Multi-Part Object Representations via Graph Structures and Co-Part Discovery

Alex Foo **Wynne Hsu** **Mong Li Lee**
School of Computing
National University of Singapore
{dcsafdw,dcshsuw,dcsleeml}@nus.edu.sg

Abstract

Discovering object-centric representations from images can significantly enhance the robustness, sample efficiency and generalizability of vision models. Works on images with multi-part objects typically follow an implicit object representation approach, which fails to recognize these learned objects in occluded or out-of-distribution contexts. This is due to the assumption that object part-whole relations are implicitly encoded into the representations through indirect training objectives. We address this limitation by proposing a novel method that uses explicit graph representations for parts and introduces a new co-part algorithm for multi-part object discovery. The part representations are refined with a part-centric regularization objective. We then introduce three benchmarks to evaluate the robustness of object-centric methods in recognizing multi-part objects within occluded and out-of-distribution settings. Experimental results on simulated, realistic, and real-world images show marked improvements in the quality of discovered objects compared to state-of-the-art methods, as well as the accurate recognition of multi-part objects in occluded and out-of-distribution contexts. We also show that the discovered object-centric representations can more accurately predict key multi-part object properties in a downstream task, highlighting the potential of our method to advance the field of object-centric representations. Code and data are available at https://alexfoo-dw.github.io/ECO-Net.

1 Introduction

Human interaction with the environment relies on the comprehension of diverse visual scenes, powered by a robust visual system which decomposes each scene into abstract and manipulable objects [25, 43]. Each object's features can be explicitly described, and different objects are recognized via the appearance and relative pose of their constituent parts at varying degrees of granularity [20]. Emulating this part-whole compositional system in the form of multi-part object representations can improve robustness, occlusion-aware perception, generalization to out-of-distribution images, and interpretability of machine learning algorithms for vision [17].

Recent work utilizes an implicit object representation approach in which scenes are decomposed either via: (a) information-bottleneck deep learning, where high-dimensional inputs are encoded into a fixed number of low-dimensional object representations then recombined for reconstruction [11, 10, 41, 42, 22], or (b) similarity clustering of pre-trained image features [40, 38, 45, 44, 14]. Information-bottleneck approaches face challenges with multi-part object recognition, while methods that rely on learned features struggle to compose part clusters into object wholes (see Figure 1). The fundamental limitation of these methods stems from their assumption that object part-whole relations are implicitly encoded into the object representations through indirect training objectives such as input reconstruction [49] or self-supervised contrastive learning [5]. While it is computationally challenging to develop algorithms that consistently compose parts into their respective objects, this

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

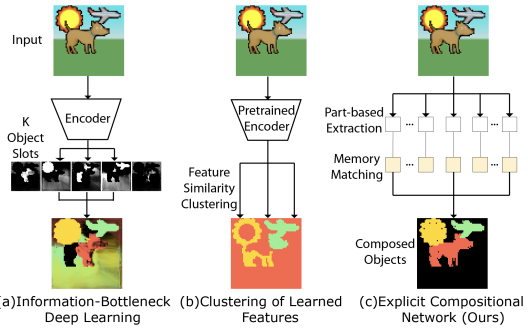


Figure 1: Comparison of existing methods. (a) Information-bottleneck approaches face challenges with multi-part object recognition, while (b) methods that rely on learned features struggle to compose part clusters into object wholes. Our proposed Explicit Compositional Network (ECO-Net) consistently and accurately composes parts into object wholes.

lack of part-whole structure in the unsupervised learning of objects leads to failure in recognizing objects in occluded and out-of-distribution contexts despite many training samples and iterations [17, 7, 23]. These limitations impede deployment of these methods to real-world applications where robust recognition of multi-part objects is often essential for deployment [48].

To address the limitations of implicit object representation, we propose Explicit Compositional Network (ECO-Net), a framework that uses explicit graph representations for parts and learns to discover multi-part objects via a novel co-part object discovery algorithm. Here, objects are defined as graphs, where the nodes define the features of each constituent part which make up the object, and the edges define the spatial relationships between the parts. Based on the intuition that neighboring parts that recurringly appear together across samples should belong to the same object [3, 29], we design a co-part object discovery algorithm which clusters parts together to form object wholes based on the repeated occurrence of parts and neighbors with similar positional relations. The part representations are refined with a part-centric regularization objective. Discovered objects are processed and integrated into a multi-part object memory module which is used for downstream tasks. Figure 1 shows that our proposed Explicit Compositional Network (ECO-Net) effectively composes parts into object wholes despite significant occlusion that confuses other methods.

Our contributions include: (1) a framework that utilizes explicit graph-based multi-part object representations, (2) a novel co-part object discovery algorithm combined with part-centric regularization to address the problem of multi-part object discovery, (3) introduction of AbsScene-O, GSO-O and AbsScene-C benchmarks for evaluating robustness of multi-part object discovery in occluded and out-of-distribution contexts, (4) experimental validation of our solution's superior performance compared to state-of-the-art methods in object discovery, occlusion-aware perception and out-of-distribution generalization, and (5) demonstration of the usefulness of learned object representations for predicting key object properties in a downstream task.

2 Related Work

Numerous works have shown remarkable success in segmenting real-world images, using supervised signals for image segmentation and object detection [19, 4, 28]. Unsupervised methods like SLIC [1] employ a modified k-means algorithm to cluster pixels, while Normalized Cut (Ncut) [39] and Felzenszwalb's method [13] use graph-based modeling on hand-crafted features for segmentation. Others use co-segmentation to jointly segment objects across images, but rely on intra-class objects being common to all images in the ground-truth [32, 6]. All these methods do not focus on obtaining useful object representations for the segmented components. Unsupervised methods to obtain object-centric representations can be categorized as follows:

• **Information-bottleneck deep learning.** This approach randomly initializes a set of compact object representations and uses iterative attention to bind these objects to different image regions. Slot Attention [34] utilizes cross-attention along the object dimension. While it is fast and can be extended to handle videos [27], it fails to discover objects when the training set is diverse. BO-QSA [22] initializes the slots of Slot Attention as learnable embeddings and uses bi-level optimization, resulting

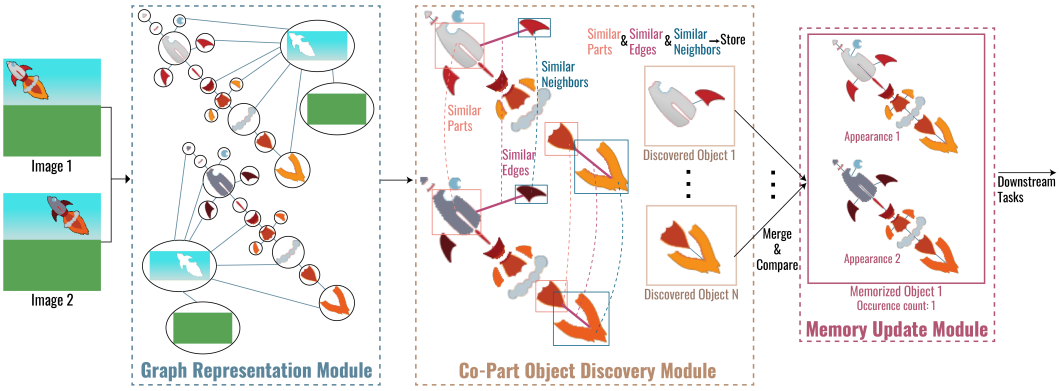


Figure 2: Overview of our proposed ECO-Net.

in more stable training. COCA-Net [30] employs a bottom-up hierarchy of compact clustering attention layers, introducing a spatial compactness prior to progressively group image features into object-centric representations. However, these methods require the number of clusters to be specified and fixed a priori, and their reliance on compact object representations limits their ability to capture part–whole relationships within multi-part objects.

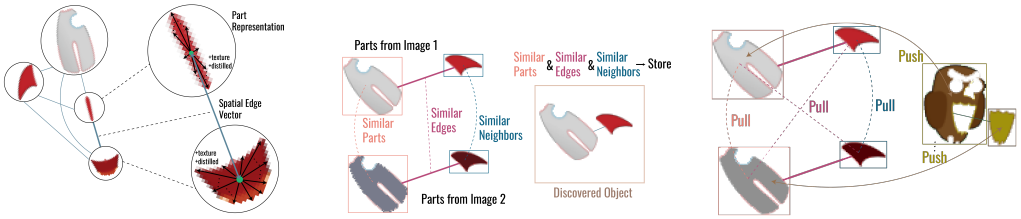
• **Clustering of learned features.** This approach clusters similar regions into objects based on learned features. DINOSAUR [38] utilizes self-supervised pre-training to obtain learned features before using iterative attention for feature reconstruction. MaskCut [44] extracts image features using DINO and applies Ncut for segmentation. OC-Net [14] leverages learned feature connectivity to discover objects and designs two object-centric regularization terms. The key limitation of feature-based clustering is the lack of modelling for complex part-whole relations of objects, which result in contrast-based segmentation of parts instead of composing parts into their proper objects.

3 Proposed Approach

Our approach extracts multi-part objects in images without relying on labeled data nor the need to specify the number of objects in the image. This is achieved by discovering part-whole relations via the recurring appearance of similarly related parts and neighbors across images in a data-driven learning manner. By not requiring the number of objects to be specified beforehand and accounting for part-whole relations, we can generalize to scenes with varying numbers of multi-part objects in real-world, occluded and out-of-distribution environments.

Figure 2 shows an overview of ECO-Net with three modules: (a) graph representation module that takes an image batch as input and clusters and represents fine-grained parts and their corresponding spatial relations in each image; (b) co-part object discovery module clusters parts and their corresponding relations that repeatedly occur into objects, where clustered parts are refined via part-centric regularization; and (c) memory update module that processes the current set of objects discovered and integrates it with the objects in memory.

Graph Representation. Felzenszwalb's algorithm has theoretical guarantees in producing segments that obey global properties. It ensures that all segments adhere to the boundary predicate, and are invariant to the order in which the segmentation is performed. The predicate is defined as follows: a segmentation boundary exists if the dissimilarity across pixels along the boundary of the two clusters is greater than the internal dissimilarity among neighboring pixels within each of the two regions. This is suitable for the initial discovery of parts since most part occurrences have homogeneous textures [6]. Given an input batch with images of N pixels whose coordinates are $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, we first process the image through a pointwise convolution with initial uniform weights to produce pixel embeddings $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$. We pass these embeddings through a modified Felzenszwalb's algorithm, where the segmentation is performed on an arbitrary number of pixel embedding dimensions, to obtain a batch-wise assignment of the input pixels into disjoint sets which we call parts $\mathcal{P} = \{P_1, \ldots, P_M\}$.



(a) Graph of parts. (b) Co-part object discovery. (c) Part-centric regularization.

