Adaptive Slot Attention: Object Discovery with Dynamic Slot Number

Ke Fan[1], Zechen Bai[2], Tianjun Xiao[3], Tong He[3], Max Horn[4],
Yanwei Fu[1,†], Francesco Locatello[5], Zheng Zhang[3]
[1]Fudan University [2]National University of Singapore [3]Amazon Web Services
[4]GSK.ai [5]Institute of Science and Technology Austria

kfan21@m.fudan.edu.cn, zechenbai@outlook.com, yanweifu@fudan.edu.cn
max.x.horn@gsk.com, Francesco.Locatello@ista.ac.at, {tianjux, htong, zhaz}@amazon.com

Abstract

Object-centric learning (OCL) extracts the representation of objects with slots, offering an exceptional blend of flexibility and interpretability for abstracting low-level perceptual features. A widely adopted method within OCL is slot attention, which utilizes attention mechanisms to iteratively refine slot representations. However, a major drawback of most object-centric models, including slot attention, is their reliance on predefining the number of slots. This not only necessitates prior knowledge of the dataset but also overlooks the inherent variability in the number of objects present in each instance. To overcome this fundamental limitation, we present a novel complexity-aware object auto-encoder framework. Within this framework, we introduce an adaptive slot attention (AdaSlot) mechanism that dynamically determines the optimal number of slots based on the content of the data. This is achieved by proposing a discrete slot sampling module that is responsible for selecting an appropriate number of slots from a candidate list. Furthermore, we introduce a masked slot decoder that suppresses unselected slots during the decoding process. Our framework, tested extensively on object discovery tasks with various datasets, shows performance matching or exceeding top fixed-slot models. Moreover, our analysis substantiates that our method exhibits the capability to dynamically adapt the slot number according to each instance's complexity, offering the potential for further exploration in slot attention research. Project will be available at https://kfan21.github.io/AdaSlot/

1. Introduction

Object-centric learning marks a departure from conventional deep learning paradigms, focusing on the extraction of structured scene representations rather than relying solely on global features. These structured representations encompass crucial attributes such as spatial information, color, texture, shape, and size, effectively delineating various regions within a scene. These regions, characterized by distinct yet cohesive properties, can be likened to objects in the human sense. These object-centric representations, often referred to as slots, are organized within a set structure that partitions the global scene information.

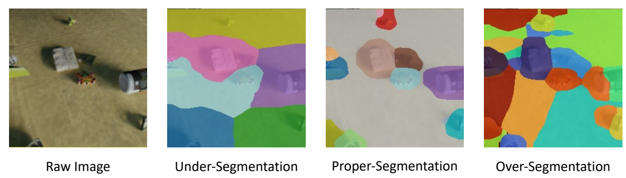

Figure 1. Illustration of raw image and three kinds of segmentation masks **under different slot numbers**. Pixels colored the same are grouped as the slot. The slot number is very important.

Traditionally, object-centric learning adopts unsupervised methods with reconstruction as the primary training objective. This process clusters distributed scene representations into object-centric features, with each cluster associated with a specific slot. Decoding these slots independently or in an auto-regressive manner yields meaningful segmentation masks. This inherent characteristic of object-centric learning has paved the way for its application across diverse tasks, including unsupervised object discovery and localization [13, 15, 24], segmentation [30] and manipulation [27]. And it can also be generalized to weakly-supervised/supervised cases [9, 14, 19]. Among these algorithms, Slot Attention [24] emerges as the most prominent and widely recognized method in the field.

However, a significant challenge within the realm of slot attention is its reliance on a predefined number of slots, which can prove problematic. On one hand, accurately determining the number of objects in a dataset can be challenging, especially when annotations are absent. On the other hand, datasets often exhibit varying object counts,

Max and Francesco did the work at Amazon; † corresponding authors.

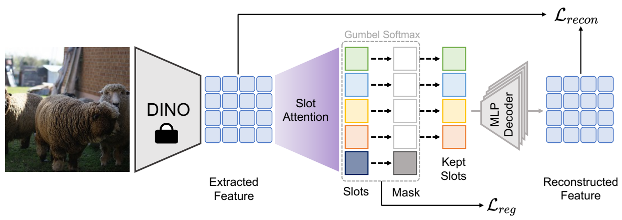

Figure 2. Illustration of our pipeline.

rendering a fixed, predefined number impractical. Incorrectly specifying the number of slots can substantially impact the results, as illustrated in Fig. 1, where an inadequate slot count leads to under-segmentation, while an excessive count results in over-segmentation.

To address this challenge, we present an approach that adaptively determines the number of slots for each instance based on its inherent complexity. Our goal is to allocate a larger slot count for instances with more objects while a smaller number for fewer objects. To achieve this, we propose a novel complexity-aware object auto-encoder framework. Within this framework, we initially generate a relatively large number of slots, denoted as K_{max}, and dynamically select a subset of slots according to each instance for the reconstruction process. Additionally, our framework incorporates a slot sparsity regularization term into the training objective, explicitly considering the complexity of each instance. This regularization term ensures a balance between reconstruction quality and the utilization of an appropriate number of slots.

