Object-centric Learning with Cyclic Walks between Parts and Whole

Anonymous Author(s)
Affiliation
Address
email

Abstract

Learning object-centric representations from complex natural environments enables both humans and machines with reasoning abilities from low-level perceptual features. To capture compositional entities of the scene, we proposed cyclic walks between perceptual features extracted from CNN or transformers and object entities. First, a slot-attention module interfaces with these perceptual features and produces a finite set of slot representations. These slots can bind to any object entities in the scene via inter-slot competitions for attention. Next, we establish entity-feature correspondence with cyclic walks along high transition probability based on pairwise similarity between perceptual features (aka "parts") and slot-binded object representations (aka "whole"). The whole is greater than its parts and the parts constitute the whole. The part-whole interactions form cycle consistencies, as supervisory signals, to train the slot-attention module. Our rigorous experiments on *seven* image datasets in *three unsupervised* tasks demonstrate that the networks trained with our cyclic walks can disentangle foregrounds and backgrounds, discover objects, and segment semantic objects in complex scenes. In contrast to object-centric models attached with a decoder for the pixel-level or feature-level reconstructions, our cyclic walks provide strong learning signals, avoiding computation overheads and enhancing memory efficiency.

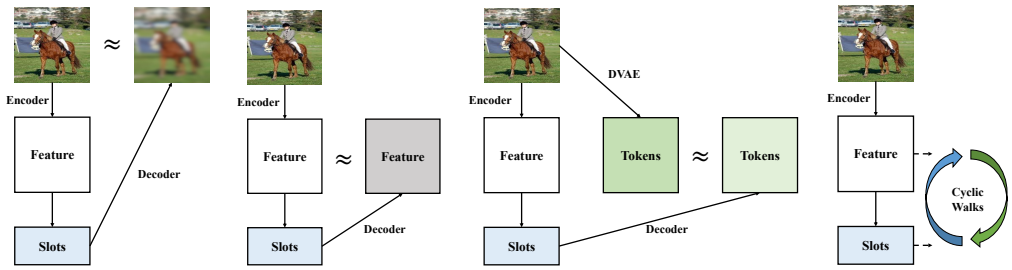
1 Introduction

Object-centric representation learning refers to the ability to decompose the complex natural scene into multiple object entities and establish the relationships among these objects [27, 22, 12, 31, 32, 30, 20]. It is important in multiple applications, such as visual perception, scene understanding, reasoning, and human-object interaction [37, 2]. However, learning to extract object-centric representations from the complex natural scenes in an unsupervised manner remains a challenge in machine vision.

Recent works attempt to overcome this challenge by relying on image or feature reconstructions from object-centric representations as supervision signals [27, 31, 20, 30] (Figure 1). However, these reconstructions have several caveats. First, these methods often require an additional decoder network, resulting in computation overheads and memory inefficiency. Moreover, these methods focus excessively on reconstructing unnecessary details at the pixel or feature levels and sometimes fail to capture object-centric representations from a holistic view.

To mitigate issues from reconstructions, some studies [38, 17] introduce mutual information maximization between predicted object-centric representations and feature maps. Building upon this idea, contrastive learning has become a powerful tool in unsupervised object-centric learning. The recent works [19, 5] propose to learn the spatial-temporal correspondences between a sequence of video frames with contrastive random walks. The objective is to maximize the likelihood of returning to the starting node by walking along the graph constructed from a palindrome of frames and repelling nodes with distinct features from adjacent frames in the graph.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.



(a) SA, BO-QSA (b) DINOSAUR (c) SLATE (d) Cyclic walks

Figure 1: **Comparison of the network structures trained with existing methods and our cyclic walks.** Supervision signal is indicated by ($\approx$) in subplots a-c. From left to right, all the methods require some forms of reconstructions as supervisory signals except for our method (d): (a) image reconstruction for the Slot-Attention (SA) [27] and BO-QSA [20] (b) feature reconstruction for DINOSAUR [30](c) tokens obtained by a Discrete Variational Autoencoder [29] (DVAE) for SLATE [31] (d). Our cyclic walks do not require any image or feature reconstructions.

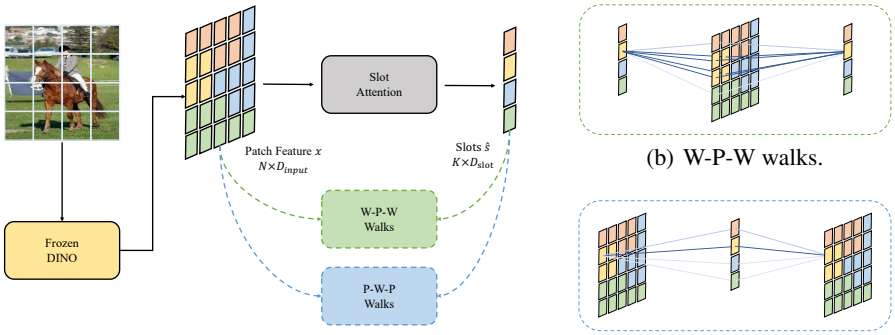
Generalizing from contrastive random walks on video sequences, we proposed cyclic walks on static images. These walks are conducted between predicted object entities and feature maps extracted from CNN or transformers. The cycle consistency of these walks serves as supervision signals to guide object-centric representation learning. First, inspired by the set encoding theory depicting that natural images can always be represented as a span of a finite set of task-dependent semantically meaningful bases, we use a slot-attention module to interface with feature maps of an image and produce a set of object-centric representations out of the slot bases. These slot bases compete with one another to bind with feature maps at any spatial location via cross-attention mechanism [7]. All the features that resemble a particular slot basis are aggregated into one object-centric representation explaining parts of the image.

