The Pursuit of Knowledge:
Discovering and Localizing Novel Categories using Dual Memory

Sai Saketh Rambhatla[1]   Rama Chellappa[2]   Abhinav Shrivastava[1]

[1]University of Maryland, College Park   [2]Johns Hopkins University

Abstract

We tackle object category discovery, which is the problem of discovering and localizing novel objects in a large unlabeled dataset. While existing methods show results on datasets with less cluttered scenes and fewer object instances per image, we present our results on the challenging COCO dataset. Moreover, we argue that, rather than discovering new categories from scratch, discovery algorithms can benefit from identifying what is already known and focusing their attention on the unknown. We propose a method that exploits prior knowledge about certain object types to discover new categories by leveraging two memory modules, namely Working and Semantic memory. We show the performance of our detector on the COCO minival dataset to demonstrate its in-the-wild capabilities.

1. Introduction

Unsupervised visual category discovery aims to automatically identify recurring "patterns," which can be objects or object parts, and consequently learn recognition models to identify them from a large collection of unlabeled images with minimal human supervision. Such algorithms can dramatically reduce annotation costs as they only need labels for already mined object clusters. Moreover, automatically discovering categories from unstructured data can help mitigate the biases that occur when manually constructing datasets by collecting images for a fixed-set of concepts. Then, why is this setup not a mainstay in recognition?

A key issue is the lack of consistent task definition, with two disparate families of approaches aiming to address unsupervised object discovery. The first set of approaches [1–4] reduces the problem to co-localization or co-segmentation, by only answering whether a pair of images share the same object and then localizing them. The other set of, arguably more generic, approaches [5–7] use clustering techniques to discover semantically similar regions and can sometimes leverage prior knowledge. Because of these different goals, each set defines evaluation protocols
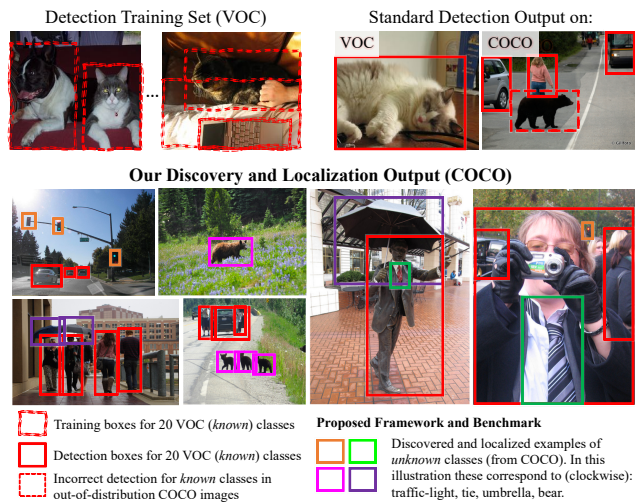

Figure 1: **Motivation**. A detector trained on 20 VOC (*known*) classes struggles on out-of-distribution images (*e.g.*, COCO) in the presence of novel (*unknown*) objects (*e.g.*, bear). Our discovery and localization framework builds on this detector and can reliably localize and group semantically meaningful "patterns" in images with both known and novel objects in challenging images. Novel objects belonging to the same class are assigned the same bounding box color. Best viewed in color.

that are incompatible with others. The second issue that limits the wide adoption of unsupervised discovery is the scalability of contemporary approaches – most works use small curated datasets and cannot scale to a realistic setup with a large number of images, a large number of object categories, and complex images (*e.g.*, a large number of objects and categories per image). Finally, a desirable property of unsupervised discovery approaches, often lacking in contemporary works, is online processing of new data as it arrives. To address these issues, it is necessary to define a standardized protocol (datasets and metrics), that more closely reflect real-world requirements of a task like unsupervised object discovery.

Towards this, this paper proposes a large-scale benchmark for evaluating unsupervised object discovery approaches and a scalable never-ending discovery approach that can deal with real-world complexities. Our approach is loosely inspired by how humans learn – continuously, utilizing prior knowledge. Our benchmark is designed to evalu-

ate such continuous and knowledge-driven learning and reflect real-world complexities and scale, and as a practical consideration, it is amenable to using available pretraining datasets in object recognition literature.

Motivation: Humans never stop learning [8, 9]. Since infancy, we continuously stumble upon objects that we have never encountered before and learn to recognize them with time [10]. This process is mostly reliant on the current knowledge we possess. For example, a toddler, with a pet dog at home, may point to a lion at a zoo and mistakenly call it a dog; but the toddler is unlikely to point to a bench, because she hasn't learned about chairs and couches yet [11]. This alludes to a continuous learning paradigm, where we notice new objects, associate them with our current knowledge [12], and update our knowledge (learn new concepts or update existing concepts). So how are we able to do this so effortlessly, and often unconsciously?

Studies have demonstrated that "memory" is the primary form of representation that brings together perception and learning [13], and knowledge, categories, *etc.*, all derive from memory [14]. Despite overwhelming evidence of its importance, memory is arguably one of the least studied components in computational visual recognition; its role often ceded to other modules; *e.g.*, pretrained weights of a neural network [15], forms of knowledge-graphs [16, 17]. In this work, we investigate how memory can be used to represent knowledge and drive the discovery of new concepts and develop a continuous learning approach that is inspired by well established memory processes [18].

Contributions: Our two complimentary contributions are: a scalable concept discovery and localization approach, and a realistic protocol for evaluating discovery approaches.

Our first contribution, a never-ending concept discovery and localization framework, is built using two memory modules: Semantic and Working memory, jointly referred to as **dual memory**. Semantic memory is the portion of long-term memory that contains concepts from past experience (current knowledge). Working memory, on the other hand, is synonymous to short-term memory and is responsible for accumulating and temporarily holding information for processing. Our final algorithm consists of carefully crafted operations which compare a new region to the dual memory, decide if it is known (or already discovered) or a novel category, and accordingly update the relevant memory modules. When sufficient concepts accumulate in the working memory, our algorithm amalgamates learned concepts from the working memory into the semantic memory, which then become part of prior knowledge. This process can continue in a never-ending, online fashion.

