This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Hierarchical Compact Clustering Attention (COCA) for Unsupervised Object-Centric Learning

Can Küçüksözen[1,2] Yücel Yemez[1,2]
[1]Department of Computer Engineering, Koç University
[2]KUIS AI Center
{ckucuksozen19, yyemez}@ku.edu.tr

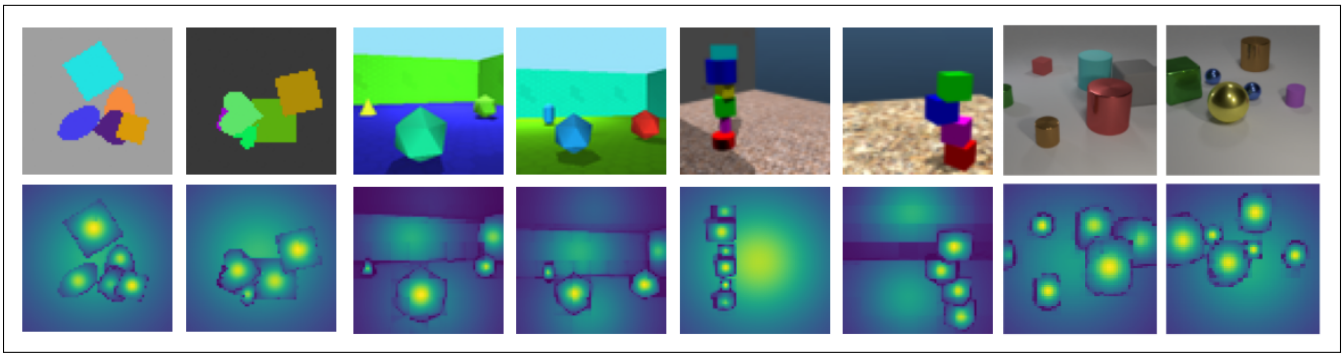


Figure 1. Compactness scores obtained for each pixel in the scene, across four different datasets. The transition from bright yellow to deep purple signifies decreasing compactness. To obtain these scores, a trained COCA-Net encoder is used to generate object masks. Each object mask is then broadcasted to pixels based on the pixel-object assignments. This operation associates every pixel with a copy of its object's mask. Finally, compactness scores for each pixel's mask are calculated via Eq. 3.

Abstract

We propose the Compact Clustering Attention (COCA) layer, an effective building block that introduces a hierarchical strategy for object-centric representation learning, while solving the unsupervised object discovery task on single images. COCA is an attention-based clustering module capable of extracting object-centric representations from multi-object scenes, when cascaded into a bottom-up hierarchical network architecture, referred to as COCA-Net. At its core, COCA utilizes a novel clustering algorithm that leverages the physical concept of compactness, to highlight distinct object centroids in a scene, providing a spatial inductive bias. Thanks to this strategy, COCA-Net generates high-quality segmentation masks on both the decoder side and, notably, the encoder side of its pipeline. Additionally, COCA-Net is not bound by a predetermined number of object masks that it generates and handles the segmentation of background elements better than its competitors. We demonstrate COCA-Net's segmentation performance on six widely adopted datasets, achieving superior or competitive results against the state-of-the-art models across nine different evaluation metrics.

1. Introduction

Object-centric learning (OCL) has emerged as a powerful paradigm in contemporary computer vision, offering advantages such as improved generalization over supervised methods, robustness to data distribution shifts, effectiveness in downstream tasks, and the ability to train without labeled data [15, 22]. These models allocate a *separate representational slot* for each object in a scene, enabling the disentanglement of individual object properties in multi-object scenarios. The benefits of OCL have driven its increasing adoption across diverse applications. While these models achieve impressive object discovery accuracy on synthetic datasets, their performance—and consequently their generalization capabilities—deteriorates significantly when applied to real-world images, which are far more complex and diverse. Yang et al., [55] argue that these models still lack the necessary inductive biases to understand 'objectness' in real-world images.

In fact, the challenges of contemporary OCL models extend beyond real-world images, with critical limitations also emerging in synthetic datasets. State-of-the-art models [11, 28, 36, 38] derived from Slot-Attention (SA) [38]

have gained popularity for their efficiency and simplicity. However, recent studies highlight key drawbacks, such as learning inconsistencies due to similar initialization of slot representations from a common latent distribution [36, 56], sensitivity to a predefined number of slots [20, 57] and poor handling of background segments. We argue that these issues stem from how *segregation*—the process of organizing raw sensory inputs into distinct, meaningful entities—is handled. When applied to our case, segregation corresponds to unsupervised instance segmentation, where the model partitions pixels into coherent object segments. Most often, unsupervised segmentation is addressed through the rich literature on cluster analysis, where SA can be seen as a *soft* version of the traditional K-Means clustering algorithm [38]. Thus, inheriting its strengths and limitations, including sensitivity to initialization, dependency on a predefined number of clusters, assumptions about cluster shape and size, and sensitivity to outliers [19, 25, 27, 33, 54], which hinder SA's suitability for complex or heterogeneous scenes.

In this work, we address the limitations of the current unsupervised OCL paradigm by developing a novel approach that avoids the aforementioned drawbacks of SA based methods. We define a notion of 'objectness', as a spatial inductive bias based on *compactness of candidate object masks* and incorporate it into our architecture. We lay the foundations of our proposed architecture on the hierarchical agglomerative clustering (HAC) based methods—which have not received much attention from the research community in recent years— despite its inherent advantages that include flexibility in the number of output clusters, robustness to noise and outliers, the ability to capture hierarchical data relationships, smooth handling of irregularly distributed clusters, and high interpretability. [19, 25, 41, 54].

To this end, we introduce the Compact Clustering Attention Layer (COCA layer), a simple and effective attention-based clustering module. When stacked hierarchically, the COCA layers form the COCA-Net image encoder. Combined with the Spatial Broadcast Decoder (SBD) [50], COCA-Net adopts a bottom-up approach for end-to-end unsupervised object discovery from scratch.

Contributions To the best of our knowledge, in the context of unsupervised OCL literature, COCA-Net is the first work to utilize a hierarchical clustering and pooling strategy within a neural network architecture to generate slot representations for a single image, in a fully unsupervised way. Due to its design, COCA-Net offers several key advantages, addressing most of the drawbacks that existing slot-based methods suffer from:

- Robustness to initial seeding: Different training runs of the COCA-Net hierarchy yield almost the same performance with minimal variance, demonstrating its reliability and robustness.
- Effective background segmentation: COCA-Net effectively handles background segments, ensuring accurate segmentation of the entire scene.
- Dynamic slot allocation: Each COCA layer can dynamically adjust the number of its output clusters, eliminating the need to predetermine the number of output object slots.
- High-quality encoder features: COCA-Net generates high-quality segmentation masks on the encoder side. This holds promise for the trained COCA-Net encoder to be repurposed as an object-centric feature extractor for downstream tasks.