Figure 3: Overview of object discovery. (a) Each node in the graph summarizes the position-invariant features (shape, texture, distilled) of each part, while each edge depicts the spatial relationships between parts. (b) Repeated occurrence of similar parts with similar spatial relationships are grouped into objects. (c) We maximize the semantic similarity between all parts which belong to the same object across images and simultaneously minimize the semantic similarity between all other parts.

Next, we define a graph of parts $\hat{\mathcal{G}} = (\mathcal{P}, \hat{\mathcal{E}})$ where each node summarizes the position-invariant features (shape, texture and distilled) of each part, and each edge has a vector that depicts the spatial relationship between parts (see Figure 3a). For each part $P_i \in \mathcal{P}$, we sample K boundary pixels with coordinates $\{\mathbf{x}_1^i, \ldots, \mathbf{x}_K^i\}$ ordered in clockwise direction. Let $\mathbf{x}_c^i$ be the coordinates of the centroid of P_i. Then the shape feature of P_i can be described in a position-invariant manner using K vectors that point from the centroid to each boundary pixel, expressed as a flattened vector $\mathbf{v}_i = \text{vec}(\left[\mathbf{x}_1^i - \mathbf{x}_c^i, \cdots, \mathbf{x}_K^i - \mathbf{x}_c^i\right])$. Let Ω_i be the set of constituent pixels in each part P_i. Then the texture feature of P_i is the average of all its constituent corresponding pixel embeddings $\mathbf{t}_i = \frac{1}{|\Omega_i|} \sum_{u \in \Omega_i} \mathbf{y}_u$. Let $\mathbf{d}_u$ be the feature vector associated by a pre-trained network [5] to pixel location u. Then the distilled feature of P_i is obtained via the average of all distilled features of the constituent pixels $\mathbf{d}_i = \frac{1}{|\Omega_i|} \sum_{u \in \Omega_i} \mathbf{d}_u$. The part representation of P_i is then a combination of shape, texture and distilled features, defined by: $\mathbf{z}_i = \mathbf{v}_i \oplus \mathbf{t}_i \oplus \mathbf{d}_i$, where $\oplus$ is the concatenation operation. An edge (P_i, P_j) exists if P_i is in the same image as P_j and the label of this edge is a vector $\mathbf{x}_c^j - \mathbf{x}_c^i$ that describes the spatial relationship of P_i and P_j.

Co-Part Object Discovery. This module finds repeated occurrences of parts with similar spatial relationships across the image batch containing inter-class multi-part objects and groups them into objects (see Figure 3b). Given the graph of parts $\hat{\mathcal{G}} = (\mathcal{P}, \hat{\mathcal{E}})$ from an input batch of images, a solution to the general problem of composing recurring parts into their respective objects requires checking all possible subgraphs. Unfortunately, there are $2^{|\hat{\mathcal{E}}|}$ possible subgraphs, so enumerating them is infeasible. This problem requires repeatedly solving the subgraph isomorphism problem as a subroutine, which is NP-complete and intractable [31]. To overcome this, we leverage the intuition that only neighboring parts that recur together across multiple samples are likely to belong to the same object. Theorem 3.1 formalizes this, with its proof provided in Appendix A.

Theorem 3.1 (Tractable pairwise iterative clustering algorithm)**.** *Given a graph of parts $\hat{\mathcal{G}} = (\mathcal{P}, \hat{\mathcal{E}})$ derived from a batch of non-artificially constructed images, there exists a pairwise iterative clustering algorithm that composes recurrent parts with similar spatial relations into foreground objects in $O(|\mathcal{P}|^2 \log |\mathcal{P}|)$ time, assuming each foreground object is represented by a connected subgraph of neighboring parts and the sufficient conditions in Appendix A hold.*

By Euler's formula [2] together with the degree-variation assumption in Appendix A, processing incident-part relations requires constant time per candidate pair. Combined with the object-update bound established there, this yields the stated runtime. Following Theorem 3.1, we design a clustering algorithm that takes as input the graph of adjacent parts $\mathcal{G} = (\mathcal{P}, \mathcal{E})$, where $\mathcal{E} \subset \hat{\mathcal{E}}$, and outputs the set of objects (see Appendix B). We uniformly sample an unprocessed part P_i and find parts that are similar based on their feature vectors. The similarity between a pair of parts is the cosine similarity between their corresponding node feature vectors: $\text{sim}(\mathbf{z}_i, \mathbf{z}_j) = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|}$. For each part P_j similar to P_i with value more than ϵ, let $\mathcal{E}_i$ and $\mathcal{E}_j$ be the set of edges connected to parts P_i and P_j respectively. For pairs of cosine-similar edges $(P_i, P_k) \in \mathcal{E}_i$ and $(P_j, P_{k'}) \in \mathcal{E}_j$, we consider that an object relation has been discovered if the corresponding neighbors P_k and $P_{k'}$ are similar. We express the discovered object as a combined set of its parts $\{\mathbf{z}_i, \mathbf{z}_k\} \in \mathcal{P}_c$ and connecting edge $\mathcal{E}_c$. If the parts $\mathbf{z}_i, \mathbf{z}_k$ have not been assigned to some object, we add the new object into the set of objects

Table 1: Summary of dataset characteristics

Dataset	Type	Ground Truth	Image Size	# Samples	Min Visible
Tetrominoes	Simulated	Pixel Mask	35×35	63K	100%
AbsScene	Simulated	Pixel Mask	64×64	63K	100%
GSO	Realistic	Pixel Mask	128×128	63K	100%
SKU-110K	Realistic	Bounding Box	128×128	11K	Varied
PASCAL VOC	Real-World	Pixel Mask	128×128	12K	Varied
MS COCO	Real-World	Pixel Mask	128×128	12K	Varied
AbsScene-O	Occluded Objects	Pixel Mask	64×64	63K	25%
GSO-O	Occluded Objects	Pixel Mask	128×128	63K	25%
AbsScene-C	Out-of-Distribution	Pixel Mask	64×64	1K	100%

found thus far $\mathcal{O}$. If either $\mathbf{z}_i$ or $\mathbf{z}_k$ already belong to some other object, we extend the first existing object by appending the new parts and connecting edge, update respective edges, and merge all other existing objects with $\mathbf{z}_i$ or $\mathbf{z}_k$ into the first. The process is repeated till all parts are processed.

Part-Centric Regularization. Intuitively, parts assigned to the same object should have more similar representations than parts assigned to different objects. This consistency should hold across object instances, both within and across images. Regularizing these representations over training iterations encourages more consistent co-part object discovery as more object samples are observed.

To this end, we perform gradient-based part-centric regularization to refine part representations across batches of images (see Figure 3c). We encourage similarity between parts assigned to the same object across images while discouraging similarity between parts assigned to different objects. For a batch of B images with M parts per image, let $\mathbf{z}_m^b$ denote the representation of part m in image b. For each representation, a positive $\hat{\mathbf{z}}_m^b$ assigned to the same object in another image is iteratively selected. Let $\mathcal{N}(b, m)$ index parts in other images assigned to different objects, excluding the selected positive. The part-centric regularization loss is defined as:

$$\mathcal{L} = -\sum_{b=1}^{B}\sum_{m=1}^{M} \log \frac{\exp(\mathbf{z}_m^b \cdot \hat{\mathbf{z}}_m^b/\tau)}{\exp(\mathbf{z}_m^b \cdot \hat{\mathbf{z}}_m^b/\tau) + \sum_{(a,h)\in\mathcal{N}(b,m)} \exp(\mathbf{z}_m^b \cdot \mathbf{z}_h^a/\tau)}.$$

where τ is the temperature constant for the regularization.

Memory Update. This module integrates the set of objects discovered in the current batch with those stored in memory. For each discovered object, we first take the union of all pixels from the constituent parts and obtain a position-invariant representation of the object shape, texture and distilled features similar to how part representations are obtained. Variations in object appearance and pose are also summarized into this object representation as different views of the same object. We then verify if the object exists in memory by comparing the similarity of its feature vectors against the feature vectors of the objects in memory. If a match is found, we increment the occurrence count. Otherwise, we add this new object to memory. Once all training samples have been processed, the object memory is sorted based on the number of occurrences to prioritize frequently occurring objects. This can then be utilized for downstream object discovery tasks.

4 Performance Study

We evaluate the performance of ECO-Net in terms of quality, occlusion-aware perception, generalizability and downstream property prediction on a diverse range of datasets:

• **Simulated datasets** Tetrominoes [24] with Tetris-like objects sampled from 6 colors and 17 shapes, and AbsScene [50] which has 10 multi-part objects.

• **Realistic datasets** GSO [8] comprises photo-realistic images of common objects with varied lighting and 3D object positions, and SKU-110K [16] is a challenging dataset with 1.7 million objects in dense retail scenes of various scales, angles and lighting.

• **Real-world multi-object datasets** PASCAL VOC [12] is the widely used benchmark for object segmentation with annotations for 20 object categories. MS COCO is the variant of the Microsoft Common Objects in Context dataset [33, 48].

• **Occluded objects datasets** AbsScene-O and GSO-O. We generate these occlusion-heavy variations from the original AbsScene and GSO datasets, where occlusion of objects is allowed up to 25% minimum visibility.

• **Out-of-distribution dataset** AbsScene-C. We generate this test set by randomly replacing the original background with 15 real-world complex-textured backgrounds.

Table 1 shows the dataset characteristics (Details in Appendix C).

Baselines. We compare ECO-Net with Ncut [39], Felzenszwalb [13], Slot Attention [34], GENESIS-V2 [11], BO-QSA [22], DINOSAUR [38], COCA-Net [30], MaskCut [44] and OC-Net [14]. Appendix D and Appendix E give the details of these baselines and the training protocols respectively. We train ECO-Net for 30K iterations with a batch size of 8, and set $\epsilon = 0.99$, $\tau = 0.1$ based on initial experiments (see Appendix E.1).

Evaluation Metrics. We evaluate object discovery using the permutation-invariant Adjusted Rand Index (ARI) [21]. We also use the Dice similarity coefficient, along with the Intersection-over-Union (IoU) between the best matching object masks (see Appendix E.7). We compute the mean Dice and the mean IoU scores, denoted as mDice and mIoU respectively, by averaging the scores across all matches. Reported $\pm$ values denote standard deviations across test images.

4.1 Quality of Discovered Objects

We first evaluate the ability of ECO-Net to discover objects in images. Table 2(a) shows the average ARI, mDice and mIoU scores based on the discovered multi-part objects in simulated datasets. ECO-Net achieves perfect score for all Tetrominoes test samples, and outperforms other methods by a large margin in AbsScene with near-perfect scores.

Table 2(b) shows the results on realistic multi-object datasets. For GSO, ECO-Net shows significantly superior performance ($p < 0.01$ by t-test) in all metrics, illustrating its capability to effectively discover common real-world objects. For SKU-110K, which contains multiple small densely-packed objects, ECO-Net significantly improves the ARI scores and more than doubles the mDice and mIoU scores over all methods, demonstrating its robustness in challenging object discovery tasks.