Our second contribution is a simple, realistic benchmark for never-ending concept discovery, not only to evaluate our approach, but also to enable future works to compare on a standardized protocol. We argue that most standard benchmarks are either not realistic, are designed for classification, are suited more for co-segmentation than discovery, are not labeled to evaluate performance, or are too small in scale. In addition, earlier works on never-ending and large-scale learning [19–21] either assume an a-priori list of concepts, which is unrealistic, utilizes crawled internet images which is not reproducible, or use proprietary datasets which cannot be released, ruling out future comparisons. Therefore, we propose a simple, realistic benchmark for never-ending concept discovery and localization.

Our benchmark is designed to leverage three existing datasets, ImageNet, Pascal VOC, and COCO. We assume ImageNet and VOC are the labeled datasets available for pretraining and detecting 20 classes, respectively, from which prior knowledge can be derived; and COCO is the in-the-wild dataset to perform concept discovery and localization. The 20 VOC classes are treated as *known* classes, and the 60 additional classes in COCO are *unknown* classes used to evaluate a subset of the discovered objects. Other discovered objects are qualitatively assessed. This setup has six desirable properties: (1) these datasets are widely used, and therefore, we know the performance if all 80 classes were labeled; (2) the discovery set consists of all known and a variety of unknown classes; (3) the images in COCO have a slightly different distribution from VOC; (4) dataset is large-scale; (5) bounding box labels are available and (6) future progress on these datasets [22] can be leveraged.

2. Related Works

Similar learning paradigms. Several paradigms have been proposed to imitate continuous learning models, including, but not limited to, incremental learning [23–27], never-ending learning [20, 21], open-world learning [28], semi-supervised [19, 29, 30], and omni-supervised learning [31]. Each of these paradigms is actively pursued by computer vision and machine learning researchers; and an equitable and comprehensive discussion with even the relevant approaches is unrealizable given the manuscript length constraints. We discuss the differences between these paradigms and the problem setup for our approach.

Most approaches mentioned above differ from ours in one of the following aspects: (a) our focus is the localization task, unlike most incremental [23–26], open-world [28], and semi-supervised learning approaches [19]; (b) we do not assume any a-priori list of concepts of interest (novel classes), unlike most incremental [23–26], semi-supervised [19], and never-ending learning frameworks [20, 21]; (c) unlike contemporary discovery methods [1–4], we operate in a real-world setup, where images contain many objects, including ones that are similar to known objects; (d) scalability to larger datasets and objects. While our work shares the goal of [20] in learning models for never-ending learning for training object detectors, there

is one technical and several practical differences. **First**, [20] is *language-driven* (ref. §1.1c in [20]), *i.e.*, it starts with an a-priori list of concepts and then searches images for that concept, and then attempts to build the detectors. As opposed to this, our approach does not start with any a-priori list, and discovers in the set of images it is given. We believe this is a crucial distinction. **Second**, [20] assumes at least a few 'canonical' images for concepts and uses it to train detectors (cf. §3.1 in [20]). In contrast, our approach can discover and localize objects directly in complex images. **Finally**, data used by [20] has not been publicly released making it impossible to compare with it fairly on the same set of images. Moreover, the released code relies on the discussed assumptions, making it unsuitable for the current setup (COCO dataset).

Object Category Discovery. Category discovery is the problem of identifying semantically similar recurring patterns in unlabelled data. Object category discovery works can be broadly divided into two categories: image [32–37] and region-based approaches [2–7, 38–49]. Our discovery framework is a region-based approach. Recent region-based methods [1–4], do not assume any prior knowledge and solve an optimization problem to discover new objects. Unlike such methods, we assume a set of known objects and leverage this knowledge to discover novel objects, similar to [5, 6]. Lee *et al*. [5] proposed object graphs to incorporate context which improves clustering for the discovery of novel categories. In contrast, we use cluster prototypes to improve clustering; and unlike [5], we do not require the number of clusters as an input. Improving on their previous work [5], Lee *et al*. [6] proposed an iterative, self-paced approach for category discovery that focuses on the easiest instances first, and then progressively expands its repertoire to include more complex objects. Our approach is implicitly self-paced and does not need a specific curriculum definition for discovery.

For an in-depth review of other object discovery methods, memory formulations, and our relationship to learning paradigms (*e.g*., supervised, incremental, open-world, never-ending), kindly refer to the **supplementary** material.

3. Approach Overview

In this section, we illustrate our approach using an example, as shown in Figure 2. For simplicity, we assume the knowledge of two objects ("known" categories), namely human and car. Our system consists of three main modules, **Encoding**, **Storage** and **Retrieval**. The **Encoding** module is a Region Proposal Network (RPN) [50] that gives candidate regions and corresponding features. The **Storage** module consists of two memory blocks, the *Semantic* and *Working* Memory. The *Semantic* memory consists of slots that store representations to identify regions of "known" or previously discovered objects while the slots in *Working* mem-
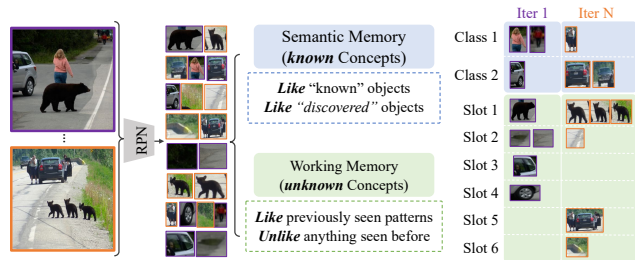

Figure 2: An example illustrating our dual-memory formulation.

ory store representations for potentially "discoverable" object. In Figure 2, the *Semantic* memory is initialized with two slots (human and car classes) and the *Working* memory is null initialized. The **Retrieval** module decides whether a region belongs to the *Semantic* ("known"/discovered) or the *Working* (to be discovered) memory. These modules are discussed in detail in the next section.

