Large-Scale Unsupervised Object Discovery

Huy V. Vo\textsuperscript{1,2} Elena Sizikova\textsuperscript{3} Cordelia Schmid\textsuperscript{1} Patrick Pérez\textsuperscript{2} Jean Ponce\textsuperscript{1,3}

\textsuperscript{1}INRIA, Département d’informatique de l’ENS, ENS, CNRS, PSL University, Paris, France
\textsuperscript{2}Valeo.ai
\textsuperscript{3}Center for Data Science, New York University
\{van-huy.vo, cordelia.schmid, jean.ponce\}@inria.fr
en5223@nyu.edu patrick.perez@valeo.com

Abstract

Existing approaches to unsupervised object discovery (UOD) do not scale up to large datasets without approximations that compromise their performance. We propose a novel formulation of UOD as a ranking problem, amenable to the arsenal of distributed methods available for eigenvalue problems and link analysis. Through the use of self-supervised features, we also demonstrate the first effective fully unsupervised pipeline for UOD. Extensive experiments on COCO \cite{lin2014microsoft} and OpenImages \cite{gupta2018openimages} show that, in the single-object discovery setting where a single prominent object is sought in each image, the proposed LOD (Large-scale Object Discovery) approach is on par with, or better than the state of the art for medium-scale datasets (up to 120K images), and over 37\% better than the only other algorithms capable of scaling up to 1.7M images. In the multi-object discovery setting where multiple objects are sought in each image, the proposed LOD is over 14\% better in average precision (AP) than all other methods for datasets ranging from 20K to 1.7M images. Using self-supervised features, we also show that the proposed method obtains state-of-the-art UOD performance on OpenImages.\footnote{Our code is publicly available at \url{https://github.com/huyvvo/LOD}}

![Figure 1: Sample UOD results obtained by LOD on the OpenImages dataset \cite{gupta2018openimages} which contains 1.7M images. Ground-truth boxes are shown in yellow, and predictions are in red. Best viewed in color.]

1 Introduction

This paper addresses the problem of identifying prominent objects in large image collections without manual annotations, a process known as unsupervised object discovery (UOD). Early approaches to UOD focused mostly on finding clusters of images featuring objects of the same category \cite{he2016unsupervised,argent2017unsupervised,arandjelovic2017objects,ross2017revisiting,li2020unsupervised}. Some of them \cite{argent2017unsupervised,arandjelovic2017objects,ross2017revisiting} also output object locations in images, but their evaluations are limited to small datasets with distinctive object classes. More recent techniques \cite{wu2019unsupervised,liu2020unsupervised,kim2021unsupervised} focus on the discovery of image links and individual object locations within much more diverse image collections. They typically rely on combinatorial optimization to select objects, or rather, object bounding boxes, among thousands of candidate region proposals \cite{singh2018unsupervised,yang2019unsupervised,kim2021unsupervised,chen2020unsupervised} given similarity scores computed for pairs of proposals associated with different images. Although these techniques achieve promising results, their computational cost and inherently sequential nature limit the size of the dataset they can be applied to. Attempts to scale up the state-of-the-art approach \cite{kim2021unsupervised} by reducing the search space size have revealed that this compromises its ability to discover multiple objects in

35th Conference on Neural Information Processing Systems (NeurIPS 2021).
each image. Other approaches to UOD focus on learning image representations by decomposing images into objects \cite{5, 12, 23, 43, 47}. These techniques do not scale up (yet) to large natural image collections, and focus mostly on small datasets containing simple shapes in constrained environments.

Table 1: Large-scale object discovery performance and comparison to the state of the art on COCO \cite{42} (C120K), OpenImages \cite{35} (Op1.7M) and their respective subsets C20K and Op50K, in three standard metrics. Using VGG16 features \cite{60}, the proposed method LOD achieves top performance in both single and multi-object discovery, and scales better to 1.7M images in Op1.7M than the previous state of the art \cite{67}. When running with self-supervised features (LOD + Self \cite{18}), it yields the best results on Op1.7M, showing the first effective fully unsupervised pipeline for UOD. See Sec. 4 for more details.

| Method             | Single-object | Multi-object |
|--------------------|---------------|--------------|
|                   | CorLoc        |              |
| EB \cite{83}       | 28.8          | 32.7         |
| Wei \cite{71}      | 38.2          | 34.8         |
| Kim \cite{32}      | 35.1          | -            |
| Vo \cite{62}       | 48.5          | 47.8         |
| Ours (LOD+Self \cite{18}) | 41.1          | 49.4         |
| Ours (LOD)         | 48.5          | 47.7         |

It is natural to cast unsupervised object discovery (UOD) as the task of finding repetitive visual patterns in image collections. Recent approaches \cite{66, 67} to UOD formulate it as a combinatorial optimization problem in a graph of images, selecting simultaneously image pairs that contain similar objects and region proposals that correspond to objects, with the corresponding computational limitations. The motivation behind our work is to formulate UOD as a simpler graph-theoretical problem with a more efficient solution, where objects correspond to well-connected nodes in a graph whose nodes are region proposals (instead of images in \cite{66, 67}), and edges are weighted by region similarity and objectness. In this scenario, finding object-proposal nodes is now a ranking problem where the goal is to rank the nodes based on how well they are connected in the graph. From another perspective, ranking is rather a natural modelization choice for UOD as in our context, discovering objects means finding the most object-like regions in a set of initial region proposals which naturally amounts to ranking them according to their “objectness”. As a result, a large array of methods available for eigenvalue problems and link analysis \cite{48} can be applied to solve UOD on much larger datasets than previously possible (Fig. 1). We consider three variants of this approach: the first one re-defines the UOD objective \cite{66, 67} as an eigenvalue problem on the graph of region proposals, the second variant explores the applicability of PageRank \cite{4, 48} for UOD, and the final variant combines the other two into a hybrid algorithm, dubbed LOD (for large-scale object discovery), which uses the solution of the eigenvalue problem to personalize PageRank. LOD offers a fast, distributed solution to object discovery on very large datasets. We show in Sec. 4.1 and Table 1 that its performance is comparable or better than the state of the art in the single object discovery setting for datasets of up to 120K images, and over 37% better than the only algorithms we are aware of that can handle up to 1.7M images. In the multi-object discovery setting, LOD significantly outperforms all existing techniques on datasets from 20K to 1.7M images. While LOD does not explicitly address discovering relationships between images \cite{4, 48}, we demonstrate that categories can be discovered as a post-processing step (see Sec. 4.2). The best performing approaches to UOD so far all use supervised region proposals and/or features. We also demonstrate for the first time in Sec. 4.1 that self-supervised features can give good UOD performance. Our main contributions can be summarized as follows:

- We propose a new formulation of UOD as a ranking problem, allowing the application of parallel and distributed link analysis methods \cite{4, 48}.
- We scale UOD up to datasets 87 times larger than those considered in the previous state of the art \cite{67}. Our novel LOD algorithm outperforms others on medium-size datasets by up to 32%.
- We propose to use self-supervised features for UOD and show that LOD, combined with these features, offers a viable UOD pipeline without any supervision whatsoever.
- We conduct extensive experiments on the COCO \cite{42} and OpenImages \cite{35} datasets to empirically validate our method. We also demonstrate applications of our approach to object category discovery and retrieval, outperforming other existing unsupervised baselines on both tasks by a large margin.
2 Problem statement and related work

