Dhruv Batra
Virginia Tech
{gneelima, harsh92, aroma.mahendru, dbatra}@vt.edu

Abstract

Object proposals have quickly become the de-facto pre-processing step in a number of vision pipelines. The standard evaluation protocol for object proposal methods involves measuring ‘recall’ of annotated instances by the proposals in an object detection dataset such as PASCAL VOC [1]. In this paper, we demonstrate that this evaluation protocol is biased. By evaluating only on a specific set of categories in a partially annotated dataset, we fail to capture the performance of the proposal algorithm on all the remaining object categories that are present in the test set, but not annotated in the ground truth. More importantly, this makes the evaluation protocol ‘gameable’ or susceptible to manipulation (both intentional and unintentional).

To alleviate this problem, we perform an exhaustive evaluation of object proposal methods on multiple densely annotated datasets including a densely annotated version of PASCAL VOC which we introduce in this paper. We also release an easy-to-use toolbox which combines various publicly available implementations of object proposal algorithms which standardizes the proposal generation and evaluation so that new methods can be added and evaluated on different datasets. We hope that the results presented in the paper will stimulate discussion in the community about the choice of the protocol used to evaluate object proposal algorithms and motivate the use of densely annotated datasets or diagnostic tools to truly evaluate the category independence of object proposal methods.

1. Introduction

In the last few years, the Computer Vision community has witnessed the emergence of a new class of techniques called Object Proposal algorithms [2–12].

Object proposals are a set of candidate regions or bounding boxes in an image that may potentially contain an object.

Object proposal algorithms have quickly become the de-facto pre-processing step in a number of vision pipelines – object detection [13–17], segmentation [18–20], object discovery [21], weakly supervised learning of object-object interactions [22, 23], content aware media re-targeting [24], and action recognition in still images [25]. Of all these tasks, object proposals have been particularly successful in object detection systems. For example, all top-performing entries [14, 26–28] in the ImageNet Detection Challenge 2014 [29] used object proposals. They are preferred over the previously-used sliding window paradigm because of their computational efficiency. Objects present in an image may vary in location, size, and aspect ratio, and performing an exhaustive search over such a high dimensional space is difficult. By using object proposals, computational effort can be focused on a small number of candidate windows.

Taking a holistic view of the literature on object proposals, it becomes clear that there are actually two distinct interpretations of the term ‘object proposals’ (although no past work seems to have explicitly made this distinction):

- **Category-independent object proposals**: where the goal is to identify all the objects in the image irrespective of their category.
- **Detection proposals**: where the goal is to improve the object detection pipeline, focusing on a chosen set of object classes (e.g. 20 PASCAL categories).

Notice that the former definition has an emphasis on object discovery. The latter definition has an emphasis on the ultimate performance of a detection pipeline.

Surprisingly, despite the two different interpretations and goals of the term ‘object proposals,’ there exists only a single evaluation protocol (with a few different evaluation metrics). Following is the widely adopted evaluation protocol:

1. Generate proposals on a dataset: The most commonly used dataset for evaluation today is the PASCAL VOC [1] detection test sets. Note that this is a partially annotated dataset where only the 20 PASCAL category instances are annotated.
2. Measure the performance of the generated proposals: typically in terms of ‘recall’ of the annotated instances. Commonly used metrics are described in Section 3.

The central thesis of this paper is that the current evaluation protocol for object proposal methods is suitable only for detection proposals and is a biased protocol for category-independent object proposals. By evaluating only on a specific set of object categories, we fail to capture the perfor-
Figure 1: (a) shows PASCAL annotations natively present in the dataset in green. Other objects that are not annotated but present in the image are shown in red; (b) shows Method 1 and (c) shows Method 2. Method 1 visually seems to recall more categories such as plates, glasses that Method 2 missed. Despite that, the computed recall for Method 2 is higher because it recalled all instances of PASCAL categories that were present in the ground truth. Note that the number of proposals generated by both methods is equal in this figure.

Figure 2: (a) shows PASCAL annotations natively present in the dataset in green. Other objects that are not annotated but present in the image are shown in red; (b) shows Method 1 and (c) shows Method 2. Method 1 visually seems to recall more categories such as plates, glasses that Method 2 missed. Clearly the recall for Method 1 should be higher. However, the calculated recall for Method 2 is significantly higher, which is counter-intuitive. This is because Method 2 recalls more PASCAL category objects.

Figs. 1, 2 illustrate this idea on images from PASCAL VOC 2010. Column (a) shows the ground-truth object annotations (in green, the annotations natively present in the dataset for the 20 PASCAL categories; in red, the annotations that we added to the dataset by marking objects originally annotated ‘background’). We can see that dataset contains annotations for PASCAL categories (chairs, tables, bottles, etc.) but does not contain annotations for other objects present in the image like picture frames, plates, glasses etc. Columns (b) and (c) show the outputs of two object proposal systems (top row shows the case when both methods produce the same number of proposals; bottom row shows unequal number of proposals). We can see that proposal method in Column (b) seems to be more “complete”, in the sense that it covers or discovers a large number of instances. For instance, in the top row it detects a number of non-PASCAL categories but misses out on finding the table. In both examples, the method in Column (c) achieves a higher accuracy metric, even in the bottom row, when it recalls strictly fewer objects, not just different ones. The reason is that Column (c) recalls/disCOVERS instances of the 20 PASCAL categories, which are the only ones annotated in the dataset.