Our framework encompasses several key strategies. Firstly, we leverage a lightweight slot selection module to acquire a sampling strategy that keeps the most informative slots and discards redundant ones. However, simply neglecting the dropped slot will not propagate the gradient. To deal with this, we employ Gumbel-Softmax [17] to achieve end-to-end training. Furthermore, simply sampling an element from the power set of slots will lead to exponentially many choices and low computational efficiency. To address this issue, we break the selection into K binary selection with the mean-field formulation [3] to overcome this problem. Finally, we introduce a masked slot decoder that adeptly removes information associated with the dropped slots. The whole pipeline is displayed in Fig. 2.

We summarize our contributions here: 1) **Novel Framework**: We propose a novel complexity-aware object autoencoder framework that dynamically determines the number of slots, addressing the limitation of fixed slot counts in object-centric learning. 2)**Efficient Slot Selection**: Our framework incorporates an efficient and differentiable slot selection module, enabling the identification of informative slots while discarding redundant ones before reconstruction. 3)**Effective Slot Decoding**: We present a masked slot decoder that efficiently removes information associated with unused slots. 4)**Promising Results**: Through extensive empirical experiments, we demonstrate the superiority of our approach, achieving competitive or superior results compared to models relying on fixed slot counts. Importantly, our method excels in instance-level slot count selection, showcasing its practical efficacy in various applications.

2. Related Work

Object-Centric Learning. Object-centric learning fundamentally revolves around the idea that natural scenes can be effectively represented as compositions of distinct objects. Current methodologies in this field mainly fall into two categories: 1) **Spatial-Attention Models** are exemplified by models like AIR [12], SQAIR [21], and SPAIR [6]. These approaches infer bounding boxes for objects, providing explicit information about an object's position and size. Typically, such methods employ a discrete latent variable z_{pres} to determine the presence of an object and infer the number of objects. However, these box-based priors often lack the flexibility needed to accurately segment objects with widely varying scales and shapes. 2) **Scene-Mixture Models** explain a visual scene by a finite mixture of component images. Methods like MONET [4], IODINE [15], and GENESIS [10] operate within the variational inference framework. They involve multiple encoding and decoding steps to process an image. In contrast, Slot Attention [24] takes a unique approach by replacing this procedure with a single encoding step using iterated attention.

Expanding on slot attention, various adaptations like SAVi [19] for video data, STEVE[28] for compositional video, and SLATE [27] for image generation have been developed. While effective on synthetic datasets, their real-world performance can be limited. DINOSAUR [26] addresses this by reconstructing deep features instead of pixels, showing enhanced results on both synthetic and real-world datasets, an approach we adopt in our work.

A common limitation among existing methods in this line is the requirement to predefine the number of slots, often treated as a dataset-dependent hyperparameter. In this context, GENESIS-V2 [11] introduces a novel approach by clustering pixel embeddings using a stochastic stick-breaking process, allowing for the output of a variable number of objects, serving as a valuable baseline method.

Differentiable Subset Sampling. Several studies have pursued the goal of achieving differentiable subset selection. Notably, Gumbel-Softmax [17, 25] introduces a continuous relaxation of the Gumbel-Max trick, enabling the selection of the top-1 element. Building upon this foundation, Gumbel Top-k [20] extends the approach to generalize top-k sampling. Another innovative approach, proposed by [7, 29], approximates top-k sampling by harnessing the Sinkhorn algorithm from Optimal Transport. Furthermore, [1, 5] employs the perturbed maximum method to achieve differentiable selection.

However, a common focus of these works lies in scenarios where the subset size is fixed at k, constraining their adaptability for slot number selection. In contrast, our method employs the common mean-field formulation to transform the subset selection problem, which does not rely on a predefined number, into a series of top-1 selections that can be efficiently resolved using Gumbel-Softmax.

3. Method

Preliminary. Slot Attention [24] stands out as one of the most prominent object-centric methods, relying on a competitive attention mechanism. In the pipeline, Slot Attention initially extracts image features with an encoder $F = f_{enc}(x) \in \mathbb{R}^{H' \times W' \times D}$, where $x \in \mathbb{R}^{H \times W \times C}$ represents the image. Rather than directly decoding F into x, the *Slot Attention Bottleneck* g_{slot} further extracts K slots, denoted as $S_1, \cdots, S_K = g_{slot}(F)$.