Next, two types of cyclic walks, along high transition probability based on the pairwise similarity between aggregated object-centric representations and feature maps, are established to learn entity-feature correspondence. The motivation of the cyclic walks draws a similar analogy with the part-whole theory stating that the whole (i.e. object-centric representations) is greater than its parts (i.e. feature maps extracted from natural images) and the parts constitute the whole. On one hand, cyclic walks in the direction from the whole to the parts and back to the whole (short for "W-P-W walks") encourage diversity of the learnt slot bases. On the other hand, cyclic walks in the direction from the parts to the whole and back to the parts (short for "P-W-P walks") broaden the coverage of the slot bases pertaining to the image content. Both W-P-W walks and P-W-P walks form cycle consistencies and serve as supervision signals to enhance the specificity and diversity of learned object-centric representations. Our contributions are highlighted below:

- We introduce cyclic walks between parts and whole to regularize object-centric learning. Both W-P-W walks and P-W-P walks form virtuous cycles to enhance the specificity and diversity of the learned object-centric representations.
- We verify the effectiveness of our method over seven image datasets in three unsupervised vision tasks. Our method surpasses the state-of-the-art by a large margin.
- Compared with the previous object-centric methods relying on reconstruction losses, our method does not require additional decoders in the architectures, greatly reducing the computation overheads and improving memory efficiency.

2 Related Works

One representative work [27] introduces the slot-attention module, where the slots compete to bind with certain regions of the image via a cross-attention mechanism and distill image features into object-centric representations. To train these slots, the networks are often attached to a decoder to reconstruct either images or features based on object-centric representations (Figure 1). Subsequent Slot-Attention methods [32, 22, 12, 4, 11] expand to video processing. To learn object-level representations, these methods often require either reconstructions from video frames or optical flows [4, 11]. Although the original Slot-Attention demonstrates to be effective in most cases, it fails to generalize to complex scenes [30, 20]. To mitigate this issue, DINOSAUR *freezes* the feature extractor while learning



(a) Network architecture of our cyclic walks.

(b) W-P-W walks.

(c) P-W-P walks.

Figure 2: **Overview of our proposed cyclic walks.** The input image is divided into non-overlapped patches and encoded into feature vectors (aka "parts") through a frozen DINO ViT pre-trained on ImageNet with unsupervised learning methods [8]. The Slot-Attention module aggregates the patch features into object-centric representations (aka "whole") based on the learnable slot bases. To train the network, we perform the contrastive random walks in two directions between the parts and the whole: (b) any node from the whole should walk to the parts and back to itself (W-P-W walks) and (c) any node from the parts should walk to the whole and back to itself (P-W-P walks).

the slot bases.Its frozen feature extractor was pre-trained on ImageNet in unsupervised learning [8]. BO-QSA [20] proposes to initialize the slots with learnable queries and optimize these queries with bi-level optimization. Both works emphasize that good initialization of network parameters is essential for object-centric learning in complex scenes. Thus, following this line of research, we also use a frozen DINO [8] as the feature extractor. However, different from these works, we introduce cyclic walks in the part-whole hierarchy of the same image, without decoders for reconstructions.

Object-centric learning for unsupervised semantic segmentation. One of the key applications in object-centric representation learning is unsupervised semantic segmentation. Several notable works include STEGO [16], DeepCut [1], ACSeg [24], and FreeSOLO [34]. All these methods employ a fixed pre-trained transformer as a feature extractor, which has proven to be effective for unsupervised semantic segmentation. In contrast to these methods which directly perform optimization on pixel clusters or feature clusters, we introduce slot-attention and perform cyclic walks between slots and feature maps. Experimental results demonstrate the superior performance of our method.

Relation to contrastive learning. Our method connects object-centric representation learning with unsupervised contrastive learning. We highlight the differences between our method and the contrastive learning methods. Infoseg [17] reconstructs an image from a set of global vectors (aka "slot bases"). The reconstructed image is pulled closer to the original image and repelled away from a randomly selected image. SlotCon [36] introduces contrastive learning on two sets of slot bases extracted from two augmented views of the same image. In contrast to the two methods, our cyclic walks curate both positive and negative pairs along the part-whole hierarchy from the *same* image. Another method, SAN [38], performs contrastive learning between slot bases to encourage slot diversity, where every slot is repelled away from one another. Sharing similar motivations, our walks from the whole to the parts and back to the whole ("W-P-W" walks) strengthen the diversity of the slot bases. Moreover, our method also introduces "P-W-P" walks, which takes the hollistic view of the entire image, and broadens the slot coverage.

3 Preliminary

Here, we mathematically formulate **Slot-Attention Module** and **Contrastive Random Walks**.