While intuitive (and somewhat obvious) in hindsight, we believe this is an important finding because it makes the current protocol gameable or susceptible to manipulation (both intentional and unintentional). Our thorough evaluation of all popular object proposal techniques suggests that no current technique seems to have ‘gamed’ or exploited the bias in the protocol. However, caution must be exercised while evaluating these and future algorithms, lest we over-fit as a community to a specific set of object classes and unknowingly lose the category independence of object proposals. To summarize, the contributions of this paper are:

• We make an explicit distinction between the two mutually co-existing but different interpretations of object proposals.
• We report the bias and ‘gameability’ of current object proposal evaluation protocols.
• We demonstrate this gameability via a simple thought experiment where we propose a ‘fraudulent’ object proposal method that outperforms all existing object proposal techniques on current metrics.
• We conduct a thorough evaluation of existing object proposal methods on three densely annotated datasets.
• We propose two ways of improving the current evaluation protocol to truly reward the category-independence of object proposals:
  1. fully annotated datasets, and
  2. evaluation metrics that look at per class performance and bias capacity of proposal generators.
For the former, we introduce a densely-annotated version of PASCAL VOC 2010 where we annotated all instances of large set of object categories occurring in all images.
• We plan to release all code and data for experiments, and an object proposals library which allows for each comparison of all popular object proposal techniques.

2. Related Work

Types of Object Proposals: Object proposals can be broadly categorized into two categories:

• Window scoring: In these methods, the space of all possible windows in an image is sampled to get a subset of the windows (e.g., via sliding window). These windows are then scored for the presence of an object based on the image features from the windows. The algorithms that fall under this category are [2,5,6,11].
• Segment based: These algorithms involve over-segmenting an image and merging the segments using some strategy. These methods include [3,4,7–10,12]. While the output of these algorithms are segments in images, they are usually converted into bounding boxes before being fed to the detection pipelines.

A table in the supplement provides a summary of the different proposal methods and the evaluation protocol used.

Beyond RGB proposals: Beyond the ones listed above, a wide variety of algorithms fall under the umbrella of ‘object proposals’. For instance, [30,31] used spatio-temporal object proposals for recognition and tracking in videos. Another direction of work [27,32,33] explored using RGB-D cuboid proposals in an object detection and semantic segmentation in RGB-D images. While the scope of this paper is limited to bounding box proposals in RGB images, the central thesis of the paper (i.e., gamability of the evaluation protocol) is broadly applicable to other settings.

Evaluating Proposals: While a variety of approaches have been proposed for generating object proposals, there has been relatively limited analysis and evaluation of these approaches or evaluation protocols. Hosang et al. [34] focus on evaluation of object proposal algorithms, in particular the stability of such algorithms on parameter changes and image perturbations. Their work shows that existing proposal algorithms generalize well to non-PASCAL categories, for instance in the ImageNet 200 category detection dataset [29]. They also introduced a new evaluation metric (Average Recall). Their argument for a new metric is the need for a better localization between generated proposals and ground truth. While this is a valid and significant concern, it is orthogonal to the ‘gameability’ and bias in evaluation protocol, which to the best of our knowledge has not been previously addressed.

3. Evaluating Object proposals

Before we describe our evaluation and analysis, let us first look at the object proposal evaluation protocol that is widely used today. The following factors are involved:

1. Evaluation Metric: The metrics used for evaluating object proposals are all typically functions of intersection over union (IOU) (or Jaccard Index) between generated proposals and ground-truth annotations. For two boxes/regions $b_i$ and $b_j$ of an image, IOU is defined as:

$$\text{IOU}(b_i, b_j) = \frac{\text{area}(b_i \cap b_j)}{\text{area}(b_i \cup b_j)}$$

(1)

The following metrics are commonly used:

• Recall @ IOU Threshold $t$: For each ground-truth instance, this metric checks whether the ‘best’ proposal from a list $L$ has IOU greater than a threshold $t$. If so, this ground truth instance is considered ‘detected’ or ‘recalled’. Then average recall is measured over all the ground truth instances:

$$\text{Recall} @ t = \frac{1}{|G|} \sum_{g_i \in G} |\{ \max_{l_j \in L} \text{IOU}(g_i, l_j) > t \}|$$

(2)

where $I[\cdot]$ is an indicator function for the logical proposition in the argument. Object proposals are evaluated using this metric in two ways:

– plotting Recall-vs-#proposals by fixing $t$
– plotting Recall-vs-t by fixing the #proposals in $L$.

• Area Under the recall Curve (AUC): AUC summarizes the area under the Recall-vs-#proposals plot for different values of $t$ in a single plot. This metric measures AUC-vs-#proposals. It is also plotted by varying #proposals in $L$ and plotting AUC-vs-t.

• Volume Under Surface (VUS): This measures the average recall by linearly varying $t$ and varying the #proposals in $L$ on either linear or log scale. Thus it merges both kinds of AUC plots into one.

• Average Best Overlap (ABO): This metric eliminates the need for a threshold. We first calculate the overlap between each ground truth annotation $g_i \in G$, and the ‘best’ object hypotheses in $L$. ABO is calculated as the average:

$$\text{ABO} = \frac{1}{|G|} \sum_{g_i \in G} \max_{l_j \in L} \text{IOU}(g_i, l_j)$$

(3)

ABO is typically calculated on a per class basis. Mean Average Best Overlap (MABO) is defined as the mean ABO over all classes.
• **Average Recall (AR):** This metric was recently introduced in [34]. Here, average recall (for IOU between 0.5 to 1)-vs-#proposals in L is plotted. AR also summarizes proposal performance across different values of t. AR was shown to correlate with ultimate detection performance better than other metrics.

2. **Dataset:** The most commonly used datasets are the the PASCAL VOC [1] detection dataset sets. Note that these are partially annotated datasets where only the 20 PASCAL category instances are annotated. Recently analyses have been shown on ImageNet [35], which has more categories annotated than PASCAL, but is still a partially annotated dataset.

4. **A Thought Experiment:**

   **How to Game the Evaluation Protocol**

   Let us conduct a thought experiment to demonstrate that the object proposal evaluation protocol can be ‘gamed’.

   Imagine yourself reviewing a paper claiming to introduce a new object proposal method – called DMP.

   Before we divulge the details of DMP, consider the performance of DMP shown in Fig. 3 on the PASCAL VOC 2010 dataset, under the AUC-vs-#proposals metric.