The slot attention pipeline proceeds to reconstruct images from these slots using a weighted-average decoder. Each slot S_i is individually decoded through an object decoder g_{object} and a mask decoder g_{mask}, subsequently integrated through weighted averaging across the slots.

$$(x_i, \alpha_i) = \left(g_{object}\left(S_i\right), g_{mask}\left(S_i\right)\right), \tag{1}$$

$$\hat{x} = \sum_{k=1}^{K} m_i \odot x_i, \quad m_i = \frac{\exp \alpha_i}{\sum_{l=1}^{K} \exp \alpha_i}, \tag{2}$$

where $x_i \in \mathbb{R}^{H \times W \times C}$ is the object reconstruction while $\alpha_i \in \mathbb{R}^{H \times W}$ is the unnormalized alpha mask. We minimize the mean squared error between x and $\hat{x}$ as $\mathcal{L}_{recon}\left(\hat{x}, x\right) = \|\hat{x} - x\|_2^2$. Here we utilize a fixed K model as our base model. Moreover, we reconstruct the RGB pixels for toy datasets, while following DINOSAUR to reconstruct features extracted by self-supervised backbones on more complicated datasets.

3.1. Complexity-aware Object Auto-Encoder

In slot attention model, predefining the slot number K profoundly affects object segmentation quality. To address this issue, we propose a complexity-aware object auto-encoder framework.

Following clustering number selection [2], we set an upper bound for the slot number as K_{max}. This represents the maximum number of objects an image may contain in the dataset. During the decoding phase, instead of decoding from all slots, our objective is to decode from the most *informative* slots. To achieve this, we learn a sampling method π for each instance $\mathbf{x}$. The probability $\pi(z_1, \cdots, z_{K_{max}})$ determines whether to keep or drop each slot $S_{1 \sim K_{max}}$, with $z_i = 0$ indicating the slot S_i should be dropped, and $z_i = 1$ indicating it should be kept during reconstruction. We introduce a masked slot decoder $\hat{x} = f_{dec}(S, Z)$ that effectively suppresses the information of the dropped slots based on Z.

To further control the slot number we retain, we incorporate a complexity-aware regularization term $\mathcal{L}_{reg}(\pi)$. This regularization term helps ensure the appropriate number of slots are retained based on the complexity of instances. The training objective can be formulated as:

$$\begin{aligned} \min \quad & \mathbb{E}_Z\, \mathcal{L}_{recon}\left(\hat{x}, x\right) + \lambda \cdot \mathcal{L}_{reg}\left(\pi\right) \\ \text{where} \quad & S_1, \cdots, S_{K_{max}} = g_{slot}\left(f_{enc}(x)\right) \\ & Z \sim \pi\left(z\right),\ \hat{x} = f_{dec}\left(S, Z\right) \end{aligned} \tag{3}$$

Naturally, without any regularization, the model tends to greedily keep all the slots, as more slots generally lead to better reconstruction quality. In contrast, our complexity regularization, as expressed in Eq. 3, compels the model to achieve the reconstruction objective while utilizing as few slots as possible. The parameter λ controls the strength of this regularization.

A natural choice of regularization is the expectation of keeping slots:

$$\mathcal{L}_{reg} = \mathbb{E}\left[\sum_{i=1}^{K} Z_i\right] = \sum_{i=1}^{K} \mathbb{E}\left[Z_i\right]. \tag{4}$$

The smaller expectation, the fewer slot left after selection.

Within this framework, we propose our *adaptive slot attention* (**AdaSlot**) and dealing with two challenges. The first is how to sample from a discrete distribution while keeping the module differentiable 3.2. The second is how to design mask slot decoder to suppress the dropped slots 3.3.

3.2. Mean-Field Sampling With Gumbel Softmax

Given K slots S, there are 2^K possible subsets $S_{sub} \subseteq S$. By mapping each subset to a number between 1 and 2^K, we transform the task of selecting a subset into a simpler top-1 choice problem, accounting for the interrelations of slots. Yet, as the number of slots increases, the exponentially growing search space complicates memory management and model optimization, often trapping the neural network in local minima. To address this, we use the mean-field formulation in variational inference [3], factoring π into a product of independent distributions for each slot:

$$\pi(z_1, \cdots, z_K) = \pi_1(z_1) \cdots \pi_K(z_K). \tag{5}$$

Therefore, the problem of selecting from 2^K space is reduced to a K binary selection problem. For each S_i, we decide drop or keep the slot individually. This mean-field slot selection approach is computational and sampling efficient. Although the relation among slots is ignored in this step, we postulate this relation can be implicitly modelled by the competition mechanism in slot attention.

To be specific, we denote $S \in \mathbb{R}^{K\times D}$. A light weight neural network $h_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^2$ is used to predict the keep/drop probability of each slot individually:

$$\pi = \mathrm{Softmax}(h_\theta(S)) \in \mathbb{R}^{K\times 2}, \tag{6}$$

where $\pi_{i,0}$ denote the soft probability to drop the i-th slot, while $\pi_{i,1}$ denote the soft probability to keep the i-th slot. By applying the Gumbel-Softmax with Straight-Through Estimation [17] on the probability dimension and take the last column, we get the hard decision slot mask Z:

$$Z = \mathrm{GumbelSoftmax}(\pi)_{:,1}. \tag{7}$$

Here, the colon (:) denotes all rows, and 1 denotes the specific column we want to extract. Since Gumbel Softmax generate onehot vector, take the column we get K-dimensional zero-one mask $Z = (Z_1, \cdots, Z_k) \in \{0,1\}^K$.