3.1 Slot-Attention Module

The slot-attention module takes an input image and outputs object-centric representations. Given an input image, a pre-trained CNN or transformer extracts its feature vectors $x \in R^{N \times D_{input}}$, where $N = W \times H$. H and W denote the height and width of the feature maps. Taking x and a set of K slot bases $s \in R^{K \times D_{slot}}$ as inputs, Slot-Attention binds s with x, and outputs a set of K object-centric feature vectors $\tilde{s}$ of the same size as s. The Slot-Attention module employs the cross-attention

mechanism [33], where x contributes to keys and values and s contributes to queries. Specifically, the inputs x are mapped to the dimension D with linear functions $k(\cdot)$ and $v(\cdot)$ and the slots s are mapped to the same dimension with linear function $q(\cdot)$. For each feature vector at every spatial location of the feature maps x, attention values are calculated with respect to all slot bases. To prevent parts of the image from being unattended, the attention matrix is normalized with the softmax function first over K slots and then over N locations:

$$\texttt{attn}_{i,j} = \frac{e^{A_{i,j}}}{\sum_{K} e^{A_{i,j}}} \text{ where } A = \frac{k(x)q(s)^{\mathsf{T}}}{\sqrt{D}} \in R^{N \times K} \tag{1}$$

$$\tilde{s} = W^T v(x) \text{ where } W_{i,j} = \frac{\texttt{attn}_{i,j}}{\sum_{N} \texttt{attn}_{i,j}} \tag{2}$$

The cross-attention mechanism introduces competitions among slot bases for explaining parts of the image. Based on the attention matrix, an object-centric feature vector from $\tilde{s}$ is distilled by the weighted sum of feature maps x. In the literature [7, 31, 27] and our method, the slot attention modules iteratively refine the output $\tilde{s}$ from the slots in a recurrent manner with a Gated Recurrent Unit (GRU) for better performances. We represent the initial slot bases as s^0. The parameters of s^0 are initialized by randomly sampling from Gaussian distribution with learnable mean μ and variance σ. The output $\tilde{s}$ at time step t acts as the slot bases and feeds back to the GRU at the next recurrent step $t+1$. Thus, the final object-centric representations $\hat{s}$ after T iterations can be formulated as:

$$\hat{s} = s^T, \text{ where } s^{t+1} = GRU(s^t, \tilde{s}^t) \tag{3}$$

After obtaining the object-centric representation $\hat{s}$, the traditional object-centric learning models decode the images from the slots with a mixture-based decoder or a transformer-based decoder [7]. The training objective of the models is to minimize the Mean Square Error loss between the output of the decoder and the original image at the feature or pixel levels.

3.2 Contrastive Random Walks

A random walk describes a random process where an independent path consists of a series of hops between nodes on a directed graph in a latent space. Without loss of generality, given any pair of feature sets $a \in R^{m \times d}$ and $b \in R^{n \times d}$, where m and n are the numbers of nodes in the feature sets and d is the feature dimension, the adjacency matrix $M_{a,b}$ between a and b can be calculated as their normalized pairwise feature similarities:

$$M_{a,b} = \frac{e^{f(a)f(b)^T/\tau}}{\sum_{n} e^{f(a)f(b)^T/\tau}} \in R^{m \times n}, \tag{4}$$

where $f(\cdot)$ is the l_2-normalization and τ is the temperature controlling the sharpness of distribution with its smaller values indicating sharper distribution.

In the original work [19], random walks serve as supervisory signals to train the object-centric learning models to capture the space-time correspondences from raw videos. The contrastive random walks are formulated as a graph. a and b are the feature maps extracted from video frames with either a CNN or a transformer. m and n are the numbers of spatial locations on the feature maps and they are often the same across video frames. On the graph, only nodes from adjacent video frames F_t and F_{t+1} at time t and $t+1$ share a directed edge. Their edge strengths, indicating the transition probabilities of a random walk between frames, are the adjacency matrix $M_{F_t,F_{t+1}}$. The objective of the contrastive random walks is to maximize the likelihood of a random walk returning to the starting node along the graph constructed from a palindrome video sequence $\{F_t, F_{t+1}, ..., F_T, F_{T-1}, ..., F_t\}$.

4 Our Method

4.1 Object-centric Feature Encoding

Following [16, 30], we use a self-supervised vision transformer trained with DINO on ImageNet, as the image feature extractor (Figure 2a). In Section 5.4, we verify that our method is also agnostic

to other self-supervised feature learning backbones. Given an input image, DINO parses it into non-overlapped patches and each patch is projected into a feature token. We keep all patch tokens except for the classification token in the last block of DINO. Same as Section 3.1, we use the same notation $x \in R^{N \times D_{input}}$ to denote feature maps extracted from a static image. The Slot-Attention module takes the feature vectors $x \in R^{N \times D_{input}}$ and a set of K learnable slot bases $s \in R^{K \times D_{slot}}$ as inputs and produces the object-centric representations $\hat{s} \in R^{K \times D_{slot}}$ (Equations 1 and 2).

4.2 Whole-Parts-Whole Cyclic Walks

Features of the image patches constitute the objects. The interactions between parts and the whole provide a mechanism for clustering and discovering objects in the scene. Motivated by this, we introduce cyclic walks in two directions: (a) from the whole to the parts and back to the whole (W-P-W walks) and (b) from the parts to the whole and back to the parts (P-W-P walks). Both serve as supervisory signals for learning slot bases s.