Our system operates sequentially, processing one image at a time. First, the **Encoding** module processes the image and outputs candidate regions and features. The **Retrieval** module assigns each region to either *Semantic* or *Working* memory. In Figure 2, after the first iteration, three regions (two humans and a car) have been assigned to the two slots of the *Semantic* memory and the remaining regions have been assigned to four different slots (Slot 1-4) in the *Working* memory. In the next iteration, the retrieval module assigns three regions (one human and two vehicles) to the semantic memory while, the remaining regions have been assigned to two previously created slots (Slot 1-2) and two new slots (Slot 5-6). Note that the retrieval module can either decide to populate an existing slot in the *Working* memory or create new patterns if necessary. This capability eliminates the need to know the number of "unknown" objects apriori. Once all the images are processed and we perform **Memory Consolidation**, which step transfers the knowledge acquired in the *Working* memory to the *Semantic* memory, this updating the list of known categories.

4. Framework Details

In this section, we describe the encoding, storage, and retrieval modules, and how they interact with each other in detail (refer to Figure 3).

Encoding: The goal of the encoding module is to process an input image and extract representations to be subsequently used by the discovery pipeline. Our encoding module is an object detector, Faster R-CNN [50], trained on a dataset (PASCAL VOC [51]) consisting of a set of known objects (20 objects). Given an image, we use N (= 150) proposals/boxes from a region proposal network [50] and their corresponding ConvNet features from the classification head (referred to as encoded representation), for subsequent discovery. This module is akin to the encoding process in human memory, which converts sensory inputs to representations to be used by other processes.
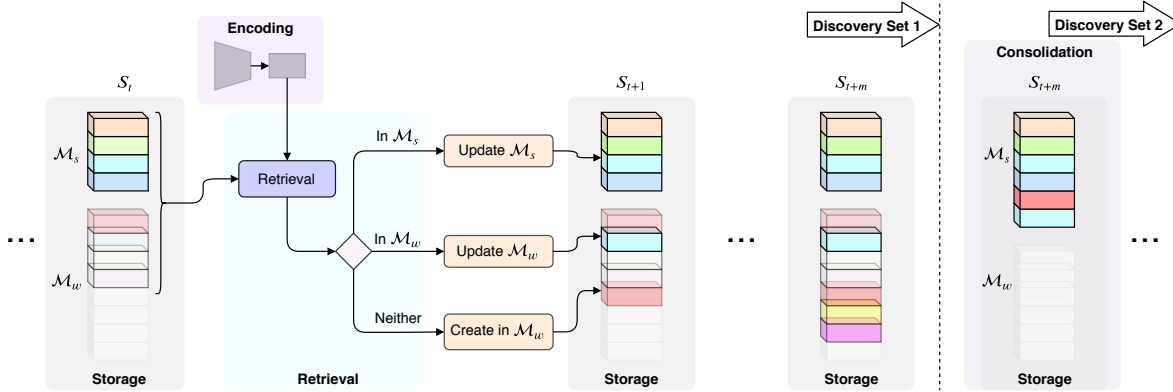
Figure 3: Overview of our Dual Memory formulation. Refer to Section 4 for details.

Storage: The storage module consists of two memory blocks: *Semantic* ($\mathcal{M}_s$) and *Working* Memory ($\mathcal{M}_w$). $\mathcal{M}_s$ serves the purpose of storing the prior and discovered concepts, and is initialized with 'semantic priors', computed using an object detector trained on known classes. It resembles the human's long-term memory in that aspect, which stores prior or acquired knowledge. On the other hand, $\mathcal{M}_w$ is used to temporarily store and manipulate representations of recently encountered objects, which can potentially be discovered, and is null initialized. The working memory resembles the short-term memory. A concept is considered discovered when we encounter enough instances of it, are able to associate them, and learn a new class from them. One of our key contributions is exploring suitable formulations for this dual memory.

Storage slots. Both the memory modules are composed of slots, where each slot represents the regions belonging to it from the data encountered so far. A **slot's representation**, computed using features of all regions belonging to the particular slot, is critical to our approach. If slots represent object classes, then the slot representations are essentially models for these classes and are 'trained' using all instances that belong to this class/slot. These representations are used by the retrieval module to decide if a new region/instance 'corresponds' to a particular slot. Representing and operating on slots are the cornerstones of our framework, therefore, the slot representation needs to be: (a) effective in representing a large-set of instances corresponding to that slot, (b) efficient in retrieval operations (*e.g.*, associate new instances, updates), and (c) scalable to a large number of object instances and classes. In this work, we explore two choices for slot representations: feature **centroids** of all the instances belonging to a slot, and a **classifier** trained on the features, each with a different speed/accuracy trade-off.

The **centroid representation** is a straightforward centroid of all the instances belonging to a slot, $\mu^{\mathbb{C}}$; which can be *updated* efficiently using the cumulative moving average. We evaluate the association of an instance feature f with the slot using cosine similarity, $\mathrm{cosine}(f, \mu^{\mathbb{C}})$.

The **classifier representation** learns a classifier using all the instances belonging to a slot and classification score is used to evaluate association. We train LDA classifier [52] for each slot. More specifically, we pose the problem of associating an instance to a slot as a two-class problem, slot *vs.* background. The LDA classifier is a linear classifier with closed form solution $w = \Sigma^{-1}(\mu^+ - \mu^-)$ and $b = \log(\frac{\pi^+}{\pi^-}) - \frac{1}{2}(\mu^+ - \mu^-)\Sigma^{-1}(\mu^+ + \mu^-)$. Here μ^+ is the mean of all instances in a slot, μ^- is the mean of background class, π^+/π^- are number of samples in positive/background classes, and Σ is the class conditional covariance matrix (assumed to be same for positive/background [52, 53]). The background class statistics, (Σ, μ_0), can be estimated efficiently offline using an online update formula operating on boxes from all images from the dataset [52, 53] (derivations are included in the supplementary material). We use the classifier score, $(w^T f + b)$, to evaluate the association of the instance feature f to a slot. An added benefit of this closed-form solution is that to *update* the classifier with a new instance, we only need to *update* π^+.