   ![Figure 3: Performance of different object proposal methods (dashed lines) and our proposed 'fraudulent' method (DMP) on the PASCAL VOC 2010 dataset.](image)

   So what is our proposed state-of-art technique DMP? It is a mixture-of-experts model, consisting of 20 experts, where each expert is a DeCAF-based [36] objectness detector. At this point, you, the savvy reader, are probably already beginning to guess what we did.

   DMP stands for ‘Detector Masquerading as Proposal generator’. We trained object detectors for the 20 PASCAL categories (in this case with RCNN [13]), and then used these 20 detectors to produce the top-M most confident detections, and declared them to be ‘object proposals’.

   The point of this ad absurdum experiment is to demonstrate the following fact – Clearly, no one would consider a collection of 20 object detectors to be a category-independent object proposal generation method. However, our existing evaluation protocols declare them to be state-of-art.

   Why did this happen? Because the protocol today involves evaluating a proposal generator on a partially annotated dataset such as PASCAL. The protocol does not reward recall of non-PASCAL categories; in fact, early recall (near the top of the list of candidates) of non-PASCAL objects results in a penalty for the proposal generator! As a result, a proposal generator that tunes itself to these 20 PASCAL categories (either explicitly via training or implicitly via design choices or hyper-parameters) will be declared a better proposal generator when it may not be (as illustrated by DMP).

   Since the asymptotic limit of this evaluation protocol is an absurd proposal generator, we should be cautious of methods proposing incremental improvements on this protocol.

   This thought experiment exposes the inability of the existing protocol to evaluate category independence. There are two ways of alleviating this problem:

   - Modify the dataset, i.e. use a fully or more densely annotated dataset.
   - Modify the metric for evaluation to be used with a partially annotated dataset, because collecting a fully-annotated dataset might be expensive and tedious.

   We explore both these directions next.

5. **Modifying the Dataset**

   As our first analysis, we evaluate the performance of various object proposal methods and two DMPs (RCNN [13] and DPM [37]) on three different denser-annotated datasets containing many more object categories. This is to quantify how much the performance of our ‘fraudulent’ proposal generators (DMPs) drops once the bias towards the 20 PASCAL categories is diminished (or completely removed).

   We begin by creating a nearly fully-annotated dataset by building on the effort of PASCAL Context [38]; followed by evaluation on other partial-but-denser-annotated datasets MS COCO [39] and NYU-Depth V2 [40].
5.1. PASCAL Context

This dataset was introduced by Mottaghi et al. [38]. It contains additional annotations for all images of PASCAL VOC 2010 dataset [41]. The annotations are semantic segmentation maps, where every single pixel previously annotated ‘background’ in PASCAL was assigned a category label. In total, annotations have been provided for 459 categories. This includes the original 20 PASCAL categories and new classes such as keyboard, fridge, picture, cabinet, plate, clock.

Unfortunately, as of the time of writing this paper, the dataset contains only category-level semantic segmentations, not instance-level segmentations. For our task, we needed instance-level bounding box annotations, which cannot be reliably extracted from category-level segmentation masks because the masks for several instances (of say chairs) may be merged together into a single ‘blob’ in the category-level mask.

Creating Instance-Level Annotations for PASCAL Context: Thus, we created instance-level bounding box annotations for all images in PASCAL Context dataset. First, out of the 459 category labels in PASCAL Context, we identified 379 categories to be ‘thing’ categories, and ignored the remaining ‘stuff’ categories or ‘ambiguous’ categories (e.g., a ‘tree’ may be a ‘thing’ or ‘stuff’ depending on viewpoint of the camera) – neither of these lend themselves to bounding-box-based object detection. A complete list is available in the supplementary material.

Next, we selected the 60 most frequent non-PASCAL categories from this list of ‘things’ and manually annotated all instances of these categories. This manual annotation was performed with the aid of the semantic segmentation maps already present in the PASCAL context annotations. Examples annotations are shown in Fig. 5, and more examples are available in the supplement.

Statistics of New Annotations: Fig. 4 shows some statistics of our new annotations. Specifically, Fig. 4a shows average number of instances for the 20 PASCAL categories, the 60 new non-PASCAL categories, and the other ignored
‘thing’ objects. It is interesting to note that the number of annotations for the new 60 categories we annotated were about the same as the number of instances for 20 PASCAL categories. This is a good indicator of the number of proposal candidates which are not being rewarded due to partially annotated nature of the PASCAL VOC 2010 dataset.

Fig. 4b shows the average size (percentage of image area) of different types of objects – 20 PASCAL categories, 60 new categories, remaining (316) ‘thing’ categories, ‘stuff’ (or background) categories, and ‘ambiguous’ categories. We can see that most non-PASCAL categories occupy a small percentage of the image. This is understandable given that the dataset was curated for the 20 PASCAL categories. The other categories just happened to be in the pictures. Unfortunately, this also makes them difficult to be detected.

5.2. MS COCO

The Microsoft Common Objects in Context (MS COCO) dataset [39] contains 91 common object categories with 82 of them having more than 5,000 labeled instances. Overall, the dataset has 2,500,000 labeled instances in 328,000 images. The dataset not only has significantly higher number of instances per category than the PASCAL VOC dataset, but also considerably more object instances per image (7.7) as compared to ImageNet (3.0) and PASCAL (2.3).

5.3. NYU-Depth V2

NYU-Depth V2 dataset [40] is comprised of video sequences from a variety of indoor scenes as recorded by both the RGB and Depth cameras from the Microsoft Kinect. It features 1449 densely labeled pairs of aligned RGB and depth images where each object is labeled with a class and an instance number. We used these 1449 densely annotated RGB images for evaluating object proposal algorithms. To the best of our knowledge, this is the first paper to compare proposal methods on such a dataset.

5.4. Evaluating Proposals on Different Datasets

Object Proposals Library: To ease the process of carrying out the experiments, we created an open source, easy-to-use object proposals library. This can be used to seamlessly generate object proposals using all the existing algorithms (for which the Matlab code has been released by the respective authors). Our library can also be used to evaluate proposals on any dataset using the commonly used metrics: Recall @ Threshold/Detection Rate, ABO, AUC. This library is publicly available.