Given feature vectors $x \in R^{N \times D_{input}}$ of all image patches (aka "parts") and the object-centric representations $\hat{s} \in R^{K \times D_{slot}}$ (aka "whole"), we apply a linear layer to map both x and $\hat{s}$ to be of the same dimension D. In practice, K is much smaller than N. Following the formulations of Contrastive Random Walks [19] in Section 3.2, random walks are conducted along high transition probability based on pairwise similarity from $\hat{s}$ to x and from x to $\hat{s}$ using Equation 4:

$$M_{wpw} = M_{\hat{s},x} M_{x,\hat{s}} \in R^{K \times K}, \text{ where } M_{\hat{s},x} \in R^{K \times N} \text{ and } M_{x,\hat{s}} \in R^{N \times K} \tag{5}$$

Ms are the adjacent matrices. Different from the original work [19] performing contrastive random walks on a palindrome video sequence, here, we enforce a palindrome walk on a part-whole hierarchy on the image. Ideally, if the W-P-W walks are successful, all the bases in the slot-attention module have to establish one-to-one mapping with certain parts of the image. This encourages the network to learn diverse object-centric representations so that each slot basis can explain certain parts of the image and every part of the image can correspond to a unique slot basis. To achieve this in W-P-W walks, we enforce that the probability of walking from $\hat{s}$ to x belonging to the object class and back to $\hat{s}$ itself should be an identity matrix I of size $K \times K$. The first loss term is defined as: $L_{wpw} = CE(M_{wpw}, I)$

4.3 Parts-Whole-Parts Cyclic Walks

Though the W-P-W walks enhance the diversity of the slot bases, there is an ill-posed situation, when there exists a limited set of slot bases failing to cover all the semantic content of an image but every trivial W-P-W walk is always successful. For example, given two slot bases and an image consisting of a triangle, a circle, and a square on a background, one slot could prefer "triangles", while the other prefers "circles". In this case, the W-P-W walks are always successful; however, the two slots fail to represent squares and the background, defeating the original intention of foreground and background extraction. To mitigate this problem and bind all the slots with all the semantic content in the entire scene, we introduce additional walks from the parts to the whole and back to the parts (P-W-P walks), complementary to W-P-W walks: $M_{pwp} = M_{x,\hat{s}} M_{\hat{s},x} \in R^{N \times N}$. As N is typically far larger than K and features x at nearby locations tend to be similar, the probabilities of random walks beginning from one feature vector of x, passing through $\hat{s}$, and returning to itself could no longer be 1. Thus, we use the feature-feature correspondence S as the supervisory signal to regularize $\hat{s}$:

$$S_{i,j} = \frac{e^{W_{i,j}}}{\sum\limits_{N} e^{W_{i,j}}} \text{ where } W_{i,j} = \begin{cases} -\infty, & \text{if } F_{i,j} <= \gamma \\ F_{i,j}, & \text{if } F_{i,j} > \gamma \end{cases}, \text{ and } F = f(x) f(x)^T \in R^{N \times N} \tag{6}$$

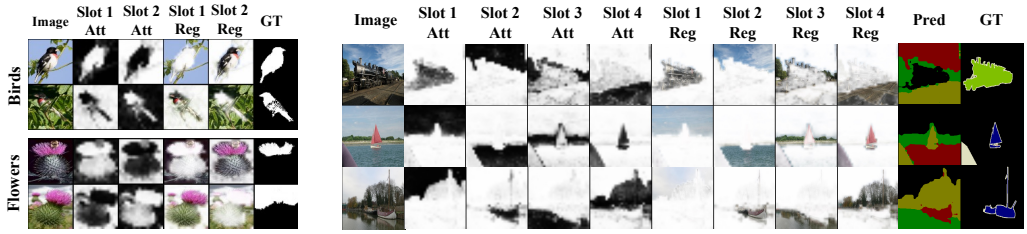
Hyper-parameter γ is the similarity threshold for preventing the random walks from returning to locations where their features are not as similar as the starting point. Thus, the overall loss of our cyclic walks can be calculated below, where α and β are the coefficients balancing the two losses.

$$L = \alpha L_{wpw} + \beta L_{pwp}, \text{ where } L_{pwp} = CE(M_{pwp}, S) \tag{7}$$

Training Configuration. We use ViT-Small pre-trained with DINO as our feature extractor and freeze the parameters of DINO throughout the entire training process. There are two justifications for freezing the feature extractor. First, pre-trained unsupervised feature learning frameworks have

Model	Birds		Dogs		Cars		Flowers	
	mIoU	Dice	mIoU	Dice	mIoU	Dice	mIoU	Dice
Slot-Attention [27]	35.6	51.5	39.6	55.3	41.3	58.3	30.8	45.9
SLATE [31]	36.1	51.0	62.3	76.3	75.5	85.9	68.1	79.1
DINOSAUR [30]	67.2	78.4	73.7	84.6	80.1	87.6	72.9	82.4
BO-QSA [20]	71.0	82.6	82.5	90.3	87.5	93.2	**78.4**	**86.1**
Cyclic walks (ours)	**72.4**	**83.6**	**86.2**	**92.4**	**90.2**	**94.7**	75.1	83.9

Table 1: **Results in the unsupervised foreground extraction task.** We report the results in mIoU and Dice on CUB200 Birds (Birds), Stanford Dogs (Dogs), Stanford Cars (Cars), and Flowers (Flowers) datasets. Numbers in bold are the best.