Table 2(c) shows the results on the real-world multi-object image dataset, highlighting the scalability of our method to challenging data where objects and their parts vary significantly across instances. ECO-Net shows superior performance for all metrics on both PASCAL VOC and MS COCO, illustrating its ability to discover challenging, commonly seen natural objects with diverse appearances. Notably, ECO-Net outperforms all baselines by a large margin in MS COCO, demonstrating its robustness in handling a large number of object categories.

Figure 4 visualizes the objects discovered by various methods for sample images. For Tetrominoes, despite the presence of scenes with connected same-color tiles that confuse existing methods, ECO-Net manages to separate each tile perfectly. For AbsScene, ECO-Net segments out the objects with multi-part relations in a fine-grained manner while other methods fail to consistently group together parts with varied appearances. For GSO, only ECO-Net is able to segment out the real-world objects and their boundaries with neighboring objects in a fine-grained manner. For SKU-110K, ECO-Net is able to segment out the small and densely-packed objects. For the real world datasets PASCAL VOC and MS COCO, ECO-Net is able to segment out the objects with highly varied appearances which other methods fail to discover. Additional samples are given in Appendix F.2.

4.2 Experiments on Occlusion-Aware Perception

Occlusion-aware perception evaluates how effectively an object-centric model can recognize objects when they are only partially visible. Importantly, when a model encounters an object with partially visible appearance, it should be able to identify the object based on memory. This can be quantified by measuring how well the model 'fills-in' the occluded parts. We assess the occlusion-aware perception of ECO-Net with the baselines by training on AbsScene and GSO, and testing on AbsScene-O and GSO-O respectively, using ground-truth objects with occluded parts completed.

Table 2: Evaluation scores of discovered foreground objects.

(a) Simulated datasets

Method	Tetrominoes			AbsScene		
	ARI	mDice	mIoU	ARI	mDice	mIoU
Ncut	66.4±19.7	62.8±17.8	56.5±19.1	23.7±17.8	25.7±8.9	15.7±6.9
Felzenszwalb	91.9±18.2	96.6±7.8	95.0±11.2	76.8±30.3	85.8±14.1	78.2±17.2
Slot Attention	99.7±1.5	41.4±1.5	26.6±1.5	94.8±11.8	74.6±10.8	63.9±13.4
GENESIS-V2	97.2±5.4	47.0±3.5	30.9±3.0	46.7±24.1	65.4±15.8	54.3±18.0
BO-QSA	99.1±3.7	40.8±1.4	25.8±1.2	93.4±13.8	83.1±9.2	74.0±12.4
DINOSAUR	92.5±9.9	40.0±2.0	25.3±1.8	95.0±8.2	51.9±6.2	35.5±5.7
COCA-Net	91.8±2.6	97.2±1.0	94.6±1.9	92.6±8.2	85.7±7.4	76.8±10.0
MaskCut	93.4±15.3	80.8±5.6	68.5±6.6	90.5±18.9	88.3±10.7	81.5±12.8
OC-Net	92.4±15.4	96.7±6.5	94.9±9.7	78.5±24.7	86.4±11.7	78.7±14.4
ECO-Net	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**98.0±2.3**	**99.5±0.9**	**99.0±1.4**

(b) Realistic datasets

Method	GSO			SKU-110K		
	ARI	mDice	mIoU	ARI	mDice	mIoU
Ncut	29.8±2.04	33.4±14.0	23.5±12.4	5.2±2.6	8.7±4.0	5.5±2.8
Felzenszwalb	61.2±34.3	70.3±16.9	58.8±18.1	-0.8±7.4	7.7±2.9	4.9±2.0
Slot Attention	64.3±28.8	39.5±13.2	27.0±10.8	-9.7±7.2	0.9±0.4	0.4±0.2
GENESIS-V2	90.1±8.3	84.3±4.3	73.6±5.9	6.1±2.7	6.4±2.2	3.5±1.3
BO-QSA	64.4±17.0	74.2±4.3	59.7±5.4	5.4±2.3	7.2±3.1	3.9±1.8
DINOSAUR	96.2±10.1	42.0±6.2	26.9±5.2	4.4±2.3	12.1±3.7	6.8±2.3
COCA-Net	86.0±14.2	89.5±8.1	80.4±9.9	5.5±1.3	7.5±2.3	4.1±1.4
MaskCut	97.5±5.6	90.9±1.8	83.4±3.0	3.7±10.0	0.7±1.1	0.3±0.7
OC-Net	86.6±22.9	86.1±10.2	77.9±13.1	7.4±3.3	13.2±4.3	7.7±2.7
ECO-Net	**99.3±1.7**	**94.6±2.1**	**90.0±3.7**	**7.8±3.1**	**32.0±8.9**	**21.7±6.6**

(c) Real-world datasets

Method	PASCAL VOC			MS COCO		
	ARI	mDice	mIoU	ARI	mDice	mIoU
Ncut	9.2±15.4	10.6±6.6	6.7±4.8	12.9±17.1	27.2±16.2	19.3±13.8
Felzenszwalb	9.2±22.8	13.6±11.5	9.4±9.6	21.1±23.4	28.6±20.4	21.6±17.5
Slot Attention	14.8±22.4	12.6±8.3	7.7±5.6	28.5±14.2	23.9±10.2	14.5±7.1
GENESIS-V2	13.2±20.1	14.6±9.5	9.4±6.8	13.2±12.5	21.8±9.7	13.9±7.2
BO-QSA	13.0±19.5	13.7±8.9	8.6±6.2	24.0±15.3	33.3±11.6	22.5±9.4
DINOSAUR	16.9±24.0	17.0±10.5	11.4±7.7	30.6±16.9	23.0±9.3	14.2±6.6
COCA-Net	14.1±14.8	18.6±10.3	13.3±9.0	26.9±12.1	50.0±12.5	36.7±12.1
MaskCut	17.7±28.5	19.2±15.0	15.5±13.7	37.5±23.4	43.0±18.6	33.2±16.2
OC-Net	16.6±25.7	20.7±13.9	15.1±11.6	26.5±19.3	48.1±13.6	35.5±15.1
ECO-Net	**19.4±19.5**	**28.4±13.0**	**20.2±10.6**	**39.3±17.2**	**64.4±11.6**	**50.7±11.9**

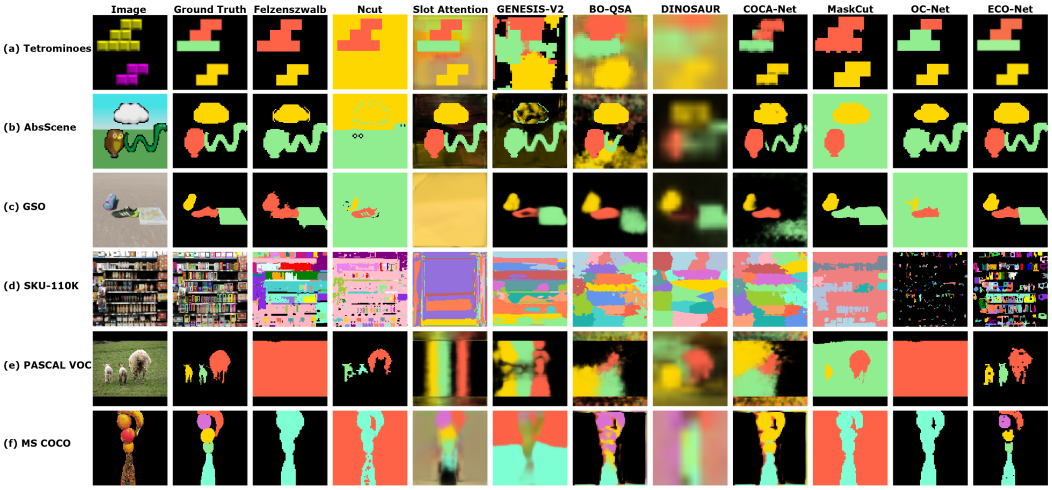


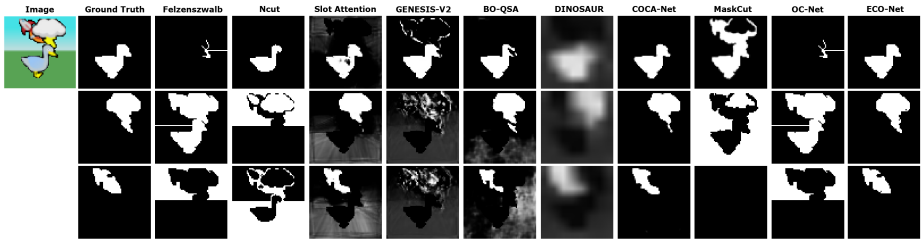
Figure 4: Visualization of discovered objects.

Table 3: Results for discovered objects with occlusion.

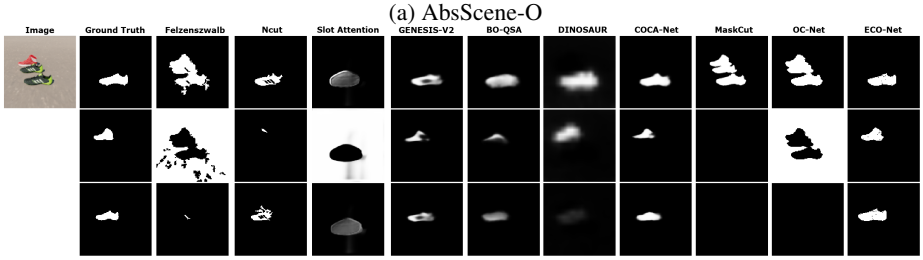
Method	AbsScene-O		GSO-O	
	mDice	mIoU	mDice	mIoU
Ncut	26.8±9.4	16.6±7.4	34.0±15.4	23.3±13.7
Felzenszwalb	70.4±20.7	59.5±22.1	71.6±19.8	60.3±20.9
Slot Attention	54.7±14.4	41.6±15.0	40.5±15.7	28.1±12.4
GENESIS-V2	48.9±19.4	40.3±20.6	81.4±5.7	69.4±7.0
BO-QSA	61.0±13.7	47.1±14.9	72.2±6.3	57.3±7.6
DINOSAUR	44.5±6.4	29.1±5.3	30.0±5.6	17.8±4.1
COCA-Net	72.7±17.0	61.6±16.7	73.9±16.3	63.2±16.4
MaskCut	68.6±21.8	59.1±23.9	85.8±3.6	76.9±4.6
OC-Net	69.9±21.0	59.2±22.5	84.5±5.3	76.2±7.0
ECO-Net	**93.2±9.3**	**90.7±12.3**	**90.4±6.5**	**83.5±8.8**

Table 4: Results for discovered objects on out-of-distribution test set AbsScene-C.