We conduct extensive experiments on the OCL benchmark provided by [15] and demonstrate that COCA-Net outperforms or performs on par with the state-of-the-art on the unsupervised object discovery task. This evaluation is based on the object slot masks obtained from both the decoder and encoder sub-networks of the pipeline, in contrast to literature on OCL where the evaluation considers only the masks generated by the decoder sub-network.

2. Related Work

2.1. Unsupervised Object Discovery

Recent years have seen a surge in *Instance Slot* models, notably Slot Attention (SA) [38] and its successors [5, 10, 11, 20, 28, 32, 36, 42, 56]. These models typically initialize a fixed number of slots from a shared latent distribution and use query-normalized cross-attention to assign input features to slots iteratively, refining slot features in the process.

However, initializing slots from a common distribution causes what is known as the *routing problem*, where symmetry among similarly initialized slots must be broken to separate distinct objects [22]. Zhang et al. [56] highlight that the routing problem – or the lack of tie-breaking in Slot Attention (SA) methods – can cause multiple slots to bind to the same input subset, hindering object separation. Similarly, Kim et al. [36] describe the *bleeding issue*, where different training runs yield inconsistent results due to SA's occasional failure to distinguish foreground objects from the background. Beyond challenges related to the routing problem, a key limitation of SA is its requirement to predetermine the number of output slots. As noted in recent studies [20, 57], predefining the number of objects in a dataset is often impractical since scenes can contain a variable number of objects. Consequently, performance of SA-based models is highly sensitive to the number of slots that are predefined; poor choices can lead to under- or over-segmentation. These limitations in SA methods arise from their reliance on a soft K-Means based clustering approach, which shares the well-documented drawbacks of traditional K-Means [19, 21, 25, 27, 33, 54].

Sequential Slot models like [6, 17, 18], on the other hand, are specifically tailored to overcome the routing problem by imposing a sequential masking – or concealing – strategy to discover distinct object masks. In each iteration, a new object mask is predicted, and the explained parts of the scene are excluded from subsequent predictions via Stick-Breaking Clustering (SBC) [18]. To generate a single object mask, for instance, [18] samples a random pixel in the scene and computes its affinity mask whereas [6] relies on a randomly initialized sub-network. Although effective to mitigate the routing problem, these models suffer from sub-optimal mask generation and high computational costs per object, limiting their performance in complex object-rich scenes. The third category of slot based methods, *Spatial Slots*, partition the scene into spatial windows and assign a distinct slot to each window. By focusing each slot on a distinct local area, these models effectively break slot symmetries and mitigate the routing problem. Notable examples include [12, 29, 37]. However, their performance deteriorates when dealing with overlapping objects or objects larger than the predefined slot windows [22].

2.2. Clustering Methods

Our work is based upon the traditional literature on *hierarchical clustering* methods [2, 45], which iteratively cluster pixels across multiple scales, often using a bottom-up agglomerative approach. Current hierarchical agglomerative clustering methods based on deep learning primarily focus on segmentation in real-world image datasets, typically optimizing weakly supervised or self-supervised objectives [7, 8, 35, 48]. Most of them make use of pre-trained backbones or sub-networks [7, 8, 48] or teacher networks to guide learning [7, 8]. Differing from these models is a recent method, [34], which leverages multi-views of a scene to provide additional supervision.

A relevant part of the literature is the methods that operate on graph structured data and address *graph-based clustering* and pooling. Some recent approaches can be listed as [4, 24, 44, 47, 49]. Notably, TokenCut [49], uses a pre-trained backbone [9] to initialize node features and applies a Normalized Cut [43] based spectral graph clustering algorithm to achieve unsupervised object discovery in scenes with a single prominent object. [47] and [3] adopt TokenCut to multi-object scenes by leveraging an SBC inspired concealing strategy. A hybrid hierarchical and graph based clustering model [48], uses TokenCut [47] to generate a top-to-bottom partition of the scene and utilizes a form of [8] as a hierarchical agglomerative clustering to produce bottom-to-top refining of the initial partitions.

3. Method

One of the key insights behind the COCA layer aligns with the principles of *perceptual grouping* in vision systems [1, 14, 43, 45], where an ideal unsupervised segmentation model (e.g., COCA-Net) is expected to produce similar feature representations for pixels within the same object, while distinct features emerge for pixels from different objects or the background. To be able to learn such a feature space, we employ a simple pixel feature encoder as our backbone. Recent findings support this point-wise handling of pixels over locality-based inductive biases, [40] reports that the local patchification process used in the state-of-the-art Vision Transformer architectures [16] is, in fact, unnecessary. Thus, our backbone initializes pixel features that encode both appearance and position information. The details of this pixel feature encoder can be found in Supplementary Material.

3.1. COCA Layer

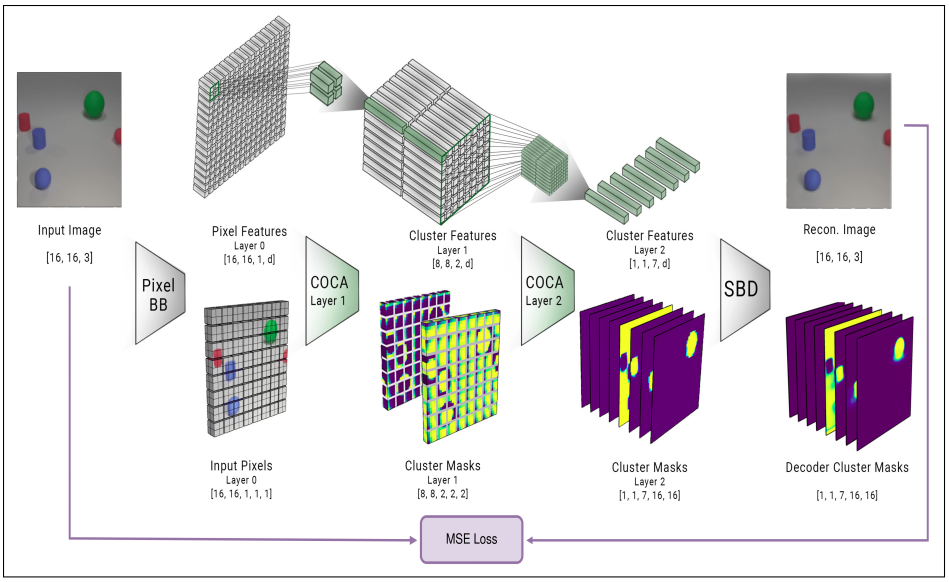
If a model achieves the ideal scenario described by the perceptual grouping literature, the task of unsupervised object discovery can be simplified to identifying a single *anchor – or representative – node* per object, whose *affinity mask* would then represent the object's segmentation. This identification of an anchor node per object could potentially benefit from insights into what constitutes a visual object and their compositional nature in the physical world. Building on such intuitions, a single COCA layer embodies a five-step approach to hierarchical agglomerative clustering, also illustrated in Figure 2b;