2.1 Problem statement

Consider a collection of \( n \) images, each equipped with a set of region proposals [66, 67]. For the sake of simplicity, we assume in this presentation that all images have exactly \( r \) region proposals. We wish to find which ones of these correspond to objects, and link images that contain similar objects, without any information other than how similar pairs of proposals are. This problem is known as unsupervised object discovery (UOD) and can be formulated as an optimization problem [67] over a graph where images are represented as nodes. Let \( e_{pq} \in \{0, 1\} \) for \( p, q = 1, 2, \ldots, n \) be a set of binary variables indicating if two images are connected in the graph, with \( e_{pq} = 1 \) when images \( p \) and \( q \) share similar visual content. Similarly, let \( x^k_p \in \{0, 1\} \) for \( p = 1, 2, \ldots, n \) and \( k = 1, 2, \ldots, r \) be indicator variables such that \( x^k_p = 1 \) when region proposal \( k \) of image \( p \) is an object-like region in image \( p \) that is similar to an object-like region in one of the neighbors of image \( p \). Let also \( x_p \) be \((x^1_p, \ldots, x^r_p)^T \), \( x \) be the \( n \times r \) matrix whose rows are \( x_p \) for \( p = 1, 2, \ldots, n \) and \( e \) be the binary adjacency matrix of the image graph. Then, the object discovery problem can be formulated as a combinatorial maximization problem:

\[
\max_{x,e} \sum_{p=1}^{n} \sum_{q \in \mathcal{N}(p)} e_{pq} x^T_p S_{pq} x_q \quad \text{s.t.} \quad \sum_{k=1}^{r} x^k_p \leq \nu \quad \text{and} \quad \sum_{q \neq p} e_{pq} \leq \tau \quad \forall \, 1 \leq p \leq n,
\]

where \( S_{pq} \in \mathbb{R}^{r \times r} \) is a matrix whose entry \( S_{pq}^{k\ell} \geq 0 \) measures the similarity between region \( k \) of image \( p \) and region \( \ell \) of image \( q \) as well as the saliency of the respective regions, \( \mathcal{N}(p) \) is a set of potential high-similarity neighbors of image \( p \), and \( \nu \) and \( \tau \) are predefined constants corresponding to the maximum number of objects in an image and the maximum number of its neighbors, respectively. Previous approaches [66, 67] to UOD solve a convex relaxation of (C) in the dual domain and/or use block-coordinate ascent on its variables \( x \) and \( e \). The similarity scores \( S_{pq}^{k\ell} \) are typically computed using the Probabilistic Hough Matching (PHM) algorithm from [8], which combines local appearance and global geometric consistency constraints to compare pairs of regions. A high PHM score between a pair of proposals is an indicator of whether the corresponding two proposals may correspond to a common foreground object. We follow this tradition and also use PHM scores (Sec. 4).

The objective of UOD as formulated in (C) is to find both the objects (variables \( x^k_p \)) and the edges linking the images that contain them (variables \( e_{pq} \)). Its combinatorial nature makes it hard to scale up to large values of \( n \) and \( r \) [67] uses a block-coordinate ascent algorithm to (C), updating variables \( x \) and \( e \) alternatively to optimize the objective. It attempts to scale up (C) with a drastic approximation, running on parts of the image collection to reduce \( r \) to only 50 before running on the entire dataset. However, using significantly reduced sets of region proposals hinders its ability to discover multiple objects (Table 1). Moreover, this algorithm is inherently sequential. According to [66], in each iteration of optimizing \( x \), an index \( i \) is chosen and \( x_i \) is updated while \( e \) and all \( x_j \) with \( j \neq i \) are kept fixed. The update of \( x_i \) depends on the updated values of other \( x_j \) if \( x_j \) is updated before \( x_i \). This is crucial to guarantee that the objective always increases. If all \( x_i \) are updated in parallel, there is no guarantee that the objective would increase. Consequently, this process is not parallelizable, preventing the algorithm from scaling up to datasets with millions of images. We therefore drop the second objective of UOD, and rely only on a fully connected, weighted graph of proposals where edge weights encode proposals’ similarity (edge weights can be zeros, see Sec. 3). In turn, we can reformulate UOD as a ranking problem [4, 30, 34, 56, 51], amenable to the panoply of large-scale distributed tools available for eigenvalue problems and link analysis. We consider two different ranking formulations: the first (C) tackles a quadratic optimization problem, and the second (D) is based on the well-known PageRank algorithm [4, 48]. We combine these two approaches into a joint formulation (LOD) that gives the best results on large-scale datasets. See Sections 3 and 4 for details.

2.2 Related work

Unsupervised object discovery. Early works on unsupervised object discovery focused on finding groups of images depicting objects of the same categories, employing probabilistic models [58, 61, 69], non-negative matrix factorization (NMF) [62] or clustering techniques [22], see [64] for a survey. In addition to finding image groups, some of these approaches, e.g., multiple instance learning (MIL) [82], graph mining [75], contour matching [39] and topic modeling [58, 61], also output object locations, but focus on smaller datasets with only a handful of distinctive object classes. Saliency detection [80] is related to UOD, but seeks to generate a per-pixel map of saliency scores, while UOD attempts to find bounding boxes around objects in each image without supervision. Object discovery
Weakly-supervised object localization and image co-localization. These problems are related to UOD but take advantage of image-level labels. Weakly-supervised object localization (WSOL) considers scenarios where the input dataset contains image-level labels [9]. Most recent WSOL methods localize objects using saliency maps obtained from convolutional features in a neural network [7, 59, 81]. Since networks tend to learn category-discriminating features, various strategies for improving the quality of features for localization have been proposed [2, 10, 76, 29]. Co-localization is another line of work where all input images are assumed to contain at least one instance of a single object category. [63] uses discriminative clustering for co-localization in noisy image sets and [29] extends this work to the video setting. [41] learns to co-localize objects by learning sparse confidence distributions, mimicking behaviors of supervised detectors. [70] and [71] observe that activation maps generated from CNN features contain information about object locations. They propose to cluster locations in the images into background and foreground using PCA and return the tight bounding box around the foreground pixels as an object. Since [71] can deal with large-scale datasets and can be easily adapted to UOD, we use it as a baseline in our experiments.

Ranking applications in computer vision. The goal of ranking is to assign a global importance rating to each item in a set according to some criterion [48]. Many computer vision problems admit a ranking formulation, including image retrieval [6], object tracking [3], person re-identification [44], video summarization [72], co-segmentation [52] and saliency detection [40]. Several techniques specifically designed for large-scale ranking problems [34, 48] have been used to explore large datasets of images [27] and shapes [16]. PageRank-based approaches in particular have been popular [27, 49, 56] due to their scalability. [31] proposed an algorithm for object discovery that combines appearance and geometric consistency with PageRank-based link analysis for category discovery. However, it does not scale beyond 600 images. A more scalable follow-up work by [32] discovers regions of interest (RoIs) from images with successive applications of PageRank. This algorithm includes two main steps. The first one attempts to find object representatives (hubs) from the current RoIs of all images using PageRank. PageRank is then utilized again in the second step to analyze the links between regions in each image and the hubs, this time to update the RoIs of the images. Finally, good RoIs are found by repeating these two steps until convergence. We compare our method to this technique in Sec. 4.1.