Method	ARI	mDice	mIoU
Ncut	40.6±23.6	51.7±18.9	40.9±19.4
Felzenszwalb	60.7±34.0	68.6±25.1	60.8±27.1
Slot Attention	37.9±21.0	41.4±12.4	28.0±11.5
GENESIS-V2	23.2±14.3	32.9±8.3	20.5±6.3
BO-QSA	32.5±16.9	38.3±10.0	25.0±8.6
DINOSAUR	82.0±17.5	54.2±6.3	37.8±6.0
COCA-Net	57.5±21.1	51.1±13.7	37.2±13.3
MaskCut	67.0±39.9	74.9±17.8	64.0±20.0
OC-Net	60.8±34.2	69.0±24.5	61.1±26.7
ECO-Net	**93.7±7.3**	**97.5±4.0**	**95.6±6.3**



(a) AbsScene-O



(b) GSO-O

Figure 5: Visualization of discovered objects with occlusion.

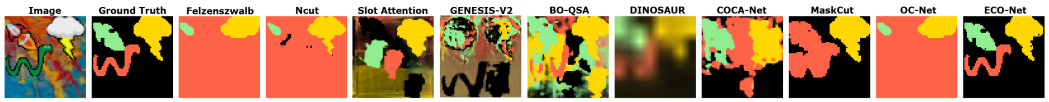


Figure 6: Visualization of discovered objects on out-of-distribution test set AbsScene-C.

Table 3 shows the results with ECO-Net outperforming all baselines by a large margin. Figure 5 and Appendix F.3 show the occluded objects discovered by the various methods. We see that the overlapping of multi-part objects continues to confuse existing SOTA. Notably, BO-QSA, DINOSAUR and COCA-Net struggle to discern the borders of overlapping objects while MaskCut and OC-Net fail to separate the objects for both AbsScene-O and GSO-O samples. In contrast, ECO-Net discovers complete objects and predicts the parts that are occluded for each object accurately.

4.3 Out-of-Distribution Discovery

An effective object-centric representation should generalize across different scenes by accurately encoding object appearances. Here, we demonstrate that the object representations learned from AbsScene by ECO-Net are robust and can be used to discover objects in AbsScene-C, which consists of random complex-texture backgrounds.

Table 4 shows a general performance decline across all models due to the shift in data distribution. However, ECO-Net has the smallest drop in performance and significantly surpasses the best performing method by a large margin. Figure 6 illustrates how, despite the complex-textured backgrounds that confuse other state-of-the-art methods, ECO-Net remains robust in discovering the complete objects. Additional visualizations are included in Appendix F.4.

Table 5: R^2 scores for object property prediction on simulated datasets.

Method	Tetrominoes						AbsScene		
	Red	Green	Blue	X-coord	Y-coord	Shape	X-coord	Y-coord	Shape
Slot Attention	85.0	80.8	93.5	99.1	98.3	36.2	97.2	96.3	76.1
GENESIS-V2	84.5	94.7	84.7	96.3	92.7	37.9	95.5	38.3	77.5
BO-QSA	97.3	98.4	98.6	98.7	99.0	52.4	90.1	98.1	96.2
DINOSAUR	91.2	89.6	86.9	93.1	94.3	20.1	97.5	98.0	95.4
COCA-Net	97.4	97.7	98.9	98.3	99.1	82.7	91.5	92.3	96.9
OC-Net	91.2	92.3	91.8	89.7	89.9	81.8	90.7	90.4	95.7
ECO-Net	**100.0**	**100.0**	**100.0**	**99.5**	**99.4**	**98.7**	**98.5**	**98.1**	**99.7**

Table 6: mIoU scores for variants of ECO-Net.

Method	Tetrominoes	AbsScene	SKU-110K	PASCAL	COCO	AbsScene-O	GSO-O	AbsScene-C
ECO-Net	**100.0±0.0**	**99.0±1.4**	**21.7±6.6**	**20.2±10.6**	**50.7±11.9**	**90.7±12.3**	**83.5±8.8**	**95.6±6.3**
w/o distilled	97.1±9.0	93.1±7.5	18.9±8.0	18.1±12.0	33.8±13.4	81.6±11.5	77.3±10.2	88.8±12.6
w/o texture	88.6±12.9	80.0±19.1	18.1±9.2	18.2±12.3	41.1±14.3	73.3±15.7	69.9±11.9	83.8±13.1
w/o shape	57.3±12.7	72.8±16.7	2.8±0.4	18.8±13.8	47.4±13.9	58.0±21.6	79.8±12.6	60.1±20.3
w/o co-part	40.6±29.2	61.6±23.4	7.8±2.8	5.9±6.3	6.8±9.3	51.6±20.1	25.1±25.8	31.9±27.1

4.4 Experiments on Object Property Prediction

A key trait of effective object-centric representation is its ability to encode object properties such as color, position and shape [37]. Here, we show that the learned object representations are disentangled and can be used to predict the values of these properties. We use the learned object representations from the simulated datasets to predict the properties of objects. Details are in Appendix G.

Table 5 shows the average R^2 scores of GBT models on the evaluation set. Predictions from ECO-Net representations achieved the highest or joint-highest R^2 scores across all properties. This suggests that ECO-Net's object representations are effective in encoding object properties.

4.5 Ablation Studies

Next, we examine the effect of co-part object discovery and of the different features on ECO-Net. We implemented four variants of ECO-Net: (a) w/o distilled. This network excludes the pre-trained feature $\mathbf{d}$ in the part representations $\mathbf{z}$. (b) w/o texture. This network excludes the texture feature $\mathbf{t}$ in the part representations $\mathbf{z}$. (c) w/o shape. This network excludes the shape feature $\mathbf{v}$ in the part representations $\mathbf{z}$. (d) w/o co-part. Here, we do not perform the clustering of parts into object wholes.

Table 6 shows the mIoU scores for all the datasets. Without the distilled feature, we observe a general decrease in performance due to the reduction of benefit in part discovery obtained via features from large-scale pre-training. Similarly, the lack of texture features leads to a general decrease in the ability to identify similar parts. Without the shape feature, we observe a significant drop in performance across almost all datasets since a reliance on the pretraining distribution causes a decrease in consistent discovery of parts from the same object. As such, ECO-Net w/o shape struggles to cluster parts accurately into object wholes. Removing the co-part object discovery module (ECO-Net w/o co-part) leads to a large performance drop in all datasets, indicating the importance of leveraging on parts to consistently discover multi-part objects especially in occluded and out-of-distribution settings.

5 Conclusion

We have described a framework that leverages explicit graph representations for parts and introduces a co-part object discovery algorithm to iteratively build up parts into objects. Part representations are refined via a part-centric regularization objective that enables more consistent discovery of parts over training iterations. Experiments on simulated, realistic and real-world datasets have demonstrated the superior quality of the objects discovered over state-of-the-art. We have also highlighted that our solution excels in occlusion-aware perception and out-of-distribution generalization. Finally, we show that the obtained object representations can predict key properties in a downstream task, indicating its potential use for applications where samples and labels are limited.

Acknowledgments and Disclosure of Funding

This research received no external grant funding. The authors declare no competing interests.

References

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. doi: 10.1109/TPAMI.2012.120.

[2] Bondy, J. A. and Murty, U. S. R. *Graph theory*. Springer Publishing Company, Incorporated, 2008.

[3] Brady, M. J. and Kersten, D. Bootstrapped learning of novel objects. *Journal of Vision*, 3(6): 2–2, 2003.

[4] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

[5] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

[6] Choudhury, S., Laina, I., Rupprecht, C., and Vedaldi, A. Unsupervised part discovery from contrastive reconstruction. *Advances in Neural Information Processing Systems*, 34:28104–28118, 2021.

[7] Dittadi, A., Papa, S. S., Vita, M. D., Schölkopf, B., Winther, O., and Locatello, F. Generalization and robustness implications in object-centric learning. In *International Conference on Machine Learning, ICML*, 2022.

[8] Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T. B., and Vanhoucke, V. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560. IEEE, 2022.

[9] Eastwood, C. and Williams, C. K. I. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

[10] Emami, P., He, P., Ranka, S., and Rangarajan, A. Efficient iterative amortized inference for learning symmetric and disentangled multi-object representations. In *International Conference on Machine Learning*, pp. 2970–2981, 2021.

[11] Engelcke, M., Parker Jones, O., and Posner, I. Genesis-v2: Inferring unordered object representations without iterative refinement. *Advances in Neural Information Processing Systems*, 34: 8085–8094, 2021.

[12] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[13] Felzenszwalb, P. F. and Huttenlocher, D. P. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[14] Foo, A., Hsu, W., and Lee, M. L. Multi-object representation learning via feature connectivity and object-centric regularization. In *NeurIPS*, 2023.

[15] Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232, 2001.

[16] Goldman, E., Herzig, R., Eisenschtat, A., Goldberger, J., and Hassner, T. Precise detection in densely packed scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5227–5236, 2019.

[17] Greff, K., Van Steenkiste, S., and Schmidhuber, J. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

[18] He, J., Yang, S., Yang, S., Kortylewski, A., Yuan, X., Chen, J.-N., Liu, S., Yang, C., and Yuille, A. Partimagenet: A large, high-quality dataset of parts. *ECCV*, 2022.

[19] He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[20] Hinton, G. Some demonstrations of the effects of structural descriptions in mental imagery. *Cognitive Science*, 3(3):231–250, 1979.

[21] Hubert, L. and Arabie, P. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[22] Jia, B., Liu, Y., and Huang, S. Improving object-centric learning with query optimization. In *The Eleventh International Conference on Learning Representations*, 2023.

[23] Jiang, J., Deng, F., Singh, G., and Ahn, S. Object-centric slot diffusion. In *NeurIPS*, 2023.

[24] Kabra, R., Burgess, C., Matthey, L., Kaufman, R. L., Greff, K., Reynolds, M., and Lerchner, A. Multi-object datasets. https://github.com/deepmind/multi-object-datasets/, 2019.

[25] Kahneman, D., Treisman, A., and Gibbs, B. J. The reviewing of object files: Object-specific integration of information. *Cognitive psychology*, 24(2):175–219, 1992.

[26] Karazija, L., Laina, I., and Rupprecht, C. Clevrtex: A texture-rich benchmark for unsupervised multi-object segmentation. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Curran, 2021.

[27] Kipf, T., Elsayed, G. F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., and Greff, K. Conditional Object-Centric Learning from Video. In *International Conference on Learning Representations (ICLR)*, 2022.

[28] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. *ICCV*, 2023.

[29] Kruger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., Rodriguez-Sanchez, A. J., and Wiskott, L. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1847–1871, 2012.

[30] Kucuksozen, C. and Yemez, Y. Hierarchical compact clustering attention (coca) for unsupervised object-centric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25388–25398, 2025.

[31] Kuramochi, M. and Karypis, G. Frequent subgraph discovery. In *Proceedings 2001 IEEE international conference on data mining*, pp. 313–320. IEEE, 2001.