3.3. Masked Slot Decoder

As mentioned in [26], the Transformer decoder is biased towards grouping semantically related instances together, while the mixture decoder is able to separate instances better. The behavior of the mixture-decoder makes it a better choice for exploring dynamic slots since we expect the model to distinguish instances rather than semantics. In this paper, we focus on mixture decoder. With the slots representations S and the keep decision vector Z, we introduce several possible design choices of suppressing less important slots based on Z.

Zero slot strategy directly multiply the zero-one keep decision vector Z with the slots S:

$$\tilde{S}_i = Z_i S_i, \tag{8}$$

which shrinks dropped slots to zero and keeps the others.

Learnable slot strategy employs a shared learnable embedding S_{mask} as the prototype of the dropped slot. The intuition is that a learnable dropped slot would offer the model more flexibility and stabilize training, and complement the information loss caused by dropping slots. This is achieved as:

$$\tilde{S}_i = Z_i S_i + (1 - Z_i) S_{mask}. \tag{9}$$

We empirically found that both the two strategies would hurt the reconstruction quality as well as the object grouping. The root cause is that when computing the alpha mask, the zero/learnable-shrinked slots are still decoded to non-zero masks which matter at the softmax operation as follows:

$$m_i = \frac{\exp \alpha_i(\tilde{S}_i)}{\sum_{l=1}^{K} \exp \alpha_l(\tilde{S}_l)}. \tag{10}$$

Zero mask strategy Instead of manipulating the slots representations, we propose to shrink the corresponding alpha masks to zero:

$$\tilde{m}_i = \frac{Z_i m_i}{\sum_{l=1}^{K} Z_l m_l + \delta}, \quad m_i = \frac{\exp \alpha_i(S_i)}{\sum_{l=1}^{K} \exp \alpha_l(S_l)}, \tag{11}$$

where δ is a small positive value for computation stability. It is worth noting that neglecting δ, Eq. 11 is equivalent to omitting the slot in the mixture decoder, except that Gumbel-Softmax is applied to ensure differentiability. The key difference is that this strategy manipulates the alpha mask directly, fully removes the information of dropped slot while the other two approaches could not.

4. Experiments

Datasets. To evaluate its performance, we utilize a toy dataset CLEVR10 [18] and two complicated synthetic MOVi-C/E [16] with high-quality scanned objects in realistic backgrounds. MOVi-C has up to 10 objects, while MOVi-E includes at most 23 objects. We treat MOVi datasets as image datasets. Additionally, we use MS COCO 2017 dataset [23] as a real-world dataset, which introduces increased complexity. Noting that we utilize COCO's instance mask instead of semantic mask.

Metrics We use three kinds of methods for evaluation. The *pair-counting metric* utilizes a pair confusion matrix to compute precision, recall, F_1 score, and Adjusted Rand Index. In the *matching-based metric*, we utilize three methods: mBO, CorLoc, and Purity. Purity assigns clusters to the most frequent class, and compute the accuracy. mBO calculates the mean intersection-over-union for matched predicted and ground truth masks, while CorLoc measures the fraction of images with at least one object correctly localized. The *information-theoretic metric* employs Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI). All metrics, except mBO and CorLoc, are computed on the foreground objects. *We use ARI to denote FG-ARI for simplicity.*

Implementation Details We employ DINO ViT/B-16 as a frozen feature extractor. We set values of K_{max} to 24 for MOVi-E, 11 for MOVi-C, and 33 for COCO. A two-layer MLP is used for each slot to determine the keeping probability. Feature reconstruction is performed using MLP mixture decoder as DINOSAUR. We use Adam optimizer, learning rate $4e-4$, 10k step linear warmup, and exponential learning rate decay. We train our model 500k steps for main experiments and 200k steps for ablation. Results are averaged over 3 random seeds. More details are in Appendix. We set λ to 0.1 for MOVi-E/C and 0.5 for COCO.