3 Proposed approach

3.1 Quadratic formulation

Let us represent region proposals by a graph $G$ with $N = nr$ nodes, where $n$ is the number of images and $r$ is the number of proposals in each image. Each node corresponds to a proposal, and any two nodes $(p, k)$ and $(q, \ell)$, corresponding to proposals $k$ and $\ell$ of images $p$ and $q$, respectively, are linked by an edge with weight $S'_{p,q}$. $G$ is represented by an $N \times N$ symmetric adjacency matrix $W$, consisting of $r \times r$ blocks $S_{pq}$ for $p, q = 1, \ldots, n$; $S_{pq}$ is defined in Sec. 2.1 and is computed via PHM algorithm [8] if $p \neq q$, and the diagonal blocks are taken to be zero since only inter-image region similarity matters in our setting. Let $y = (y_1, \ldots, y_N)^T$. Define the support of node $i$ given $y$ as $z_y(i) = \sum_j W_{ij} y_j$ so that taking $z_y = (z_y(1), \ldots, z_y(N))^T$ we have $z_y = W y$. Intuitively, given $y$, $z_y(i)$ quantifies how well $i$ is connected to (or “supported by”) the rest of the nodes $j$ in the graph,
taking into account the similarity $W_{ij}$ between $i$ and $j$ as well as the importance $y_j$ of that node. We would like to find the importance scores that rank the nodes as well as possible, so that the order corresponds to their amount of support. As shown by the following lemma, it turns out that this “chicken-and-egg” problem admits a simple solution.

**Lemma 1.** Suppose $W$ is irreducible (i.e., represents a strongly connected graph $G$). The solution $y^*$ of the quadratic optimization problem:

$$y^* = \arg\max_{\|t\| \leq 1, t \geq 0} t^T W t \quad (Q)$$

is the unique unit, non-negative eigenvector of $W$ associated with its largest eigenvalue.

This is a classic result and can be proved using Perron-Frobenius theorem [15, 50]. We include the complete proof in the supplemental material. In our context, $W$ is not, in general, irreducible (i.e., for all pairs $(i, j)$, there exists $m \geq 1$ such that $W^m(i, j) > 0$) since some proposal similarities may be zero. Reminiscent of PageRank [48], we add a small term $\gamma e e^T / N$ to $W$, with $e$ being the vector with all entries equal to 1 in $\mathbb{R}^N$ and $\gamma = 10^{-4}$, deliberately chosen so that the added term does not influence the similarity score much, to make $W$ irreducible. This term ensures that the resulting ranking is unique and serves the same purpose as the similar term in PageRank. Note: since $y^*$ is associated with $W$’s largest eigenvalue $\lambda^*$, which is positive according to the Perron-Frobenius theorem, we have $\lambda^* y^* = W y^* = z y^*$. Hence, the importance score $y^*_i$ of each node is, up to a positive constant, equal to its support, and can thus be used to rank the nodes as desired. Notice that (C) and (Q) are closely related problems when the graph of images in (C) is assumed to be complete. In this case, (C) can be written as $\max_{x \in \{0,1\}^N} x^T W x$, s.t., for all $p$ from 1 to $n$, $\sum_{k=1}^{m} x_{p(r+1)+k} \leq \nu$. Here, we stack $x_i$ ($i = 1, \ldots, n$) into a vector $x$. (Q) can thus be seen as a continuous relaxation of (C) where the binary variables are replaced by continuous ones, and the linear constraints attaching the proposals to their source images are dropped. The order induced by the dominant eigenvector $y^*$ of $W$ on the nodes of $G$ is reminiscent of the PageRank approach [4, 48] to link analysis. This remark leads to a second approach to UOD through ranking, discussed next.

### 3.2 PageRank formulation

When defining PageRank, [48] does not start from an optimization problem like (Q), but directly formulates ranking as an eigenvalue problem. Following [37], let $A$ denote the transition matrix of the graph associated with a Markov chain, such that $a_{ij} \geq 0$ is the probability of moving from node $j$ to node $i$. In our context, $A$ can be taken as $W D^{-1}$ where $D$ is the diagonal matrix with $D_{jj} = \sum_i W_{ij}$. By definition [4, 48], the PageRank vector $\nu$ of the matrix $A$ is the unique non-negative eigenvector $\nu$ of the matrix $P$, associated with its largest (unit) eigenvalue, where $P$ is defined as:

$$P = (1 - \beta) A + \beta u e^T, \quad (P)$$

where $\beta$ is a damping factor. Here, $u$, the so-called personalized vector, is an element of $\mathbb{R}^N$ such that $e^T u = 1$. As noted earlier, the second term ensures that $P$ is irreducible, so that, by the Perron-Frobenius theorem, the eigenvector $\nu \geq 0$ is unique [38]. The vector $u$ is typically taken equal to $e / N$, but can also be used to “personalize” the ranking by attaching more importance to certain nodes. This leads to the hybrid formulation proposed in the next section. (Q) and (P) are closely related, and the $\nu$ vector can also be seen as the solution of a quadratic optimization problem [45]. Besides this formal similarity, the goals of the two formulations are also similar. Quoting [48], “a page has a high rank (according to PageRank) if the sum of the ranks of its backlinks is high”. The solution of both (Q) and (P), as an eigenvector associated with the largest eigenvalue, provides a ranking based on the support function and can be found with the power iteration algorithm [49]. This algorithm involves only matrix-vector multiplications and can be implemented efficiently in a distributed way.

### 3.3 Using (Q) to personalize PageRank

The above discussion suggests combining the two approaches. We thus propose to use the maximizer of (Q) to generate the personalized vector for (P). (Q) and (P) are two different optimization problems for ranking region proposals, and combining them may help improve the final performance. Intuitively, region proposals with high scores given by (Q) are reliable and we should be able to rank the “objectness” of other regions more accurately based on the “feedback” of these top-scoring proposals. We compute the personalized vector from the solution of (Q) as follows. Given a factor $\alpha$, the top region in each image is chosen as candidates, then the top $\alpha$ percent of regions amongst these candidates are selected. Since only regions that have a high probability of being correct are beneficial, we choose $\alpha$ sufficiently small (see Sec. 4 to select only the most likely correct regions. Given the set
of selected regions, the personalized vector \( u \) is the \( L_1 \)-normalized indicator vector with \( u_i = 1/K \) where \( K \) is the total number of selected regions if proposal \( i \) is selected and \( u_i = 0 \) otherwise. We set the initialization \( v_0 \) of the power iteration algorithm to \( u \) to further bias \( (P) \) toward reliable regions found by \( (Q) \). In what follows, we refer to this hybrid algorithm as Large-Scale Object Discovery (LOD).