1. Refining the features of each input node within its spatial neighborhood to relate and contrast appropriate elements,
2. Partitioning these refined nodes into non-overlapping windows,
3. Generating a set of *affinity masks* within each window in parallel,
4. Evaluating these affinity—candidate object—masks based on their *compactness scores*,
5. Running an SBC-based sequential concealing strategy to iteratively identify the most compact mask, assign its constituent nodes as the next output cluster, and conceal these already clustered nodes to prevent potential duplicate masks in subsequent iterations.

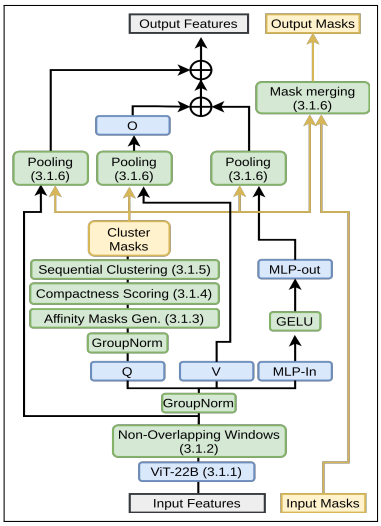
The layer is finalized by pooling the features of nodes that correspond to distinct output clusters, merging masks that are produced in the last two consecutive layers and aggregation of these elements to the next level of the hierarchy.

3.1.1. Feature Refinement

For effective and adaptive feature refinement, COCA leverages the dynamic message-passing capabilities of self-attention [46] layers in Vision Transformers (ViTs) [16]. The widespread success of ViTs across various tasks and modalities makes them an ideal choice for refining features among neighboring nodes within the same object. Specifically, we leverage the self-attention operation that is used in



(a) Overview of the COCA-Net Hierarchy: COCA-Net processes a 16×16 input image and corresponding features from the pixel encoder backbone. The first COCA layer partitions these features into 2×2 non-overlapping windows, performing clustering and pooling in parallel to produce $8 \times 8 \times 2 \times d$ cluster features. Layer 1 outputs two clusters, with 2×2 cluster masks aggregated to the next layer along with pooled features. Similarly, the second COCA layer partitions its input into an 8×8 window and outputs 7 clusters. Finally, the image is reconstructed by the SBD, where the loss is Mean Squared Error.



(b) Best viewed in color. Overview of a single COCA layer, which can be seen as a clustering variant of the ViT-22B layer [13]. Blue boxes indicate layers with learnable parameters, green boxes denote non-parametric operations, and yellow boxes represent operations involving masks.

Figure 2. Overall summary of the COCA-Net hierarchy and a single COCA layer.

a recent version, called ViT-22B [13], which modifies and scales the original model to 22 billion parameters. This refinement process enhances feature similarity between nodes that are likely to be clustered together, first locally at earlier hierarchy levels and then globally at higher levels with an expanded receptive field, allowing spatial patterns in the data that correspond to distinct objects to be gradually distinguished. Implementation details of the feature refinement stage can be found in Supplementary Material.

3.1.2. Operating on Non-Overlapping Windows

At each layer l, where $l = 1, 2, ..., L$, COCA handles a total of $h'_{l-1} \times w'_{l-1} \times k_{l-1}$ nodes refined by the vision transformer, where h'_{l-1}, w'_{l-1} are input spatial dimensions and k_{l-1} is input cluster dimension. COCA partitions these nodes into t_l^2 non-overlapping windows, where each window is composed of $h_l \times w_l \times k_{l-1}$ nodes. Note that $h_l = h'_{l-1}/t_l$ and $w_l = w'_{l-1}/t_l$. This partitioning results in an unfolded feature tensor for each window t at layer l, which we represent by $\mathbf{X}^{l,t} \in \mathbb{R}^{n_l \times d_l}$ where $n_l = h_l \times w_l \times k_{l-1}$ and d_l represents the feature dimension of a node. In effect, COCA maps $n_l = h_l \times w_l \times k_{l-1}$ input nodes to $1 \times 1 \times k_l$ distinct output clusters, for each window, in parallel. The clusters generated at the highest layer L yield the final object slots. Note that we henceforth omit the window index t for ease of notation since each window is treated simultaneously in the same manner.

3.1.3. Generating Candidate Affinity Masks

The *affinity masks* in a COCA layer, denoted by $\mathbf{\Lambda}^l \in \mathbb{R}^{n_l \times n_l}$, represent the pairwise affinities between nodes i and j within a window, where $i = 1, 2, ..., n_l$ and $j = 1, 2, ..., n_l$. Each entry Λ^l_{ij} in this affinity mask tensor is a soft binary value between 0 and 1, indicating how similar nodes i and j are. To compute pairwise affinities between each node pair, we begin by projecting the refined input node features onto a common high-dimensional space and apply GroupNorm normalization [53]. Next, we calculate the Euclidean distance E^l_{ij} between nodes i and j based on their feature representations after projection and normalization:

$$\mathrm{E}^l_{ij} = \frac{\tau_l}{\sqrt{n_l \cdot d_l}} \cdot \left\| \mathbf{Y}^l_i - \mathbf{Y}^l_j \right\|^2 \quad (1)$$

where τ_l is the temperature hyper-parameter for layer l and $\mathbf{Y}^l \in \mathbb{R}^{n_l \times d_l}$ denotes the projected and normalized feature vectors of nodes. Each column vector $\mathbf{E}^l_i \in \mathbb{R}^{n_l}$ in $\mathbf{E}^l$ can be viewed as a distance mask of a single node i measured against all other nodes within the window. So we apply soft-argmin normalization followed by min-max scaling to each distance mask, obtaining the affinity mask $\mathbf{\Lambda}^l_i \in \mathbb{R}^{n_l}$ for each node i (see Supplementary Material for details). This normalization procedure dynamically builds associations between nodes within a window and maps each affinity value into the required range $[0, 1]$ for compactness scoring and sequential clustering, which we describe next.