[32] Li, W., Hosseini Jafari, O., and Rother, C. 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*, pp. 638–653. Springer, 2019.

[33] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

[34] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

[35] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024.

[36] Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 2019.

[37] Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. doi: 10.1109/JPROC.2021.3058954.

[38] Seitzer, M., Horn, M., Zadaianchuk, A., Zietlow, D., Xiao, T., Simon-Gabriel, C.-J., He, T., Zhang, Z., Schölkopf, B., Brox, T., et al. Bridging the gap to real-world object-centric learning. *arXiv preprint arXiv:2209.14860*, 2022.

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

[40] Siméoni, O., Puy, G., Vo, H. V., Roburin, S., Gidaris, S., Bursuc, A., Pérez, P., Marlet, R., and Ponce, J. Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279*, 2021.

[41] Singh, G., Deng, F., and Ahn, S. Illiterate dall-e learns to compose. In *International Conference on Learning Representations*, 2021.

[42] Singh, G., Kim, Y., and Ahn, S. Neural systematic binder. In *The Eleventh International Conference on Learning Representations*, 2023.

[43] Spelke, E. S. and Kinzler, K. D. Core knowledge. *Developmental science*, 10(1):89–96, 2007.

[44] Wang, X., Girdhar, R., Yu, S. X., and Misra, I. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3124–3134, 2023.

[45] Wang, Y., Shen, X., Hu, S. X., Yuan, Y., Crowley, J. L., and Vaufreydaz, D. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14543–14553, 2022.

[46] Wright, S. Correlation and causation. *Journal of agricultural research*, 1921.

[47] Xia, J., Wu, Y., Huang, W., Zhang, J., and Zhang, J. Unsupervised part discovery via descriptor-based masked image restoration with optimized constraints. In *ICCV*, 2025.

[48] Yang, Y. and Yang, B. Promising or elusive? unsupervised object segmentation from real-world single images. *Advances in Neural Information Processing Systems*, 35:4722–4735, 2022.

[49] Yuan, J., Chen, T., Li, B., and Xue, X. Compositional scene representation learning via reconstruction: A survey. *arXiv preprint arXiv:2202.07135*, 2022.

[50] Zitnick, C. L. and Parikh, D. Bringing semantics into focus using visual abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3009–3016, 2013.

A Tractable Co-Part Object Discovery

In this section, we provide the proof to Theorem 3.1 by formally analyzing the completeness and tractability of co-part object discovery via pairwise comparisons of neighboring parts.

Proof. Let $\hat{\mathcal{G}} = (\mathcal{P}, \hat{\mathcal{E}})$ be a graph of parts from an input batch of images, where $\mathcal{P} = \{P_1, \ldots, P_M\}$ is the set of parts with features $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$, and $\hat{\mathcal{E}} = \{(P_i, P_j)\}$, where $1 \leq i < j \leq M$, is the set of edges with corresponding spatial labels. Define the graph of adjacent parts $\mathcal{G} = (\mathcal{P}, \mathcal{E})$, where $\mathcal{E} \subset \hat{\mathcal{E}}$, and parts are adjacent if they share a boundary of nonzero length (i.e., there exists at least one pair of adjacent pixels).

Sufficient conditions. The guarantees of Theorem 3.1 are conditional on the following assumptions:

1. Each initial part lies entirely within one foreground object or within the background.
2. Each foreground object contains at least two parts. The within-image part pairs accepted by the algorithm's similarity tests contain a spanning tree of that object.
3. No accepted within-image part pair crosses an object boundary or contains a background part.
4. The graph of adjacent parts is simple and planar. Its maximum degree d satisfies $d \leq C\bar{d}$, where $\bar{d}$ is its average degree and C is a constant independent of the total number of parts M.
5. Feature dimensions and the number of parts per foreground object are bounded by constants independent of M. Object membership is maintained using indexed lookups.

An accepted within-image pair is (P_i, P_k) supported by a matching pair $(P_j, P_{k'})$ through the three similarity tests. The matching pair provides evidence for grouping; it does not cause parts from different images to be merged into the same object instance.

These are sufficient conditions for the theoretical guarantees, not properties ensured by segmentation or planarity alone.

Consider the following pairwise iterative algorithm that clusters neighboring parts with similar relations into objects:

1. **Pairwise Comparison:** For each pair of parts $[P_i, P_j]$, where $1 \leq i < j \leq M$, perform the following checks:
   - Compare the features $\mathbf{z}_i$ and $\mathbf{z}_j$ for similarity.
   - Compare the similarity of their edges with neighboring parts.
   - For each similar edge $(P_i, P_k) \in \mathcal{E}$ and $(P_j, P_{k'}) \in \mathcal{E}$, compare the features of the incident parts P_k and $P_{k'}$.
2. **Object Assignment:** If all three similarity conditions are satisfied, process the accepted within-image pair (P_i, P_k):
   (a) If neither part belongs to an existing object, create an object containing P_i, P_k, and their connecting edge.
   (b) Otherwise, add both parts and their connecting edge to an existing object containing either part. Merge any other object containing either part into this object, and complete its adjacency edges.

   The supporting pair $(P_j, P_{k'})$ is processed through the same procedure when considered as the within-image candidate pair.
3. **Iteration:** Repeat the above steps for every pairs of parts until all pairs have been considered.

Proof of Completeness. By condition (3), each accepted within-image part pair belongs to a single foreground object. Thus, creating clusters from accepted pairs and merging clusters that share a part cannot combine different objects or include background parts.

By condition (2), the accepted pairs within each foreground object contain a spanning tree. Since the algorithm examines all candidate part pairs and their incident-edge pairs, every edge of this spanning tree is eventually processed. The resulting merges therefore connect all parts of that object into one cluster. Condition (1) ensures that no initial part straddles different objects. Consequently, the final clusters recover the foreground objects under the stated conditions.

Proof of Tractability: Let $|\mathcal{P}| = M$ be the number of parts (segments), and let d be the maximum degree (number of neighbors) of a part in the graph of parts $\mathcal{G}$.

1. **Pairwise Comparison.**
   - The number of unordered pairs $[P_i, P_j]$ is $\binom{M}{2} = O(M^2)$.
   - For each pair:
     - Comparing features $\mathbf{z}_i$ and $\mathbf{z}_j$ is $O(1)$.
     - Comparing the similarity of their neighbor edges involves up to d neighbors each, i.e., $O(d^2)$ operations per pair.
     - For each such neighbor pair $((P_i, P_k), (P_j, P_{k'}))$, comparing P_k and $P_{k'}$ is $O(1)$.
   - Total work per pair is $O(d^2)$.
2. **Object Assignment.**
   - Under condition (5), indexed object-membership lookups and bounded-size object insertions and merges require at most $O(\log M)$ amortized time per accepted relation.
   - Completing adjacency edges requires scanning only a bounded number of object parts under condition (5). Since each candidate pair has at most d^2 accepted relations, these updates require $O(d^2 \log M)$ amortized time per candidate pair.
3. **Total Time Complexity:** $O(M^2 d^2) + O(M^2 d^2 \log M)$.

Degree bound from Euler's formula. For a simple planar graph with $M \geq 3$, Euler's formula implies $|E| \leq 3M - 6$. Hence its average degree satisfies

$$\bar{d} = \frac{2|E|}{M} < 6.$$

By condition (4),

$$d \leq C\bar{d} < 6C = O(1).$$

Graphs with fewer than three vertices also have bounded degree. Thus, each candidate pair requires only a constant number of incident-edge comparisons.

Conclusion. Under conditions (1)–(5), the algorithm correctly composes recurrent parts into foreground objects with total runtime

$$O(M^2 d^2) + O(M^2 d^2 \log M) = O(M^2 \log M).$$

This establishes the claimed completeness and tractability under the stated assumptions.

□

B Co-part Object Discovery Algorithm

We design a co-part object discovery algorithm which iteratively clusters recurrent parts into objects. Details of the algorithm is given below.

Algorithm 1 Co-Part Object Discovery.

Input: Set of parts $\mathcal{P} = \{P_1, \ldots, P_M\}$; set of part features $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$; set of edges for adjacent parts $\mathcal{E} = \{(P_i, P_j) \mid P_i \text{ and } P_j \text{ are adjacent}\}$
Output: Set of objects $\mathcal{O} = \{\mathcal{O}_1, \ldots, \mathcal{O}_c\}$
$\mathcal{U} \leftarrow \mathcal{P}$; $\mathcal{O} \leftarrow \emptyset$; $c \leftarrow 1$ // Initialization
repeat
  $P_i \leftarrow$ UniformSampling($\mathcal{U}$)
  let $\mathcal{Z}' \subset \mathcal{Z} - \{\mathbf{z}_i\}$ be the set of part features with similarity more than ϵ to $\mathbf{z}_i$
  for each $\mathbf{z}_j \in \mathcal{Z}'$ **do**
    let $\mathcal{E}_i, \mathcal{E}_j \subset \mathcal{E}$ be the set of edges connected to parts P_i and P_j respectively
    for each pair of edges $((P_i, P_k), (P_j, P_{k'})) \in \mathcal{E}_{i,j} \subset \mathcal{E} \times \mathcal{E}$ with similarity more than ϵ **do**
      if $\text{sim}(\mathbf{z}_k, \mathbf{z}_{k'}) > \epsilon$ **then**
        if $\mathbf{z}_i$,$\mathbf{z}_k$ do not exist in some object **then**
          $\mathcal{P}_c \leftarrow \{\mathbf{z}_i, \mathbf{z}_k\}$
          $\mathcal{E}_c \leftarrow \{(P_i, P_k)\}$
          $\mathcal{O}_c \leftarrow \{\mathcal{P}_c, \mathcal{E}_c\}$
          $\mathcal{O} \leftarrow \mathcal{O} \cup \{\mathcal{O}_c\}$; $c \leftarrow c + 1$
        else
          let $\mathcal{O}_c = \{\mathcal{P}_c, \mathcal{E}_c\}$ be the object that contains $\mathbf{z}_i$ or $\mathbf{z}_k$
          $\mathcal{P}_c \leftarrow \mathcal{P}_c \cup \{\mathbf{z}_i, \mathbf{z}_k\}$
          $\mathcal{E}_c \leftarrow \mathcal{E}_c \cup \{(P_i, P_k)\}$
          For each $P_y \in \mathcal{P}_c$, if $(P_y, P_i) \in \mathcal{E}$, add to $\mathcal{E}_c$; same for P_k
          For any other objects $\mathcal{O}_y$ containing $\mathbf{z}_i$ or $\mathbf{z}_k$ respectively, merge all such into $\mathcal{O}_c$
  $\mathcal{U} \leftarrow \mathcal{U} - \{P_i\}$
until $|\mathcal{U}| == \emptyset$

C Datasets

1. Tetrominoes [24] (Apache-2.0 License): This dataset has 35×35 images, each consisting of 3 Tetris-like shapes sampled from 6 colors and 17 shapes. We download the data from the original github page[1]. Ground truth segmentation masks and properties are provided. We follow [34, 11, 14] and use the first 60K samples for training and the next 320 samples for testing.
2. Abstract Scenes (AbsScene) [50]: This dataset consists of 64×64 images with 10 multi-part objects. Each object consists of multiple parts with different appearances, and objects vary in position across images. We download the data from the original release[2]. Ground truth segmentation masks are provided.
3. Abstract Scenes with Occlusion (AbsScene-O) [50]: We generated this variation of the AbsScene dataset, where occlusion of objects is allowed up to 25% minimum visibility for each object. Ground truth masks of the full objects are provided.
4. Abstract Scenes with Complex Backgrounds (AbsScene-C) [50]: We generated this out-of-distribution test set of the AbsScene dataset, where we randomly replace the original background with 15 real-world complex-textured backgrounds. Ground truth masks of the full objects are provided.
5. Google Scanned Objects (GSO) [8]. We generated this dataset which consists of photo-realistic images made via compositions of common real-world objects and backgrounds. Each generated image contains realistic variations of lighting, shadows, 3D object positions and object-occlusion. All raw images and their ground truths are resized and cropped to

[1] *https://github.com/deepmind/multi_object_datasets*
[2] *https://www.microsoft.com/en-sg/download/details.aspx?id=52035*

size 128×128. We downloaded the raw data from the official website[3], and generated the dataset by using the Kubric setup following the official github repository[4] from [49].