4 Experimental analysis

Datasets. We consider two large public datasets: C120K, a combination of all images in the training and validation sets of the COCO 2014 dataset [42], except those contain only “crowd” objects, with approximately 120,000 images depicting 80 object classes and OpenImages (Op1.7M) [35], the largest dataset ever evaluated for UOD so far, with 1.7 million images. The latter dataset is 87 times the size of the previous largest dataset evaluated by the state-of-the-art UOD method [67]. We resize all images in this dataset so that their largest side does not exceed 512 pixels. To facilitate ablation studies and comparisons, we also evaluate our methods on C20K, a subset of C120K containing 19,817 images used by [67] and Op50K, a subset of Op1.7M containing 50,000 images.

Implementation details. We use the proposal generation method of [67] since it gives the best object discovery performance among the unsupervised region proposal extraction methods [67]. We use VGG16 [60], trained with and without image class labels (Sec. 4.1) on the ImageNet [11] dataset, to both generate (with [67]) and represent (extracting with RoiPool [20]) proposals. We have also experimented with VGG19 [60] and ResNet101 [25], but found they give worse performance, possibly because they are more discriminative and less helpful in localizing entire objects. We compute the similarity score between proposals with the PHM algorithm [8] similar to prior work [8, 66, 67]. For large datasets, computing all score matrices \( S_{pq} \) is intractable. In this case, we only compute the similarity scores for the 100 nearest neighbors of each image, computed based on the Euclidean distance between image features from the fc6 layer. For optimization, we choose \( \beta = 10^{-4} \) in \( (P) \) and \( \alpha = 10\% \) in LOD. To select objects from ranked proposals in an image, we choose proposal \( i \) as an object if it has the highest score in the image or the intersection over union (IoU) between \( i \) and each of the previously selected object regions is at most 0.3. When using proposals from [67], which are divided into disjoint groups, we additionally impose that the newly chosen region must be in a group different from the groups of the previously selected objects. See supplemental material for discussions on LOD’s sensitivity to hyper-parameters and more implementation details.

Metrics and evaluation settings. We consider two settings: single- and the multi-object discovery. In the single-object setting, we return \( m = 1 \) region per image, which is the region most likely to be an object. In the multi-object setting, we return up to \( M \) regions per image, where \( M \) is the maximum number of objects in any image in the dataset. Following [43], we assume \( M \) is known during evaluation. In a real application, one could use a rough “budget estimate” of the upper bound on how many objects per image one may try to detect. Measuring performance of UOD is always a difficult task due to the ambiguity of the notion of an object in an unsupervised setting: object parts vs. objects, individual objects vs. crowd objects, etc. Following previous works [8, 66, 67], we consider the annotated bounding boxes in the tested datasets as the only correct objects and use them to evaluate our methods. We evaluate UOD results according to the following metrics:

1. Correct localization score (CorLoc) – percentage of images correctly localized, i.e., where the IoU score between one of the ground-truth regions and the top predicted region is at least \( \sigma = 0.5 \). Note that it is equivalent to precision of returned regions. This metric is commonly used to evaluate single-object discovery [8, 66, 67].

2. Average Precision (AP) – the area under the precision-recall curve with precision and recall computed at each value of \( m \) from 1 to \( M \). A ground-truth object is considered discovered if its intersection with any predicted region is at least \( \sigma \). This metric is used to evaluate multi-object discovery. We report AP50 where \( \sigma = 50 \) and AP@[50:95], where we average AP at 10 equally spaced values of \( \sigma \) from 0.5 to 0.95. Note that AP is different from [67]’s metrics for multi-object discovery, which is the object recall (detection rate) at a predefined value of \( m \). This metric depends on the number of selected regions per image \( m \) while our metrics do not. In contrast, AP is a standard metric in object detection-like tasks [19, 21, 24, 32, 55, 57]. Note that since the precision decreases significantly with increasing \( m \), AP appears much smaller than CorLoc.

4.1 Large-scale object discovery

In this section, we compare our methods to the state of the art in UOD [32, 67, 71]. We also compare to Edgeboxes (EB) [85], an unsupervised method which outputs regions with an importance score.
Actual run

Table 1: Since state-of-the-art approaches to UOD report results using supervised features [60], we have used these features as well in our comparisons. We additionally report LOD’s performance with self-supervised features [18] on these datasets. Overall, LOD obtains state-of-the-art object discovery performance in all settings and datasets. Using VGG16 features [60], it outperforms Kim [32], Wei [71], and EB [83] by large margins: by 26% in single-object discovery and by 14% in multi-object discovery settings. In comparison to Vo [67], LOD performs similarly in the single-object setting, but outperforms Vo [67] by at least 19% in the multi-object setting. This is likely due to the fact that our proposed LOD method considers the full proposal graph and does not reduce the number of region proposals (see supplementary material). It is also noteworthy that LOD scales better than [67] and runs much faster on the large datasets C120K and Op1.7M (Fig. 2). On the Op1.7M dataset, it takes 53.7 hours to run while Vo [67] needs more than a month to finish. It is also interesting that self-supervised features [18] works better with LOD than supervised ones [60], yielding the state-of-the-art performance on Op1.7M dataset.

Run time. Next, we compare scalability and run times of the proposed technique and of the baselines. All tested methods ([32, 67, 71, 83] and LOD) use similar pre-processing steps: feature extraction, proposal generation and similarity computation, which are done separately across all images. This is followed in [32, 67] and LOD by an optimization stage. The optimization step in [67] is inherently sequential, but [32] and LOD can be parallelized. In our experiments, we use 4,000 CPUs for preprocessing for all methods, and 48 CPUs for the optimization step in [32] and LOD. The maximum possible with the MatLab parallel toolbox used in our implementation. The timings in Fig. 2 include both pre-processing and optimization, when the latter is used. It can be seen that [32, 71, 83] and LOD scale nearly linearly with the number of images, while [67] exhibits a superlinear pattern. Note that [83] and [71] are 70 times faster than LOD, but at a significant decrease in performance. These methods are not initially designed for object discovery, but serve as good, scalable baselines. Compared to previous top UOD methods, LOD runs at least 2.8 times faster than [32] on all datasets, at least 2 times faster than [67] on datasets between 120K and 1.7 million images. Here, we evaluate only the parallel implementation typical for modern computing setups. In a serial implementation, compute times will be similar between top performing UOD methods [32, 67] and LOD, but none of the methods would be able to run on 1.7M images in reasonable time. Note also that additional computational resources can further speed up processing for both [32] and LOD.

Qualitative evaluation. We present sample qualitative multi-object discovery results of LOD on C120K and Op1.7M in Fig. 4 (additional Op1.7M results are presented in Fig. 1). LOD discovers both the larger objects (people in the first and sixth images, food items in the second and third images) and the smaller ones (tennis balls and racquet in the first image). It may fail of course, and two typical failure cases are shown on the right of Fig. 4. In the first case, objects are too small and in the second case, LOD returns object parts instead of entire objects. Note that there is some ambiguity in what parts of the image are labelled as ground truth objects. For example, the leaves in the bottom left image are not labelled as objects, while the flowers are.

