Wholly Unsupervised! Segmenting Objects by Contrast and Context

Fei Pan*
University of Michigan, Ann Arbor
feipanir@gmail.com

Yixing Wang*
University of Michigan, Ann Arbor
yixingw@umich.edu

Sangryul Jeon
Pusan National University
srjeonn@pusan.ac.kr

Stella X. Yu
University of Michigan, Ann Arbor
University of California, Berkeley
stellayu@umich.edu

Abstract

We study the problem of unsupervised object segmentation, with the aim of discovering *whole* objects–including both distinctive and less salient parts, rather than just visually striking fragments. Existing unsupervised methods often identify only distinctive parts (e.g., *head* but not *torso*), resulting in incomplete objects. Our key insight is that whole objects can emerge from the interplay of *similarity among parts* and *contrast with surrounding context*, both within and across images. This contrastive and contextual grouping process enables the discovery of heterogeneous object parts as unified wholes, without any predefined notion of object structure.

To this end, we propose Contrastive Contextual Grouping (CCG), a three-step framework for unsupervised whole object segmentation: 1) identifying semantically similar yet visually diverse image pairs, 2) performing co-segmentation using joint graph cuts with pairwise attraction and repulsion, and 3) distilling the results into a single-image segmentation model. Our approach achieves state-of-the-art results across four benchmarks: unsupervised saliency detection, unsupervised object discovery, unsupervised video object segmentation, and unsupervised nuclei segmentation. Remarkably, in some settings it even rivals or exceeds the performance of a supervised foundation model, SAM2, at whole object segmentation given box prompts.

1 Introduction

We consider the task of segmenting *whole* objects from a collection of *unlabeled* images, without external supervision. Unlike prior approaches that often highlight the most visually distinctive parts, our goal is to recover whole objects, including less salient regions that are equally essential for coherent perception.

This problem is crucial for three key reasons. **1)** Pixel-level segmentation annotations remain costly and labor intensive, making unsupervised methods highly desirable in practice. **2)** It offers insights into the fundamental processes by which infants and other cognitive systems learn to perceive and conceptualize objects from unstructured sensory input. **3)** Many downstream applications, such as salient object detection, unsupervised object discovery, and video object segmentation, rely on identifying complete object masks rather than fragmented parts, making accurate whole-object segmentation especially valuable.

*Equal Contribution

Accepted to the NeurIPS 2025 workshop on Space in Vision, Language, and Embodied AI.

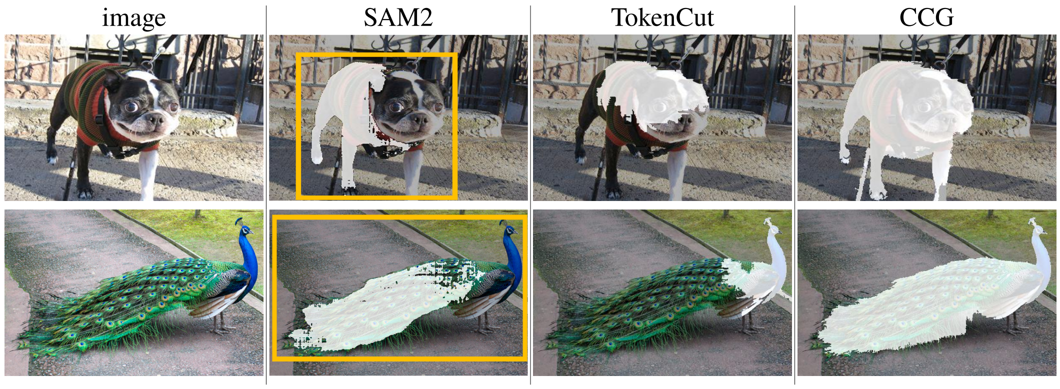

Figure 1: **Unsupervised whole object segmentation is extremely challenging and our CCG method excels. Col.1)** Can we discover and segment whole objects in object-centric images? **Col.2)** Even the latest, largest, extensively supervised model, SAM2 [35], with *the right bounding box prompt* can only delineate visually salient parts (*dog's brown fur, peacock's green train*). **Col.3)** Unsupervised methods such as TokenCut [56] rely on features unsupervisedly learned to optimize certain image-level criteria, discovering only statistically distinctive parts (*face/head*). **Col.4)** Our insight: Objects emerge as wholes through not only *intrinsic part similarity*, but also *extrinsic context contrast*; our CCG discovers *distinctive and unremarkable* parts in a whole without supervision.

Despite progress, whole object segmentation is still challenging, even for supervised foundation models [17, 35, 37]. For example, SAM2 [35] is trained on massive collections of annotated, high-resolution images. Yet, even with *perfect, tight object bounding box* prompts, SAM2 often delineates only visually salient parts (e.g., *a dog's brown fur*, *a peacock's green train*) rather than the entire object (e.g., *the whole dog*, *the whole peacock*).

Unsupervised object segmentation in general has been widely explored, ranging from low-level salient cues to high-level statistical clustering. Key developments include objectness [2], category-independent object proposals [8], exemplar-based recognition through associations [26], multiscale combinatorial grouping [3, 33], object discovery via matching [38, 50], unsupervised feature learning [14, 16], slot attention [22, 39]. Some approaches leverage motion cues in unlabeled videos [64, 60, 21], assuming pre-trained optical flow detectors or piece-wise constant object motion models.

Unsupervised *whole* object segmentation has been explored earlier using matting or boundary cues [46, 25] and, more recently, through feature similarity or attention maps [56, 28, 68, 45, 67] from self-supervised models like DINO [5]. However, because these features are optimized for image-level objectives, existing methods, e.g., TokenCut [56], tend to highlight only statistically distinctive parts, rather than capturing the object as a whole.

Our novel approach to whole object discovery shifts the focus from *what the object is* to *how it contrasts with its context*. The key insight is that an object, even when composed of distinctive parts, can emerge as a cohesive whole through both *intrinsic similarity among its parts* and *extrinsic contrast with its surroundings*. This contextual relationship is crucial for binding diverse object parts

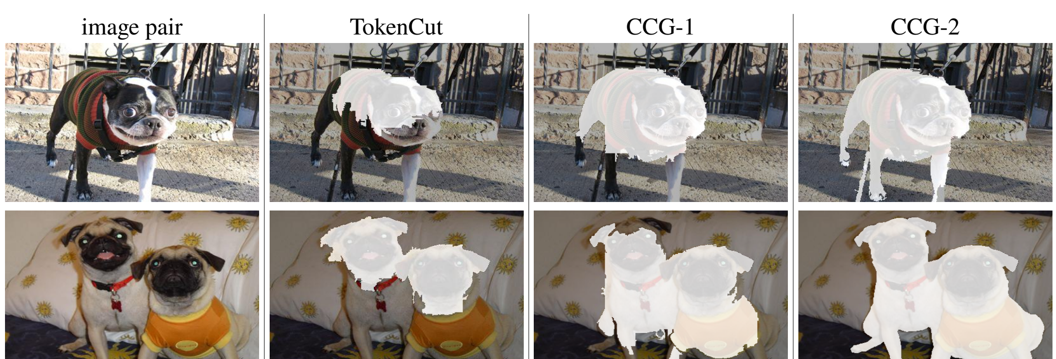

Figure 2: **Our CCG benefits from co-segmenting semantically similar yet visually distinct image pairs, identified without supervision.** CCG-1 (2) denotes single(two)-image (co-)segmentation results. Contexts and contrasts from paired images significantly enhance whole object discovery.

into a unified entity in a bottom-up, data-driven manner [1]. In Fig. 1, while the green peacock train and blue peacock head have different textures, their colors starkly contrast with the gray background. Echoing the adage "*The enemy of my enemy is my friend*", the two distinctive parts become allies through their shared contrast with the background, allowing the peacock to emerge as a unified whole.

For richer grouping relationships, we introduce a co-segmentation setting using semantically similar yet visually different image pairs (Fig.2). These pairs can be derived from unlabeled data, such as images or videos of the same scene, or by clustering self-supervised ViT features [5, 31, 6] that capture semantic similarities. By leveraging co-segmentation, we gain additional contrastive and contextual grouping cues across image pairs, enabling more robust and accurate whole object segmentation.