Different *slot representations* exhibit different performance and scalability characteristics, and understanding them are crucial for the decision making process of the retrieval module. Centroids are fast to calculate, but not as accurate, whereas classifiers are slow, but accurate.

Recall that $\mathcal{M}_s$ is the stable long term memory, representing classes (known or discovered) with enough samples; and $\mathcal{M}_w$ is the volatile short term memory, storing 'novel' concepts we encounter, adding/deleting slots frequently, and often with few instances. Therefore, we use classifiers for $\mathcal{M}_s$, where a slot i is (w_i, b_i) or $(\mu_i^+, \pi_i^+, \mu^-, \Sigma, \pi^-)$; and centroids for $\mathcal{M}_w$, where a slot j is $(\mu_j^{\mathbb{C}}, \pi_j)$ (π_j is number of instances, needed for cumulative moving average). Note that since (μ^-, π^-, Σ) are shared across all slots in $\mathcal{M}_s$, the implementation of both $\mathcal{M}_s$ and $\mathcal{M}_w$ are similar. As previously mentioned, $\mathcal{M}_s$ is initialized with 'semantic priors', which are computed using the detector's outputs on the training set; *i.e.*, we take detections with score > 0.9 and compute a slot representa-

tion for each class in $\mathcal{M}_s$. We limit the total number of slots in $\mathcal{M}_s$ and $\mathcal{M}_w$ combined to 2000.

Retrieval: The retrieval module is tasked with making decisions for our method. It takes as input the current state of storage and the features of the region being considered, and makes a decision. For decision making, we draw inspiration from the intuitive item recognition model [54]. Given a region, we invoke the Semantic Memory $\mathcal{M}_s$ with classifier slot representations for each known (includes previously discovered) objects and use it to determine if the new region corresponds to a known class. As mentioned above, the association with a slot in $\mathcal{M}_s$ is evaluated using the classifier score. If so, we update the matching (μ_i^+, π_i^+) in $\mathcal{M}_s$ ('Update $\mathcal{M}_s$' in Figure 3). If the instance does not correspond to any known object, it likely implies that the region potentially contains a 'novel' object. The retrieval process continues by invoking the Working Memory $\mathcal{M}_w$ to check if this region corresponds to a slot in $\mathcal{M}_w$ or a new singleton slot needs to be created ('Update $\mathcal{M}_w$' or 'Create in $\mathcal{M}_w$' respectively, in Figure 3). This is done by computing the cosine similarity between the region's feature and centroids of all the slots in $\mathcal{M}_w$. If a slot in $\mathcal{M}_w$ has high similarity with the region, then that slot centroid is updated using a moving average. If the region cannot be associated with any of the slots in $\mathcal{M}_w$, a new slot is created.

Memory Consolidation: The Working Memory $\mathcal{M}_w$ is responsible for discovering new concepts, and after consistent/repeated occurrence of these concepts, they should be amalgamated with the Semantic memory $\mathcal{M}_s$. Towards this, we propose a memory consolidation step, where representations formed in the Working Memory are added to the Semantic Memory, extending our repertoire of known categories. To move slots from $\mathcal{M}_w$ to $\mathcal{M}_s$, we train an LDA classifier on the centroids in $\mathcal{M}_w$. A naïve consolidation step is to use these new classifiers as slots in $\mathcal{M}_s$. However, since the slots in $\mathcal{M}_w$ are essentially online learned clusters, we often encounter fragmented clusters for one concept. To merge multiple clusters of the same categories, we perform an iterative **merge step** formulated as a graph clustering problem, where each slot is a node and edge weights are based on how well classifiers fire on other slots. We obtain connected components using MinCut and proceed to merge the samples from slots in the same component. This step gives us the flexibility of re-assigning samples of one slot to another based on its relative affinity. Though less fragmented, the resulting slots after the merge operation can be incoherent and lead to semantic drift [19]. *E.g.*, samples from a smaller slot (say fire hydrant) can get merged with a larger slot (say bottle) due to their visual affinity to the bottle centroid. To ensure that slots consolidated into $\mathcal{M}_s$ are coherent, we perform a final **refine step**, where we remove all samples from the slot that do not activate its classifier (*i.e.*, $w^T x + b < 0$). After memory consolidation, $\mathcal{M}_w$ is

reinitialized to its original state (*i.e.*, null), and the number of slots increase in $\mathcal{M}_s$. Refer to the supplementary material for detailed merge, refine, and consolidation algorithms.

Discovery set: Given a dataset, we perform our discovery process in a sequential manner. This allows us to abstain from making assumptions about the availability of data all at once. To discourage the network from memorizing [56], and to leverage the fact that errors made by classifiers on different images will be different [30], we split the discovery dataset into two disjoint sets (*c.f.* [56]): $\mathcal{D}_1$ and $\mathcal{D}_2$. First, we run discovery and localization on $\mathcal{D}_1$ (discovery set) and fill the storage slots. Then, after memory consolidation, we use classifiers in $\mathcal{M}_s$ on $\mathcal{D}_2$ (validation set) to find more examples. The expectation is that the classifiers will find correct and diverse samples, but not detect error modes from the first set $\mathcal{D}_1$. After updating $\mathcal{M}_s$ using these samples, we swap $\mathcal{D}_1$ and $\mathcal{D}_2$, and continue the process.