4.1. Main Results on Each Dataset

Toy Dataset. We compare a fixed 11-slot model ($K_{max} = 11$) on the toy dataset CLEVR10 in Fig. 4, with pixel re-

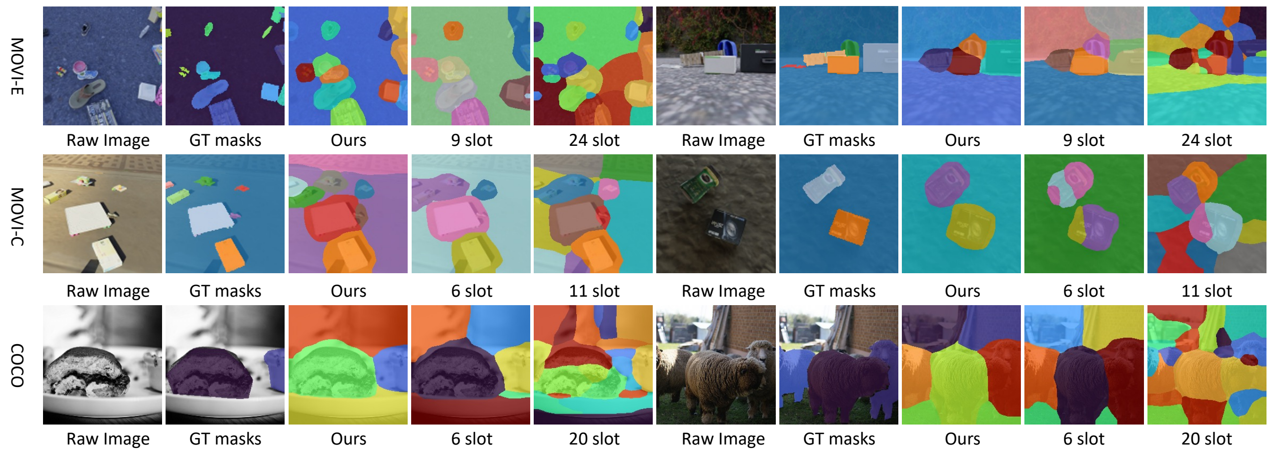

Figure 3. **Visualization of instance-level adaptive slot number selection.** We compare our models and the fixed-slot DINOSAUR on three datasets. For each dataset, we select two examples and compare our model with a small slot number and a large slot number.

Table 1. Results on MOVi-C. (P., R. for Precision, and Recall).

		Pair-Counting				Matching			Information	
Model	K	ARI	P.	R.	F1	mBO	CorLoc	Purity	AMI	NMI
GENESIS-V2	6	39.65	71.02	52.34	58.23	11.58	1.29	59.83	52.56	52.70
	11	26.63	65.36	37.61	45.72	14.44	6.97	49.58	40.16	40.42
DINOSAUR	3	42.98	61.42	79.06	66.87	10.75	4.94	67.88	49.53	49.61
	6	*73.23*	83.06	*84.98*	*82.56*	33.85	*73.86*	*83.19*	*76.44*	*76.51*
	9	69.11	*87.50*	75.53	79.08	*35.00*	71.26	79.77	75.43	75.50
	11	66.42	**88.42**	71.31	76.73	34.72	68.69	77.43	74.31	74.39
AdaSlot (Ours)		**75.59**	84.64	**86.67**	**84.25**	**35.64**	**76.80**	**85.21**	**78.54**	**78.60**

construction. The ordinary 11-slot model lacks knowledge of the object number and tends to allocate slots for segmenting the background, resulting in slot duplication. In contrast, AdaSlot accurately groups pixels according to the actual number of ground truth objects. Surprisingly, our AdaSlot exhibits the ability to determine the object count and resolve slot duplication on the toy dataset. Please refer to the appendix for detailed results.

Results on MOVi-C/E. Compared to our model, vanilla slot attention in DINOSAUR uses a pre-defined fixed slot number. The selection of slot numbers is subject to the dataset statistics. Note that for data in the wild, we don't have access to the ground-truth statistics. Here, we access the number only for comparison. We established baselines for the MOVi-E dataset with an average of 12 objects (max 23) using small (3, 6, 9), medium (13), and large (18, 21, 24) slot numbers. For the MOVi-C dataset with a maximum of 10 objects, we used slot numbers 3, 6, 9, and 11. Besides, GENESIS-V2 is compared. The results are displayed in Tab. 1, Tab. 2 and Fig. 3.

For *Object Grouping*, our algorithm demonstrates its benefits through three different kinds of metrics. Our method outperforms GENESIS-V2 by a large margin. When compared to the fixed-slot DINOSAUR, our complexity-aware model achieves the highest ARI and F_1 score, indicating that it can effectively group sample pairs within the same cluster as defined by the ground truth. In terms of Purity, AdaSlot yields the highest results, showing the greatest overlap between our predictions and the foreground in the ground truth. Additionally, the information-based metrics AMI and NMI indicate that our model shares the most amount of information with the ground truth. Overall, AdaSlot outperforms fixed slot models across all five mentioned metrics. For *Localization*, our model have the highest CorLoc and as good as best mBO compared with fixed slot models. Improper slot number will oversegment or undersegment the objects, and decrease the IoU, leading to poor spatial localization.

In MOVi-E, 18-24 slots model keeps the precision at a higher level. Our model can decide the slot number according to the instance and further merge the oversegmented clusters together to improve the recall rate by a large amount. On MOVi-E, our model keeps the same level of precision as 18-slot model but has around 12 points higher recall. Therefore, our model reaches best F_1 and ARI scores.