We present an unsupervised whole-object segmentation algorithm based on *Contrastive Contextual Grouping*. Our CCG operates in three steps: **1)** Identify semantically similar yet visually distinctive image pairs for co-segmentation. Identical images reduce the task to single-image segmentation, while unrelated pairs hinder co-segmentation. **2)** Perform co-segmentation via joint graph partitioning, where patches are nodes and edges encode two types of pairwise relationships: feature similarity and dissimilarity. The objective is not only to *discover friends through similarity*, but also to *discover allies through shared dissimilarity*, enabling robust whole-object discovery. **3)** Distill co-segmentation results into a single-image segmentation model, with a ViT backbone and lightweight segmentation head, enabling efficient inference on individual images without requiring paired inputs. CCG achieves state-of-the-art performance on unsupervised saliency detection, object discovery, video object segmentation, and nuclei segmentation.

Our contributions. **1)** We tackle the problem of *unsupervised whole object segmentation*, addressing the underexplored challenge of discovering both salient/characteristic and unremarkable parts in cohesive wholes. **2)** We propose a novel, fully unsupervised framework for bottom-up whole-object discovery, driven by data rather than labels. It operates via dual forces: *grouping by similarity* and *segregation by dissimilarity*, enhanced by co-segmentation, feature learning, and model distillation. **3)** We achieve consistent, significant gains over prior unsupervised methods across four benchmarks, In some cases, CCG even outperforms the supervised foundation model SAM2 in segmenting whole objects given box prompts.

2 Related Work

Unsupervised Object Discovery. Most works leverage self-supervised features from visual transformers [5, 6, 4]. TokenCut [56] constructs a weighted graph using feature similarities (attraction) and performs graph cuts to separate objects from backgrounds. Unlike TokenCut, we introduce pairwise attraction and repulsion in a joint weighted graph for co-segmentation, enabling whole object localization and segmentation. SelfMask [43] clusters multiple self-supervised features to extract object masks, while LOST [44] localizes object seeds and expands them to similar patches. FreeSOLO [54] generates FreeMask predictions from feature similarities, and FOUND [45] uses heuristics to search for background seeds. HEAP [67] employs contrastive learning for clustered feature embeddings. PEEKABOO [68] localizes objects by hiding parts of images. However, these methods are limited to discovering descriptive parts of objects. In contrast, our CCG uses pairwise attraction and repulsion in co-segmentation to segment whole objects.

Unsupervised Video Object Segmentation. [63] proposes an adversarial-based method to predict object masks from images and optical flow maps. [23] adopts co-attention layers based on siamese networks for segmentation, requiring expensive training resources. [58] uses optical flow and contrastive motion clustering to segment moving objects in videos. However, these methods rely on externally supervised motion estimation networks [49, 47]. VideoCutLER [55] segments video objects via graph cuts on attractions and refines masks through training. While AMD [21] jointly learns segmentation and motion estimation end-to-end, its segment-wise constant motion assumption is too simplistic to yield fine segmentations with details and complete parts. In contrast, our CCG , trained on unlabeled videos, delivers more accurate whole-object segmentation without a pre-trained optical flow detector.

Segmentation by Graph Cuts. Normalized cuts [41] frames segmentation as a graph partitioning problem, optimizing similarity within partitions. [29] derives partitions using stacked eigenvectors of the graph Laplacian matrix. [66] applies graph cuts to affinities of key, query, and value features of ViTs, revealing visual semantics and spatial locations of segments. Earlier work [65] introduces the role of repulsion for single-image segmentation based on fixed low-level features. [24] conduct segmentation using graph neural networks. In contrast, CCG is the first to address unsupervised whole object segmentation using data-driven learned features in a co-segmentation and model distillation framework.

Co-Segmentation. [13] leverages color histogram similarities to segment common objects from similar image pairs. [20] employs a Siamese network to segment shared objects across image pairs. [15] introduces a unified ViT framework for joint co-segmentation and co-detection. However, existing co-segmentation methods lack contextual relationship analysis and do not address whole object segmentation. In contrast, our approach incorporates attraction and repulsion across a related image pair, enabling whole object segmentation through contrastive contextual grouping.

3 Contrastive Contextual Grouping

We aim to discover and segment whole objects without supervision, based on *intrinsic similarity* between parts and *extrinsic contrast* with their surroundings.

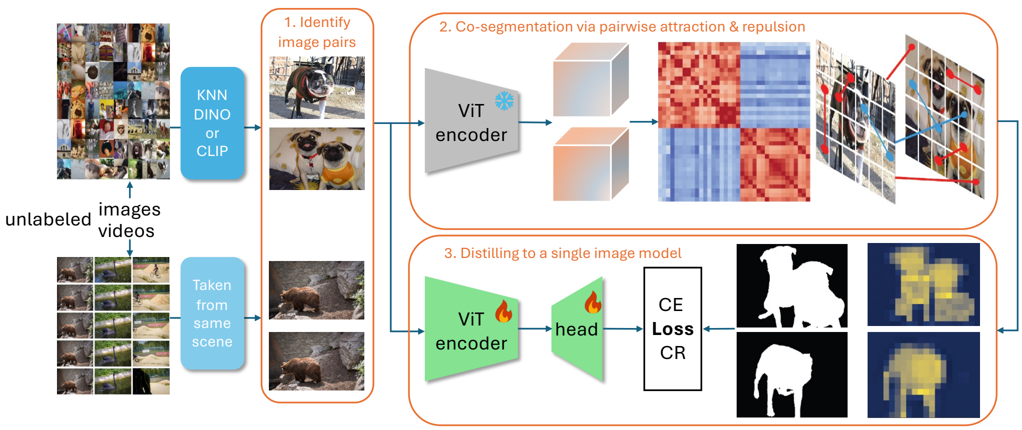

Figure 3: **Overview of our three-step CCG algorithm for unsupervised whole object discovery**. **1)** Identify semantically similar yet visually different image pairs. For unlabeled videos, they are simply consecutive video frames, whereas for unlabeled images, they are k-nearest neighbors in some unsupervisedly learned feature space. **2)** Co-segmentation based on pairwise similarity (attraction) and dissimilarity (repulsion) of image patch features extracted from a self-supervised ViT encoder. **3)** Distill co-segmentation results to a single-image model with a ViT encoder and a segmentation head, trained with cross-entropy (CE) and contrastive (CR) losses.

Our CCG has three steps (Fig 3): **1)** identifying semantically similar yet visually different image pairs, **2)** performing co-segmentation through joint graph cuts with pairwise attraction and repulsion, and **3)** distilling the results into a single-image segmentation model.

Primer: Graph Cuts with Attraction and Repulsion. We apply prior work [65] to a ViT patch graph, where each node represents a square image patch used in ViT, and the edge between nodes i, j is attached with an attraction weight A_{ij} and a repulsion weight R_{ij}, both derived from the cosine similarity $\boldsymbol{S}_{ij}$ between their ViT patch features $\boldsymbol{F}_i$, $\boldsymbol{F}_j$:$\boldsymbol{S}_{ij} = \frac{<\boldsymbol{F}_i,\boldsymbol{F}_j>}{\|\boldsymbol{F}_i\|\|\boldsymbol{F}_j\|}$. The larger S_{ij}, the larger the attraction A_{ij} and the smaller the repulsion R_{ij}. $\boldsymbol{A}$ and $\boldsymbol{R}$ are defined as Gaussian functions of $\boldsymbol{S}$ (Fig.A1). Object segmentation is then formulated as a two-way node partitioning problem. Let $\mathbb{V}$ denote the set of all patch nodes, and $\mathbb{V}_1, \mathbb{V}_2$ two disjoint subsets: $\mathbb{V}_1 \cup \mathbb{V}_2 = \mathbb{V}, \mathbb{V}_1 \cap \mathbb{V}_2 = \varnothing$. We seek an optimal partitioning with dual forces: Group by similarity and segregate by dissimilarity. Given attraction $\boldsymbol{A}$ and repulsion $\boldsymbol{R}$, we maximize the following with hyperparameter $\omega \in [0, 1]$:

$$\xi_{AR} = \frac{\text{within-group } \boldsymbol{A}}{\text{total degrees of } \boldsymbol{A}, \boldsymbol{R}} + \omega \frac{\text{between-group } \boldsymbol{R}}{\text{total degree of } \boldsymbol{A}, \boldsymbol{R}}. \tag{1}$$

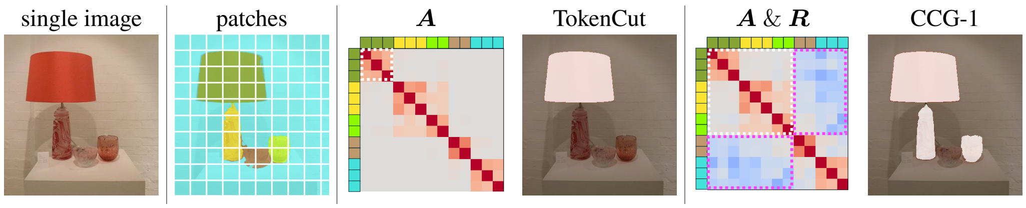

Figure 4: **Pop out whole objects by contrastive contextual grouping of patches within a single image.** Patches are color coded. **Middle column)** By attraction $\boldsymbol{A}$ alone (values shaded in red, outlined in white boxes), object parts are too weakly similar to be grouped as one; TokenCut [56] can thus only segment out the most distinctive part: *lamp shade*. **Right column)** By repulsion $\boldsymbol{R}$ (values shaded in blue, outlined in magenta boxes) in addition, *lamp shade, lamp base* are both dissimilar to the background and need to be separated from it; our CCG can thus segment out the *whole lamp* and a similar item.