5. Experimental Setup

5.1. Benchmark Datasets

Labeled and Discovery Datasets: Our benchmark builds on two datasets: (1) Pascal **VOC** 2007 [51], which has 10k images with annotations for 20 classes, and (2) **COCO** 2014 [57] which has ∼80k train and 5k validation images with annotations for 80 classes. Our benchmark assumes VOC is the *known* set, with 20 *known* classes, and COCO represents the in-the-wild real-world dataset which we use to benchmark concept discovery and localization. Out of the 80 COCO classes, 60 classes that do not overlap with VOC are treated as *novel* or *unknown* for evaluation. We use ImageNet [58] for pretraining, but do not use any class similar to the 60 *unknown* classes (discussed later).

Justification and Discussion. *Difficulty:* Even with 60 unknowns, this is an extremely challenging problem setup, and no other discovery benchmark matches its scale and difficulty; and no current approach can scale to this realistic setup. Another challenge is the different distribution of images – COCO images are more complex, have more objects per image, and there is a large portion of the discovery dataset which do not have any known class (distractor images). This is also a departure from standard discovery datasets, where all images contain at least one of the objects of interest, providing a strong signal. *Semantic drift:* Another benefit is that there are unknown categories (*e.g.*, bear, zebra, truck, laptop) available, which are visually quite similar to the 20 known categories (*e.g.*, dog, horse, car, tv, monitor) from VOC. A discovery and never-ending approach needs to successfully separate these classes to avoid semantic drift [19]. *Localization focus:* Most importantly, these are detection datasets with bounding box labels, which are suited for evaluating localization. *Unlabeled unknowns:* Note that an approach can certainly dis-

Table 1: **Large-scale Object discovery** on the **entire COCO train2014** (80k images). Comparisons with scalable clustering methods using **AuC** for *unknown* classes.

Method	AuC@0.5	AuC@0.2	#disc. objs
K-means	3.34	7.23	42
FINCH [55]	3.03	6.99	42
Ours	**3.60**	**9.11**	**46**

Table 2: **Smaller-scale Object discovery** on **subsets of COCO train2014** (2.5k/20k images). Comparison with contemporary discovery methods using **AuC** for *unknown* classes.

Method	#imgs.	CorLoc	CorRet	DetRate
OSD [2]	2.5k	6.62	80.00	4.73
OSD[†]	2.5k	6.34	70.00	5.17
Ours	2.5k	43.00	64.22	48.56
rOSD [1]	20k	15.77	100.0	11.56
Ours	20k	41.41	64.60	46.81

[†]: OSD with ResNet-101 Faster R-CNN proposals and classification-head features (same as **Ours**).

Figure 4: **Qualitative results (labeled *unknown*)** (entire COCO train2014): Concepts discovered by our approach which we can evaluate using the ground-truth annotations for the 60 'unknown' classes.

Figure 5: **Qualitative results (unlabeled *unknown*)** (entire COCO train2014): Concepts discovered by our approach which we cannot evaluate since they are unlabeled (not in the 60 'unknown' classes).

cover more than the available 60 unknowns objects. However, due to the lack of labels, we can only evaluate the quantitative performance of part of the discovered and localized objects and visualize other discovered unknowns.

Pretraining Dataset: Pretraining on ImageNet 1000 classes, a mainstay in object recognition [50, 59–63], creates an issue for any discovery setup – these 1000 classes might overlap with unknown classes being evaluated. Therefore, to avoid all ambiguity, we identify and eliminate 68 ImageNet classes similar to the 60 *novel* COCO classes; and refer to this split as ImageNet$^-$. For the complete list of classes removed from Imagenet for pretraining, training and performance details of the backbone and known detector (Faster-RCNN), we direct the readers to the supplementary material.

5.2. Evaluation Metrics

Evaluating concept discovery and localization is quite challenging. The two families of approaches discussed in Section 1 evaluate on different sets of metrics. For completeness, we report on all metrics used by related approaches, despite their shortcomings, and highlight some metrics that are useful for future works.

Co-localization/Recall Metrics: Several contemporary approaches [1, 3] used three co-segmentation/localization metrics to evaluate concept discovery. **CorLoc** (correct localization) [3] is defined as the percentage of images in which *any single object* is correctly localized with intersection-over-union (IoU)>0.5 with the ground-truth. **CorRet** (correct retrieval) [3] is defined as the mean percentage of k nearest neighbors (k=10), identified by retrieval for each image, that belong to the same (ground-truth) class as the image itself. **DetRate** (detection rate) [1] is the standard recall measure. There are important shortcomings of the metrics described above. CorLoc and DetRate measure the localization capabilities and *do not* reflect

the discovery performance. Such metrics are more suitable for tasks like co-segmentation/localization (for which they were originally proposed). Neither of these metrics measure the number of objects actually discovered by the algorithm. Moreover, CorLoc and CorRet assume the number of instances per image is 1, which is overly simplistic and does not hold for datasets like COCO (or even for most VOC images). Finally, DetRate is simply the recall of the region proposal network [50] and does not give any useful information about the discovery algorithm itself. We report these metrics for completeness, but they are not appropriate for our benchmark.

Discovery/Pattern-mining Metrics: Some approaches in object discovery [5, 6] and visual pattern mining [38, 56] explored forms of Purity-Coverage plots and/or mean Average Prevision (mAP) to evaluate their mining or discovery methods. Following [5, 6, 38], we argue that it is natural to evaluate concept discovery methods using such clustering-based metrics as opposed to co-segmentation metrics. We report Area-under-the-curve (**AuC**) of the Cumulative Purity *vs*. Coverage plots from [38]. Along with this, we also report the number of novel objects discovered to evaluate the efficacy of the discovery process. We contend that these metrics together offer a fair assessment across methods. Other plots (*e.g*., # of discovered objects *vs*. coverage, # of discovered objects/cumulative purity/coverage *vs*. # of clusters) offer ablative insights but have limited comparative benefit, and we provide them in the supplementary.