(a) Foreground extraction (b) Object discovery

Figure 3: **Visualization of attention maps predicted by slot bases.** In (a) unsupervised foreground extraction, we present two examples each from Birds, and Flowers datasets. In each example, for all the slots, we provide their attention maps (Col.2-3) and their corresponding attended image regions (Col.4-5). Ground truth foreground and background masks are shown in the last Col. In (b) unsupervised object discovery, we provide 3 examples on the PASCAL VOC dataset. In each example, for all the slots, we provide their attention maps (att., Col.2-5) and their corresponding attended image regions (reg., Col. 6-9). Their combined object discovery masks from all the slots are shown in Col.7 (pred.). Each randomly assigned color denotes an image region activated by one slot. Ground truth object masks are presented in the last Col.

already generated semantically consistent content in nearby locations [16, 1, 24]. Object-centric learning addresses its follow-up problem of capturing compositional entities out of the semantic content. Second, recent object-centric learning works [30, 20] have emphasized the importance of freezing the pre-trained weights of feature extractors (see Section 2). For a fair comparison with these methods, we keep the parameters of the feature extractor frozen. We list out the choices of all hyperparameters and elaborate on the training details in **Supp. Material**. All models are trained on 4 Nvidia RTX A5000 GPUs with a total batch size of 128. We report the mean $\pm$ standard deviation of 5 runs with 5 random seeds for all our experiments. All our source code and data will be publicly available upon publication.

5 Experiments

5.1 Unsupervised Foreground Extraction

Task Setting, Metrics, Baselines and Datasets. In the task, all models learn to output binary masks separating foregrounds and backgrounds in an unsupervised manner. See **Supp. Material** for implementations of foreground and background mask predictions during the inference. We evaluate the quality of the predicted foreground and background masks with mean Intersection over Union (mIoU) [20] and Dice [20]. The mIoU measures the overlap between predicted masks and the ground truth. Dice is similar to mIoU but replaces the union of two sets with the sum of the number of their elements and doubles the intersection. We used publicly available implementations of Slot-Attention [27] and SLATE [31] from [20] and replicated DINOSAUR [30] by ourselves. Following the work of BO-QSA [20], we include Stanford Dogs [21], Stanford Cars [23], CUB 200 Birds [35], and Flowers [28] as benchmark datasets.

Results and Analysis. The results in mIoU and Dice are presented in Table 1. Our method achieved the best performance on Birds, Dogs, and Cars datasets and performs competitively well as BO-QSA on Flowers dataset. Slot-Attention, SLATE, and DINOSAUR use the pixel-level,

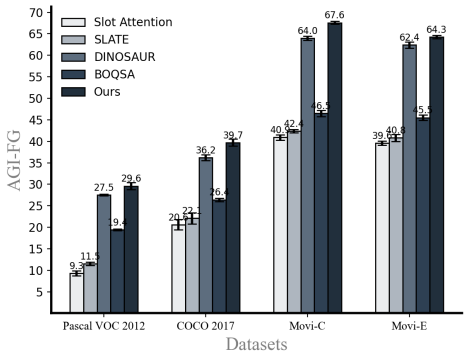


Figure 4: **Performance of Object Discovery on Pascal VOC 2012 [13] (red), COCO 2017 [25] (blue), Movi-C [15] (yellow) and Movi-E [15] (cyan).** We report the performance in foreground adjusted rand index (ARI-FG). The higher ARI-FG, the better. We compare our method (cyclic walks) with Slot-Attention [27], SLATE [31], DINOSAUR [30] and BO-QSA [20].

Model	Pascal VOC 2012	COCO Stuff-27	COCO Stuff-3
ACSeg [24]	**47.1**	16.4	-
MaskDistill [14]	42.0	-	-
PiCIE + H [10]	-	14.4	-
STEGO [16]	-	**26.8**	-
SAN [38]	-	-	80.3
InfoSeg [17]	-	-	73.8
Cyclic walks (ours)	43.3 $\pm$ 1.6	22.5 $\pm$ 1.1	**82.4** $\pm$ 0.7

Table 2: **Results of Unsupervised Semantic Segmentation in IoU on Pascal VOC 2012, COCO-Stuff-27 [6] and COCO-Stuff-3 [6].** ($\pm$) standard deviations of IoU over 5 runs are also reported. The '-' indicates that corresponding results are not provided in the original papers and the source codes are unavailable. Best is in bold.

feature-level, and token-level reconstruction losses from object-centric representations as supervisory signals. The inferior performance of Slot-Attention over SLATE and DINOSAUR implies that pixel-level reconstructions focus excessively on unnecessary details, impairing the object-centric learning performance. We also note that DINOSAUR outperforms Slot-Attention and SLATE by a significant margin. This aligns with the previous findings that freezing pre-trained feature extractors facilitates object-centric learning. In addition to reconstructions, BO-QSA imposes extra constraints on learnable slot queries with bi-level optimizations, which leads to better performances. This emphasizes the necessity of searching for better supervisory signals for regularizing object-centric learning beyond reconstructions. Indeed, even without any forms of reconstruction, we introduce cyclic walks between parts and whole as an effective training loss for object-centric learning, and our method yields the best performance among all comparative methods. The result analyses hold true in subsequent tasks (Section 5.2 and 5.3).