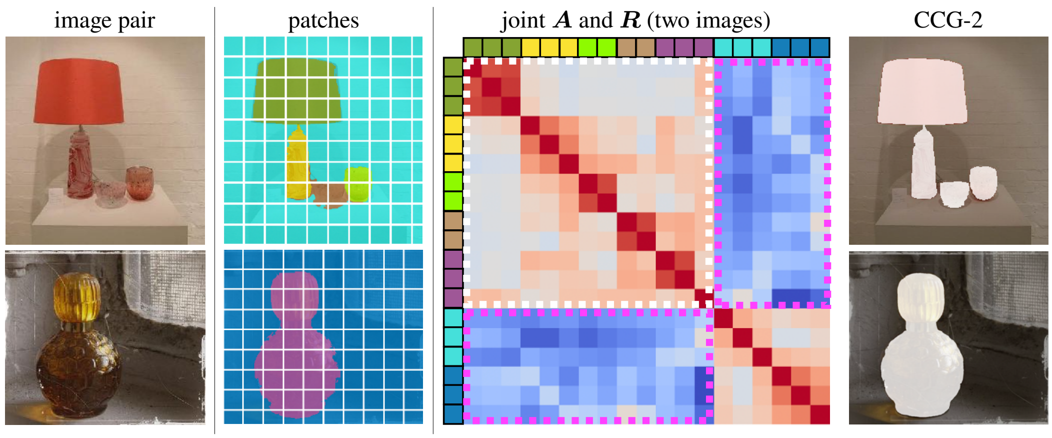

Figure 5: **Pop out whole objects more accurately with co-segmentation.** Image pairs are obtained by unsupervised clustering, or simply videos of the same scene. A joint graph is constructed using patches from both images. Patches are color-coded. To visualize the effects of attraction and repulsion, we sort patches by foreground then background. Strong foreground-background repulsion (values shaded in blue, outlined in magenta boxes) across these two images, strong attraction within foreground and background respectively, help our CCG discover the *whole lamp set* and the *whole vaze* together.

ω weighs the relative importance between attraction and repulsion. [65] shows that:

$$\max \quad \xi_{AR}(\boldsymbol{p}) = \sum_{t=1}^{2} \frac{\boldsymbol{p}_t^T \boldsymbol{W} \boldsymbol{p}_t}{\boldsymbol{p}_t^T \boldsymbol{D} \boldsymbol{p}_t}, \tag{2}$$

$$\boldsymbol{W} = \boldsymbol{A} - \boldsymbol{R} + \boldsymbol{D}_R, \quad \boldsymbol{D} = \boldsymbol{D}_A + \boldsymbol{D}_R, \tag{3}$$

where $\boldsymbol{p}_t$ is a binary partition indicator for $\mathbb{V}_t$, $\boldsymbol{D}_A(\boldsymbol{D}_R)$ is a diagonal degree matrix with each diagonal entry indicating total $A(R)$ weights a patch node has. The optimum in the relaxed continous domain is the second largest eigenvector $\hat{\boldsymbol{z}}$ of the following eigensystem:

$$\boldsymbol{D}^{-1} \boldsymbol{W} \boldsymbol{z} = \lambda \boldsymbol{z}. \tag{4}$$

See Supplementary for more technical details. Our CCG uses both A and R, whereas TokenCut [56] uses only A, a special case of ours when $\omega = 0$.

Bipartitioning imposes an important bottleneck: Each region must commit to one of two camps, limiting grouping variability. **1)** Strict attraction-based bipartitioning precludes indirect grouping, which is essential for assembling whole objects composed of diverse parts. **2)** Repulsion enables such indirect grouping by aligning parts not because they are similar to each other, but because they are dissimilar to the same background - realizing "*The enemy of my enemy is my friend*". Fig. 4 shows that attraction alone may isolate a single homogeneous region, but it is *repulsion* that allows visually distinct parts to emerge together as a coherent whole, without any preconception of object structure.

Step 1. Identify Related Image Pairs. We adopt an image co-segmentation setting to facilitate whole object discovery. Ideally, image pairs should be semantically similar yet visually distinct to enhance within-group similarity and between-group dissimilarity, facilitating clearer figure-ground segregation (Fig.2). Such pairs can be found in unlabeled data, e.g., from videos of the same scene or by clustering self-supervised ViT features[5, 31, 6] that capture semantic similarity. Examples of k-nearest neighbors from DINO as well as pre-trained CLIP features are shown in Fig. A2.

Step 2. Co-Segmentation by Attraction and Repulsion. We construct a joint graph with patches from both images as nodes, compute attraction and repulsion as edge weights, and perform graph cuts accordingly. The joint partitioning finds not only two regions within each image, but also region correspondence across images. We follow TokenCut and select the foreground as the region with the maximum absolute value of the eigenvector components. Note that if the two images are identical, then the two-image co-segmentation based on attraction and repulsion within and across images is reduced to the single-image segmentation based on within-image attraction and repulsion only. For clarity, we denote the two-image and one-image cases as CCG-2 and CCG-1 respectively. Fig. 5 shows that co-segmentation not only brings out two related whole objects, but also enhances the whole object segmentation within individual images. Compared to the partial lamp set discovered by CCG-1 in Fig. 4, the entire lamp set is now segmented out by CCG-2 .

Step 3. Distill to A Single-Image Segmentation Model. We distill co-segmentation results into a single-image segmentation model with a ViT encoder (shared with DINO) and a lightweight head composed of a $1{\times}1$ convolution followed by softmax. The model is trained using a combination of cross-entropy (CE) loss and contrastive loss [59, 48, 40, 53].

The CE loss refines ViT features using the whole object masks: For pixel i in the image, given the predicted probability map $\hat{y}$ and the binary mask y from co-segmentation, the total pixel-wise CE loss is

$$\mathcal{L}_{CE} = -\sum_i y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i). \tag{5}$$

The contrastive loss aims to sharpen the mask by reducing the feature distance within each region and increasing the feature distance between regions. Given features $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ extracted from the distillation ViT encoder, the contrastive loss is formulated as

$$\mathcal{L}_{CR} = -\frac{1}{|P|}\sum_{(i,j)\in P} \log \frac{\exp(\frac{f_i \cdot f_j}{\tau})}{\sum_{q\in Q(i)} \exp(\frac{f_i \cdot f_q}{\tau})}, \tag{6}$$

where P is the set of positive (foreground-forground) pixel pairs, $Q(i)$ is the set of negative (foreground-background) pixel pairs, and τ is a temperature parameter. Fig. 3 shows the entire workflow. While we can close the loop by replacing the initial ViT encoder with the distillation ViT encoder, we find such iterations unnecessary, as the model converges fast and brings little further gain.

4 Experiments

Our CCG aims to discover and segment whole objects without any supervision. In our framework, CCG-1 denotes the segmentation results for an individual image, whereas CCG-2 represents the segmentation outcomes obtained from image pairs (in the co-segmentation setting). We evaluate its performance and benefits in four tasks: 1) unsupervised saliency detection, 2) unsupervised object discovery, 3) unsupervised video object segmentation, and 4) unsupervised unclei segmentation.

Implementation Details. Our ViT encoder follows the same architecture as DINO `ViT-S/8` during the distillation stage. The segmentation head consists of a single `conv` 1×1 layer. We train the ViT encoder using the AdamW optimizer with a learning rate of 0.001, while the segmentation head is optimized with AdamW at a learning rate of 0.05. Training is conducted over 300 epochs with a batch size of 16, utilizing four A40 NVIDIA GPUs. The repulsion weight ω is set to 0.2. For video frame pair selection, we use a frame interval of 10 to generate reference image pairs for co-segmentation, such as [(00.jpg, 10.jpg), (01.jpg, 11.jpg), (02.jpg, 12.jpg), $\cdots$]. See more details in Supplementary.