3.1.4. Compactness Scoring

Compactness can be viewed as a scoring function that operates on affinity masks. A single affinity mask has dimensions equal to the number of input nodes within a window, inherently forming a two-dimensional shape. Compactness scoring function is intended to guide the model to prioritize nodes that produce affinity masks corresponding to compact shapes, assigning them higher scores. Intuitively, a prominent foreground object is expected to have a compact and convex spatial form, while background elements are often scattered, with holes and concave edges. Various approaches exist across diverse disciplines for measuring compactness, as discussed in Supplementary Material. A notable approach [39, 51, 52] for calculating the *mass normalized compactness* of a shape – or affinity mask – $\mathbf{\Lambda}$, is based on the concept of moment of inertia (MI), which physically quantifies shape dispersion relative to a point, in a scale-invariant and additive manner. The compactness functional $\mathcal{C}^{\mu}(\mathbf{\Lambda})$ of $\mathbf{\Lambda}$ around point μ can be computed by comparing its MI to that of the most compact two-dimensional shape in this scenario: a circle $\mathbf{\Theta}$ with the same effective area as $\mathbf{\Lambda}$ and centered at μ:

$$\mathcal{C}^{\mu}(\mathbf{\Lambda}) = \frac{\mathcal{I}^{\mu}(\mathbf{\Theta}_{\mu})}{\mathcal{I}^{\mu}(\mathbf{\Lambda})} \tag{2}$$

where $\mathcal{I}^{\mu}(\mathbf{\Theta}_{\mu})$ represents the MI computed around μ, for the circle $\mathbf{\Theta}_{\mu}$ that is centered at μ. This compactness measure is bounded within $(0, 1]$, with larger values indicating more compact shapes. A perfect circle achieves the maximum compactness value of 1.

Formally, we treat the affinity masks tensor $\mathbf{\Lambda}^l \in \mathbb{R}^{n_l \times n_l}$ as a collection of singular affinity masks $\mathbf{\Lambda}^l_i \in \mathbb{R}^{n_l}$, each associated with a node i. A single affinity mask can be considered as a flattened two dimensional grid of affinities that constitute a non-uniform density shape. To simulate the physical properties of nodes that construct the affinity masks, during the execution of the first COCA layer, we declare five attributes for each pixel; area $\mathbf{A}^1 \in \mathbb{R}^{n_1 \times 1}$, mass $\mathbf{M}^1 \in \mathbb{R}^{n_1 \times 1}$, density $\mathbf{D}^1 \in \mathbb{R}^{n_1 \times 1}$, moment of inertia $\mathbf{I}^1 \in \mathbb{R}^{n_1 \times 1}$ and mean position in the original image resolution $\mathbf{P}^1 \in \mathbb{R}^{n_1 \times 2}$. Note that $D^l_i = M^l_i / A^l_i$. After the clustering is completed, these attributes are pooled according to the cluster masks and aggregated to the next levels of the hierarchy, just as the cluster feature vectors.

In order to distribute a copy of these physical attributes of nodes to each affinity mask and scale these attributes based on the corresponding affinities, we compute the intermediate attributes, $\tilde{\mathbf{A}}^l, \tilde{\mathbf{D}}^l, \tilde{\mathbf{I}}^l, \tilde{\mathbf{M}}^l \in \mathbb{R}^{n_l \times n_l}$, by first broadcasting then computing an element-wise product with the affinity masks $\mathbf{\Lambda}^l \in \mathbb{R}^{n_l \times n_l}$. The details for this broadcasting and scaling operations are shared in Supplementary Material. With the intermediate attributes of each affinity mask are computed, inspired by [39, 51, 52], we compute the *mass normalized compactness*, $\mathcal{C}^i(\mathbf{\Lambda}^l_i)$, of the shape described by the affinity mask $\mathbf{\Lambda}^l_i \in \mathbb{R}^{n_l}$ around the axis passing through node i as:

$$\mathcal{C}^i(\mathbf{\Lambda}^l_i) = \frac{\sum_j \tilde{\mathrm{M}}^l_{ij} \tilde{\mathrm{A}}^l_{ij} + \sum_{j<v} 2 \min(\tilde{\mathrm{D}}^l_{ij}, \tilde{\mathrm{D}}^l_{iv}) \tilde{\mathrm{A}}^l_{ij} \tilde{\mathrm{A}}^l_{iv}}{2\pi \cdot \left(\sum_j \tilde{\mathrm{I}}^l_{ij} + \tilde{\mathrm{M}}^l_{ij} \Delta^l_{ij} \right)} \tag{3}$$

where $\mathbf{\Delta}^l_i \in \mathbb{R}^{n_l}$ represents the Euclidean distances from the position of node i to all other nodes, with entries Δ^l_{ij} computed as:

$$\Delta^l_{ij} = \left\| \mathbf{P}^l_i - \mathbf{P}^l_j \right\|^2_2 \tag{4}$$

The compactness measure derived from the moment of inertia offers several advantages for our hierarchical clustering architecture. First, it is additive, meaning that the compactness of a complex shape can be efficiently computed as a linear combination of the compactness values of its parts. Additionally, as noted in [52], this measure is insensitive to shape size and complex boundaries. By assigning higher scores to compact shapes, the model is encouraged to prioritize regions corresponding to well-defined foreground objects. It is important to note that we use Equation 3 to compute compactness for all affinity masks across all windows in layer l. These computations are performed in parallel and only once per layer, prior to the sequential cluster generation process detailed next. This strategy provides an efficiency advantage over existing Sequential Slot models or recent graph based clustering methods, possibly much more when dealing with crowded scenes.

3.1.5. Sequentially Discovering Object Centroids

A key feature of MI-based compactness measurement emerges when considering the node around which the MI is calculated. When this node coincides with the *centroid* γ of the shape $\mathbf{\Lambda}$, the MI value $I^{\gamma}(\mathbf{\Lambda})$ reaches its minimum, which translates to maximum compactness achievable for any node within $\mathbf{\Lambda}$ [52]. This property of MI-based compactness offers a significant advantage for unsupervised scene segmentation. Nodes corresponding to the centroids of distinct objects will yield the highest compactness scores within their respective objects, as illustrated with Figure 1. By leveraging this spatial inductive bias introduced by compactness measurement, the clustering algorithm can effectively focus on the centroids of distinct objects, facilitating object separation even in challenging cases where closely located objects have similar appearances.