6. Google Scanned Objects with Occlusion (GSO-O) [8]: We generated this variation of the GSO dataset, where occlusion of objects is allowed up to 25% minimum visibility for each object. Ground truth masks of the full objects are provided.
7. SKU-110K (SKU) [16]: This challenging real-world dataset provides 11,762 images with 1.7 million annotated objects captured in densely packed store-keeping scenarios. The images are collected from supermarket stores and are of various scales, viewing angles, lighting conditions, and noise levels. All the images and their ground truth bounding boxes are cropped and resized into size 128×128. We download the data from the official github page[5]. Following [16], we split SKU-110K into 8233 images for training and 588 images for testing.
8. PASCAL VOC [12]: The PASCAL VOC (Visual Object Classes) challenge 2012 dataset is a widely used benchmark for object segmentation with ground truth annotations for 20 object classes and 1 background class. The images are collected from real-world settings and contain objects which are in natural environments of various scales, viewing angles, lighting conditions, and noise levels. We train on the trainaug variant with 10582 images and we evaluate on the trainaug validation set with 1499 images. All the images and their ground truth bounding boxes are cropped and resized into size 128×128. We download the data from the official website[6]. We use the augmented training set with 10582 samples and 1499 validation samples for testing.
9. MS COCO [33]: The MS COCO (Microsoft Common Objects in Context) dataset is a large-scale dataset for object segmentation, containing over 330,000 images annotated with 80 object categories. We use the variant that focuses on multi-object segmentation as proposed by [48], which we download from here[7]. All the images and their ground truth bounding boxes are cropped and resized into size 128×128. We use the first 10K samples for training and the last 2K samples for testing.

D Baseline Models

1. SLIC [1] is a clustering algorithm that clusters pixels into superpixels by using an efficient adaptation of the k-means algorithm. We use the python implementation and for each dataset, we perform a grid search for the optimal hyperparameters which produce the best results. For example, for the Tetrominoes dataset, we set the optimal clustering threshold value as 10 and the initial number of clusters as 12.
2. Felzenszwalb's Algorithm [13] is a graph-based segmentation algorithm that groups pixels together through a hand-crafted boundary detection procedure. We use the python implementation and for each dataset, we perform a grid search for the optimal hyperparameters which produce the best results. For example, for the Tetrominoes dataset, we set the optimal clustering threshold 1000, the minimum cluster size as 10, and the image smoothening value as 0.1.
3. Normalized Cut (Ncut) algorithm [39] is an unsupervised image segmentation algorithm which treats image segmentation as a graph partitioning problem and uses a hand-crafted criterion to measure both the total dissimilarity between the different groups as well as the total similarity within the groups in order to determine the final segmentation groups.
4. Slot Attention [34] initializes a set of random object representations called slots which are iteratively refined by slot-normalized cross-attention on the outputs of a simple convolutional neural network (CNN). The slots are then decoded individually and combined to reconstruct the input. All Slot Attention baselines are trained with 300,000 iterations. We use the default training hyperparameters from the official reference implementation.

[3] *https://app.gazebosim.org/GoogleResearch/fuel/collections/Scanned%20Objects%20by%20Google%20Research*

[4] *https://github.com/jinyangyuan/compositional-scene-representation-datasets*

[5] *https://github.com/eg4000/SKU110K_CVPR19*

[6] *http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html*

[7] *https://www.dropbox.com/sh/u1p1d6hysjxqauy/AACgEh0K5ANipuIeDnmaC5mQa?dl=0*

5. GENESIS-V2 [11] obtains pixel embeddings through a U-Net which are then clustered using a stochastic stick-breaking process. The clusters are then decoded to reconstruct the input. We similarly fine-tune the model to use the output standard deviation of 0.7 and the equivalent per-pixel GECO reconstruction target as EfficientMORL and achieved much higher results than those reported in the CLEVRTEX paper [26].
6. BO-QSA [22] initializes Slot Attention's object representations as learnable embeddings instead of sampling from a learnable Gaussian distribution and supplements the training with bi-level optimization. All BO-QSA baselines are trained with 300,000 iterations. We use the default training hyperparameters from the official reference implementation. We train all datasets with both the mixture-based decoder and autoregressive transformer-based decoder and report the highest scores.
7. DINOSAUR [38] leverages on pre-trained features and Slot Attention to scale to real-world images. DINOSAUR first extracts learned features of the input image by using the DINO model, which was pre-trained on the ImageNet dataset using self-supervised learning techniques [5], then uses Slot Attention to attend to different sets of features before reconstructing the self-supervised features. All DINOSAUR baselines are trained with 300,000 iterations. We use the default training hyperparameters from the validated reference implementation.
8. COCA-Net [30] discovers objects through hierarchical compact clustering attention. It progressively groups image features into object-centric clusters by combining attention-based feature refinement with compactness-based cluster selection and feature pooling. The resulting object representations are decoded using a spatial broadcast decoder to reconstruct the input image, and the model is trained end-to-end using a pixel-reconstruction loss without segmentation supervision.
9. MaskCut [44] was recently proposed to perform unsupervised image segmentation by leveraging on a pre-trained model. MaskCut first extracts learned features of the input image by using the DINO model, which was pre-trained on the ImageNet dataset using self-supervised learning techniques [5], before applying the Ncut algorithm on the extracted features to determine the segmentation.
10. OC-Net [14] leverages on learned feature connectivity to discover objects in an unsupervised manner. OC-Net also uses two designed object-centric regularization terms to ensure good separation between object representations and suitable disentanglement within representations. We use similar training iterations, thresholds and loss weights as proposed in the original paper for all the datasets.

E Additional Implementation Details

E.1 Hyperparameters

Our ECO-Net framework has 2 main hyperparameters: latent-space size K and similarity constant ϵ. For latent-space size K which determines the size of each part representation, we follow the baseline models and set the representation latent-space size to $K = 128$. For the similarity constant ϵ, we describe our initial experiment to determine its value as follows: we first generate 2 samples of the AbsScene dataset, similar to the input images in Figure 2. After running the co-part object discovery algorithm with grid searched values of ϵ, we find that the object can be accurately discovered when $0.2 \leq \epsilon \leq 1$. Considering that a lower ϵ increases the number of similarity matches which increases the runtime, we set $\epsilon = 0.99$ for efficiency and accuracy.

We further present the results for the different dataset types when varying ϵ, from which we see that our method can take a wide range of threshold values without affecting performance:

E.2 Object Similarity Computations

Our ECO-Net framework uses a graph-based representation for objects, where each object in memory is defined by its part nodes, part edges and object node. Nodes are derived by a combination of the shape, texture and distilled feature of each part, while edges are vectors which point from the centroid of one part to an adjacent part. Object nodes are derived in a similar manner to part nodes. In order to

Table 7: Performance of ECO-Net with different values of ϵ on AbsScene, GSO, and COCO.

Method	AbsScene	GSO	COCO
ECO-Net ($\epsilon = 0.1$)	89.9 ± 1.7	74.1 ± 4.2	44.8 ± 14.3
ECO-Net ($\epsilon = 0.2$)	97.0 ± 1.6	88.0 ± 4.0	48.6 ± 14.0
ECO-Net ($\epsilon = 0.55$)	96.9 ± 1.6	87.3 ± 3.9	49.1 ± 13.8
ECO-Net ($\epsilon = 0.9$)	97.4 ± 1.5	87.8 ± 3.9	49.1 ± 13.7
ECO-Net ($\epsilon = 0.99$)	$\mathbf{99.0 \pm 1.4}$	$\mathbf{90.0 \pm 3.7}$	$\mathbf{50.7 \pm 11.9}$

determine whether a part or an object is similar to another, we compute the similarity between the part nodes, part edges and object node. We then take the mean. We have attached a simple demo for this computation using the shape feature for parts and objects in the supplementary material.

E.3 Settings for Felzenszwalb's Algorithm

We use Felzenszwalb's algorithm to obtain the parts in ECO-Net's Graph Representation Module. Since we are concerned with fine-grained parts formed by clustering together pixels with similar features, we fix the threshold to 10 and the minimum cluster size as 1 for all experiments.

E.4 Training Details

We train ECO-Net for 30K iterations with a batch size of 8, and set $\epsilon = 0.99$, $\tau = 0.1$ based on initial experiments

For all baselines, we set the number of foreground objects to 4 for Tetrominoes, 5 for AbsScene, 8 for GSO, 50 for SKU-110K, 6 for PASCAL VOC and MS COCO. ECO-Net does not require the number of foreground objects for either training or evaluation. We train information-bottleneck methods for 300K iterations with a batch size of 64, using Adam optimizer with a learning rate of 4×10^{-4}. We set the size of the latent space to be $D = 64$ for all baseline models.

E.5 Third-Party Assets

ECO-Net is implemented using PyTorch [36]. In addition to various open-sourced Python packages, we make use of the following third-party assets:

- Locatello et al. [34] (Apache-2.0 license): Implementation of Slot Attention[8],
- Engelcke et al. [11] (GPL-3.0 license): Implementation of GENESIS-V2[9],
- Jia et al. [22]: Implementation of BO-QSA[10],
- Seitzer et al. [38] (MIT License): Implementation of DINOSAUR[11].
- Wang et al. [44] (CC-BY-NC 4.0 License): Implementation of MaskCut[12].

E.6 Computational Resources

An 8X GPU server is used to train ECO-Net and all comparison baselines for our experiments.

E.7 Evaluation Metrics.