Experimental Setup: On MS COCO and PASCAL Context datasets we conducted experiments as follows:

- Use the existing evaluation protocol for evaluation, i.e., evaluate only on the 20 PASCAL categories.

- Evaluate on all the annotated classes.

- For the sake of completeness, we also report results on all the classes except the PASCAL 20 classes.

On NYU-Depth V2 we only evaluate the performance on all categories. This is because only 8 (of the 20) PASCAL categories are present in NYU-Depth V2 (since it comprises of indoor scenes).

The two DMPs we use are based on two popular object detectors - DPM [37] and RCNN [13]. We train a DPM on 20 pascal categories and use it as an object proposal method. To generate large number of proposals, we chose a low value of threshold in Non-Maximum Suppression (NMS). Proposals are generated for each category and a score is assigned to them by the corresponding DPM for that category. These proposals are then mergesorted on the basis of this score. Top M proposals are selected from this sorted list where M is the number of proposals to be generated.

Another (stronger) DMP is RCNN which is a detection pipeline that uses 20 SVMs (each for one PASCAL category) trained on decaf features extracted on selective search boxes. Since RCNN itself uses selective search proposals, it should be viewed as a trained reranker of selective search boxes. As a consequence, it ultimately equals selective search performance once the number of candidates become large. We used the pretrained SVM models released with the RCNN code, which were trained on the 20 classes of PASCAL VOC 2007 trainval set. For every test image, we generate the Selective Search proposals using the ‘FAST’ mode and calculate the 20 SVM scores for each proposal. The ‘objectness’ score of a proposal is then the maximum of the 20 SVM scores. All the proposals are then sorted by this score and top M proposals are selected.

Results and Observations: We now explore how changing the evaluation protocol affects the results of the thought experiment from Section 4.

Figs. 6, 7 compare the performance of DMPs with a number of existing object proposal methods [2–7, 9, 11, 12] on PASCAL Context and MS COCO respectively.

We can see in Column (a) that when evaluated on only 20 PASCAL categories the DMPs trained on these categories appear to significantly outperform all proposal generators. However, we can see that they are not category independent because they suffer a big drop in performance when evaluated on 60 non-PASCAL categories in Column (b).

Notice that on PASCAL context, all proposal generators suffer a drop in performance between the 20 PASCAL categories and 60 non-PASCAL categories. As we previously discussed, this is due to the fact that the non-PASCAL categories tend to be generally smaller than the PASCAL categories (which were the main targets of the dataset curators). However, the DMPs suffer the biggest drop.

1 Code for the Object Proposals library is available at: https://github.com/batra-mlp-lab/object-proposals/
It is interesting to note that due to the ratio of instances of 20 PASCAL categories vs other 60 categories, DMPs continue to slightly outperform proposal generators when evaluated on all categories, as shown in Column (c).

Fig. 8 shows results for NYU-Depth V2. We can see that when many classes in the test dataset are not PASCAL classes, DMPs tend to perform poorly, although interesting still not as poor as the worst proposal generators. Results on other evaluation criteria are in the supplementary document.

**Take-away messages:**

1. The drop in performance from after the adding new non-PASCAL categories exposes DMPs’ poor generalization on “unseen” categories. This drop can be used to detect and guard against ‘biased’ proposal generators.

2. The slight advantage that DMPs continue to hold over existing proposals (even when all categories are annotated) is due to the imbalance in the number of annotations of PASCAL classes and non-PASCAL classes. This exposes the problem of a metric which is class agnostic, used on a dataset which is unbalanced, and suggests the need for dataset with more coverage.

6. Modifying the Metric

To recap what we have discussed so far – we began by introducing a ‘fraudulent’ object proposal algorithm that could exploit or fool the existing evaluation protocols. We discussed how one way of detecting such bias is by changing the annotations in the dataset. However, annotating datasets is an expensive and time-consuming process. In this section, we analyze whether we can continue to use existing partially annotated datasets but modify the metrics used.
To gain other such insights, we look at a more fine-grained section, we saw that the performance of DMPs was different on non-PASCAL categories as compared to 20 PASCAL classes on the modified PASCAL Context dataset.

To gain other such insights, we look at a more fine-grained per category performance of proposal methods and DMPs. We evaluate by plotting recall values and area under the recall curve for all 80 (20 PASCAL + 60 non-PASCAL) categories for the modified PASCAL-Context dataset. We sorted/clustered all categories on the basis of:

- Average size (fraction of image area) of the category,
- Frequency (Number of instances) of the category,
- Membership in ‘super-categories’ defined in MS COCO dataset (electronics, animals, appliance, etc.).

**Trends:** Fig. 9 shows the performance of different proposal methods and DMPs along each of these dimensions. These plots can be used to answer if some proposal methods are optimized for larger or frequent categories. Interestingly, we noticed that all methods follow similar trends with respect to different attributes, suggesting that either no such optimization has taken place, or all methods have been optimized. It is reasonable to believe the former.

In Fig. 9a, we find that recall value steadily improves as the relative size of image object increases for all proposal methods as well as DMPs. This shows that perhaps as expected, bigger objects are typically easier to find than smaller objects. In Fig. 9b, we see that the recall values generally increases as the number of instances increase except for one outlier category. This category was found to be ‘pole’ which appears to be quite difficult to recall, since poles are often occluded and have a long elongated shape.

Overall, while this fine-grained analysis is useful, we found that it can not be used to detect a ‘fraudulent’ proposal generator, since DMPs behave similar to the other methods.

### 6.1. Measuring Fine-Grained Recall

One way of developing a fine-grained understanding of the performance of various proposal generation methods and to detect bias is by plotting Recall per category. In the previous section, we saw that the performance of DMPs was different on non-PASCAL categories as compared to 20 PASCAL classes on the modified PASCAL Context dataset.