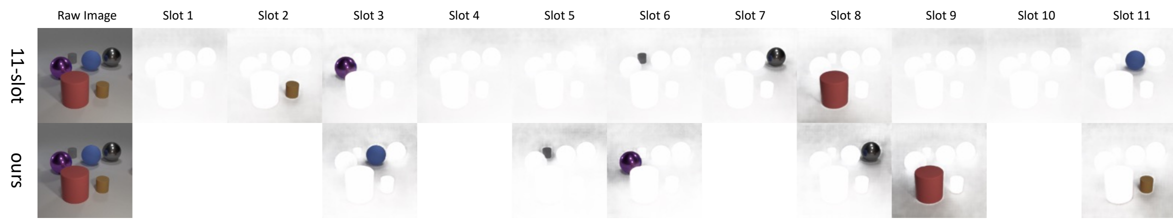

Figure 4. **Visualization of instance-level adaptive slot number selection** by per-slot segmentation, comparing the fixed 11-slot model(first row) and our model(second row). Dropped slot are left empty.

Table 2. Experiments on MOVi-E. (P., R. for Precision, and Recall)

Model	K	Pair-Counting				Matching			Information	
		ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
GENESIS-V2	9	48.19	61.52	58.86	58.14	11.16	12.38	60.07	65.16	65.35
	24	34.27	62.97	34.87	43.32	16.12	21.13	48.34	57.57	58.06
DINOSAUR	3	36.78	41.37	85.27	54.10	6.23	1.67	53.19	50.31	50.42
	6	68.68	68.20	**88.66**	75.66	12.04	27.92	73.81	76.52	76.62
	9	*76.01*	77.29	*87.83*	*81.16*	25.41	87.45	*79.57*	81.17	81.28
	13	73.74	83.73	77.35	78.93	29.08	*90.02*	78.41	*81.53*	*81.67*
	18	68.89	86.08	68.36	74.46	29.57	86.71	74.60	80.19	80.35
	21	66.15	*87.15*	63.87	71.86	*30.01*	85.57	72.39	79.33	79.51
	24	61.98	**88.09**	57.82	67.91	**30.54**	85.15	68.96	77.93	78.14
AdaSlot (Ours)		**76.73**	85.21	80.31	**81.42**	29.83	**91.03**	**81.28**	**83.08**	**83.20**

Results on COCO. MS COCO has a problem of extreme imbalance in its validation set: most images have less than 10 objects. This makes it difficult to determine the correct number of slots. To address this, we conducted experiments using a wide range of slot numbers with non-uniform spacing. The results can be found in Table 3 and Fig 3.

When it comes to object grouping, MS COCO is highly sensitive to the number of slots in the fixed-slot DINOSAUR. The experiment showed that the best results were achieved with 6 slots. However, increasing the number of slots led to a rapid decline in performance, especially in object grouping. For example, just going from 6 to 8 slots resulted in a significant drop of around 4 points in ARI, which is about a 10% reduction from the maximum score.

Our models, set $K_{max} = 33$ and equipped with complexity-aware regularization, effectively surpass the performance of the 33-slot model. Specifically, our model achieves approximately 20 points higher in terms of ARI. Although the improvement in localization is comparatively smaller, our model still outperforms the 33-slot model by three points in terms of mBO.

It is worth noting that on the MS COCO dataset, the best results obtained with fixed slot numbers are marginally superior to our results. COCO's nature images present greater challenges than MOVi-C/E due to incomplete labeling, cluttered compositions without clear backgrounds, and a vast range of object sizes and varieties. Despite these challenges, our complexity-aware module enables our model to achieve results comparable to top-performing fixed-slot methods, highlighting its effectiveness.

4.2. Revealing the insights of AdaSlot

Statistical Results Stratified by Ground-truth Object Number. The above sections reflect the average performance of models on the whole validation datasets. However, the model may over-fit a specific slot number to improve the final average. *To eliminate this possibility*, we used stratified sampling method on MOVi-C/E to display the values of various metrics of images with different ground truth object number in Fig. 5 . For MOVi-C, we compare our models with fixed 11 slots(the upper bound of object number) and fixed 6 slots(high ARI and mBO simultaneously). Similarly, for MOVi-E, compare our models with fixed 13-slot and 24-slot models.

Precision&Recall are inversely related to the number of objects present in an image. As the number of objects increases, precision decreases while recall increases. In the case of our model, it falls somewhere in between high-slot and low-slot models in terms of precision. However, regarding the recall, our model outperforms high-slot models significantly and performs just as well as low-slot models for image with different objects number.