Specifically, our cluster generation algorithm can be considered as a variation of SBC implemented in [6, 18]. Motivated by these works, we start by initializing a *scope* tensor $\mathbf{Z}^l_0 \in \mathbb{R}^{n_l}$ at iteration $m = 0$ as a tensor full of ones with the same dimensions as the compactness scores. Following [6, 18], the scope tensor $\mathbf{Z}^l_m$ represents a map of input nodes which are still unassigned and awaiting clustering in iteration m of the sequential clustering process. Given

$\mathbf{C}_0^l \in \mathbb{R}^{n_l}$ and $\mathbf{Z}_0^l \in \mathbb{R}^{n_l}$, the algorithm proceeds as follows at each iteration m: i) The compactness scores are eroded by element-wise multiplication with the current scope tensor $\mathbf{Z}_m^l$, ensuring that only unassigned nodes (with non-zero scope values) contribute to subsequent calculations, (Eq. 5a). ii) The node ω_m that generates the affinity mask $\mathbf{\Lambda}_{\omega_m}^l$ with the highest compactness score is selected as the anchor of the current cluster, (Eq. 5b). iii) The affinity mask $\mathbf{\Lambda}_{\omega_m}^l$ associated with the selected anchor node is concealed by the current scope and designated as the next output cluster $\mathbf{\Pi}_m^l \in \mathbb{R}^{n_l}$, (Eq. 5c). iv) The scope tensor $\mathbf{Z}_m^l$ is updated by masking out nodes included in $\mathbf{\Pi}_m^l$, excluding these nodes from further iterations, (Eq. 5d). One iteration of the cluster generation process can be summarized as follows (omitting the layer index l):

$$\mathbf{C}_m = \mathbf{C}_{(m-1)} \odot \mathbf{Z}_{(m-1)} \tag{5a}$$
$$\omega_m = \text{argmax}_i \left(\mathrm{C}_{mi}\right) \tag{5b}$$
$$\mathbf{\Pi}_m = \mathbf{\Lambda}_{\omega_m} \odot \mathbf{Z}_{m-1} \tag{5c}$$
$$\mathbf{Z}_m = \mathbf{Z}_{(m-1)} \odot \left(\mathbf{1}^{n_l} - \mathbf{\Pi}_m \right) \tag{5d}$$

3.1.6. Pool, Aggregate and Skip Connect

Following the generation of output cluster masks $\mathbf{\Pi}^l \in \mathbb{R}^{k_l \times n_l}$, COCA layer completes its functionality by using $\mathbf{\Pi}$ to pool and aggregate all output cluster features and attributes to the next layer. To enforce unsupervised feature learning throughout the COCA-Net hierarchy, we make use of skip connections *across* layers, starting from layer $l = 2$ up to the final layer $l = L$. In each layer $l \geq 2$, the inter-layer skip connection procedure involves merging the output cluster masks generated in the last two layers, specifically $\mathbf{\Pi}^{(l-1)}$ and $\mathbf{\Pi}^l$. This merging operation allows COCA-Net to skip-connect the feature tensors between every second layer, $\mathbf{X}^{(l-2)}$ and $\mathbf{X}^{(l)}$, facilitating unsupervised learning with a hierarchical deep learning model. Details on the aggregation and skip connection procedures are provided in Supplementary Material.

3.2. COCA-Net

Following a procedure similar to most sequential, agglomerative, hierarchical, and non-overlapping clustering methods [41], each COCA layer partitions its inputs into non-overlapping windows, then performs clustering and pooling within each window in parallel to produce a reduced number of output clusters. This strategy is repeated at each hierarchical level, progressively reducing the number of input nodes to ultimately obtain object clusters at the last layer. The clustering assignments generated at each layer of the COCA-Net hierarchy are combined into a tree-structure called dendrogram, similar to most HAC algorithms. This dendrogram is used to evaluate the segmentation performance of COCA-Net's encoder sub-network.

4. Experiments and Results

4.1. Baselines and Datasets

With the aim of establishing a comprehensive benchmark for our experiments, we utilize the OCL library provided by [15]. We compare our architecture against three of the state-of-the-art object-centric models that generate scene segmentation masks in an unsupervised way: GEN-v2 [18], INVSA [5] and BOQSA [28]. The evaluation of these baselines and the proposed architecture is carried out across six widely adopted synthetic object discovery datasets, namely, Tetrominoes, Multi-dSprites, ObjectsRoom [31], Shapestacks [23], CLEVR6 and CLEVRTex [30].

To establish a fair comparison between different methods, we utilize the SBD decoder version of each method, as it can be considered as the standard practice for OCL methods on these datasets. Moreover, we optimize solely the pixel-reconstruction loss for all methods and train all architectures from scratch. In addition, we set the number of output slots equal to the maximum number of objects in a dataset for all methods. For the baseline methods, we implement the default hyper-parameters shared in their original work. Remaining training details for COCA-Net and the baselines can be found in Supplementary Material.

4.2. Metrics and Evaluation Configurations

In line with previous work [18, 28], we adopt two performance metrics namely Adjusted Rand Index (ARI) [26] and mean Segmentation Covering (mSC) [17] for segmentation performance and Mean Squared Error (MSE) for reconstruction success. In accordance with the customs of the literature, we provide the segmentation performance of the object slot masks that are generated at the end of the decoder sub-networks. To assess COCA-Net's potential to be leveraged as a stand-alone unsupervised scene segmentation encoder, we also compare object slot masks that are obtained at the encoder sub-networks of the overall pipeline. Finally, to analyze COCA-Net's overall scene segmentation performance, we include background segments in our evaluation criteria. We trained all models using three random seeds and report results as mean $\pm$ standard deviation.