Self-supervised features vs. supervised features. LOD and all of the optimization-based baselines [32, 67, 71] rely on a VGG [60]-based classifier trained on ImageNet [11]. In this section, we investigate their performance when the underlying classifier is trained with (Sup) and without (Self) image labels, i.e., in a self-supervised fashion. To obtain self-supervised features, we use a VGG16 model trained with OBoW [18], a recent method which yields state-of-the-art performance in

![Figure 2: Comparison of run time as a function of input images. LOD achieves significant improvement in performance and/or savings in run time compared to previous works. EB [83] and Wei [71] are linear in the number of images but their run time are very small compared to other methods and look flat in the figure.](image-url)
object detection after fine-tuning. This model is tested for both the proposal generation and similarity computation steps in optimization-based methods. The results of several variants of each optimization method, depending on the proposal generation algorithm (EB [83], [67]+Self or [67]+Sup) and the region proposal representation (Self or Sup) are presented in Table 2 (left). [67] generates proposals from local maxima of the image’s saliency map obtained with CNN features. To evaluate [67]+Self and [67]+Sup for UOD, we assign each proposal a score equal to the saliency of the local maximum it is generated from. If two regions have the same score, the larger one is ranked higher so that entire objects instead of object parts are selected. Finally, when EB [83] proposals are used for [67] and LOD, we multiply their features with their EB scores before computing their similarity.

In general, variants with supervised features perform better in UOD than those with self-supervised features, except for Wei [71] and LOD in single-object discovery on Op50K. Kim [32] is the most dependent on supervised features. Its performance drops by at least 63% when switching to self-supervised features. It is also noteworthy that the performance of Vo [67] and LOD with supervised and self-supervised features on Op50K are much closer than on C20K. This is likely due to the fact that the supervised features [60] are trained on the 1000 ImageNet object classes which contain all of the COCO classes and thus offer a stronger bias toward these classes than the self-supervised features. Using self-supervised features, variants of LOD are the best performer in both single-object discovery (with Vo [67]+Self proposals) and multi-object discovery (with EB proposals). They yield reasonable results on both datasets compared to variants with supervised features. In particular, self-supervised object proposals [67] and self-supervised features, combined with LOD, give the best results of all tested methods on Op50K in single-object discovery. These results show that LOD combined with self-supervised features is a viable option for UOD without any supervision whatsoever.

Comparing ranking formulations. We compare the UOD performance of Q, P and LOD with different proposals and features in Table 2 (right). It can be seen that LOD outperforms Q and P in almost all datasets and settings. These results confirm the merit of our proposed method, using Q’s solution to personalize PageRank.

4.2 Category discovery

Contrary to [8, 66, 67], our work aims specifically at localizing objects in images and omits the discovery of the image graph structure, i.e., identifying image pairs that contain objects of the same category. However, objects localized by our methods can be used to perform this task in a post-processing step. To this end, we define similarity between two images as the maximum similarity between pairs of selected proposals. Similarity is measured using cosine distance between features extracted from fc6 layer. We compare LOD to [8] in image neighbor retrieval task on VOC_all, a subset of Pascal VOC2007 dataset [13] used as a benchmark in [8]. Similar to these works, we retrieve 10 nearest neighbors per image. Then, CorRet [8] (the average % of retrieved image neighbors that are actual neighbors in the ground-truth image graph over all images) is used to compare different methods. Results are shown in Fig. 4. LOD outperforms [8] in almost all datasets and settings. These results confirm the merit of our proposed method, using Q’s solution to personalize PageRank.

Figure 3: Examples where our method (LOD) succeeds (left) and fails (right) to discover ground-truth objects in the Op1.7M dataset [35]. Ground-truth objects are in yellow, our predictions are in red. Best viewed in color.
Table 2: Left: UOD performance with supervised [60] (Sup) and self-supervised features [18] (Self) on C20K and Op50K datasets. Region proposals are generated by methods from EB [83] and Vo [67] with different types of features. LOD with self-supervised features yields reasonable results compared to supervised features. Variants of our proposed method LOD yield state-of-the-art performance in all settings. Right: A comparison of different ranking methods for UOD. LOD is better than [2] and [4] in most of the cases.

Table 3: Purity (↑) of our clustering method compared to the state of the art in category discovery on the SIVAL dataset [53]. Following prior work, we perform the task on a partition of the dataset and report the average purity on its parts as the final result. Results of other methods are from [75].

Conduct experiments on the SIVAL [53] dataset, a popular benchmark for this task. This dataset consists of 25 object categories, each containing about 60 images. Following [82], we partition the 25 object classes into 5 groups, named SIVAL1 to SIVAL5, and use purity (average percentage of the dominant class in the clusters) as an evaluation metric. Intuitively, purity measures the extent to which a cluster contains images of a single dominant class. A comparison between our method and other popular object category discovery methods is given in Table 2. It can be seen that our method outperforms the state of the art by a significant margin, attaining an average purity of 95.3. It is also noteworthy that the performance drops to 23.7 when the features of entire images are used instead of the representative top proposals. This finding shows that our performance gain is in great part due to the object localization performance of our method. Since individual images in the SIVAL dataset [53] contain only one object, we conduct a similar experiment on the more challenging VOC_all [13] dataset. In this experiment, a histogram is computed for each cluster, showing the score of each ground-truth object category (a category score is the sum of contributions of all its images). An image contribution is computed as \(1/n_c\), with \(c\) is the number of object categories appearing in the image and \(n\) is the number of images in the cluster. We then match the clusters to the ground-truth categories by solving a stable marriage problem with the Gale–Shapley algorithm [17] using the preference orders induced by the histograms.

The confusion matrix generated by combining these histograms, revealing the correspondence between the clusters and the classes, is shown in Fig. 4. Our method is able to discover 17 categories (which are dominant in at least one cluster) out of 20 ground-truth categories. As for the three undiscovered categories: sheep is dominated by similar class cow in cluster 10; sofa is dominated by co-occurring class chair in cluster 9; diningtable suffers from being often largely occluded in images. Interestingly, it seems that our method might be used to discover pairs of categories that often appear together. For instance: bicycle and person, horse and person, motorbike and person (clusters 2, 13 and 14 have two corresponding dominating classes each). Quantitatively, using the top

![Comparison to prior work on the image neighbor retrieval task (CorRet, ↑).](image-url)
extracted proposals from our method achieves a purity of 68.6 on this dataset, which is better than the purity of 61.8 obtained when features of entire images are used.

4.3 Discussions

Without a formal definition of objects, casting objects as frequently appearing salient visual patterns is natural. However, findings could be biased toward popular object classes and ignore rare classes in image collections that contain a long-tail distribution of object classes. To have an insight to this potential bias, we compute LOD’s performance by object category on C20K dataset. Surprisingly, we have observed little correlation between the performance on an object class and its appearance frequency (the corresponding correlation is only $-0.09$, see Figure 6). A possible explanation is that even though we rank all regions in the image collection at once, we choose objects (based on the ranking) on the image level. Therefore, regions can be selected as objects if they stand out more from the background and are better connected in the graph than other regions in the same image, even if they represent objects of a rare class.