Details. To compute *cumulative purity*, we first compute purity of all the clusters and sort them in the descending order of their purity. Then, for the kth point on the curve, we compute the average purity of top-k clusters and plot it against coverage. We define *purity* of a cluster as $\max_{c \in \mathcal{C}} \frac{\text{\# of samples assigned to class } c}{\text{Total \# of samples in cluster}}$, where $\mathcal{C}$ is the set of all classes in COCO. A sample of a cluster is assigned to class

Table 3: **Detection performance (mAP)** for object detectors on COCO minival, trained using *oracle* labels for clusters. †: mAP of classes with AP greater than chance.

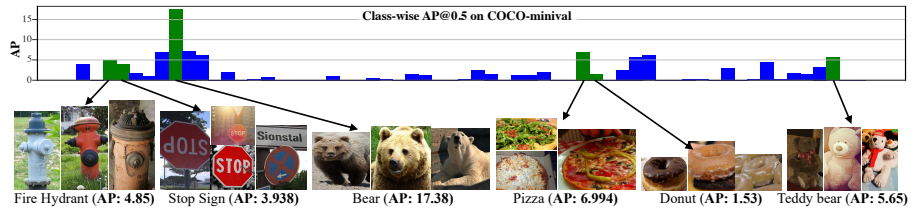| Classes (#)	GT-IoU: 0.5		GT-IoU: 0.2	
	AP@0.5	AP@0.2	AP@0.5	AP@0.2
All (80)	2.69	4.44	2.62	4.37
Novel (60)	1.87	3.50	1.76	3.42
Novel†	5.23	6.47	5.45	6.40

Figure 6: **Sample detections and class-wise AP on COCO minival** using our object detectors trained novel classes using *oracle* labels. Many more detection results are provided in the supplementary.

c if it has IoU $\geq$ threshold with ground-truth box (of class c) in the image. We define *coverage* as the fraction of ground-truth boxes covered by our method, *i.e.*, if there is at least one proposal with IoU $\geq$ threshold with the ground-truth.

***Oracle* Detection Metric:** We also evaluate our approach by training an object detector on the discovered clusters and evaluating it on an held-out set (COCO-minival). We assume the availability of an 'oracle,' in the form of ground-truth annotations to assign class labels to each cluster based on majority voting. We use different IoU thresholds for class assignment (0.2/0.5). After training a detector, all boxes rejected by all the classifiers are treated as background. We evaluate using AP at 0.2 and 0.5 IoU. We do not perform any bounding box regression. This provides proxy results, as if we used a human-annotator to provide a single label (out of 60 *unknown* classes) for each cluster. Since these are reproducible 'oracle' labels, we encourage future works to continue reporting this metric.

6. Results and Analysis

6.1. Baselines

Object discovery baselines: We compare our approach with two recent state-of-the-art object discovery methods: OSD [2] and rOSD [1]. **OSD** is an unsupervised method that solves an optimization problem to discover and match object classes among images in a collection. **rOSD** builds on [2] and is the first large-scale method to perform discovery of multiple objects.

Comparing our approach with other related methods such as [1–4] is not appropriate due several reasons: (1) they perform discovery on VOC, which is eight times smaller in scale compared to COCO, and (2) they do not assume knowledge of known classes. Unfortunately, adapting their official public code to scale to COCO was unsuccessful. [6, 44] is closest to our setting, but these methods also suffer from computational scalability and are unable to perform discovery on cluttered environments in COCO.

Clustering baselines: In addition, we compare with two clustering methods: K-means and FINCH [55]. **K-means** requires the number of clusters as input, which is hard to estimate; therefore, we use the number of discovered slots from our approach for K-means for fairness. **FINCH** [55] is a parameter-free clustering method that automatically discovers groupings in the data based on the first nearest neighbors. We pick the clustering that is closest to our discovered number of slots for fairness. Both K-means and FINCH do not scale well to large datasets like COCO with millions of regions. To circumvent this issue, we cluster 150 proposals in 10000 images and do a greedy label propagation to get cluster assignments for all the images [64].

6.2. Object Discovery Results

Large-scale Quantitative Evaluation: We report the **AuC** results (*unknown* classes) of our large-scale setup on the COCO2014 train set in Table 1, which uses all the 80k images. We recommend this setup for future comparisons. Since no other discovery approach can scale to these many boxes and images, we can only compare our method with K-means and FINCH. Both baselines perform worse compared to our approach, because similarity computation of visual features for clustering is not sufficient to form coherent clusters [53]. Low AuC values across approaches suggest there is tremendous scope for improvement in this domain. Our approach also discovers more objects than these two baselines (46 *vs*. 42), but there are fourteen categories that are are not discovered.

Smaller-scale Quantitative Evaluation: To compare with [2] (OSD), we modified their code (with some assumptions) to run on 2.5k images from COCO. We run our method on the same split and report the results in Table 2. Originally, OSD [2] used Randomized Prim proposals with whitened HoG features (OSD in Table 2). For fairer comparison, we replace their proposals and features with those used by our approach (ResNet-101+Faster R-CNN) and report the results as OSD† in Table 2. Our approach handily outperforms OSD on CorLoc and DetRate, whereas OSD performs better on CorRet. In fact, OSD with proposals and features from Faster R-CNN performs worse compared to original OSD, consistent with [2]. One explanation is that deep features are typically trained for image classification while [2] requires region matching. Moreover, since only few regions are localized by OSD, it results in fewer false positives, thereby resulting in a higher CorRet.

Next, we compare with rOSD [1], the first method to scale up to 20k images from COCO (we use the same split from [1]) and report the results in Table 2 (20k). We observe a similar pattern for each metric as earlier. We would like to reiterate that none of these metrics (CorLoc, CorRet, and DetRate) adequately measure the object discovery performance and are only provided for completeness.

Table 4: Ablation analysis for different initializations for $\mathcal{M}_s$ performed on mini-train set.