Some proposal methods [3–8, 10–12] rely on explicit training to learn an “objectness” model, in a manner similar to our DMPs. Depending upon which and how many categories are trained on, these methods could have a biased view of “objectness”. One way of measuring the bias capacity in an proposal method to plot the performance vs. the number of categories trained on. A method that involves little or no training will be a flat curve on this plot. Biased methods such as DMPs will get better and better as more categories are seen in training. Thus, this analysis can help us find ‘biased’ or ‘bias-prone’ methods like DMPs that are tuned to specific classes.

In this experiment, we compared the performance of one DMP method (RCNN [13]), one learning-based proposal method (Objectness [5]), and two non-learning based proposal methods (Selective Search [9], EdgeBoxes [2]) as a function of the number of ‘seen’ categories (the categories trained on). Method names ‘rcnnTrainN’, ‘objectnessTrainN’ indicate that they were trained on images that contain annotations for only N categories. Once trained, these methods were evaluated on a randomly-chosen set of 2396 images taken from MS COCO [39] dataset. Fig. 10 shows the results. We can see that with even a modest increase in training data, performance improvement of RCNN is much more than objectness. There is no change in Selective Search [9] and EdgeBoxes [2] since there is no training involved at all. It is thus reasonable to conclude that object independent methods are less prone to be affected by training with more data as compared to methods like DMPs. This analysis could hence serve as a diagnostic tool for differentiating between such methods.

### 6.2. Assessing Bias Capacity

In Fig. 9a, we find that recall value steadily improves as the relative size of image object increases for all proposal methods as well as DMPs. This shows that perhaps as expected, bigger objects are typically easier to find than smaller objects. In Fig. 9b, we see that the recall values generally increases as the number of instances increase except for one outlier category. This category was found to be ‘pole’ which appears to be quite difficult to recall, since poles are often occluded and have a long elongated shape.

Overall, while this fine-grained analysis is useful, we found that it can not be used to detect a ‘fraudulent’ proposal generator, since DMPs behave similar to the other methods.

### 7. Conclusion

In this paper, we make an explicit distinction between the two mutually co-existing but different interpretations of object proposals. The current evaluation protocol for object proposal methods is suitable only for detection proposals and is a biased ‘gameable’ protocol for category-
independent object proposals. By evaluating only on a specific set of object categories, we fail to capture the performance of the proposal algorithm on all the remaining object categories that are present in the test set, but not annotated in the ground truth. We demonstrate this gameability via a simple thought experiment where we propose a ‘fraudulent’ object proposal method that outperforms all existing object proposal techniques on current metrics. We conduct a thorough evaluation of existing object proposal methods on three densely annotated datasets. We introduce a fully-annotated version of PASCAL VOC 2010 where we annotated all instances of all object categories occurring in all images. We hope this dataset will be broadly useful.

Furthermore, since densely annotating the dataset is a tedious and costly task; we proposed a set of diagnostic tools to plug the vulnerability of the current protocol.

Fortunately, we find that none of existing proposal methods seem to be biased, most of the existing algorithms and do generalize well to different datasets and in our experiments even on densely annotated datasets. In that sense, our findings are consistent with results in [34]. However, that should not prevent us from recognizing and safeguarding against the flaws in the protocol, lest we over-fit as a community to a specific set of object classes.

8. APPENDIX

The main paper demonstrated how the object proposal evaluation protocol is gameable and performed some experiments to resolve this gameability. Here we present additional details and results which support the arguments presented in the main paper.

In Section 8.1, we list the various object proposal algorithms used in our experiments.

Section 8.2 provides details of the annotation we created for PASCAL Context.

Section 8.3 presents extended results for comparison of different proposal methods on densely annotated datasets under different evaluation metrics.

Finally, we show the per category performance of various methods on MS COCO (to accompany the PASCAL Context results in the main paper) in section 8.4.

8.1. Related Work

Table 1 provides an overview of all the object proposal algorithms that fall under the scope of this paper and the evaluation protocol used by them. The character * indicates methods we have evaluated in this paper. Note that a majority of the approaches are learning based.

8.2. Details of PASCAL Context Annotation

As explained before, PASCAL Context provides full annotations for PASCAL VOC 2010 dataset in the form of semantic segmentations. A total of 459 object classes have labeled in this dataset. We split these into three categories namely Objects/Things, Background/Stuff and Ambiguous as shown in Tables 2, 4 and 3. Most object classes (396) were put in the ‘Objects’ category. 20 of these are PASCAL categories. Of the remaining 376, we selected the most frequently occurring 60 categories and added instance level annotations for the same.
| Method       | Code Source                        | Approach             | Learning Involved                                           | Metric                                                                 | Datasets                                      |
|--------------|-----------------------------------|----------------------|------------------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------|
| objectness* | Source code from [42]             | Window scoring       | Supervised, train on 6 PASCAL classes and their own custom dataset of 50 images | Recall @ t ≥ 0.5 vs # proposals                                       | PASCAL VOC 07 test set, test on unseen 16 PASCAL classes |
| selectiveSearch* | Source code from [43]         | Segment based        | No                                                         | Recall @ t ≥ 0.5 vs # proposals, MABO, per class ABO                 | PASCAL VOC 2007 test set, PASCAL VOC 2012 train val set |
| rahtu*       | Source code [44]                  | Window Scoring       | Yes, two stages. Learning of generic bounding box prior on PASCAL VOC 2007 train set, weights for feature combination learnt on the dataset released with [42] | Recall @ t > various IoU thresholds and # proposals, AUC              | PASCAL VOC 2007 test set                       |
| randomPrim*  | Source code from [45]             | Segment based        | Yes supervised, train on 6 PASCAL categories               | Recall @ t > various IOU thresholds using 10k and 1k proposals        | Pascal VOC 2007 test set/2012 trainval set on 14 categories not used in training |
| mcg*         | Source code from [46]             | Segment based        | Yes                                                        | NA, only segments were evaluated                                     | NA                                           |
| edgeBoxes*   | Source code from [47]             | Window scoring       | No                                                         | AUC, Recall @ t > various IOU thresholds and # proposals, Recall vs IoU | PASCAL VOC 2007 testset                       |
| bing*        | Source code from [48]             | Window scoring       | Supervised, on PASCAL VOC 2007 train set, 20 object classes/6 object classes | Recall @ t > 0.5 vs # proposals                                       | PASCAL VOC 2007 detection complete test/14 unseen object categories |
| rantalankila  | Source code [49]                  | Segment based        | Yes                                                        | NA, only segments are evaluated                                      | NA                                           |
| Geodesic     | Source code from [50]             | Segment based        | Yes, for seed placement and mask construction on PASCAL VOC 2012 Segmentation training set | VUS at 10k and 2k windows, Recall vs IoU threshold, Recall vs proposals | PASCAL 2012 detection validation set          |
| Rigor        | Source code from [51]             | Segment based        | Yes, pairwise potentials between super pixels learned on BSDS-500 boundary detection dataset | NA, only segments were evaluated                                     | NA                                           |
| endres       | Source code from [52]             | Segment based        | Yes                                                        | NA, only segments are evaluated                                      | NA                                           |