4.1 Unsupervised Saliency Detection

Datasets & **Eval Metrics.** We consider three datasets: ECSSD [42] containing 1000 images (train: 700, val: 150, test: 150), DUT-OMRON [62] including 5186 images (train: 3630, val: 778, test:

Table 1: **CCG outperforms existing methods for unsupervised saliency detection task**. In the *w/o. training* setting, CCG outperforms the *SoTA* method TokenCut across all three datasets (performance gap in blue). In the *w/. training* setting, with initial object masks by attraction and repulsion, CCG surpasses the *SoTA* method HEAP (performance gap in green).

Method	Training?	`ViT`	ECSSD			DUTS			DUT-OMRON		
			max F_β	IoU	Acc.	max F_β	IoU	Acc.	max F_β	IoU	Acc.
FUIS [27]	×		–	71.3	91.5	–	52.8	89.3	–	50.9	88.3
LOST [44]	×	`S/16`	75.8	65.4	89.5	61.1	51.8	87.1	47.3	41.0	79.7
DSS [28]	×	–	–	**73.3**	–	–	51.4	–	–	**56.7**	–
TokenCut [56]	×	`S/16`	80.3	71.2	91.8	67.2	57.6	90.3	60.0	53.3	88.0
CCG-1	×	`S/16`	82.7(+2.4)	72.8(+0.6)	93.1(+1.3)	**69.5**(+2.3)	60.2(+2.6)	92.8(+2.5)	62.6(+2.6)	55.3(+2.0)	**90.7**(+2.7)
CCG-2	×	`S/16`	**83.1**(+2.8)	73.2(+2.0)	**94.7**(+2.9)	69.3(+2.1)	**60.5**(+2.9)	**93.2**(+2.9)	**63.3**(+3.3)	56.4(+3.1)	90.6(+2.6)
SelfMask [43]	✓	`S/8`	–	78.1	94.4	–	62.6	92.3	–	58.2	90.1
FOUND [45]	✓	`S/8`	**95.5**	80.7	94.9	71.5	64.5	93.8	66.3	57.8	91.2
PEEKABOO [68]	✓	`S/8`	95.3	79.8	94.6	**86.0**	64.3	93.9	**80.4**	57.5	91.5
HEAP [67]	✓	`S/8`	93.0	81.1	94.5	75.7	64.4	94.0	69.0	59.6	92.0
CCG-1	✓	`S/8`	94.1(+1.1)	83.6(+2.5)	95.2(+0.7)	78.0(+2.3)	65.9(+1.5)	**94.6**(+0.6)	70.7(+1.7)	60.8(+1.2)	93.5(+1.5)
CCG-2	✓	`S/8`	94.5(+1.5)	**83.9**(+2.8)	**95.8**(+1.3)	78.2(+2.5)	**66.5**(+2.1)	94.4(+0.4)	71.2(+2.2)	**61.3**(+1.7)	**93.8**(+1.8)

778), and DUTS [52] with 1580 images (train: 7373, val: 1580, test: 1580). We adopt three standard metrics: mean intersection-over-union (IoU) with a threshold set at 0.5, pixel accuracy (Acc), and the maximal F_β score (max F_β), where β^2 is set to 0.3, in accordance with [56], [45], and [67].

Baselines. In the setting of *w/o. training*, we compare CCG-1 and CCG-2 directly – without distillation – against non-training baseline methods including FUIS [27], LOST [44], DSS [28], and TokenCut [56]. We also compare CCG-1 with SAM2 [35] on DUTS given bounding boxes as the prompts. In the setting of *w/. training*, we apply distillation to the results from both CCG-1 and CCG-2, and benchmark these against methods that require network training, namely SelfMask [43], FOUND [45], PEEKABOO [68], and HEAP [67].

Results. Table 1 shows that in the *w/o. training* setting, both CCG-1 and CCG-2 outperform TokenCut using the same `ViT-S/16` architecture. While TokenCut – employs graph cut using only attraction – segments merely discriminative object parts, CCG leverages a weighted graph that combines attraction and repulsion to capture complete objects. This indicates that the *combined use of attraction and repulsion promotes the segmentation of whole objects from unlabeled images*. In the *w/. training* setting, CCG with distillation achieves higher scores than the current state-of-the-art model HEAP using the same `ViT-S/8` architecture, confirming that distillation with initial object masks by attraction and repulsion greatly refines whole object segmentation and builds new SoTA on saliency detection(Fig. A4). CCG-2 outperforms CCG-1 overall, highlighting the benefits of co-segmentation: similar image pairs bring stronger contextual information for unsupervised whole object segmentation.

We further compare CCG-1 with SAM2 [35] on DUTS in a zero-shot setting. Since SAM2 requires prompts for segmentation, we begin by providing the ground-truth bounding boxes as prompts. However, using ground-truth boxes undermines the purpose of saliency detection, so we gradually enlarge the box prompts until they cover the entire image. To eliminate the effect of object size, we only evaluate images with medium-sized ground-truth boxes (diagonal ratio between 50~60%), where the ratio is defined as the diagonal length of the box over that of the image. For each box prompt, we feed the corresponding region to CCG-1 for fair comparison. As shown in Fig. 6, even with ground-truth boxes, SAM2 often fails to segment whole objects. As the box expands from tightly enclosing the object to covering the full image, SAM2 struggles to consistently identify the salient object. We attribute this to increasing heterogeneity within the prompted region. We measure heterogeneity as the standard deviation of the normalized l_2 distance between each patch feature and the mean feature within the box. As shown in Table 2, the heterogeneity of the boxed region grows with the box size, indicating that more complex regions hinder SAM2's ability to segment whole objects. In contrast, our method remains robust by leveraging both patch-level similarity and dissimilarity cues to discover complete objects even in heterogeneous images.

4.2 Unsupervised Object Discovery

Datasets & **Eval Metric.** We use three widely recognized benchmarks: VOC07 [9] containing 5011 images (train: 3507, val: 752, test: 752), VOC12 [10] that includes 11540 images in total (train: 8078, val: 1731, test: 1731), and COCO20K [51] consisting of 19,817 images (train: 13873, val: 2972, test:

Bbox ratio	50~60%	60~70%	70~80%	80~90%	90~100%
SAM2 mIoU	**84.0**	**76.0**	46.3	15.3	1.70
CCG-1 mIoU	59.3	64.3	**67.0**	**67.8**	**67.2**
Heterogeneity	0.151	0.172	0.192	0.205	0.212

Table 2: Mean IoU of SAM2 v.s. CCG-1 on selected DUTS images given different sizes of box prompts. Increasing the box size significantly degrades SAM2's performance, while CCG-1 remains consistently strong.

Figure 6: As the bounding box expands, SAM2 often fails to capture the salient object and produces noticeable artifacts, whereas our unsupervised CCG-1 consistently segments whole objects when given only a single box-covered region each time as input.

Table 3: Both CCG-1 and CCG-2 outperform existing methods on unsupervised object discovery in both *w/o. training* (performance gap in blue) and *w/. training* settings (performance gap in green).

Method	Training?	ViT	VOC07	VOC12	COCO20K
DINO-seg [5]	×	S/16	45.8	46.2	42.0
LOST [44]	×	S/16	61.9	64.0	50.7
DSS [28]	×	S/16	62.7	66.4	52.2
TokenCut [56]	×	S/16	68.8	72.1	58.8
CCG-1	×	S/16	71.4 (+2.6)	**73.8** (+1.7)	60.3 (+1.5)
CCG-2	×	S/16	**72.3** (+3.5)	73.7 (+1.6)	**61.7** (+2.9)
SelfMask [43]	✓	S/8	72.3	75.3	62.7
FOUND [45]	✓	S/8	72.5	76.1	62.9
PEEKABOO [68]	✓	S/8	72.7	75.9	64.0
HEAP [67]	✓	S/8	73.2	77.1	63.4
CCG-1	✓	S/8	76.4 (+3.2)	79.8 (+2.7)	65.6 (+2.2)
CCG-2	✓	S/8	**77.7** (+4.5)	**80.8** (+3.7)	**66.2** (+2.8)

Table 4: **CCG is a strong unsupervised video object segmenter.** In *w/o. training* setting, CCG outperforms TokenCut (performance gap in blue). In *w/. training* setting, CCG-1 and CCG-2 surpass VideoCutLER which relies solely on attraction for object discovery (performance gap in green). They also achieve competitive results compared with models leveraging optical flows.