5 Conclusion and future work

We have demonstrated a novel formulation of unsupervised object discovery (UOD) as a ranking problem, allowing application of efficient and distributed algorithms used for link analysis and ranking problems. In particular, we have shown how to apply the personalized PageRank algorithm to derive a solution, and proposed a new technique based on eigenvector computation to identify the personalized vector in Pagerank. The proposed LOD algorithm naturally admits a distributed implementation and allows us to scale up UOD to the OpenImages [35] dataset (Op1.7M) with 1.7M images, 87 larger than datasets considered in the previous state-of-the-art technique [67], and outperforms (in single- and multi-object discovery) all existing algorithms capable of scaling to this size. In multi-object discovery, LOD is better than all other methods on medium and large-scale datasets. State-of-the-art solutions to UOD rely on supervised region proposals [8] or features [71, 67], thus their output requires at least in part on some sort of supervision. We have proposed to combine LOD with self-supervised features, offering a solution to fully unsupervised object discovery. Finally, we have shown that LOD yields state-of-the-art results in category discovery which is obtained as a post-processing step. A limitation of our method is that it works well only with VGG-based features which prevents it from benefiting from more powerful features [25, 26]. Future work will be dedicated to investigating these features for LOD. Another interesting avenue for future research is to better discover smaller objects.

Effective solutions to UOD have the potential for a great impact on existing visual classification, detection, and interpretation technology by harnessing the vast amounts of non-annotated image data available on the Internet today. In turn, the application of this technology has its known potential benefits (from natural human computer interfaces to X-ray image screening in medicine) and risks (from state-sponsored surveillance or military target acquisition). We believe that such concerns are ubiquitous in machine learning in general and computer vision in particular, and beyond the scope of this scientific presentation.
Acknowledgments and Disclosure of Funding

This work was supported in part by the Inria/NYU collaboration, the Louis Vuitton/ENS chair on artificial intelligence and the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute). Elena Sizikova was supported by the Moore-Sloan Data Science Environment initiative (funded by the Alfred P. Sloan Foundation and the Gordon and Betty Moore Foundation) through the NYU Center for Data Science. Huy V. Vo was supported in part by a Valeo/Prairie CIFRE PhD Fellowship. We thank Spyros Gidaris for providing the VGG16-based OBoW model. Finally, we thank anonymous reviewers for their helpful suggestions and feedback for the paper.
References

[1] Francis Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 2013.

[2] Kyungjune Baek, Minhyun Lee, and Hyunjung Shim. Psynet: Self-supervised approach to object localization using point symmetric transformation. In *AAAI*, 2020.

[3] Yancheng Bai and Ming Tang. Robust tracking via weakly supervised ranking SVM. In *CVPR*, 2012.

[4] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search. *Computer Networks*, 1998.

[5] Christopher Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv:1901.11390*, 2019.

[6] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *CVPR*, 2019.

[7] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, 2018.

[8] 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*, 2015.

[9] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *CVPR*, 2020.

[10] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *CVPR*, 2019.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[12] Martin Englecke, Adam Kosiorek, Owain Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *ICLR*, 2020.

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes challenge 2007 (VOC2007) results, 2007.

[14] Jie Feng, Yichen Wei, Litian Tao, Chao Zhang, and Jian Sun. Salient object detection by composition. In *ICCV*, 2021.

[15] Ferdinand G. Frobenius. Über matrizen aus nicht negativen elementen. 1912.

[16] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3D models. *ACM Trans. on Graphics*, 2003.

[17] David Gale and Lloyd Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 1962.

[18] Spyros Gidaris, Aandrei Bursuc, Gilles Puy, Nikos Komodakis, Matthieu Cord, and Patrick Pérez. Online bag-of-visual-words generation for unsupervised representation learning. In *CVPR*, 2021.

[19] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *ICCV*, 2015.

[20] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[21] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[22] Kristen Grauman and Trevor Darrell. Unsupervised learning of categories from sets of partially matching image features. In *CVPR*, 2006.

[23] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *ICML*, 2019.

[24] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[26] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[27] Yushi Jing and Shumeet Baluja. Visualrank: Applying Pagerank to large-scale image search. *IEEE Trans. Pattern Anal. Machine Intell.*, 2008.

[28] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[29] Armand Joulin, Kevin Tang, and Li Fei-Fei. Efficient image and video co-localization with Frank-Wolfe algorithm. In *ECCV*, 2014.
[30] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43, 1953.
[31] Gunhee Kim, Christos Faloutsos, and Martial Hebert. Unsupervised modeling of object categories using link analysis techniques. In *CVPR*, 2008.
[32] Gunhee Kim and Antonio Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *NIPS*, 2009.
[33] Gunhee Kim and Eric P Xing. On multiple foreground cosegmentation. In *CVPR*, 2012.
[34] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 1999.
[35] Ivan Krasin, Tom Duerig, Neil Aldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyansh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017.
[36] Endmund Landau. Zur relativen wertbemessung der turnierresultate. *Deutsches Wochenschach*, 11:366–369, 1895.
[37] Amy Langville and Carl Meyer. Deeper inside Pagerank. *Internet Mathematics*, 2004.
[38] Amy Langville and Carl Meyer. *Google’s PageRank and beyond: The science of search engine rankings*, 2011.
[39] Yong Jae Lee and Kristen Grauman. Shape discovery from unlabeled image collections. In *CVPR*, 2009.
[40] Changyang Li, Yuchen Yuan, Weidong Cai, Yong Xia, and David Dagan Feng. Robust saliency detection via regularized random walks ranking. In *CVPR*, 2015.
[41] Yao Li, Linqiao Liu, Chunhua Shen, and Anton van den Hengel. Image co-localization by mimicking a good detector’s confidence score distribution. In *ECCV*, 2016.
[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.
[43] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020.
[44] Chen C. Loy, Chunxiao Liu, and Shaogang Gong. Person re-identification by manifold ranking. In *ICIP*, 2013.
[45] Michael W. Mahoney and Lorenzo Orecchia. Implementing regularization implicitly via approximate eigenvector computation. 2010.
[46] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim’s algorithm. In *ICCV*, 2013.
[47] Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised layered image decomposition into object prototypes. *arXiv:2104.14575*, 2021.
[48] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
[49] Shahram Payandeh and Eddie Chiu. Application of modified Pagerank algorithm for anomaly detection in movements of older adults. *IJTI*, 2019.
[50] Oskar Perron. Grundlagen für eine theorie des jacobischen kettenbruchalgorithmus. 1907.
[51] Gabriel Pinski and Francis Narin. Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. *Information Processing & Management*, 12:297–312, 1976.
[52] Rong Quan, Junwei Han, Dingwen Zhang, and Feiping Nie. Object co-segmentation via graph optimized-flexible manifold ranking. In *CVPR*, 2016.
[53] Rouhollah Rahmani, Sally A Goldman, Hui Zhang, John Krettek, and Jason E Fritts. Localized content-based image retrieval. *IEEE Trans. Pattern Anal. Machine Intell.*, 2008.
[54] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.
[55] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *CVPR*, 2017.
[56] Qinghua Ren and Renjie Hu. Saliency detection via pageank and local spline regression. *Journal of Electronic Imaging*, 2018.
[57] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.
[58] Bryan Russell, William Freeman, Alexei Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.
[59] Ramprasaath Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-Cam: Visual explanations from deep networks via gradient-based...
localization. In ICCV, 2017.