Table 1: Properties of existing bounding box approaches. * indicates the methods which have studied in this paper.
Object/Thing Classes in PASCAL Context Dataset

| accordion | candleholder | drainer | funnel | lightbulb | pillar | sheep | tire |
|-----------|--------------|---------|--------|-----------|--------|-------|------|
| aeroplane | cap           | drop    | furnace | lighter | pillow | shell | toaster |
| airconditioner | car      | drinkdispenser | gamecontroller | line | pipe | shoe | toilet |
| antenna | card         | drinkingmachine | gamemachine | lion | pitcher | shoppingcart | tong |
| ashtray | cart         | drop    | gascylinder | lobster | plant | shovel | tool |
| babycarriage | case      | drug    | gashold | lock | plate | sidecar | toothbrush |
| bag       | cassettrecorder | drum    | gasstove | machine | player | sign | towel |
| ball      | cashregister | drumkit | gifbox | mailbox | mailers | signallight | toy |
| balloon   | cat          | duck    | glass | mannequin | plume | sunk | toycar |
| barrel    | cd           | dumbell | glassmarble | map | poker | skateboard | train |
| baseballbat | cdplayer   | earphone | globe | mask | pokercach | ski | trampoline |
| basket    | cellphone    | earrings | glove | mat | pole | sled | trashbin |
| basketballbackboard | cello | egg | gravestone | matchbook | pooltable | slippers | tray |
| bathtub   | chair        | electricfan | guitar | matte | postcard | snail | tricycle |
| bed       | chair        | electricfan | gun | menu | poster | snake | tripod |
| beer      | chessboard  | electricpot | hammer | meterbox | pot | snowmobiles | trophy |
| bell      | chicken      | electricsaw | handcart | microphone | pottedplant | sofa | truck |
| bench     | chopstick    | electronicskeyboard | handle | microwave | printer | spanner | tube |
| bicycle   | clip         | engine | hanger | mirror | projector | spatula | turtle |
| binoculars | clippers    | envelope | hardiskdrive | missile | pumpkin | speaker | tvmonitor |
| bird      | clock        | equipment | hat | model | rabbit | spacecontainer | sweater |
| birdcage  | closet       | extinguisher | headphone | money | racket | spoon | typewriter |
| birdfeeder | cloth        | eyeglass | heater | monkey | radiator | sprayer | umbrella |
| birdnest  | coffee       | fan | helicopter | mop | radio | squirrel | vacuumcleaner |
| blackboard | coffeemachine | faucet | helmet | motorcycle | rake | stapler | vendingmachine |
| board     | comb         | faxmachine | holder | mouse | ramp | stick | videocamera |
| boat      | computer     | ferriiswheel | hook | mousepad | ringhede | sticknote | videogameconsole |
| bone      | container    | firehydrant | horse | musicinstrument | receiver | stone | videoencoder |
| book      | controller   | fire-hydrant | horse-drawncarriage | kazoo | recorder | steed | videotape |
| bottle    | controller   | firelace | hot-airballoon | newspaper | remoteccontrol | straw | wakeboard |
| bottleopener | cooker      | fish | hydrovalve | newspaper | remoteccontrol | straw | wakeboard |
| bow       | copymachine  | fishtank | inflatable | oar | robot | stretcher | wallet |
| box       | cork          | fishbowl | ipod | ornament | rock | sun | wardrobe |
| bracelet  | corkscrew    | fishingnet | iron | oven | rocket | sunglass | washingmachine |
| brick     | cow           | fishingpole | ironingboard | oxygenbottle | rockinghorse | sunshade | watch |
| broom     | cratck       | flag | jay | pack | rope | surveilancecamera | waterdispenser |
| brush     | crane         | flagstaff | kart | pan | rag | swimwear | waterpiper |
| bucket    | crate         | flashlight | kettle | paper | ruler | swimmer | watermelon |
| bus       | cross         | flower | key | paperbox | saddle | swimming | whale |
| cabinet   | clutch        | fly | keyboard | paperpouch | saw | sweet | whirlpool |
| cabinetdoor | cup          | food | keyboard | parachute | scale | switch | wheel |
| cage      | curtain       | forceps | knife | parapet | scale | switch | wheel |
| cake      | cushion       | fork | knifeblock | pen | scissors | table | wheelchair |
| calculator | cuttingboard | forklift | ladder | penci | container | scoon | tank |
| calendar  | disc          | fountain | laddertruck | pencil | screen | tap | wireglass |
| camel     | discisease    | fox | ladle | person | screwdriver | tape | wire |
| camera    | dislahware   | frame | laptop | photo | sculpture | tarp | wire |
| cameralens | dog           | fridges | lid | piano | scythe | telephone | wire |
| can       | dolphin       | frog | lifebuoy | picture | sewer | telephonebooth | wire |
| candle    | door          | fruit | light | pig | sewingmachine | tent |