Method	Training?	Use Optical Flow?	DAVIS	FBMS	SegTV2
TokenCut [56]	×	×	64.3	60.2	59.6
CCG-1	×	×	66.4 (+2.1)	62.5 (+2.3)	61.2 (+1.6)
CCG-2	×	×	**67.9** (+3.6)	**64.1** (+3.9)	**62.1** (+2.5)
CIS [63]	✓	✓	71.5	63.6	62.0
CMC [58]	✓	✓	**75.4**	66.8	62.6
AMD [21]	✓	×	45.7	28.7	42.9
VideoCutLER [55]	✓	×	68.4	64.6	62.5
CCG-1	✓	×	71.8 (+3.4)	66.4 (+1.8)	64.5 (+2.0)
CCG-2	✓	×	72.4 (+4.0)	**67.9** (+3.3)	**66.1** (+3.6)

2972). Following the evaluation protocol [57, 7], results are reported using the correct localization (*CorLoc*) metric, which measures the percentage of images where objects are correctly localized.

Baselines. In the *w/o. learning*, both CCG-1 and CCG-2 are tested without distillation and compared against non-training approaches such as DINO-seg [5], DSS [28], LOST [44], and TokenCut [56]. In the *w/. learning*, we access the results of distillation from CCG-1 and CCG-2 against the training-dependent methods including SelfMask [43], FOUND [45], PEEKABOO [68], and HEAP [67].

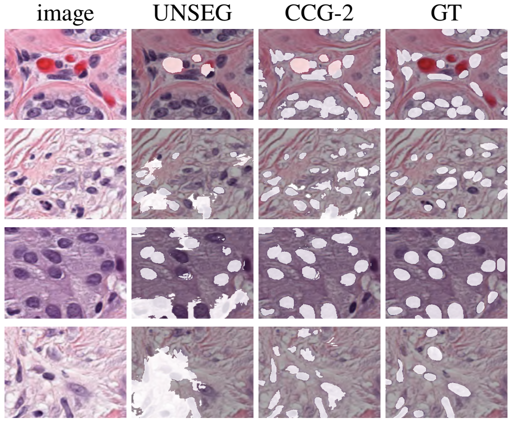

Figure 7: **Our CCG surpasses the state-of-the-art unsupervised nuclei segmentation method UNSEG [18]**. CCG using attraction and repulsion shows better nuclei segmentation than UNSEG as a Baysina-based method that predicts prior distribution of nuclei.

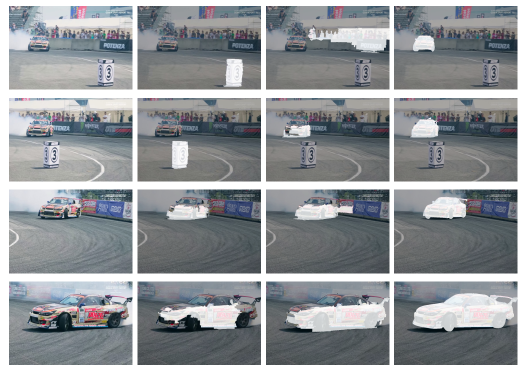

Figure 8: **CCG outperforms TokenCut and FOUND in segmenting the racing car as a whole from these unlabeled video frames**. Both TokenCut and FOUND only use attractions and fail to pop out the vehicle from the background, whereas CCG using co-segmentation on video frames segments the whole car body from the background.

Results. Table 3 shows that, in *a w/o. training* setting, CCG-1 outperforms TokenCut (using `ViT-S/16`) by using both attraction and repulsion forces. CCG-2 further improves performance over CCG-1, highlighting the advantage of co-segmentation for unsupervised object discovery. In *w/. training* setting, both CCG-1 and CCG-2 present higher scores than the current *SoTA* model HEAP (using `ViT-S/8`).

4.3 Unsupervised Video Object Segmentation

Datasets & **Eval Metric.** We utilize there datasets for evaluation: DAVIS [32] including 50 videos in total (train: 30, val: 10, test: 10), FBMS [30] that has 59 videos (train: 25, test: 30), and SegTV2 [19] containing 14 videos (train: 6, test: 7). We merge the annotations of all moving objects into a single mask for both the FBMS and SegTV2 datasets following [56, 61]. We also adopt the object centric CO3D dataset [36]. The performance is assessed using the Jaccard index ($\mathcal{J}$), which quantifies the intersection over union (IoU) between the predicted segmentation masks and the ground-truth.

Baselines. CCG is evaluated against several unsupervised video object segmentation methods, many of which rely on optical flow during training. These include AMD [21], CIS [63], CMC [58], and VideoCutLER [55]. Notably, VideoCutLER predicts object masks using only feature similarity (attraction). TokenCut, though training-free, still requires optical flow as input.

Results. Table 4 shows that, in the *w/o learning* setting, CCG-1 with attraction and repulsion within a single frame, outperforms TokenCut. CCG-2 further boosts performance by incorporating attraction and repulsion across adjacent frames. This demonstrates that CCG is an effective zero-shot segmenter from unlabeled video, without relying on optical flow. Visualizations are shown in Fig.8.

4.4 Unsupervised Nuclei Segmentation

We apply CCG to unsupervised nuclei segmentation on the PanNuke dataset [11], which contains 7,904 H&E-stained images (2,657 train / 2,524 val / 2,732 test), each sized 256×256. We compare against the SoTA UNSEG [18], which uses Bayesian inference to model nuclei priors for segmentation. Performance is evaluated using pixel accuracy, mean IoU, and F_1 score (Table A3). Even without distillation, both CCG-1 and CCG-2 outperform UNSEG by over 20%, demonstrating that *CCG generalizes well from natural to medical images (Fig. 7), including nuclei segmentation.*

5 Performance Analysis and Ablation

Repulsion Weight. We analyze the effect of ω. Fig.A3 shows an ablation on unsupervised saliency detection (ECSSD). When $\omega = 0$ (red line), CCG reduces to TokenCut[56]. Optimal performance -

measured by pixel accuracy, mean IoU, and maximal F_β - occurs near $\omega = 0.2$. We adopt this setting for all subsequent experiments, *removing the need for per-task tuning*.

Image Pair Discovery. We explore discovering similar image pairs from unlabeled data using k-nearest neighbors on DINO and CLIP features. DINO is a self-supervised ViT model trained without external labels. Results are shown in Table A1. To minimize dependence on additional models, we use DINO features for all main experiments. We also ablate CLIP and ResNet-50 (ImageNet pre-trained) features. On ECSSD, CLIP achieves the best performance, likely due to its supervised training on large-scale labeled data.

Video Frame Pair Discovery. CCG utilizes a pair of frames captured from the same video clip. These two frames are possibly located at different timestamps in the video. We analyze the effect of varying frame intervals on unsupervised video object segmentation, as shown in Fig. A5 in the supplementary. When the video frame interval is set to 0, CCG-2 is reduced to CCG-1 as the two reference images are the same. The video frame intervals between 8 to 18 yield the best results. Consequently, we set the frame interval to 10 for all unsupervised video object segmentation experiments.

Segmentation Head. We investigate the impact of distillation with different # of `conv` layers in the segmentation head. Table A2 presents the results of various head designs. Performance improves when using a 2×`conv`(1,1) configuration but degrades with a 3×`conv`(1,1) setup, suggesting a trade-off between model complexity and effectiveness.

Eigenvectors. Fig. A6 shows that the eigenvectors of CCG-2 pop out the whole body of the dogs, while TokenCut's eigenvectors can only segment out the dog heads.

6 Summary and Limitation

We formulate unsupervised whole object segmentation as graph bi-partitioning using both attraction and repulsion. By maximizing within-group attraction and between-group contrast, our method segments entire objects with both distinctive and unremarkable parts, and outperforms prior methods on object discovery, saliency detection, and video segmentation. This simple approach draws inspiration from principles of early perceptual organization–grouping by similarity and contrast–and offers insights into how complex visual scenes can be parsed without supervision.

Currently, our CCG framework performs co-segmentation on image pairs. Future work could explore large-scale graph partitioning with attraction and repulsion to enable co-segmentation across thousands of images efficiently. Moreover, our current formulation addresses binary segmentation; extending it to multi-way segmentation tasks, such as semantic or panoptic segmentation, would further enhance its applicability and impact.