4.3. Results

Quantitative results are summarized in Tables 1, 2, whereas qualitative results are displayed across Figure 3. Note that an extension of Table 1 can be found in Supplementary Material. As it can be observed from the overall results, COCA-Net outperforms the state-of-the-art on all datasets across most metrics and displays highly competitive performance in others. Regarding the evaluation of foreground object masks produced by the decoder sub-network, COCA-Net achieves the highest ARI and mSC performance in almost all datasets, sharing the lead with [5] in Object-

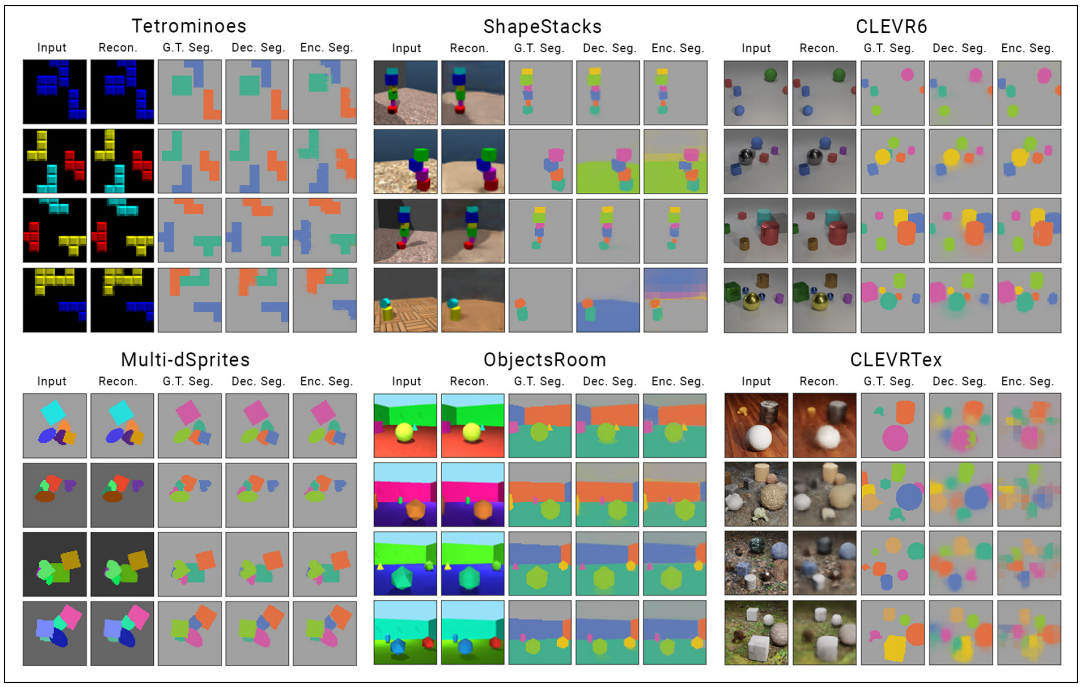


Figure 3. Qualitative results of the COCA-Net architecture across six datasets examined in this work. For each dataset, we present four challenging samples along with COCA-Net's reconstructions. Ground truth segmentation masks, as well as segmentation masks from the decoder and encoder sub-networks, are also included. The mask with the highest intersection with the background segment is shown in gray.

Table 1. Quantitative Results of proposed COCA-Net and baseline BOQSA on Tetrominoes and Multi-dSprites datasets. Evaluation configurations are: masks generated by encoder, (1) only foreground objects, (2) including background segments. Scores are reported as mean ± standard deviation for 3 seeds.

	ENC-FG Only		ENC-BG Included	
Name	ARI↑	mSC↑	ARI↑	mSC↑
Tetrominoes				
BOQSA [28]	0.60±0.03	0.42±0.01	0.29±0.02	0.47±0.00
COCA-Net (ours)	**0.74±0.04**	**0.70±0.03**	**0.71±0.09**	**0.75±0.06**
Multi-dSprites				
BOQSA [28]	0.75±0.01	0.55±0.01	0.34±0.06	0.56±0.02
COCA-Net (ours)	**0.96±0.01**	**0.95±0.01**	**0.98±0.00**	**0.96±0.00**

sRoom. Inspecting the foreground object masks generated by the encoder sub-networks, a dendrogram in our case, COCA-Net significantly surpasses its competitors, achieving an improvement of nearly thirty percent in both metrics on the ShapeStacks dataset. Notably, COCA-Net is more robust compared to its baselines in almost every dataset, exhibiting less variance across different training runs. Examining the segmentation results including the background segments, one can observe that COCA-Net is a better option for holistic scene segmentation, providing approximately thirty percent performance increase over [28] and [5] on ObjectsRoom. The only exception is COCA-Net's ARI results on background segments of the ShapeStacks dataset. Both ShapeStacks and ObjectsRoom datasets contain multiple background segments but ShapeStacks dataset provides a single ground truth mask for all background segments. Therefore, COCA-Net's poor ARI performance on background segments of ShapeStacks is expected. Analyzing the qualitative results depicted in Figure 3, one can conclude that COCA-Net produces coherent segments that match with human intuition across all datasets, including ShapeStacks, where this intuitive behavior of COCA-Net is also supported by the quantitative mSC results.

4.4. Ablation Studies

We conduct two ablation studies for our design choices across the COCA-Net architecture. These ablation studies are designed as follows; (1) to verify the effectiveness of compactness based mask selection and SBC based clustering, we consider the original recipe laid out in GEN-v2 [18], where anchor node selection is carried out by sampling a random pixel from a uniform distribution that is concealed by the current scope. We name this version as COCA-Net-RAS to indicate "random anchor node selection". (2) To assess COCA-Net's flexibility on generating a varying number of output slots, we develop a version of COCA-Net that keeps on generating object masks until a stopping condi-

Table 2. Unsupervised scene segmentation results of three baseline models and proposed COCA-Net on four multi-object datasets based on a total of nine performance evaluation metrics assessed across four different configurations. Scores are reported as mean ± standard deviation for 3 seeds. The four different evaluation configurations are: (1) masks generated by decoder, only foreground objects, (2) masks generated by decoder, with background segments, (3) masks generated by encoder only foreground objects, (4) masks generated by encoder with background segments.