### Table 2: Object/Thing Classes in PASCAL Context

| Ambiguous Classes in PASCAL Context Dataset |
|---------------------------------------------|
| artillery | escalator | ice | speedbump |
| bedclothes | exhibitionbooth | leaves | stair |
| clothetheater | flame | outlet | tree |
| coral | guardrail | rail | unknown |
| dais | hand | shelves |

### Table 3: Ambiguous Classes in PASCAL Context

| Background/Stuff Classes in PASCAL Context Dataset |
|-----------------------------------------------|
| atrium | floor | parterre | sky |
| bamboo | weaving | foam | patio | smoke |
| bridge | building | footbridge | pelage | snow |
| ceiling | concrete | grandstand | platform | swimmingpool |
| control | control | booth | ground | road |
| court | counter | hay | runway | water |
| dock | door | kitchen | sand | wharf |
| fence | mountain | metal | shed | wood |
| ground | mountain | rope | sail | wood |

### Table 4: Background/Stuff Classes in PASCAL Context

**Statistics of PASCAL vs Non PASCAL annotations**

We noted in the take-away message in section 5.4 of our paper that DMPs were still doing better than other proposals on fully annotated datasets. This is because of the imbalance in the number of PASCAL and non PASCAL annotations. Table 5 summarizes the statistics of annotations on various datasets.

**8.3. Evaluation of Proposals on Other Metrics**

In this section, we show the performance of different proposal methods and DMPs on MS COCO dataset on var-
ious metrics. Fig. 11a shows performance on Recall-vs-IoU metric at 1000 #candidates on PASCAL 20 categories. Fig. 11b, Fig. 11c show performance on Recall-vs-#candidates metric at 0.5 and 0.7 IoU respectively.

Similarly in Fig. 12 and Fig. 13, we can see the performance of all proposal methods and DMPs on these three metrics where 60 non-PASCAL and all categories respectively are annotated in the MS COCO dataset.

These metrics also demonstrate the same trend as shown by the AUC-vs-#candidates in the main paper. When only PASCAL categories are annotated (Fig. 11), DMPs outperform all proposal methods. However, when other categories are also annotated (Fig. 13) or the performance is evaluated specifically on the other categories (Fig. 12), DMPs cease to be the top performers.

For the sake of completeness, we also report results on different metrics PASCAL Context (Fig. 14, Fig. 15 and Fig. 16) and NYU-Depth v2 (Fig. 17). They also show similar trends backing our claim.

### 8.4. Measuring Fine-Grained Recall

To look at a more fine-grained per category performance of proposal methods and DMPs, we presented the plots of recall values for all 80 (20 PASCAL + 60 non-PASCAL) categories for the modified PASCAL Context dataset. We have done the same experiment on MS COCO data set. 20 PASCAL categories are the same. However, the other 60 categories are different for MS COCO. It can be seen in Fig. 18, the trends on MS COCO are more or less similar to PASCAL Context.

### References

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,” http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[2] C. L. Zitnick and P. Dollar, “Edge boxes: Locating object proposals from edges,” in ECCV, 2014.

[3] I. Endres and D. Hoiem, “Category-independent object proposals with diverse ranking,” PAMI, vol. 36, no. 2, pp. 222–234, 2014.

[4] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, “Multiscale combinatorial grouping,” in CVPR, 2014.

[5] B. Alexe, T. Deselaers, and V. Ferrari, “Measuring the objectness of image windows,” PAMI, 2012.

[6] E. Rahtu, J. Kannala, and M. B. Blaschko, “Learning a category independent object detection cascade,” in ICCV, 2011.

[7] S. Manen, M. Guillaumin, and L. Van Gool, “Prime object proposals with randomized prim’s algorithm,” in ICCV, 2013.

[8] P. Rantalankila, J. Kannala, and E. Rahtu, “Generating object segmentation proposals using global and local search,” in CVPR, 2014.

[9] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, “Selective search for object recognition,” IJCV, 2013.

[10] A. Humayun, F. Li, and J. M. Rehg, “Rigor- recycling inference in graph cuts for generating object regions,” in CVPR, 2014.

[11] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, “Bing: Binarized normed gradients for objectness estimation at 300fps,” in CVPR, 2014.

[12] P. Krähenbühl and V. Koltun, “Geodesic object proposals,” in ECCV, 2014.

[13] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” CoRR, vol. abs/1311.2524, 2013.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” CoRR, 2014.

[15] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov, “Scalable, high-quality object detection,” CoRR, vol. abs/1412.1441, 2014.

[16] X. Wang, M. Yang, S. Zhu, and Y. Lin, “Regionlets for generic object detection,” in ICCV, 2013.

[17] R. G. Cimbis, J. Verbeek, and C. Schmid, “Segmentation Driven Ob-
Recall at IoU threshold 0.5
Recall at IoU threshold 0.7
Recall

0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

IoU overlap threshold @ 1000
Recall

DPM
RCNN
edgeBoxes
objectness
randomPrim
rahtu
mcg
selectiveSearch
bing

(a) Recall vs IOU at 1000 proposals for all non-PASCAL categories annotated in MS COCO validation dataset

(b) Recall vs number of candidates at 0.5 IoU for 60 non-PASCAL categories annotated in MS COCO validation dataset

(c) Recall vs number of candidates at 0.7 IoU for 60 non-PASCAL categories annotated in MS COCO validation dataset

(b) Recall vs number of candidates at 0.5
tion dataset

Figure 12: Performance of various object proposal methods on different evaluation metrics when evaluated on MS COCO dataset containing annotations for only 60 non-PASCAL categories

(c) Recall vs number of candidates at 0.7

Figure 13: Performance of various object proposal methods on different evaluation metrics when evaluated on MS COCO dataset containing annotations for all categories

[18] J. Carreira and C. Sminchisescu, “CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts,” *PAMI*, vol. 34, 2012.

[19] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. D. Bourdev, and J. Malik, “Semantic segmentation using regions and parts,” in *CVPR*, 2012.