Acknowledgements

This project was supported, in part, by NSF 2215542, NSF 2313151, and Bosch gift funds to S. Yu at UC Berkeley and the University of Michigan.

References

[1] Ralph Adolphs, Lauri Nummenmaa, Alexander Todorov, and James V Haxby. Data-driven approaches in the investigation of social perception. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371(1693):20150367, 2016.

[2] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2928–2935, 2010.

[3] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Computer Vision and Pattern Recognition*, 2014.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[7] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1201–1210, 2015.

[8] Ian Endres and Derek Hoiem. Category independent object proposals. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 575–588, 2010.

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[10] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Anal. Stat. Model. Comput. Learn., Tech. Rep*, 2007(1-45):5, 2012.

[11] Jevgenij Gamper, Navid Alemi Koohbanani, Ksenija Benes, Simon Graham, Mostafa Jahanifar, Syed Ali Khurram, Ayesha Azam, Katherine Hewitt, and Nasir Rajpoot. Pannuke dataset extension, insights and baselines. *arXiv preprint arXiv:2003.10778*, 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1943–1950. IEEE, 2010.

[14] Tsung-Wei Ke, Jyh-Jing Hwang, Yunhui Guo, Xudong Wang, and Stella X. Yu. Unsupervised hierarchical semantic segmentation with multiview cosegmentation and clustering transformers. In *CVPR*, New Orleans, Louisiana, 19-24 June 2022.

[15] Tsung-Wei Ke, Jyh-Jing Hwang, Yunhui Guo, Xudong Wang, and Stella X Yu. Unsupervised hierarchical semantic segmentation with multiview cosegmentation and clustering transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2571–2581, 2022.

[16] Tsung-Wei Ke, Sangwoo Mo, and Stella X. Yu. Learning hierarchical image segmentation for recognition and by recognition. In *ICLR*, 2024.

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[18] Bogdan Kochetov, Phoenix D Bell, Paulo S Garcia, Akram S Shalaby, Rebecca Raphael, Benjamin Raymond, Brian J Leibowitz, Karen Schoedel, Rhonda M Brand, Randall E Brand, et al. Unseg: unsupervised segmentation of cells and their nuclei in complex tissue samples. *Communications Biology*, 7(1):1062, 2024.

[19] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE international conference on computer vision*, pages 2192–2199, 2013.

[20] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 638–653. Springer, 2019.

[21] Runtao Liu, Zhirong Wu, Stella Yu, and Stephen Lin. The emergence of objectness: Learning zero-shot segmentation from videos. *Advances in neural information processing systems*, 34:13137–13152, 2021.

[22] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538. Curran Associates, Inc., 2020.

[23] Xiankai Lu, Wenguan Wang, Jianbing Shen, David Crandall, and Jiebo Luo. Zero-shot video object segmentation with co-attention siamese networks. *IEEE transactions on pattern analysis and machine intelligence*, 44(4):2228–2242, 2020.

[24] Xiankai Lu, Wenguan Wang, Jianbing Shen, David J Crandall, and Luc Van Gool. Segmenting objects from relational visual data. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7885–7897, 2021.

[25] Aiyesha Ma, Nilesh Patel, Mingkun Li, and Ishwar K Sethi. Confidence based active learning for whole object image segmentation. In *Multimedia Content Representation, Classification and Security: International Workshop, MRCS 2006, Istanbul, Turkey, September 11-13, 2006. Proceedings*, pages 753–760. Springer, 2006.

[26] Tomasz Malisiewicz and Alexei A Efros. Recognition by association via learning per-exemplar distances. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[27] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Finding an unsupervised image segmenter in each of your deep generative models. *arXiv preprint arXiv:2105.08127*, 2021.

[28] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8364–8375, 2022.

[29] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[30] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2013.

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[32] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

[33] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. In *arXiv:1503.00848*, March 2015.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[35] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[36] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021.

[37] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[38] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[39] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[40] Jinhwan Seo, Wonho Bae, Danica J Sutherland, Junhyug Noh, and Daijin Kim. Object discovery via contrastive learning for weakly supervised object detection. In *European conference on computer vision*, pages 312–329. Springer, 2022.

[41] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[42] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):717–729, 2015.

[43] Gyungin Shin, Samuel Albanie, and Weidi Xie. Unsupervised salient object detection with spectral cluster voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3971–3980, 2022.

[44] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279*, 2021.

[45] Oriane Siméoni, Chloé Sekkat, Gilles Puy, Antonín Vobecký, Éloi Zablocki, and Patrick Pérez. Unsupervised object localization: Observing the background to discover objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3176–3186, 2023.

[46] Andrew N Stein, Thomas S Stepleton, and Martial Hebert. Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[47] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[48] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 347–365. Springer, 2020.

[49] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[50] 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.

[51] Huy V Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 779–795. Springer, 2020.

[52] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 136–145, 2017.

[53] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7303–7313, 2021.

[54] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14176–14186, 2022.

[55] Xudong Wang, Ishan Misra, Ziyun Zeng, Rohit Girdhar, and Trevor Darrell. Videocutler: Surprisingly simple unsupervised video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22755–22764, 2024.

[56] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *IEEE transactions on pattern analysis and machine intelligence*, 2023.

[57] Xiu-Shen Wei, Chen-Lin Zhang, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Unsupervised object discovery and co-localization by deep descriptor transformation. *Pattern Recognition*, 88:113–126, 2019.

[58] Lin Xi, Weihai Chen, Xingming Wu, Zhong Liu, and Zhengguo Li. Online unsupervised video object segmentation via contrastive motion clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(2):995–1006, 2023.

[59] Yunqiu Xu, Chunluan Zhou, Xin Yu, and Yi Yang. Cyclic self-training with proposal weight modulation for cross-supervised object detection. *IEEE Transactions on Image Processing*, 32:1992–2002, 2023.

[60] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7177–7188, 2021.

[61] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7177–7188, 2021.

[62] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013.

[63] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 879–888, 2019.

[64] Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. Deformable sprites for unsupervised video decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2657–2666, 2022.

[65] Stella X Yu and Jianbo Shi. Understanding popout through repulsion. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–II. IEEE, 2001.

[66] Xiao Zhang, David Yunis, and Michael Maire. Deciphering'what'and'where'visual pathways from spectral clustering of layer-distributed neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4165–4175, 2024.

[67] Xin Zhang, Jinheng Xie, Yuan Yuan, Michael Bi Mi, and Robby T Tan. Heap: Unsupervised object discovery and localization with contrastive grouping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7323–7331, 2024.

[68] Hasib Zunair and A Ben Hamza. Peekaboo: Hiding parts of an image for unsupervised object localization. *arXiv preprint arXiv:2407.17628*, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: It accurately reflects our contribution to unsupervised whole object segmentation, using attraction and repulsion as binding power.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We disucsse the limitation that our framework requires paired images. Nevertheless, this limiatation can be properly handeled in real-applications.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We have introduced clear and substential assumptions and proofs in our methodology part.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have detailed contents for reproducibility at the supplementary part.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Justification: The datasets used in this paper are all publicly available. We plan to release the code after the paper being reviewed.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the details about settings are included in the experiment section in our paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The reported error bars are also included in the supplementary part (Fig. A5).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Can be found in the implementation part in the supplementary section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This research conducted conforms in every respect with the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This work could help improve how machines perceive and interpret entire object structures, potentially leading to more accurate and reliable visual perception systems.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: This research has very little chance to be misused.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: These are all properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: they are well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [No]

Justification: This paper is not related to crowdsourcing experiments nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Haven't found any potential risks.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Technical Appendices and Supplementary Material

A.1 Unsupervised Whole Objectness by Contrastive Contextual Grouping

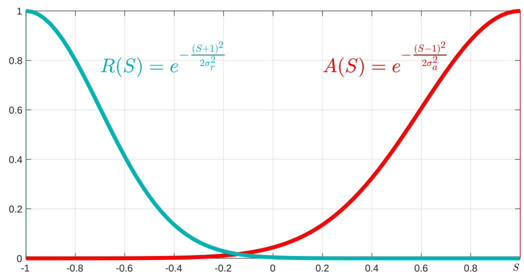

Figure A1: **We define attraction $\boldsymbol{A}$ and repulsion $\boldsymbol{R}$ as the Gaussian functions of pairwise feature similarity $\boldsymbol{S}$.** The larger (smaller) the similarity, the larger the attraction (repulsion). Here $\sigma_a = 0.4$. $\sigma_r = 0.3$.