Name	DEC-FG Only		DEC-BG Included		ENC-FG Only		ENC-BG Included		
	ARI↑	mSC↑	ARI↑	mSC↑	ARI↑	mSC↑	ARI↑	mSC↑	MSE↓
ObjectsRoom									
GEN-v2 [18]	0.86±0.01	0.57±0.03	0.11±0.00	0.38±0.01	0.67±0.07	0.18±0.02	0.13±0.00	0.23±0.00	0.003±0.000
INV-SA [5]	**0.88±0.00**	0.80±0.01	0.66±0.12	0.68±0.07	0.80±0.02	0.44±0.02	0.36±0.01	0.39±0.01	0.001±0.001
BOQ-SA [28]	0.87±0.01	**0.83±0.00**	0.57±0.00	0.63±0.00	0.59±0.02	0.69±0.03	0.52±0.01	0.56±0.01	**0.001±0.000**
COCA-Net (ours)	**0.88±0.00**	0.82±0.01	**0.95±0.01**	**0.87±0.02**	**0.87±0.01**	**0.82±0.01**	**0.94±0.02**	**0.86±0.02**	**0.001±0.000**
ShapeStacks									
GEN-v2 [18]	0.83±0.01	0.70±0.01	**0.87±0.00**	0.77±0.01	0.54±0.04	0.47±0.03	**0.60±0.04**	0.58±0.03	0.003±0.000
INV-SA[5]	0.65±0.13	0.64±0.07	0.12±0.02	0.55±0.05	0.55±0.11	0.37±0.04	0.09±0.01	0.36±0.04	0.004±0.002
BOQ-SA [28]	0.83±0.09	0.80±0.09	0.21±0.07	0.73±0.10	0.49±0.16	0.59±0.13	0.18±0.09	0.58±0.13	**0.001±0.000**
COCA-Net (ours)	**0.91±0.01**	**0.85±0.01**	0.31±0.09	**0.79±0.01**	**0.82±0.02**	**0.85±0.01**	0.25±0.04	**0.78±0.01**	0.006±0.003
CLEVR6									
GEN-v2 [18]	0.43±0.10	0.18±0.00	0.08±0.01	0.21±0.00	0.17±0.24	0.07±0.05	0.05±0.07	0.16±0.01	0.004±0.000
INV-SA [5]	0.96±0.01	**0.87±0.03**	0.74±0.35	**0.87±0.07**	0.64±0.04	0.54±0.02	0.30±0.07	0.55±0.04	**0.000±0.000**
BOQ-SA [28]	0.86±0.21	0.34±0.12	0.08±0.07	0.33±0.11	0.78±0.26	0.30±0.12	0.08±0.07	0.31±0.13	**0.000±0.000**
COCA-Net (ours)	**0.97±0.02**	0.82±0.06	**0.88±0.04**	0.85±0.05	**0.96±0.03**	**0.76±0.02**	**0.82±0.08**	**0.79±0.02**	**0.000±0.000**
CLEVRTex									
GEN-v2 [18]	0.22±0.08	0.10±0.00	0.02±0.04	0.14±0.00	0.00±0.00	0.04±0.00	0.00±0.00	0.15±0.00	0.009±0.000
INV-SA [5]	0.60±0.12	0.35±0.13	0.21±0.11	0.38±0.12	0.53±0.11	0.28±0.08	0.14±0.05	0.31±0.06	0.009±0.003
BOQ-SA [28]	0.69±0.06	0.44±0.07	0.18±0.05	0.43±0.07	**0.53±0.03**	0.34±0.07	0.14±0.06	0.34±0.08	**0.005±0.002**
COCA-Net (ours)	**0.76±0.02**	**0.54±0.03**	**0.50±0.06**	**0.57±0.03**	0.44±0.03	**0.39±0.03**	**0.39±0.03**	**0.45±0.03**	0.008±0.001

tion is met, e.g., number of remaining elements in the scope $\mathbf{Z}$ drops below a threshold, similar to one described in [18]. This version, that we refer to as COCA-Net-Dyna, is trained on CLEVR6 with a fixed number of output slots and evaluated on CLEVR10 with dynamic number of output slots. Results for these ablation studies are displayed in Table 3. The integration of compactness scoring significantly enhances the performance of the COCA layer in multi-object segmentation. Furthermore, COCA-Net demonstrates the ability to dynamically adjust its output representation capacity without compromising performance.

Table 3. Results of the Ablation Studies (ASs) conducted for the proposed COCA-Net architecture. For both studies, we use decoder-generated foreground segmentation accuracy as the comparison criterion.

AS 1: Effectiveness of Compactness Measurement		
Model	ARI	mSC
COCA-Net on ObjectsRoom	**0.894**	**0.832**
COCA-Net-RAS on ObjectsRoom	0.832	0.739
AS 2: Variable Number of Output Slots		
Model	ARI	mSC
COCA-Net on CLEVR6	0.985	0.881
COCA-Net-Dyna on CLEVR10	0.978	0.840

5. Conclusion

COCA-Net provides state-of-the-art segmentation results across various configurations and datasets. Notably, it is more robust compared to its competitors, yielding significantly lower variance across training runs. Segmentation masks generated by COCA-Net, both in the encoder and decoder sub-networks of its pipeline, exhibit accurate and highly competitive performance, highlighting its potential as a stand-alone unsupervised image segmentation encoder. In addition, COCA-Net can produce more coherent background segments compared to its baselines and can dynamically adjust the number of output slots.

A current shortcoming of COCA-Net is its bottom-to-top clustering approach, which lacks top-down feedback. This prevents the model from correcting errors made in earlier clustering layers. Future work might consider implementing a top-down feedback pathway, possibly replacing the standard SBD decoder architecture. Another drawback of the COCA layer, as currently framed, is the calculation of moment of inertia of the reference circle. This mass-normalized compactness measurement involves pairwise comparison of densities within each affinity mask, creating a heavy memory requirement. Future work can address this shortcoming by implementing an efficient approximation for the MI calculation of the reference shape.

References

[1] Pablo Arbelaez, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

[2] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. 3

[3] Shahaf Arica, Or Rubin, Sapir Gershov, and Shlomi Laufer. CuVLER: Enhanced Unsupervised Object Discoveries through Exhaustive Self-Supervised Transformers . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23105–23114, Los Alamitos, CA, USA, 2024. IEEE Computer Society. 3

[4] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *International Conference on Machine Learning*, 2019. 3

[5] Ondrej Biza, Sjoerd Van Steenkiste, Mehdi S. M. Sajjadi, Gamaleldin Fathy Elsayed, Aravindh Mahendran, and Thomas Kipf. Invariant slot attention: Object discovery with slot-centric reference frames. In *Proceedings of the 40th International Conference on Machine Learning*, pages 2507–2527. PMLR, 2023. 2, 6, 7, 8, 5

[6] Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew M. Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *ArXiv*, abs/1901.11390, 2019. 3, 5

[7] Shengcao Cao, Dhiraj Joshi, Liangyan Gui, and Yu-Xiong Wang. HASSOD: Hierarchical adaptive self-supervised object detection. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[8] Shengcao Cao, Jiuxiang Gu, Jason Kuen, Hao Tan, Ruiyi Zhang, Handong Zhao, Ani Nenkova, Liangyan Gui, Tong Sun, and Yu-Xiong Wang. SOHES: Self-supervised open-world hierarchical entity segmentation. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. 3