We also visualize the attention masks predicted by our slot bases in Figure 3(a). We observed that the model trained with our cyclic walks outputs reasonable predictions. However, we also noticed several inconsistent predictions. For example, our predicted masks in the Flowers dataset (Row 3 and 4) group the pedals and the sepals of a flower altogether, because these two parts often co-occur in a scene. Moreover, we also spotted several wrongly annotated ground truth masks. For example, in Row 4, the foreground mask mistakenly incorporates the sky as part of the ground truth. It is also interesting to see that the foreground object belonging to the same object class is not always identified by a fixed slot basis (compare Row 1 and 2 in Birds dataset). During the inference, we randomly sample parameters from learnable mean μ and variance σ as the slot bases (Section 3.1), which could cause the reverse of foreground and background masks. Applying the same set of parameters for slot bases across the experiments avoids such issues.

5.2 Unsupervised Object Discovery

Task Setting, Metrics, Baselines and Datasets. The task aims to segment the image with a set of binary masks, each of which covers similar parts of the image, (aka "object masks"). However, different from image segmentation, each of the masks does not have to be assigned specific class labels. We follow recent works and evaluate all models [27, 31, 20, 30] with ARI on foreground objects (ARI-FG). ARI reflects the proportion of sampled pixel pairs on an image, correctly classified into the same class or different classes. All baseline implementations are based on the open-source codebase [26]. Same as Section 5.1, we use the $M_{x,\hat{s}}$ as the object masks. We benchmark all methods on the common datasets Pascal VOC 2012 [13], COCO 2017 [25], Movi-C [15] and Movi-E[15].

Dataset	Full Model	Random Walk Direction		Temperature of Random Walks			Number of slots			Similarity Threshold			Feature Extractor		
		P-W-P	W-P-W	0.07	0.3	1	5	6	7	-inf	0.3	1	Moco-v3 [9]	MAE [18]	MSN [3]
Pascal VOC 2012	**29.6**	27.1	28.4	23.5	27.3	21.9	27.7	26.2	24.3	22.5	26.5	26.2	29.3	26.8	29.0
		P-W-P	W-P-W	0.07	0.3	1	10	12	13	-inf	0.3	1	Moco-v3 [9]	MAE [18]	MSN [3]
COCO 2017	**39.7**	34.8	36.4	35.5	36.7	29.6	37.9	35.7	35.3	29.8	32.3	31.8	38.4	34.5	38.2

Table 3: **Ablation Studies and Method Analysis on Pascal VOC 2012 and COCO 2017 in terms of ARI-FG.** Full model refers to our default method introduced in Section 4. See **Supp. Material** for the empirically determined hyper-parameter details of our full model. We vary one factor at a time and study its effect on ARI-FG performances in the Pascal VOC 2012 (Row 3) and COCO 2017 (Row 5). The best is our full model highlighted in bold. See Section 5.4 for details.

Results and Analysis The results in ARI-FG on the four datasets are shown in Figure 4. The unsupervised object discovery task is harder than the unsupervised foreground extraction task (Section 5.1) due to the myriad scene complexity and object class diversity. Our method still consistently outperforms all SOTA methods and beats the second-best method DINOSAUR by a large margin of 2 - 4% over all four datasets. Different from all other methods attached with decoders for the image or feature reconstructions, our model trained with cyclic walks learns better object-centric representations. We visualized some example object mask predictions in Figure 3(b). Our method can clearly distinguish semantic areas in complex scenes. For example, trains, trees, lands, and sky are segmented in Row 1 in an unsupervised manner, although only the foreground train masks are provided in the ground truth annotations in PASCAL VOC. However, we also notice several failure cases. When the number of semantic classes in the image is less than the number of slots, our method segments the edges of semantic classes (see **Supp. Material** for detailed analysis).

5.3 Unsupervised Semantic Segmentation

Task Setting, Metrics, Baselines and Datasets. In the task, each pixel of an image has to be classified into one of the pre-defined object categories. See **Supp. Material** for implementations of obtaining the category labels for each predicted mask during inference. We report the intersection over union (IoU) between the predicted masks and the ground truth over all categories. We include ACSeg [24], MaskDistill [14], PiCIE [10], STEGO [16], SAN [38] and InfoSeg [17] for comparison. All the results are directly obtained from their original papers. We evaluate all competitive methods on Pascal VOC 2012, COCO Stuff-27 [6], and COCO Stuff-3 [6]. COCO Stuff-27 and COCO Stuff-3 contain 27 and 3 supercategories respectively.

Results and Analysis We report the results in terms of IoU in Table 2. Our method has achieved the best performance on COCO-Stuff-3 and the second best on Pascal VOC 2012 and COCO-Stuff-27. This highlights that our cyclic walks in the part-whole hierarchy on the same images act as effective supervision signals and distill image pixels into diverse object-centric representations rich in semantic information.

5.4 Ablation Study and Method Analysis

We conduct all the ablations and network analysis illustrated below on Pascal VOC 2012 [13] and COCO 2017 [25] datasets and report ARI-FG scores for all the model variations (Table 3).