Attraction and Repulsion. Given the similarity matrix $\boldsymbol{S}$, attraction and repulsion matrices $\boldsymbol{A}$ and $\boldsymbol{R}$ are defined as Gaussian functions of $\boldsymbol{S}$ (Fig.A1). To adjust the relative importance between attraction and repulsion, we introduce a repulsion weight factor ω, where $\omega \in [0, 1]$. We take $\omega = 0.2$ and the ablation study for the repulsion weight ω is shown in Fig. A3.

Segmentation by Only Attraction. Previous methods [56, 28] formulate unsupervised object discovery as a graph partitioning problem and use normalized cut [41] to divide the graph into two parts. Let $C_A(\mathbb{V}_1, \mathbb{V}_2)$ as total connections of attraction from $\mathbb{V}_1$ to $\mathbb{V}_2$: $\sum_{i \in \mathbb{V}_1, j \in \mathbb{V}_2} \boldsymbol{A}(i, j)$. The normalized cut is equivalent to maximizing the attraction within partitioned groups by

$$\max \xi_A = \sum_{u=1}^{2} \frac{C_A(\mathbb{V}_u, \mathbb{V}_u)}{C_A(\mathbb{V}_u, \mathbb{V})} \tag{7}$$

The features from self-supervised Visual Transformers present strong feature attraction in discriminative parts of objects. TokenCut [56] utilizes attraction for graph cut which can only segments out characteristic local regions, not whole objects. An example of illustrating how TokenCut segment object parts is in Fig. 4.

Segmentation by Attraction and Repulsion. Instead of using normalized cut by using only attraction, we investigate whether attraction and repulsion can jointly contribute to pop out whole objects. Given attraction $\boldsymbol{A}$ and repulsion $\boldsymbol{R}$, we follow [65] and conduct a binary segmentation by using a unified grouping criterion

$$\begin{aligned} \max \xi_{AR} =& \frac{\text{within-group } \boldsymbol{A}}{\text{total degree of } \boldsymbol{A} \& \boldsymbol{R}} + \frac{\text{between-group } \boldsymbol{R}}{\text{total degree of } \boldsymbol{A} \& \boldsymbol{R}} \\ =& \sum_{u=1}^{2} \frac{C_A(\mathbb{V}_u, \mathbb{V}_u)}{C_A(\mathbb{V}_u, \mathbb{V}) + C_R(\mathbb{V}_u, \mathbb{V})} + \\ & \frac{C_R(\mathbb{V}_u, \mathbb{V} \backslash \mathbb{V}_u)}{C_A(\mathbb{V}_u, \mathbb{V}) + C_R(\mathbb{V}_u, \mathbb{V})}, \end{aligned} \tag{8}$$

where $C_R(\mathbb{V}_1, \mathbb{V}_2)$ represents total connections of repulsion from $\mathbb{V}_1$ to $\mathbb{V}_2$. It's easy to discover that ξ_{AR} is equivalent to ξ_A when the strength of repulsion $\boldsymbol{R}$ is not considered for grouping (we set up $\omega_r = 0$). Let $\boldsymbol{D}_A, \boldsymbol{D}_R$ represent the diagonal degree matrix of $\boldsymbol{A}, \boldsymbol{R}$:

$$\begin{aligned} \boldsymbol{D}_A &= \mathtt{diag}(\mathtt{sum}(\boldsymbol{A}, \mathtt{dim} = 1)), \\ \boldsymbol{D}_R &= \mathtt{diag}(\mathtt{sum}(\boldsymbol{R}, \mathtt{dim} = 1)). \end{aligned} \tag{9}$$

According to [65], the joint attraction and repulsion criterion is equivalent to

$$\begin{aligned} \max \xi_{AR}(\boldsymbol{p}) &= \sum_{u=1}^{2} \frac{\boldsymbol{p}_u^T \boldsymbol{W} \boldsymbol{p}_u}{\boldsymbol{p}_u^T \boldsymbol{D} \boldsymbol{p}_u}, \\ \boldsymbol{W} = \boldsymbol{A} - \boldsymbol{R} &+ \boldsymbol{D}_R, \ \boldsymbol{D} = \boldsymbol{D}_A + \boldsymbol{D}_R, \end{aligned} \tag{10}$$

Table A1: Discovering image pairs using CLIP and pre-trained ResNet-50, both supervised methods, yields better performance than DINO, which is trained in a self-supervised manner.

feature	$\max F_\beta$	IoU	Acc.
DINO [5]	83.1	73.2	94.7
ResNet-50 [12]	83.4	**74.2**	95.6
CLIP [34]	**83.8**	73.8	**95.8**

Table A2: The ablation study to evaluate the impact on the distillation process by utilizing additional `Conv` layers in the segmentation head. Our implementation uses a single `Conv` layer in all tasks.

seg. head	$\max F_\beta$	IoU	Acc.
$1 \times$ Conv(1,1)	94.5	83.9	95.8
$2 \times$ Conv(1,1)	**95.2**	**84.4**	**96.3**
$3 \times$ Conv(1,1)	92.3	81.5	92.7

Table A3: **CCG demonstrates exceptional performance in unsupervised unclei segmentation on Pan-Nuke dataset.** We surpass the state-of-the-art method, UNSEG, by at leas 20% without requiring any training nor annotations.

	accuracy	mIoU	F_1 score
UNSEG [18]	43.6	41.4	48.2
CCG-1	58.3 (+14.7)	54.5 (+13.1)	57.9 (+9.7)
CCG-2	61.1 (+17.5)	56.9 (+15.5)	58.6 (+10.4)

where $\boldsymbol{p}_u$ is a binary membership vector for $\mathbb{V}_u$. The real valued solution to this partition problem is finding the second largest eigenvector $\mathbf{z}^*$ of the eigensystem

$$\boldsymbol{D}^{-1}\boldsymbol{W}\boldsymbol{z} = \lambda \boldsymbol{z}. \tag{11}$$

Attraction and Repulsion within a Single Image. Given an unlabeled image x, we assume it contains at least one object, and segment the whole objects by attraction and repulsion from x.

Attraction and Repulsion across Image Pairs. So far we consider attraction and repulsion within a single image. It is straightforward to extend it to a co-segmentation setting, where two or more related images need to be jointly segmented.

A.2 Implementation Details

We choose `ViT-S/16` as the architecture for evaluation with the baselines in *w/o. training* setting and `ViT-S/8` to compare with the baselines in *w/. training* setting. To find semantically similar but visually distinct images as image pairs, we extract the features from DINO (`ViT-S/8`) and run k-nearest neighbors. It takes less than 1 hour to run k-nearest neighbors on 100,000 images as a preprocessing step. To find video frame pairs, we use a frame interval of 10 to create reference image pairs for co-segmentation: [(00.jpg, 10.jpg), (01.jpg, 11.jpg), (02.jpg, 12.jpg), $\cdots$]. Our ViT encoder at the distillation stage takes the same architecture as DINO `ViT-S/8`. The segmentation head contains a single `conv` 1×1 layer. During the distillation, our ViT encoder is trained using AdamW optimizer with a learning rate of 0.001, and our segmentation head trained using AdamW optimizer with a learning rate of 0.05. We set the batch size to 16 and have 300 training epochs. The repulsion weight ω is set to 0.2. The segmentation head contains a single `conv` 1×1 layer. During the distillation process, we set the batch size to 16 and have 300 training epochs. The training is run on 4 A40 NVIDIA GPUs. The repulsion weight ω is set to 0.2.

A.3 Ablation study

Video Frame Pair Discovery. CCG employs a pair of frames taken from the same video clip, which may be captured at different timestamps. We examine how varying frame intervals affect unsupervised video object segmentation, as illustrated in Fig. A5 in the supplementary material. When the frame interval is set to 0, CCG-2 becomes equivalent to CCG-1 , as the two reference images are identical. The best results are obtained with video frame intervals ranging from 8 to 18. Therefore, we set the frame interval to 10 for all unsupervised video object segmentation experiments.

Eigenvectors. As shown in Fig. A6, the eigenvectors of CCG-2 highlight the entire body of the dogs, whereas TokenCut's eigenvectors are only able to segment the dog heads.