Init. ($\mathcal{M}_s$)	CorLoc	CorRet	AuC		#disc. objs
			@0.5	@0.2	
ϕ (Null)	40.21	61.31	2.31	8.00	27
Det. Scores	40.88	61.83	2.97	8.46	27
GT Overlap	40.96	61.77	2.77	8.65	30

Table 5: Memory consolidation component analysis performed on mini-train set.

Method	CorLoc	CorRet	AuC		#disc. objs
			@0.5	@0.2	
Naïve	42.57	63.12	3.70	9.60	44
+ Merge (M)	42.57	62.21	3.80	9.79	44
+ M + Refine	41.20	65.88	4.02	10.68	43

Table 6: Recall of VOC2007 detectors on the COCO2014 train set.

Classes (#)	ImageNet		ImageNet$^-$	
	@0.5	@0.2	@0.5	@0.2
All (80)	44.26	57.49	46.33	59.29
VOC (20)	71.26	80.07	71.67	80.47
Novel (60)	35.26	49.96	37.88	52.23

Qualitative Results: We show some discovered clusters in Figure 4 from the COCO train set which we evaluated in this section (60 labeled *unknown* in COCO). In Figure 5, we show some discovered clusters which we could not evaluate because they are unlabeled in COCO (unlabeled *unknown*). Many more examples are provided in the supplementary.

6.3. Object Detection Results

To demonstrate the performance of our approach on novel data, we evaluate detectors obtained from our approach on COCO-minival using the *Oracle* Detection Metric methodology described in §5.2. Note that this is a purposeful benefit of our VOC-COCO setup, where we can assume the availability of an 'oracle' for the 60 *unknown* classes and train a detector from our discovered clusters. In Table 3, we report results by assigning labels with different ground-truth IoU thresholds (0.2/0.5) and evaluating COCO AP at 0.2 and 0.5 IoU. We report AP results for all 80 COCO objects, all 60 novel objects, and for novel objects whose performance is greater than chance (Novel†). Novel† objects achieve >5 AP performance, which is reminiscent of early days of complex object detection.

Next, we show the per-class AP, for the model evaluated at 0.5 IoU for the discovered classes in Figure 6. We also visualize the detections on COCO-minival for few random categories. Evidently, the detectors display a lot of intraclass variation. We achieve the highest AP of $\sim 17.38\%$ for the bear class and a lowest mAP of 0.08% for traffic lights.

6.4. Ablation Analysis

Finally, we evaluate several design choices in our framework. For tractability, we perform all ablation studies on a randomly selected subset of 5000 training images from the COCO2014 train set. To ensure that results from the analysis translate to the entire dataset, we compare the distribution of objects in this mini-train and the entire train set in the supplementary.

Semantic Memory Initialization: To understand how the initialization of Semantic Memory $\mathcal{M}_s$ with features of known categories influences the discovery process, we run the discovery method without this initialization and report results in Table 4 (rows 1-2), which demonstrates the importance of prior knowledge. We use detection results to initialize $\mathcal{M}_s$ (see §4). Here, we discuss an alternate, higher-quality initialization, where we assume access to ground-truth annotations for the 20 known classes in COCO (which might not be available in most scenarios). For each known category, we compute the feature centroid of all boxes with

IoU > 0.5 with ground-truth. Results in Table 4 (row 3) shows that this helps discover more objects.

Memory Consolidation Component Analysis: We analyze the contribution of different components of our memory consolidation step in Table 5. For evaluation, all clusters with regions from less than 5 images are dropped. The merge step improves the AuC for *unknown* classes because it reassigns cluster memberships based on affinity scores. The refine step further improves the purity of a cluster by dropping all incoherent samples, which increases AuC and CorRet, but drops CorLoc slightly.

Recall Analysis: Since most discovery approaches, including ours, rely on region proposals, we explore how good are VOC-trained proposal generators for unseen 'novel' classes. We compute the recall of these proposals at IoU thresholds of 0.2 and 0.5 on the entire COCO2014 train split and report results in Table 6 for all classes ('All'), for 20 known VOC classes ('VOC'), and for 60 unknown classes ('Novel'). As we can see, the recall for unknown classes significantly lags behind known classes. This gives us a performance upper-bound for all discovery methods discussed in this work. This also strongly suggests that future research in better proposal generators is needed that can generalize to unseen categories.

Discovery Rounds: Since our approach can run in a never-ending fashion, we assess how many rounds are needed for convergence and how the performance changes across iterations. These results are reported in the supplementary.

Conclusion and Future Work. We presented a dual memory formulation, which can exploit prior knowledge about known objects to discover and localize novel categories in-the-wild. We perform extensive experiments to validate our claims. However, as the raw numbers indicate, there is a lot of scope for improving current object discovery and localization methods, especially for complex scenes and realistic benchmarks. One immediate future work is to adapt deep neural network-based region proposal methods to generalize beyond their training datasets and seen classes. Another exciting future direction is a new paradigm of supervised recognition, which is more suited for adaptation to in-the-wild discovery setups.

References

[1] Huy V. Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[2] Huy V. Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez, and Jean Ponce. Unsupervised image matching and object discovery as optimization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8287–8296, 2019.

[3] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised Object Discovery and Localization in the Wild: Part-based Matching with Bottom-up Region Proposals. In *CVPR - IEEE Conference on Computer Vision & Pattern Recognition*, pages 1201–1210, Boston, United States, June 2015. IEEE.

[4] Suha Kwak, Minsu Cho, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Unsupervised Object Discovery and Tracking in Video Collections. In *ICCV - IEEE International Conference on Computer Vision*, pages 3173–3181, Santiago, Chile, Dec. 2015. IEEE.

[5] Y. J. Lee and K. Grauman. Object-graphs for context-aware category discovery. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2010.