Ablation on Random Walk Directions. To explore the effects of random walk directions, we use either P-W-P or W-P-W walks to train the network separately and observe the changes (Table 3, Col.3-4). We found that either direction of random walks can make the network converge. The W-P-W walks perform slightly better than the P-W-P walks, but neither of these walks surpasses random walks in both directions. This aligns with the design motivation of our method (Section 4).

Analysis on Temperature of Random Walks. We titrate the temperate in Equation 4 from 0.07 to 1 and observe a non-monotonic performance trend versus temperature choices (Col.5-7). On one hand, the lower the temperature, the more concentrated the attention distribution over all the slots and the faster the network can converge due to the steeper gradients. On the other hand, the attention distribution becomes too sharp with much lower temperatures, resulting in optimization instability.

Analysis on Number of Slots. The number of slots imposes constraints on the diversity of objects captured in the scene. We trained models with various numbers of slots (Col.8-10). More slots do not always lead to better performances. With more slots, it is possible that all slots compete with one another for each image patch and the extra slots only capture redundant information, hurting the object discovery performance. in all the experiments.

Analysis on Similarity Threshold in P-W-P Walks. During P-W-P cyclic walks, we introduce similarity threshold t (Equation 6). From Col.11-13, we can see that with the increase of threshold γ from $-inf$ to 0.7, the P-W-P random walks become more selective, enforcing the slot bases to learn more discriminative features; hence, better performance in object discovery tasks. However, when the threshold approaches 1, the ability to walk to neighboring image patches with high semantic similarity to the starting patch is impaired, leading to overfitting of slot bases.

Analysis on Different Feature Extractors. We replace pre-trained DINO transformer (Section 4.1) with MOCO-V3 [9], MAE [18], and MSN [3]. Together with DINO, these feature extractors are state-of-the-art unsupervised representation learning frameworks. From Col.14-15, we observe that various backbones with our method consistently achieve high ARI-FG scores over both datasets. This suggests that our method is agnostic to backbone variations and it can be readily adapted to any general SOTA unsupervised learning frameworks.

5.5 Efficiency Analysis in Model Sizes, Training Speed, and GPU Usages

Method	Parameters (10^3)	Training speed (image / sec)	GPU usage (M)
Slot-Attention	3144	114	4371
SLATE	14035	27	16327
DINOSAUR	11678	124	3443
BO-QSA	14223	25	16949
Cyclic walks (ours)	**1927**	**208**	**2331**

Table 4: **Method Comparison in the number of parameters, training speed, and GPU memory usage during training.** From top to bottom, we include Slot-Attention, SLATE, DINOSUAR and BO-QSA. The best is in bold.

We report the number of network parameters, training speed, and GPU usage required during training in Table 4. All these experiments are run with the same hardware specifications and method configurations: (1) one single RTX-A5000 GPU; and (2) input image size of 224 $\times$ 224, batch size of 8, and 4 slots. In comparison with all transformer decoder-based methods (SLATE, DINOSAUR and BO-QSA), our method uses the fewest parameters and yet, achieves the best performance in all the tasks (Sections 5.1, 5.2, 5.3). Compared to other decoder-based methods, the Slot-Attention requires fewer number of parameters by sampling from a mixture of slot bases for image reconstruction. However, our method even requires twice fewer parameters than Slot-Attention (Col. 1). Moreover, although the transformer-based networks are slower in training compared with the CNN-based networks, DINOSAUR and our method are much faster than SLATE and BO-QSA due to freezing the feature extractor (Col. 2). By additionally benefiting from cyclic walks and bypassing decoders for feature reconstruction, our method runs twice as much faster as DINOSAUR. With the same reasoning, the networks trained with our cyclic walks only require half of the GPU memory usage of DINOSAUR without sacrificing the performance in all the three tasks (Col. 3).

6 Discussion

We propose cyclic walks in the part-whole hierarchy for unsupervised object-centric representation learning. In the slot-attention module, a finite set of slot bases compete to bind with certain regions of the image and distill into object-centric representations. Both P-W-P and W-P-W cyclic walks serve as implicit supervision signals for training the networks to learn compositional entities of the scene. Our experiments demonstrate that our cyclic walks outperform all competitive baselines over seven datasets in three unsupervised tasks while being memory-efficient and computation-efficient during training. As our method builds a connection between object-centric learning and unsupervised clustering on static images, it has the potential in contributing to many practical applications, such as scene understanding, reasoning, and explainable AIs. So far, our method has been developed on a frozen unsupervised feature extractor. In the future, hierarchical contrastive walks can be explored in any feed-forward architectures, where the models can simultaneously learn both pixel-level and object-centric representations incrementally over multiple layers.

References

[1] A. Aflalo, S. Bagon, T. Kashti, and Y. C. Eldar. Deepcut: Unsupervised segmentation using graph neural networks clustering. *CoRR*, abs/2212.05853, 2022.

[2] R. Assouel, P. Taslakian, D. Vazquez, P. Rodriguez, and Y. Bengio. Object-centric compositional imagination for visual abstract reasoning. In *Workshop at the International Conference on Learning Representations (ICLR)*, 2022.