[20] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, “Semantic segmentation with second-order pooling,” in *ECCV*, 2012.

[21] T. Deselaers, B. Alexe, and V. Ferrari, “Weakly supervised localization and learning with generic knowledge,” *IJCV*, no. 3, pp. 275–293, 2012.

[22] R. G. Cinbis and S. Sclaroff, “Contextual object detection using set-based classification,” in *ECCV*, 2012.

[23] A. Prest, C. Schmid, and V. Ferrari, “Weakly supervised learning of interactions between humans and objects,” *PAMI*, vol. 34, no. 3, pp. 601–614, 2012.

[24] J. Sun and H. Ling, “Scale and object aware image retargeting for thumbnail browsing,” in *ICCV*, 2011.

[25] F. Sener, C. Bas, and N. Ikizler-Cinbis, “On recognizing actions in still images via multiple features,” in *ECCV*, 2012.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *arXiv preprint arXiv:1409.4842*, 2014.

[27] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*, 2013.

[28] W. Ouyang, P. Luo, X. Zeng, S. Qiu, Y. Tian, H. Li, S. Yang, Z. Wang, Y. Xiong, C. Qian, *et al.*, “Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection,” *arXiv preprint arXiv:1409.3505*, 2014.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” 2014.

[30] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid, “Spatio-temporal object detection proposals,” in *ECCV*, 2014.

[31] K. Fragiadaki, P. Arbelaez, P. Felsen, and J. Malik, “Spatio-temporal moving object proposals,” *arXiv preprint arXiv:1412.6504*, 2014.

[32] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning Rich Features from RGB-D Images for Object Detection and Segmentation,” *arXiv preprint arXiv:1407.5736*, 2014.

[33] D. Banica and C. Sminchisescu, “CPMC-3D-O2P: semantic segmentation of RGB-D images using CPMC and second order pooling,” *CoRR*, vol. abs/1312.7715, 2013.

[34] J. Hong, R. Benenson, P. Dollár, and B. Schiele, “What makes for effective detection proposals?,” *arXiv preprint arXiv:1502.05082*, 2015.

[35] J. Hong, R. Benenson, and B. Schiele, “How good are detection proposals, really?,” in *BMVC*, 2014.

[36] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” *arXiv preprint arXiv:1310.1531*, 2013.
Figure 14: Performance of various object proposal methods on different evaluation metrics when evaluated on PASCAL Context dataset containing annotations for only 20 PASCAL categories

(a) Recall vs IOU at 1000 proposals for non-PASCAL categories annotated in PASCAL Context dataset
(b) Recall vs number of candidates at 0.5 IOU for 20 PASCAL annotated in PASCAL Context dataset
(c) Recall vs number of candidates at 0.7 IOU for 20 PASCAL categories annotated in PASCAL Context dataset

Figure 15: Performance of various object proposal methods on different evaluation metrics when evaluated on PASCAL Context dataset containing annotations for only non-PASCAL categories

(a) Recall vs IOU at 1000 proposals for non-PASCAL categories annotated in PASCAL Context dataset
(b) Recall vs number of candidates at 0.5 IOU for non-PASCAL categories annotated in PASCAL Context dataset
(c) Recall vs number of candidates at 0.7 IOU for non-PASCAL categories annotated in PASCAL Context dataset

[37] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” PAMI, vol. 32, no. 9, pp. 1627–1645, 2010. 4, 6
[38] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, “The role of context for object detection and semantic segmentation in the wild,” in CVPR, 2014. 4, 5
[39] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in ECCV, 2014. 4, 6, 8
[40] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in ECCV, 2012. 4, 6
[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results.” http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html. 5
[42] B. Alexe, T. Deselaers, and V. Ferrari, “Objectness measure v2.2.” http://groups.inf.ed.ac.uk/calvin/objectness/objectness-release-v2.2.zip. 10
[43] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” http://koen.me/research/downloads/SelectiveSearchCodeIJCV.zip. 10
[44] E. Rahtu, J. Kannala, and M. B. Blaschko, “Learning a category independent object detection cascade.” http://www.ee.oulu.fi/research/imag/object_ detection/ObjectnessICCV_ver02.zip. 10
[45] S. Manen, M. Guillaumin, and L. V. Gool, “Prime object proposals with randomized prims algorithm.” https://github.com/smanenfr/rp#rp. 10
[46] P. Arbelaez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, “Multiscale combinatorial grouping,” https://github.com/jPonttuset/mcg/archive/v2.0.zip. 10
[47] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges.” https://github.com/pdollar/edges. 10
[48] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, “Bing: Binarized normed gradients for objectness estimation at 300fps.” https://github.com/varun-nagaraja/BING-Objectness. 10
[49] P. Rantalankila, J. Kannala, and E. Rahtu, “Generating object segmentation proposals using global and local search.” http://www.cse.oulu.fi/~erahtu/ObjSegProp/spaggrom_01.zip. 10
[50] P. Krähenbühl and V. Koltun, “Geodesic object proposals.” http://www.philkr.net/home/gop. 10
[51] A. Humayun, F. Li, and J. M. Rehg, “Rigor: Recycling inference in graph cuts for generating object regions.” http://cpl.cc.gatech.edu/projects/RIGOR/resources/rigor_src.zip. 10
[52] I. Endres and D. Hoiem, “Category-independent object proposals with diverse ranking.” http://vision.cs.uiuc.edu/proposals/data/PROP_code.tar.gz. 10
Recall @ 0.7 IOU, #candidates=1000

(a) Recall vs number of instances.
(b) Sorted by the number of instances.
(c) MS COCO super-categories.

Figure 16: Performance of various object proposal methods on different evaluation metrics when evaluated on PASCAL Context dataset containing annotations for all categories.

Figure 17: Performance of various object proposal methods on different evaluation metrics when evaluated on NYU2 dataset containing annotations for all categories.

Figure 18: Recall at 0.7 IOU for categories sorted/clustered by (a) size, (b) number of instances, and (c) MS COCO ‘super-categories’.