[10] Ayush K Chakravarthy, Trang Nguyen, Anirudh Goyal, Yoshua Bengio, and Michael Curtis Mozer. Spotlight attention: Robust object-centric learning with a spatial locality prior. *ArXiv*, abs/2305.19550, 2023. 2

[11] Michael Chang, Tom Griffiths, and Sergey Levine. Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation. In *Advances in Neural Information Processing Systems*, pages 32694–32708. Curran Associates, Inc., 2022. 1, 2

[12] Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *AAAI Conference on Artificial Intelligence*, 2019. 3

[13] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *Proceedings of the 40th International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023. 4, 2

[14] Zhiwei Deng, Ting Chen, and Yang Li. Perceptual group tokenizer: Building perception with iterative grouping. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[15] A. Dittadi, S. S. Papa, M. De Vita, B. Schölkopf, O. Winther, and F. Locatello. Generalization and robustness implications in object-centric learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 5221–5285. PMLR, 2022. 1, 2, 6, 5

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 3

[17] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *ArXiv*, abs/1907.13052, 2019. 3, 6

[18] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. In *Neural Information Processing Systems*, 2021. 3, 5, 6, 7, 8

[19] B. Everitt, S. Landau, M. Leese, D. Stahl, and an O'Reilly Media Company Safari. *Cluster Analysis, 5th Edition*. John Wiley & Sons, 2011. 2

[20] Ke Fan, Zechen Bai, Tianjun Xiao, Tong He, Max Horn, Yanwei Fu, Francesco Locatello, and Zheng Zhang. Adaptive slot attention: Object discovery with dynamic slot number, 2024. 2

[21] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., USA, 2006. 2

[22] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *ArXiv*, abs/2012.05208, 2020. 1, 2, 3

[23] Oliver Groth, Fabian B. Fuchs, Ingmar Posner, and Andrea Vedaldi. Shapestacks: Learning vision-based physical intuition for generalised object stacking. In *Computer Vision – ECCV 2018*, pages 724–739, Cham, 2018. Springer International Publishing. 6

[24] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision GNN: An image is worth graph of nodes. In *Advances in Neural Information Processing Systems*, 2022. 3

[25] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009. 2

[26] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985. 6

[27] Anil K. Jain and Richard C. Dubes. Algorithms for clustering data. 1988. 2

[28] Baoxiong Jia, Yu Liu, and Siyuan Huang. Improving object-centric learning with query optimization. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 6, 7, 8, 5

[29] Jindong Jiang, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. In *International Conference on Learning Representations*, 2019. 3

[30] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997, 2017. 6

[31] Rishabh Kabra, Chris Burgess, Loic Matthey, Raphael Lopez Kaufman, Klaus Greff, Malcolm Reynolds, and Alexander Lerchner. Multi-object datasets. https://github.com/deepmind/multi-object-datasets/, 2019. 6

[32] Sandra Kara, Hejer Ammar, Florian Chabot, and Quoc-Cuong Pham. The Background Also Matters: Background-Aware Motion-Guided Objects Discovery . In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1205–1214, Los Alamitos, CA, USA, 2024. IEEE Computer Society. 2

[33] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 2009. 2

[34] Tsung-Wei Ke, Jyh-Jing Hwang, Yunhui Guo, Xudong Wang, and Stella X. Yu. Unsupervised hierarchical semantic segmentation with multiview cosegmentation and clustering transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2561–2571, 2022. 3

[35] Tsung-Wei Ke, Sangwoo Mo, and Stella X. Yu. Learning hierarchical image segmentation for recognition and by recognition. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[36] Jinwoo Kim, Janghyuk Choi, Ho-Jin Choi, and Seon Joo Kim. Shepherding slots to objects: Towards stable and robust object-centric learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19198–19207, 2023. 1, 2

[37] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*, 2020. 3

[38] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, pages 11525–11538. Curran Associates, Inc., 2020. 1, 2, 5

[39] Bryan H. Massam and Michael F. Goodchild. Temporal trends in the spatial organization of a service agency. *Canadian Geographies / Géographies canadiennes*, 15(3):193–206, 1971. 5

[40] Duy-Kien Nguyen, Mahmoud Assran, Unnat Jain, Martin R. Oswald, Cees G. M. Snoek, and Xinlei Chen. An image is worth more than 16x16 patches: Exploring transformers on individual pixels. *ArXiv*, abs/2406.09415, 2024. 3

[41] Xingcheng Ran, Yue Xi, Yonggang Lu, Xiangwen Wang, and Zhenyu Lu. Comprehensive survey on hierarchical clustering algorithms and the recent developments. *Artificial Intelligence Review*, 56:8219 – 8264, 2022. 2, 6

[42] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[43] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2002. 3

[44] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *J. Mach. Learn. Res.*, 24(1), 2024. 3

[45] J R R Uijlings, K E A van de Sande, T Gevers, and A W M Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. 3

[46] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017. 3, 2

[47] Xudong Wang, Rohit Girdhar, Stella X. Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3124–3134, 2023. 3

[48] Xudong Wang, Jingfeng Yang, and Trevor Darrell. Segment anything without supervision. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[49] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L. Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12):15790–15801, 2023. 3

[50] Nicholas Watters, Loïc Matthey, Christopher P. Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *ArXiv*, abs/1901.07017, 2019. 2

[51] Elizabeth A. Wentz Wenwen Li, Tingyong Chen and Chao Fan. Nmmi: A mass compactness measure for spatial pattern analysis of areal features. *Annals of the Association of American Geographers*, 104(6):1116–1133, 2014. 5

[52] Michael F. Goodchild Wenwen Li and Richard Church. An efficient measure of compactness for two-dimensional shapes and its application in regionalization problems. *International Journal of Geographical Information Science*, 27 (6):1227–1250, 2013. 5, 3

[53] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 4

[54] Rui Xu and Donald C. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16:645–678, 2005. 2

[55] Yafei Yang and Bo Yang. Benchmarking and analysis of unsupervised object segmentation from Real-World single images. *International Journal of Computer Vision*, 132(6): 2077–2113, 2024. 1

[56] Yan Zhang, David W. Zhang, Simon Lacoste-Julien, Gertjan J. Burghouts, and Cees G. M. Snoek. Unlocking slot attention by changing optimal transport costs. In *Proceedings of the 40th International Conference on Machine Learning*, pages 41931–41951. PMLR, 2023. 2

[57] Roland S. Zimmermann, Sjoerd van Steenkiste, Mehdi S. M. Sajjadi, Thomas Kipf, and Klaus Greff. Sensitivity of slot-based object-centric models to their number of slots. *CoRR*, abs/2305.18890, 2023. 2