ARI&mBO. Different advantages can be observed for large and small slot models. Our model's curve encom-

Table 3. Experiments on COCO datasets. (P., R. for Precision, and Recall)

Model	K	Pair-Counting				Matching			Information	
		ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
GENESIS-V2	6	25.39	58.95	40.49	44.60	15.42	7.77	52.39	33.55	34.15
	33	9.74	63.61	10.77	15.28	10.19	0.41	21.26	24.08	26.08
DINOSAUR	4	30.85	75.95	61.93	62.86	17.75	17.95	61.09	37.30	37.35
	6	**41.89**	82.00	**70.12**	**70.66**	*27.46*	**50.81**	**69.07**	**46.11**	**46.16**
	7	*39.95*	82.87	65.69	68.00	**27.77**	*50.09*	66.40	*45.25*	*45.31*
	8	37.60	83.83	59.86	64.38	26.93	45.68	62.93	44.36	44.43
	10	35.25	85.29	54.05	60.43	27.19	44.18	59.15	43.66	43.73
	12	32.70	86.44	48.63	56.53	27.02	42.42	55.55	42.64	42.71
	20	26.55	*88.93*	36.31	46.00	25.43	35.28	46.18	40.00	40.10
	33	20.83	**90.96**	26.63	36.50	24.09	32.09	37.87	37.10	37.23
AdaSlot (Ours)		39.00	81.86	*66.42*	*68.37*	27.36	47.76	*67.28*	44.11	44.17

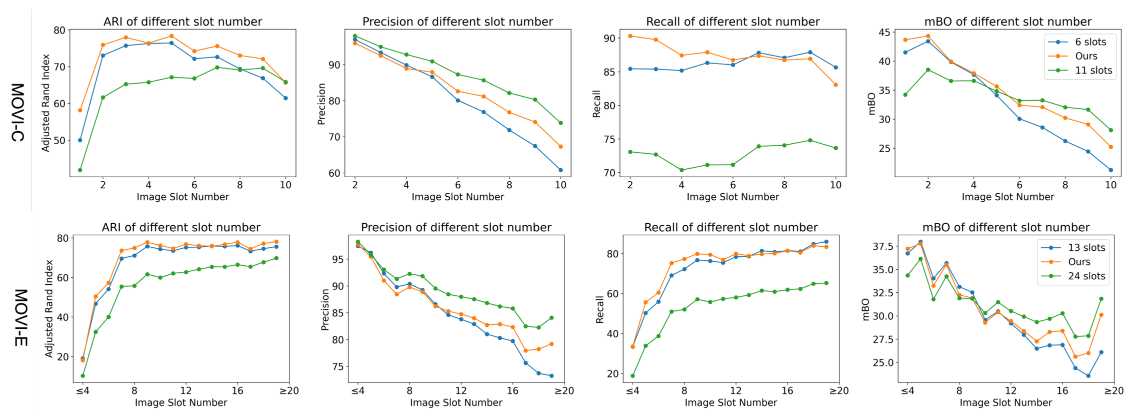

Figure 5. **Stratified statistics** of four metrics of our models and two fixed slot models, one set the slot number to the upper bound and another set to slot with both high ARI and mBO. We apply stratified sampling according to ground truth object number the image have. The first row is MOVi-C while second row is MOVi-E. *The visualizations prove that our model do not over-fit a specific slot number to improve the performance.*

passes the metric curve of the two fixed-slot models for ARI, indicating a wider range of effectiveness. For mBO, our model achieves a performance comparable to the better-performing fixed-slot models across the entire range. This demonstrates the efficacy of our dynamic slot selection approach, as it consistently delivers favorable results.

Comparison between ground truth and predicted object numbers. We reveal the insights of our model by showing some examples in Fig. 3, and heatmap and slot distribution in Fig. 6. The predictions of fixed-slot models tend to be concentrated within a narrow range, forming a sharp peak which deviates from ground truth distribution. In contrast, our models exhibit a smoother prediction distribution that closely aligns with the ground truth.

On MOVi-C/E, fixed-slot models may generate fewer masks due to the one-hot operation. However, most of their predictions are concentrated around the predefined slot number, resulting in a heatmap exhibiting a distinct vertical pattern. Our model instead exhibits an approximately diagonal pattern on the heatmap. In other words, our model can predict more masks for images with more objects, and the number of predicted masks roughly matches the ground truth number. Though the diagonal relationship is imperfect, and the prediction on images with an extremely large or small number of objects is slightly poorer than other images, our model first achieves the adaptive slot selection.

Figure 3 demonstrated the adaptability of slot numbers at the instance level with illustrative examples. In particular, on the MOVi-E dataset, our model generates 13 and 6 slots for two different images, highlighting a significant discrepancy in slot counts. Noteworthy, our results effectively group pixels based on image complexity, resulting in accurate and appropriate segmentation.