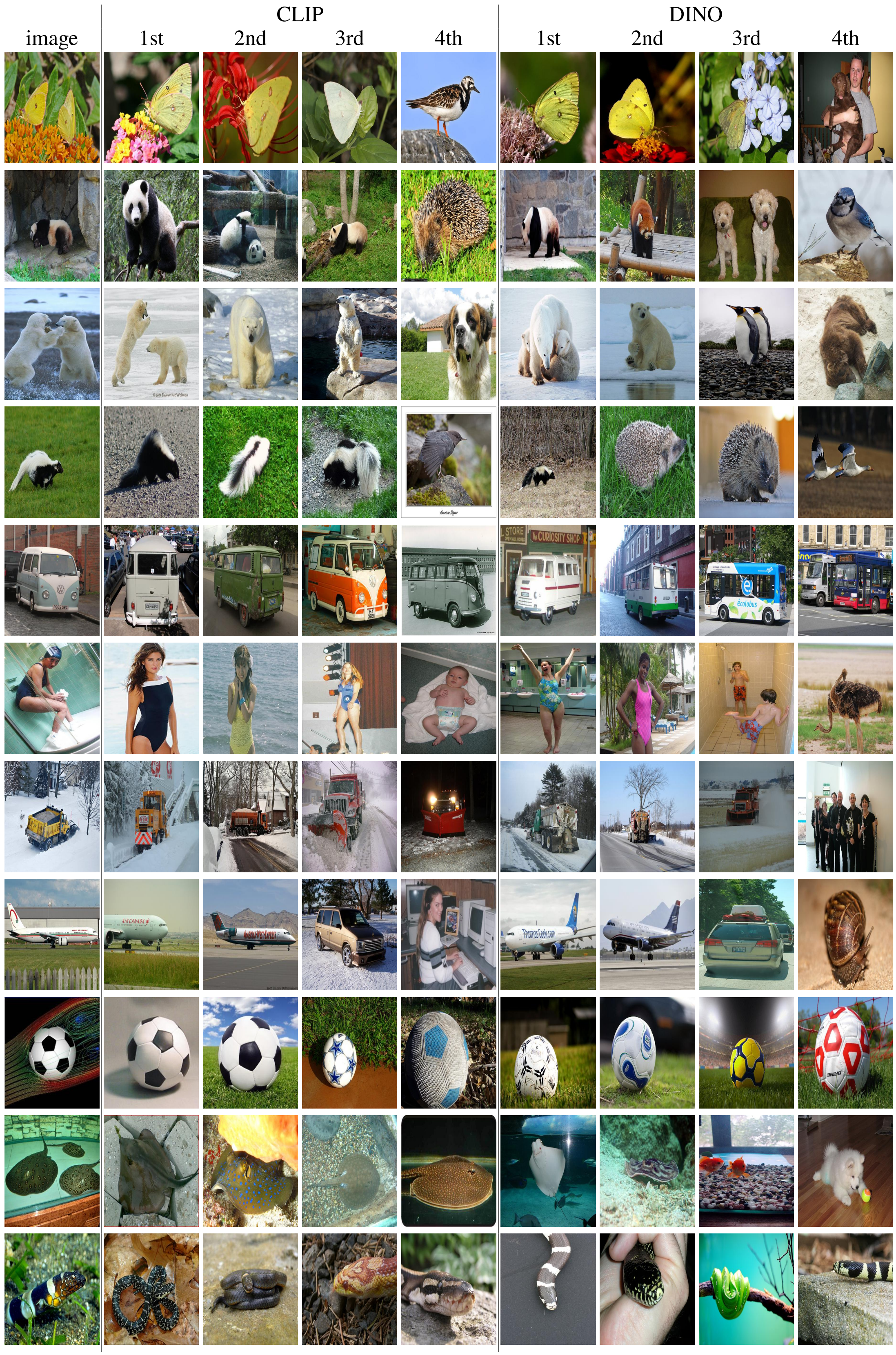

Figure A2: Identifying image pairs from unlabeled images using CLIP image embeddings or self-supervised ViT features from DINO.

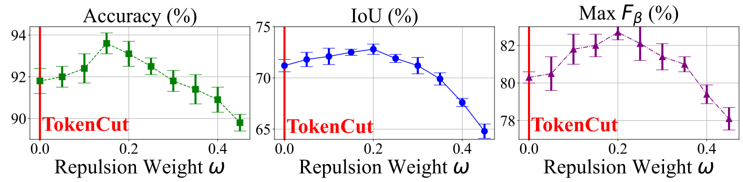

Figure A3: The unsupervised saliency detection performance of CCG on ECSSD dataset with different values of repulsion weight ω. CGR is the same as TokenCut when $\omega = 0$ because the impact of repulsion is set to zero in grouping.

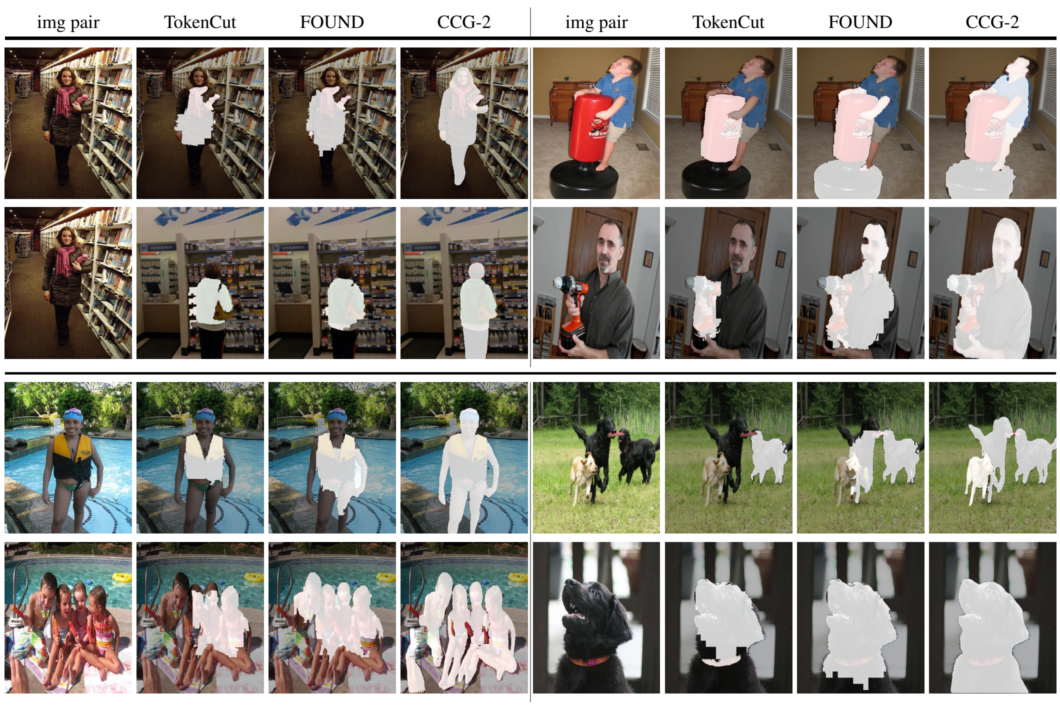

Figure A4: Qualitative results of CCG outperforming both TokenCut and FOUND on unsupervised saliency detection.

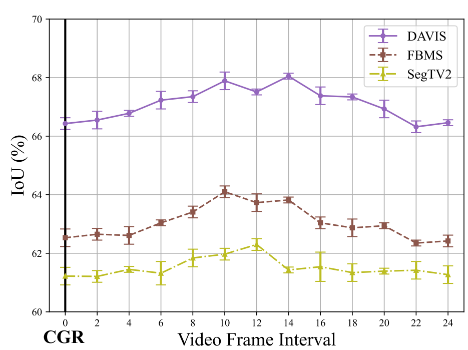

Figure A5: The performance of CCG-2 with video frames at different video frame intervals for unsupervised video object segmentation. CCG-2 is the equivalent to CCG-1 when the video frame interval is 0.

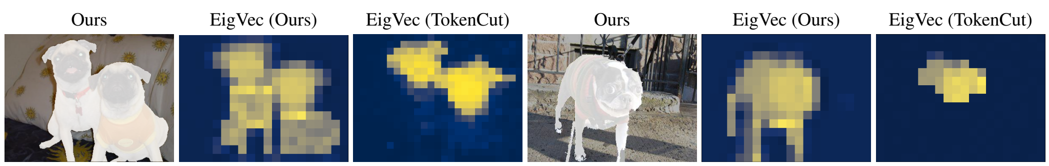

Figure A6: The eigenvectors of CCG-2 using attraction and repulsion across reference images pop out the whole body of the dogs while the eigenvectors of TokenCut utilizing attraction pop out only the head part of the dogs.