[6] Yong Jae Lee and Kristen Grauman. Learning the easy things first: Self-paced visual category discovery. In *CVPR*, pages 1721–1728. IEEE, 2011.

[7] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 370–377 Vol. 1, Oct 2005.

[8] Edward L. Bennett, Marian C. Diamond, David Krech, and Mark R. Rosenzweig. Chemical and anatomical plasticity of brain. *Science*, 146(3644):610–619, 1964.

[9] Angeline S Lillard and Alev Erisir. Old dogs learning new tricks: Neuroplasticity beyond the juvenile period. *Developmental review : DR*, 31 4:207–239, 2011.

[10] Scott P. Johnson, Dima Amso, and Jonathan A. Slemmer. Development of object concepts in infancy: Evidence for early learning in an eye-tracking paradigm. *Proceedings of the National Academy of Sciences*, 100(18):10568–10573, 2003.

[11] Gibson J J. *"The theory of affordances," in Perceiving, Acting, and Knowing. Towards an Ecological Psychology.* Number eds Shaw R., Bransford J. Hoboken, NJ: John Wiley & Sons Inc., 1977.

[12] Scott P. Johnson. How infants learn about the visual world. *Cognitive science*, 34 7:1158–1184, 2010.

[13] Graham Nuthall. The role of memory in the acquisition and retention of knowledge in science and social studies units. *Cognition and Instruction*, 18(1):83–139, 2000.

[14] Katherine Nelson. Language in cognitive development: Emergence of the mediated mind. *Mind*, Culture:76–79, 01 1998.

[15] Xiaodan Liang, Si Liu, Yunchao Wei, Luoqi Liu, Liang Lin, and Shuicheng Yan. Towards computational baby learning: A weakly-supervised approach for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[16] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[17] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, 2018.

[18] R.C. Atkinson and R.M. Shiffrin. Human memory: A proposed system and its control processes. volume 2 of *Psychology of Learning and Motivation*, pages 89–195. Academic Press, 1968.

[19] Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *European Conference on Computer Vision*, 2012.

[20] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Neil: Extracting visual knowledge from web data. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2013.

[21] Santosh K. Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, page 3270–3277, USA, 2014. IEEE Computer Society.

[22] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5356–5364, 2019.

[23] E. Belouadah and A. Popescu. Il2m: Class incremental learning with dual memory. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 583–592, Oct 2019.

[24] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5533–5542, 2017.

[25] Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, pages 241–257, 2018.

[26] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[27] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3420–3429, 2017.

[28] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[29] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc., 2014.

[30] Ishan Misra, Abhinav Shrivastava, and Martial Hebert. Watch and learn: Semi-supervised learning of object detectors from videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[31] Ilija Radosavovic, Piotr Dollár, Ross B. Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4119–4128, 2018.

[32] Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[33] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Deep image category discovery using a transferred similarity function. *CoRR*, abs/1612.01253, 2016.

[34] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[35] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[36] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[37] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *CVPR*, 2019.

[38] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Context as supervisory signal: Discovering objects with predictable context. In *European Conference on Computer Vision (ECCV)*, pages 362–377. Springer, 2014.

[39] Hongwen Kang, Martial Hebert, and Takeo Kanade. Discovering object instances from scenes of daily living. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, page 762–769, USA, 2011. IEEE Computer Society.

[40] Aljoša Ošep, Paul Voigtlaender, Jonathon Luiten, Stefan Breuers, and Bastian Leibe. Large-scale object mining for object discovery from unlabeled video. *ICRA*, 2019.

[41] Amirreza Shaban, Amir Rahimi, Shray Bansal, Stephen Gould, B. Boots, and R. Hartley. Learning to find common objects across few image collections. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5116–5125, 2019.

[42] Z. Wang and J. Yuan. Simultaneously discovering and local-

[43] Christopher Xie, Yu Xiang, Zaid Harchaoui, and Dieter Fox. Object discovery in videos as foreground motion clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[44] C. Galleguillos, B. McFee, S. Belongie, and G. Lanckriet. From region similarity to category discovery. In *CVPR 2011*, pages 2665–2672, June 2011.

[45] A. Karpathy, S. Miller, and Fei-Fei Li. Object discovery in 3d scenes via shape analysis. In *International Conference on Robotics and Automation (ICRA)*, 2013.

[46] Alvaro Collet Romea, Bo Xiong, Corina Gurau, Martial Hebert, and Siddhartha Srinivasa. Herbdisc: Towards lifelong robotic object discovery. In *Proceedings of International Journal of Robotics Research (IJRR)*, January 2014.

[47] Carolina Galleguillos, Brian McFee, and Gert R. G. Lanckriet. Iterative category discovery via multiple kernel metric learning. *International Journal of Computer Vision*, 108(1-2):115–132, 2014.

[48] R. Arandjelović and A. Zisserman. Object discovery with a copy-pasting GAN. *CoRR*, abs/1905.11369, 2019.

[49] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Enriching Visual Knowledge Bases via Object Discovery and Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[52] Bharath Hariharan, Jitendra Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012.

[53] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. "data-driven visual similarity for cross-domain image matching". *"ACM Transaction of Graphics (TOG) (Proceedings of ACM SIGGRAPH ASIA)"*, 30(6), 2011.

[54] Roger Ratcliff. A theory of memory retrieval. *PSYCHOL. REV*, 85(2):59–108, 1978.

[55] Vivek Sharma M. Saquib Sarfraz and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2019.

[56] Saurabh Singh, Abhinav Gupta, and Alexei A. Efros. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision*, 2012.

[57] 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 David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–

755, Cham, 2014. Springer International Publishing.

[58] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, San-jeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[59] Ross Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society.

[60] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, Oct 2017.

[61] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[62] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.

[63] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):318–327, 2020.

[64] Rob Fergus, Yair Weiss, and Antonio Torralba. Semi-supervised learning in gigantic image collections. In *NIPS*, volume 1, page 2. Citeseer, 2009.