[60] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[61] Josef Sivic, Bryan Russell, Alexei Efros, Andrew Zisserman, and William Freeman. Discovering objects and their location in images. In ICCV, 2005.

[62] Jiayu Tang and Paul H Lewis. Non-negative matrix factorisation for object class discovery and image auto-annotation. In CIVR, 2008.

[63] Kevin Tang, Armand Joulin, and Li-jia Li. Co-localization in real-world images. In CVPR, 2014.

[64] Tinne Tuytelaars, Christoph Lampert, Matthew Blaschko, and Wray Buntine. Unsupervised object discovery: A comparison. Int. Journal on Computer Vision, 2010.

[65] Jasper Uijlings, Karin van de Sande, Theo Gevers, and Arnold Smeulders. Selective search for object recognition. Int. Journal on Computer Vision, 2013.

[66] 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 CVPR, 2019.

[67] Huy V. Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In ECCV, 2020.

[68] Richard von Mises and Hilda Pollaczek-Geiringer. Praktische verfahren der gleichungsauflosung. ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 1929.

[69] Markus Weber, Max Welling, and Pietro Perona. Towards automatic discovery of object categories. In CVPR, 2000.

[70] Xiu-Shen Wei, Chen-Lin Zhang, Yao Li, Chen-Wei Xie, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Deep descriptor transforming for image co-localization. In IJCAI, 2017.

[71] Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In CVPR, 2016.

[72] Dan Zhang, Fei Wang, Luo Si, and Tao Li. Maximum margin multiple instance clustering with applications to image and text clustering. Transactions on Neural Networks, 2011.

[73] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In CVPR, 2018.

[74] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In ECCV, 2018.

[75] Jun-Yan Zhu, Jiajun Wu, Yan Xu, Eric Chang, and Zhuowen Tu. Unsupervised object class discovery via saliency-guided multiple class learning. In CVPR, 2012.

[76] Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In ECCV, 2014.
A Appendix

We include in this appendix additional information about the proposed method, including implementation details, experimental results, visualizations and proofs.

A.1 Additional implementation details

**PHM algorithm.** We use the probabilistic Hough matching (PHM) algorithm [8] to compute region similarity scores in our implementation due to its effectiveness in object discovery [8, 66, 67]. Given two images \(i\) and \(j\), the PHM algorithm [8] computes the match between region \(k\) of image \(i\) and region \(l\) of image \(j\) a score defined as:

\[
S_{kl}^{ij} = a_{kl}^{ij} \sum_{k',l'} K_{kl,k'} K_{kl,l'} a_{kl}^{ij},
\]

where \(a_{kl}^{ij}\) is the appearance similarity between the two regions and \(K_{kl,k'} K_{kl,l'}\) measures how compatible two potential matches \((k,l)\) and \((k',l')\) are geometrically. The “Hough” part of the algorithm’s name comes from the associated geometric voting procedure. In our implementation, \(a_{kl}^{ij}\) is the dot product of the unnormalized CNN features associated with the two regions, and \(K_{kl,k'} K_{kl,l'}\) is computed by comparing the matches \((k,l)\) and \((k',l')\) against a set of discretized geometric transformations. See [8, 66] for more details.

**Parallel power iterations.** We solve Q, P, and LOD with the power iteration method [68] (Algorithm 1 below). Since the adjacency matrix (in Q) and the PageRank matrix (in P) are very large, we divide them into chunks of consecutive rows of approximately equal size. At iteration \(t\) in the optimization, these chunks are loaded in parallel into multiple processors’ memories for multiplication with the current iterate \(x_t\). The results of these operations are chunks of the new vector \(x_{t+1}\) which is then assembled from them. We run up to \(T = 50\) iterations of the power method in each experiment.

**Algorithm 1:** Parallel power iterations for finding the first eigenvector of a matrix \(A\).

**Result:** The first eigenvector of \(A\).

**Input:** Number \(M\) of matrix chunks, chunks of matrix rows \(A_1, \ldots, A_M\) of \(A\), number of region proposals \(N\), norm \(L_p\) (\(p = 1\) for P and \(p = 2\) for Q), number of iterations \(T\).

**Initialization:** \(x_0 = \frac{1}{\|e_N\|_p} e_N\).

for \(t = 0\) to \(T - 1\) do

    In parallel in multiple processors:
        for \(i = 1\) to \(M\) do
            Load matrix chunk \(A_i\) into memory.
            Compute the \(i\)-th chunk \(x_{t+1,i} = A_i x_t\).
        end

    In the main processor, assemble \(x_{t+1}\) from its chunks: \(x_{t+1} = [x_{t+1,1}; x_{t+1,2}; \ldots; x_{t+1,M}]\).
    Normalize: \(x_{t+1} = \frac{1}{\|x_{t+1}\|_p} x_{t+1}\).

end

Return \(x_T\).

A.2 Influence of hyper-parameters

The proposed method has two important hyper-parameters, the damping factor \(\beta\) in PageRank and the scalar \(\alpha\) used to select reliable object candidates in LOD. In practice, \(\beta\) should be small so as not to change much the weight matrix \(A\) and \(\alpha\) should also be small since we only want to select a few top-scoring proposals. We have evaluated PageRank for object discovery on C20K and Op50K datasets\(^3\) with increasing values of \(\beta\), ranging from \(10^{-5}\) to \(10^{-1}\), and present the results in Table 4 (left).

\(^3\)We remind that C20K is a subset of COCO [42] (C120K) and that Op50K is a subset of OpenImages [35] (Op1.7M).
Table 4: Influence of the damping factor $\beta$ on PageRank’s performance (left) and of the selection factor $\alpha$ on LOD’s performance (right) on the C20K and Op50K datasets.

| $\beta$ | C20K | Op50K | C20K | Op50K |
|---------|------|-------|------|-------|
| $10^{-3}$ | 48.0 | 47.8 | 6.3 | 6.13 |
| $10^{-4}$ | 48.0 | 47.8 | 6.29 | 6.19 |
| $10^{-5}$ | 47.9 | 47.7 | 6.22 | 6.08 |
| $10^{-6}$ | 47.0 | 47.0 | 5.82 | 5.69 |
| $10^{-7}$ | 40.0 | 38.8 | 4.45 | 4.14 |

| $\alpha$ | C20K | Op50K | C20K | Op50K |
|---------|------|-------|------|-------|
| $0.05$ | 48.4 | 48.2 | 6.63 | 6.5 |
| $0.10$ | 48.5 | 48.1 | 6.63 | 6.46 |
| $0.15$ | 48.5 | 48.2 | 6.64 | 6.49 |
| $0.20$ | 48.5 | 48.2 | 6.64 | 6.48 |

This experiment shows that the performance of PageRank begins to drop when $\beta$ becomes larger than $10^{-3}$ and deteriorates significantly when it exceeds $10^{-2}$. It does not depend much on $\beta$ when this parameter is small enough (less than $10^{-3}$). We choose $\beta = 10^{-4}$ in our implementation.

We have also evaluated LOD with different values of $\alpha$, taken in {$0.05, 0.1, 0.15, 0.2$}, which amounts to selecting 5%, 10%, 15% and 20% of candidates respectively, and show the results in Table 4 (right). As long as $\alpha$ is reasonably small, its value does not significantly affect the performance of LOD. We choose $\alpha = 0.1$ in our implementation.