We use the Adjusted Rand Index (ARI) to measure the quality of objects discovered [21]. ARI is a measure of similarity between two data clusterings that takes into account the permutation-invariant nature of the predicted segmentation masks and their corresponding object ground-truth masks. We

[8] *https://github.com/google-research/google-research/tree/master/slot_attention*
[9] *https://github.com/applied-ai-lab/genesis*
[10] *https://github.com/YuLiu-LY/BO-QSA*
[11] *https://github.com/gorkaydemir/DINOSAUR*
[12] *https://github.com/facebookresearch/CutLER*

also use the Dice similarity coefficient, along with the Intersection-over-Union (IoU) between the best matching object masks X and Y:

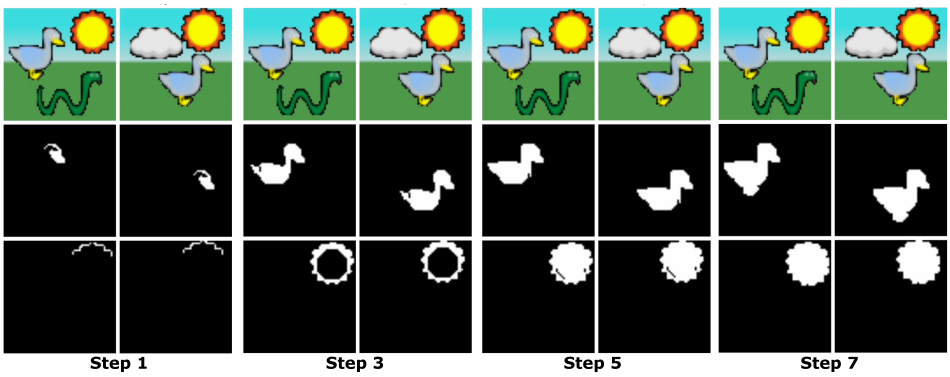
$$\text{Dice}(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \; ; \; \text{IoU}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{1}$$

where X the set of object pixels extracted and Y is the set of annotated object pixels in the ground truth. We compute the mean Dice and the mean IoU scores, denoted as mDice and mIoU respectively, by averaging the individual Dice and IoU scores across all matches.

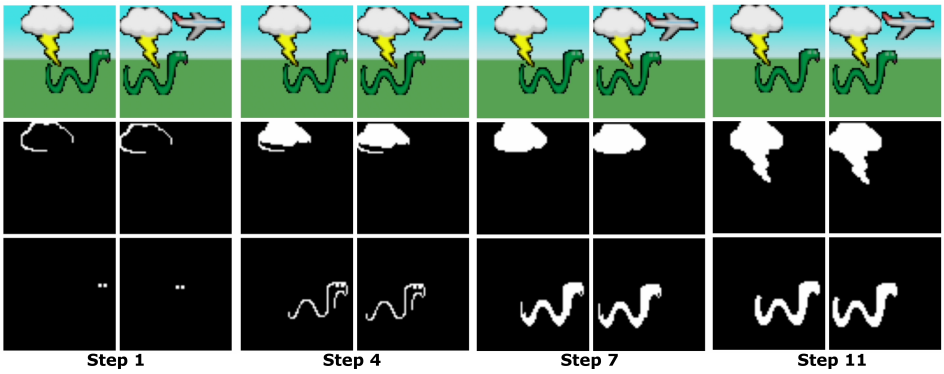
F Additional Visualizations

F.1 Co-Part Object Discovery Algorithm

In Figure 7, we visualize the runtime merging of parts into objects using our co-part object discovery algorithm with samples from AbsScene.



(a) Sample 1



(b) Sample 2

Figure 7: Supplementary visualization of Co-Part Object Discovery Algorithm.

F.2 Object Discovery

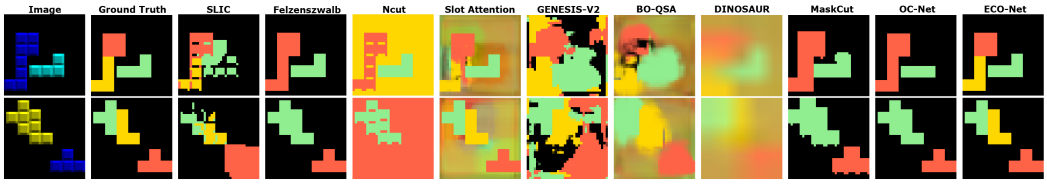
We visualize the results of additional samples from simulated datasets in Figure 8, realistic datasets in Figure 9 and real-world datasets in Figure 10.

F.3 Occlusion-Aware Perception

We visualize the results of additional samples from occluded-objects datasets in Figure 11.

F.4 Out-of-Distribution Object Discovery

We visualize the results of additional samples from AbsScene-C in Figure 12.

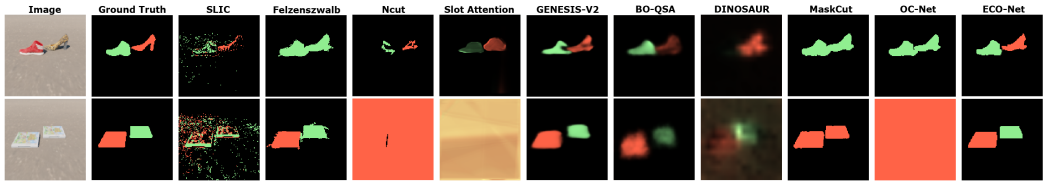


(a) Tetrominoes

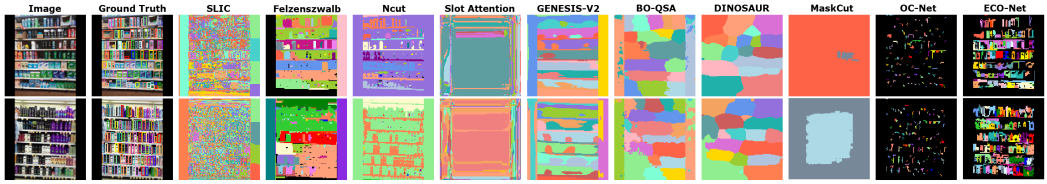


(b) AbsScene

Figure 8: Supplementary visualization of discovered objects on simulated datasets.

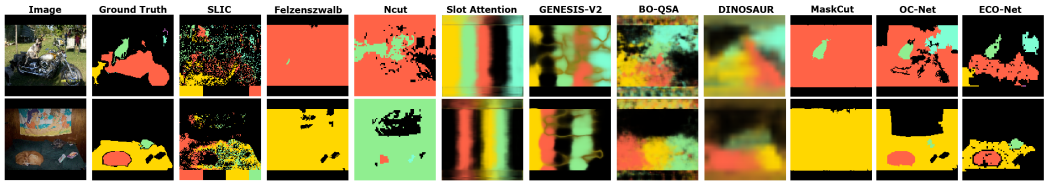


(a) GSO

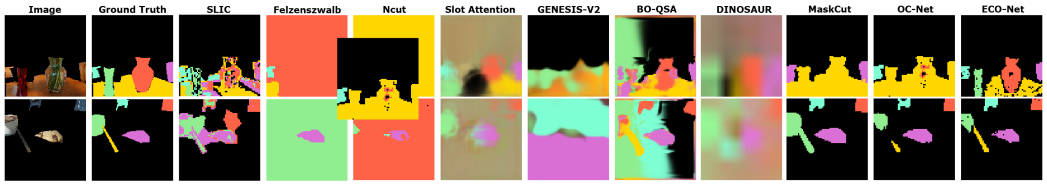


(b) SKU-110K

Figure 9: Supplementary visualization of discovered objects on realistic datasets.

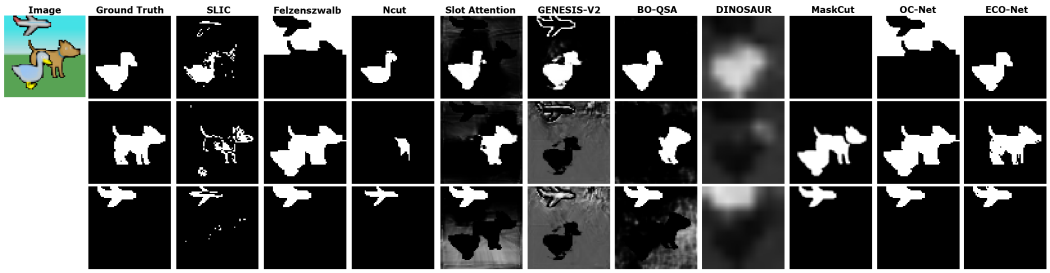


(a) PASCAL VOC

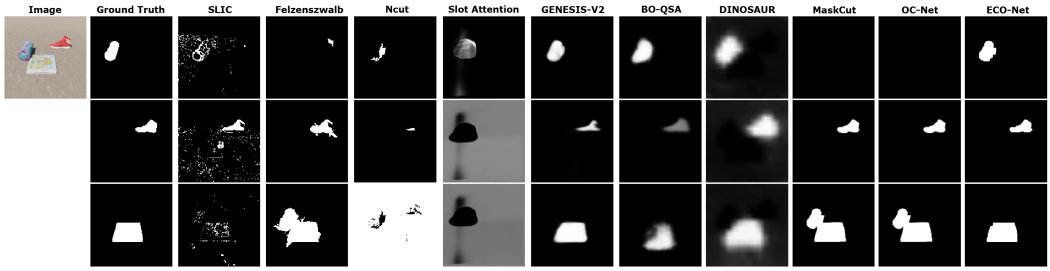


(b) MS COCO

Figure 10: Supplementary visualization of discovered objects on real-world datasets.



(a) AbsScene-O



(b) GSO-O

Figure 11: Supplementary visualization of discovered objects on occluded objects datasets.

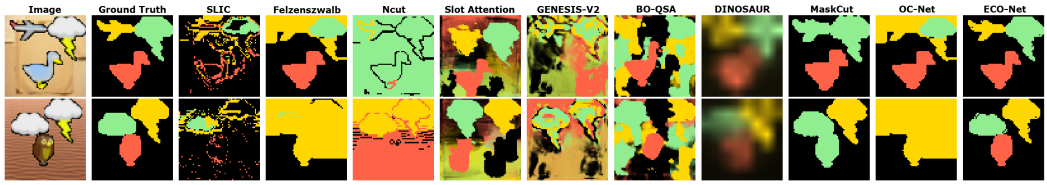


Figure 12: Supplementary visualization of discovered objects on out-of-distribution test set AbsScene-C.

G Downstream Property Prediction

A key trait of effective object-centric representation is its ability to encode object properties such as color, position and shape [37]. Here, we show that the learned object representations are disentangled and can be used to predict the values of these properties.

Given the representations and their corresponding ground truth values of a target object property, we employ them as features to train a gradient boosted tree (GBT) [15]. While linear projection is a possible alternative, we use GBT for ease of comparison with past works [14, 10, 9]. Also, GBT has the added strength of interpretability which provides a means of quantifying disentanglement of learned features. To evaluate how well the properties of unseen objects are predicted by the GBT, we use the coefficient of determination R^2 [46].