[3] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas. Masked siamese networks for label-efficient learning. In S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXI*, volume 13691 of *Lecture Notes in Computer Science*, pages 456–473. Springer, 2022.

[4] Z. Bao, P. Tokmakov, A. Jabri, Y. Wang, A. Gaidon, and M. Hebert. Discovering objects that can move. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 11779–11788. IEEE, 2022.

[5] Z. Bian, A. Jabri, A. A. Efros, and A. Owens. Learning pixel trajectories with multiscale contrastive random walks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 6498–6509. IEEE, 2022.

[6] H. Caesar, J. R. R. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. *CoRR*, abs/1612.03716, 2016.

[7] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020.

[8] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE, 2021.

[9] X. Chen, S. Xie, and K. He. An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9620–9629. IEEE, 2021.

[10] J. H. Cho, U. Mall, K. Bala, and B. Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 16794–16804. Computer Vision Foundation / IEEE, 2021.

[11] S. Choudhury, L. Karazija, I. Laina, A. Vedaldi, and C. Rupprecht. Guess what moves: Unsupervised video and image segmentation by anticipating motion. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*, page 554. BMVA Press, 2022.

[12] G. F. Elsayed, A. Mahendran, S. van Steenkiste, K. Greff, M. C. Mozer, and T. Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. *CoRR*, abs/2206.07764, 2022.

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

[14] W. V. Gansbeke, S. Vandenhende, and L. V. Gool. Discovering object masks with transformers for unsupervised semantic segmentation. *CoRR*, abs/2206.06363, 2022.

[15] K. Greff, F. Belletti, L. Beyer, C. Doersch, Y. Du, D. Duckworth, D. J. Fleet, D. Gnanapragasam, F. Golemo, C. Herrmann, T. Kipf, A. Kundu, D. Lagun, I. H. Laradji, H. D. Liu, H. Meyer, Y. Miao, D. Nowrouzezahrai, A. C. Öztireli, E. Pot, N. Radwan, D. Rebain, S. Sabour, M. S. M. Sajjadi, M. Sela, V. Sitzmann, A. Stone, D. Sun, S. Vora, Z. Wang, T. Wu, K. M. Yi, F. Zhong, and A. Tagliasacchi. Kubric: A scalable dataset generator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 3739–3751. IEEE, 2022.

[16] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[17] R. Harb and P. Knöbelreiter. Infoseg: Unsupervised semantic image segmentation with mutual information maximization. In C. Bauckhage, J. Gall, and A. G. Schwing, editors, *Pattern Recognition - 43rd DAGM German Conference, DAGM GCPR 2021, Bonn, Germany, September 28 - October 1, 2021, Proceedings*, volume 13024 of *Lecture Notes in Computer Science*, pages 18–32. Springer, 2021.

[18] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE, 2022.

[19] A. Jabri, A. Owens, and A. A. Efros. Space-time correspondence as a contrastive random walk. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[20] B. Jia, Y. Liu, and S. Huang. Unsupervised object-centric learning with bi-level optimized query slot attention. *CoRR*, abs/2210.08990, 2022.

[21] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[22] T. Kipf, G. F. Elsayed, A. Mahendran, A. Stone, S. Sabour, G. Heigold, R. Jonschkowski, A. Dosovitskiy, and K. Greff. Conditional object-centric learning from video. *CoRR*, abs/2111.12594, 2021.

[23] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[24] K. Li, Z. Wang, Z. Cheng, R. Yu, Y. Zhao, G. Song, C. Liu, L. Yuan, and J. Chen. Acseg: Adaptive conceptualization for unsupervised semantic segmentation, 2022.

[25] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[26] Y. Liu. https://bo-qsa.github.io/, 2022.

[27] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[28] M. Nilsback and A. Zisserman. Delving deeper into the whorl of flower segmentation. *Image Vis. Comput.*, 28(6):1049–1062, 2010.

[29] J. T. Rolfe. Discrete variational autoencoders. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[30] M. Seitzer, M. Horn, A. Zadaianchuk, D. Zietlow, T. Xiao, C. Simon-Gabriel, T. He, Z. Zhang, B. Schölkopf, T. Brox, and F. Locatello. Bridging the gap to real-world object-centric learning. *CoRR*, abs/2209.14860, 2022.

[31] G. Singh, F. Deng, and S. Ahn. Illiterate DALL-E learns to compose. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[32] G. Singh, Y. Wu, and S. Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. *CoRR*, abs/2205.14065, 2022.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[34] X. Wang, Z. Yu, S. De Mello, J. Kautz, A. Anandkumar, C. Shen, and J. M. Alvarez. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14176–14186, 2022.

[35] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. 09 2010.

[36] X. Wen, B. Zhao, A. Zheng, X. Zhang, and X. Qi. Self-supervised visual representation learning with semantic grouping. *arXiv preprint arXiv:2205.15288*, 2022.

[37] J. Yang, J. Mao, J. Wu, D. Parikh, D. D. Cox, J. B. Tenenbaum, and C. Gan. Object-centric diagnosis of visual reasoning. *CoRR*, abs/2012.11587, 2020.

[38] D. Zhang, C. Li, H. Li, W. Huang, L. Huang, and J. Zhang. Rethinking alignment and uniformity in unsupervised image semantic segmentation. *CoRR*, abs/2211.14513, 2022.