We have also assessed on the C20K and Op50K datasets the sensitivity of LOD to $N$, the number of initial image neighbors, and to $\gamma$, the parameter controlling the strength of the small perturbation added to the score matrix $W$. For $N$, we have tried different values from 100 to 500 and found that the performance improves only slightly when more neighbors are considered. However, the computational cost increases linearly with $N$ and we find that using 100 neighbors is a good compromise for our datasets. As discussed later, the number of neighbors might need to be changed in case of (undetected) near duplicates or in the case of videos (with successive, highly similar frames). For $\gamma$, we have varied its value in {$10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}$} and found that the performance does not vary much ($\leq 0.5\%$) when $\gamma \leq 10^{-4}$ and slightly degrades when $\gamma \geq 10^{-3}$. This shows that LOD is insensitive to $\gamma$ as long as it is small enough.

A.3 Influence of the number of object proposals on object discovery performance

Unlike [67], we are able to use almost all the regions produced by the proposal algorithm (2000 regions per image at most) thanks to the good scalability of our formulation. On average, we have 814 and 850 regions per image on C20K and Op50K, respectively. We have evaluated LOD on C20K and Op50K using different numbers of proposals (see Table 5) and observed that its performance improves with additional region proposals, notably in the multi-object setting. This observation partly explains our better performance compared to [67] (which places a limit on the number of regions for computational reasons) and the benefit of using all region proposals.

Table 5: Influence of the number of object proposals on performance of LOD.

| # of regions | C20K | Op50K |
|--------------|------|-------|
|              | CorLoc | AP50 | AP@[50:95] | CorLoc | AP50 | AP@[50:95] |
| 50           | 40.9 | 4.5 | 1.22 | 42.0 | 4.55 | 1.31 |
| 100          | 44.0 | 5.38 | 1.47 | 43.4 | 5.1 | 1.4 |
| 200          | 46.5 | 6.13 | 1.71 | 45.6 | 5.83 | 1.61 |
| 400          | 48.0 | 6.6 | 1.91 | 47.1 | 6.32 | 1.77 |
| All          | 48.5 | 6.63 | 1.98 | 48.1 | 6.46 | 1.88 |

A.4 Influence of underlying features

We use features from a VGG16 [60] model trained for image classification on ImageNet [11] in our main experiments. We have also tested LOD with features from VGG19 and ResNet50 [25] and present the results on C20K and Op50K in Table 6. Although VGG19 and ResNet50 give better results in image classification [25, 60], they perform worse than VGG16 in object discovery with LOD. This may be due to the fact that they are more discriminative, focusing mostly on the
most prominent object parts thus less helpful in localizing entire objects, although we do not have a definitive answer (yet) for this.

Table 6: LOD performance with VGG19 [60], ResNet50 [25] and VGG16 [60] features on C20K and Op50K datasets. Although the latter are more powerful in image classification, VGG16 features yield the best results in object discovery with LOD.

| Features  | Single-object | Multi-object |
|----------|---------------|-------------|
|          | CorLoc        |             |
|          | C20K Op50K    | C20K Op50K  |
| VGG19    | 47.4 45.1     | 6.27 5.57   |
| ResNet50 | 35.4 45.9     | 4.08 5.59   |
| VGG16    | 48.5 48.1     | 6.63 6.46   |

A.5 Multi-object discovery performance according to a detection rate metric

Contrary to [67], we have evaluated multi-object discovery performance using average precision (AP) instead of detection rate [67] (DetRate), which can also be thought of recall over ground-truth objects. We argue (in the main body of our submission) that plain DetRate is not a good metric for multi-object discovery since it depends on the number \( m \) of regions returned per image, which is pre-defined. Beside the fact that there is a priori no optimal choice for \( m \), evaluating the performance at a single value of \( m \) does not capture the range of possible performances. AP, on the other hand, summarizes the performance at different values of \( m \).

Despite these remarks, we present here for completeness the multi-object discovery performance in DetRate for LOD and the baselines in Table 7. In addition to computing DetRate at \( m = 5 \) as in [67], we also consider \( m = \bar{m} \) where \( \bar{m} \) is the average number of ground-truth objects per image in the dataset, which is 7 for C20K and C120K, and 8 for Op50K and Op1.7M. The results show that LOD significantly outperforms the baselines in all datasets when detection rate is computed at \( m = 5 \). It also performs better than the others when detection rate is computed at \( m = \bar{m} \), except for Edgeboxes (EB) [83] on Op1.7M dataset. However, we stress again that we think detection rate is not a natural metric for multi-object discovery. We show in the main paper that LOD is significantly better than all baselines in all datasets according to AP, which we think is a more appropriate metric for object discovery.

Table 7: Large-scale multi-object discovery performance and comparison to the state of the art on COCO [42], OpenImages [35] and their respective subsets C20K and Op50K, as measured by detection rate

| Method  | DetRate \((m = 5)\) | DetRate \((m = \bar{m})\) |
|---------|---------------------|--------------------------|
|         | C20K C120K Op50K Op1.7M | C20K C120K Op50K Op1.7M |
| EB [83] | 12.0 12.1 12.5 12.5 | 14.5 14.5 16.0 16.0 |
| Wei [71] | 6.8 6.9 5.7 5.7 | 6.8 6.9 5.7 5.7 |
| Kim [32] | 10.5 10.6 10.8 - | 12.1 12.2 12.9 - |
| Vo [67]  | 12.3 11.8 11.8 - | 13.3 12.7 13.1 - |
| Ours (LOD) | 14.2 14.2 14.0 13.7 | 15.7 15.7 16.2 15.8 |

A.6 More qualitative results

We show additional examples on COCO [42] in Fig. 7 and on OpenImages [35] in Fig. 8 for which LOD successfully discovers objects. We also present some failure cases in Fig. 9. LOD typically fails to discover objects that are too small (images 1 to 5) or only discovers the most discriminative object parts instead of entire objects (images 6 to 8). In some cases, LOD discovers objects that are not annotated: entrance in image 1, tower in image 2 and flower branch in image 4.
Figure 7: Examples in the COCO [42] dataset where LOD successfully discovers ground-truth objects. Ground-truth boxes are in yellow and our predictions are in red.

Figure 8: Examples in the OpenImages [35] dataset where LOD successfully discovers ground-truth objects. Ground-truth boxes are in yellow and our predictions are in red.

Figure 9: Examples in the COCO [42] and OpenImages [35] datasets where LOD fails to discover ground-truth objects. Ground-truth boxes are in yellow and our predictions are in red.

A.7 Proof of Lemma 1.

Proof. Since $W$ is symmetric, all its eigenvalues are real and it can be diagonalized by an orthonormal basis of its eigenvectors. The maximizer of $t^TWt$ in the unit ball is the unit eigenvector of $W$ associated with its largest eigenvalue $\lambda^*$. Given that $W$ is irreducible, it has a unique, unit, non-negative eigenvector associated with its largest eigenvalue, according to the Perron-Frobenius theorem [15, 50].

Note: This is a classic result, only included here for completeness.