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}, \tag{2}$$

where $\hat{y}_i$ is the predicted value of the i-th object property and y_i is the corresponding true value for total n unseen samples, and $\bar{y}_i = \frac{1}{n}\sum_{i=1}^{n} y_i$.

We use the learned object representations from the simulated datasets to predict the properties of objects. We use the mIoU score to match the discovered object to the ground truth object. We split the test images equally into two sets, one for training the GBT model for each object property, and the other for evaluation. We use the learned object representations from the simulated datasets to predict the properties of objects.

H Additional Experiments

H.1 Experiments on Part Information

In this section, we run experiments to quantify the presence of part-level information in widely used pretrained ViT models in comparison to ECO-Net. We perform these experiments by comparing with the recent MPAE method [47], which adds a learnable part-discovery head on top of a frozen DINOv2 model [35]. We present the mDice and mIoU results for part-discovery on the PartImageNet out-of-distribution test set [18] in the table below. Although MPAE directly optimizes for part-discovery, we see that ECO-Net significantly outperforms ($p < 0.001$ under t-test).

Table 8: Evaluation scores of discovered parts on PartImageNet.

Method	ARI	NMI	mDice	mIoU
MPAE	73.5±21.5	53.5±12.2	43.8±19.0	32.5±17.7
ECO-Net	**74.6±18.9**	**55.6±11.3**	**48.9±16.3**	**34.6±14.3**

H.2 Experiments on Model Speed

We perform experiments to evaluate the speed of ECO-Net. Table 9 shows comparison wall-clock times for inference time per iteration (Time / IT) in seconds and training time (Train) in hours. We see that ECO-Net performs competitively against the two main categories of approaches in terms of speed. The image resolution of GSO is twice that of AbsScene.

Table 9: Processing speed and training time across various methods.

	AbsScene		GSO		COCO	
Method	Time / IT (s)	Train (hrs)	Time / IT (s)	Train (hrs)	Time / IT (s)	Train (hrs)
Slot Attention	0.20	29.3	0.12	77.23	0.10	64.27
MaskCut	11.95	–	48.45	–	57.75	–
ECO-Net	3.88	22.67	20.42	122.7	19.86	119.47

H.3 Experiments on Image Resolution Scaling

Table 10 presents inference time per iteration and training time at the original AbsScene resolution of 64×64, alongside for 128×128.

Table 10: Processing speed and training time at different image resolutions.

	64×64		128×128	
Method	Time / IT (s)	Train (hrs)	Time / IT (s)	Train (hrs)
Slot Attention	0.20	29.38	0.23	74.19
MaskCut	11.95	–	41.12	–
ECO-Net	3.88	22.67	16.32	108.56

H.4 Additional Experiments on Reconstruction-Based Approaches

We further evaluate COCA-Net [30] against representative object-centric learning methods that employ an information bottleneck to obtain compact, object-centric latent representations.

COCA-Net is a recent unsupervised object-centric learning method that discovers objects through *hierarchical compact clustering attention*. It progressively groups image features into object-centric clusters at multiple levels of granularity, allowing it to capture both fine-grained parts and whole objects without supervision.

Its key idea is to combine attention-based feature refinement and aggregation with a compactness-based clustering prior: spatially coherent features are encouraged to form compact clusters, while distinct objects are separated. This produces object masks and representations that can be evaluated with segmentation metrics such as ARI, mDice, and mIoU.

In Tables 11, 12 and 13, we compare COCA-Net with information-bottleneck-style object-centric approaches trained with similar reconstruction objectives, including Slot Attention [34], GENESIS-V2 [11], BO-QSA [22], and DINOSAUR [38]. The results show that ECO-Net significantly outperforms COCA-Net on all datasets, while not requiring to fix the number of objects during training.

Table 11: Evaluation scores of discovered foreground objects compared with COCA-Net.

(a) Simulated datasets

Method	Tetrominoes			AbsScene		
	ARI	mDice	mIoU	ARI	mDice	mIoU
Slot Attention	99.7±1.5	41.4±1.5	26.6±1.5	94.8±11.8	74.6±10.8	63.9±13.4
GENESIS-V2	97.2±5.4	47.0±3.5	30.9±3.0	46.7±24.1	65.4±15.8	54.3±18.0
BO-QSA	99.1±3.7	40.8±1.4	25.8±1.2	93.4±13.8	83.1±9.2	74.0±12.4
DINOSAUR	92.5±9.9	40.0±2.0	25.3±1.8	95.0±8.2	51.9±6.2	35.5±5.7
COCA-Net	91.8±2.6	97.2±1.0	94.6±1.9	92.6±8.2	85.7±7.4	76.8±10.0
ECO-Net	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**98.0±2.3**	**99.5±0.9**	**99.0±1.4**

(b) Realistic datasets

Method	GSO			SKU-110K		
	ARI	mDice	mIoU	ARI	mDice	mIoU
Slot Attention	64.3±28.8	39.5±13.2	27.0±10.8	-9.7±7.2	0.9±0.4	0.4±0.2
GENESIS-V2	90.1±8.3	84.3±4.3	73.6±5.9	6.1±2.7	6.4±2.2	3.5±1.3
BO-QSA	64.4±17.0	74.2±4.3	59.7±5.4	5.4±2.3	7.2±3.1	3.9±1.8
DINOSAUR	96.2±10.1	42.0±6.2	26.9±5.2	4.4±2.3	12.1±3.7	6.8±2.3
COCA-Net	86.0±14.2	89.5±8.1	80.4±9.9	5.5±1.3	7.5±2.3	4.1±1.4
ECO-Net	**99.3±1.7**	**94.6±2.1**	**90.0±3.7**	**7.8±3.1**	**32.0±8.9**	**21.7±6.6**

(c) Real-world datasets

Method	PASCAL VOC			MS COCO		
	ARI	mDice	mIoU	ARI	mDice	mIoU
Slot Attention	14.8±22.4	12.6±8.3	7.7±5.6	28.5±14.2	23.9±10.2	14.5±7.1
GENESIS-V2	13.2±20.1	14.6±9.5	9.4±6.8	13.2±12.5	21.8±9.7	13.9±7.2
BO-QSA	13.0±19.5	13.7±8.9	8.6±6.2	24.0±15.3	33.3±11.6	22.5±9.4
DINOSAUR	16.9±24.0	17.0±10.5	11.4±7.7	30.6±16.9	23.0±9.3	14.2±6.6
COCA-Net	14.1±14.8	18.6±10.3	13.3±9.0	26.9±12.1	50.0±12.5	36.7±12.1
ECO-Net	**19.4±19.5**	**28.4±13.0**	**20.2±10.6**	**39.3±17.2**	**64.4±11.6**	**50.7±11.9**

Table 12: Results for discovered objects with occlusion.

Method	AbsScene-O		GSO-O	
	mDice	mIoU	mDice	mIoU
Slot Attention	54.7±14.4	41.6±15.0	40.5±15.7	28.1±12.4
GENESIS-V2	48.9±19.4	40.3±20.6	81.4±5.7	69.4±7.0
BO-QSA	61.0±13.7	47.1±14.9	72.2±6.3	57.3±7.6
DINOSAUR	44.5±6.4	29.1±5.3	30.0±5.6	17.8±4.1
COCA-Net	72.7±17.0	61.6±16.7	73.9±16.3	63.2±16.4
ECO-Net	**93.2±9.3**	**90.7±12.3**	**90.4±6.5**	**83.5±8.8**

Table 13: Results for discovered objects on out-of-distribution test set AbsScene-C.

Method	ARI	mDice	mIoU
Slot Attention	37.9±21.0	41.4±12.4	28.0±11.5
GENESIS-V2	23.2±14.3	32.9±8.3	20.5±6.3
BO-QSA	32.5±16.9	38.3±10.0	25.0±8.6
DINOSAUR	82.0±17.5	54.2±6.3	37.8±6.0
COCA-Net	57.5±21.1	51.1±13.7	37.2±13.3
ECO-Net	**93.7±7.3**	**97.5±4.0**	**95.6±6.3**

H.5 Additional Evaluation with Mean Segmentation Covering

We additionally report mean Segmentation Covering (mSC) alongside ARI, mDice, and mIoU for COCA-Net and ECO-Net on Tetrominoes and AbsScene in Table 14. While mSC provides a complementary measure of segmentation quality, it weights segment matches by their area, giving larger objects greater influence on the overall score. We therefore retain mIoU as our primary overlap metric and include mSC here to provide a more comprehensive comparison.

Table 14: mIoU and mean Segmentation Covering (mSC) scores of discovered foreground objects.

	Tetrominoes		AbsScene	
Method	mIoU	mSC	mIoU	mSC
COCA-Net	94.6±1.9	94.7±3.9	76.8±10.0	74.6±8.9
ECO-Net	**100.0±0.0**	**100.0±0.0**	**99.0±1.4**	**99.1±1.7**

I Limitations

There are still obstacles that have to be overcome for successful application of our framework to the visual complexity of the real world. A natural next step would be to extend ECO-Net to more efficient implementations of the co-part object discovery algorithm especially for applications which require real-time object discovery. It is also promising to dynamically learn an appropriate clustering similarity constant given the training data. Lastly, real-world scenes containing multi-part objects with variations of illumination, viewpoints and pose is still of higher visual complexity than the datasets considered here and reliably bridging this gap is an open problem.

J Ethics Statement

Our analysis has no immediate impact on general society. To the best of our knowledge, none of the datasets used in this study contain personally identifiable information or offensive content. ECO-Net performs scene understanding, where potential negative societal impact such as in the area of surveillance cannot be fully excluded upon future research. However, there remains challenging open questions that need to be reliably addressed before enabling the deployment of ECO-Net to real-world settings, and hence a direct application for malicious purposes is currently unlikely.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: All claims are validated by theoretical and experimental results in the paper.

   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: Limitations are discussed in Section I.

   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: Assumptions and proofs are provided in Section A.

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: All details required for reproducibility are included in the paper and appendix. Code for object similarity is also included in the supplementary material.

   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: [Yes]

Justification: Code and data are publicly available at `https://github.com/alexfoo-dw/ECO-Net`. Code for object similarity is included in the supplementary material for review.

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: Details are in the main paper and the appendix.

   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: All error bars are included in the experimental results.

   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: Computer resources are provided in Section E.6.

   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: None of the datasets used in this study contain personally identifiable information or offensive content.

   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: Broader impacts are discussed in Section J.

    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: [N/A]

    Justification: There remains challenges to be overcome before the deployment ECO-Net and its counterparts to real-world settings.

    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: All creators and owners are properly cited, along with their licenses mentioned, where the assets are stated to be used.

    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: Code and data to be released 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: [N/A]

    Justification: The paper does not involve crowdsourcing 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: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    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: [N/A]

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.