Results on Object Property In addition to object discovery, we study the usefulness of adaptive slot attention for other downstream tasks. Following the setting of [8], we provide experiments of object category prediction on the MOVi-C dataset. Our experiments employ a two-layer MLP

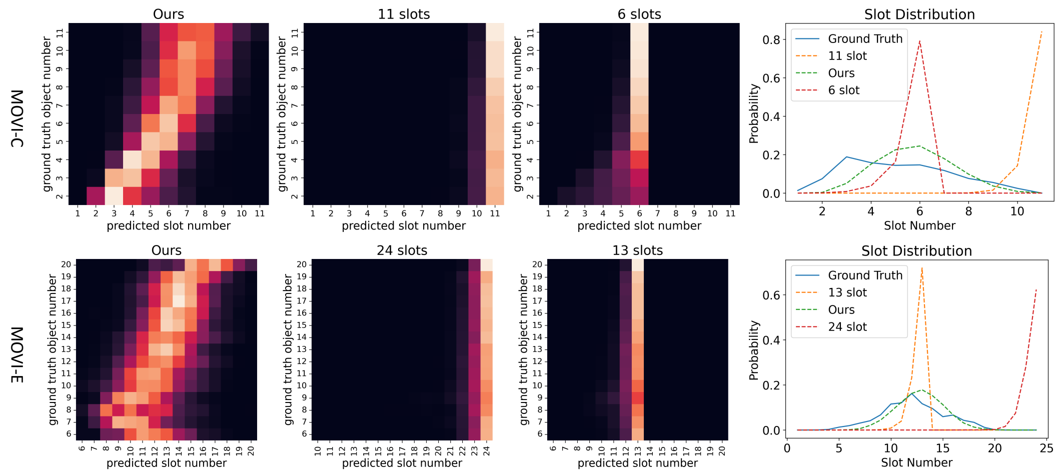

Figure 6. **Comparison between ground truth and predicted object numbers.** Heatmap of confusion matrix and slot distribution of our models and two fixed slot models on MOVi-C/E. For heatmap, y-axis corresponds to the number of objects of ground truth, and x-axis is the predicted object number by models. Due to imbalanced ground truth object numbers, we normalized the row and visualize the percentage. The brighter the grid, the higher the percentage. The slot distribution graph shows the probability density of grounded and predicted object numbers.

Table 4. Experiments of **object property prediction** on MOVi-C.

Slot	Recall	Precision	Jaccard
3	36.16	74.89	36.16
6	58.62	60.69	51.43
9	70.34	48.55	47.93
11	77.88	43.98	43.98
Ours	59.25	63.08	54.10

as the downstream model. Our model only makes predictions on the retained slots. We employ cross-entropy loss and align predictions with targets with the Hungarian algorithm [22], minimizing the total loss of the assignment. To better compare the results for models with different slot numbers, we provide the precision, recall and the Jaccard index. The results are provided in Tab. 4.

In the fixed slot model, an increase in the number of slots typically leads to the discovery of more objects, thus enhancing recall. However, models with a larger number of slots also tend to generate more redundant objects, adversely affecting precision. The Jaccard index, which takes slot redundancy into account, offers a more comprehensive evaluation. In our experiments on MOVi-C dataset, the 6-slot model achieved the best Jaccard index among fixed-slot models. Notably, our model yields a superior Jaccard index to all fixed slot models. This demonstrates the effectiveness of our adaptive slot attention mechanism.

More Ablation Study in Appendix. We conduct thorough ablation studies in the appendix to assess our framework, including comparing three masked decoder designs and examining the impact of λ. These studies demonstrate our model's effectiveness.

Limitations. Our model excels in scenarios with well-segmented objects but may struggle with complex, densely packed scenes like COCO, where annotations are *incomplete* and learned objects don't always align with manual labels. Its performance on small, dense objects is limited, and the complexity of real-world part-whole hierarchies poses additional challenges. We aim to address these issues in future work.

5. Conclusion

We have introduced adaptive slot attention (AdaSlot) that can dynamically determine the appropriate slot number according to the content of the data in object-centric learning. The framework is composed of two parts. A slot selection module is first proposed based on Gumbel-Softmax for differentiable training and mean-field formulation for efficient sampling. Then, a masked slot decoder is further designed to suppress the information of unselected slots in the decoding phase. Extensive studies demonstrate the effectiveness of our AdaSlot in two folds. First, our AdaSlot achieves comparable or superior performance to those best-performing fixed-slot models. Second, our AdaSlot is capable of selecting appropriate slot number based on the complexity of the specific image. The instance-level adaptability offers potential for further exploration in slot attention.

Acknowledgements: Yanwei Fu is the corresponding authour. Yanwei Fu is with School of Data Science, Fudan University, Shanghai Key Lab of Intelligent Information Processing, Fudan University, and Fudan ISTBI-ZJNU Algorithm Centre for Brain-inspired Intelligence, Zhejiang Normal University, Jinhua, China.

References

[1] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020. 3

[2] David M Blei and Michael I Jordan. Variational inference for dirichlet process mixtures. 2006. 3

[3] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017. 2, 3

[4] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. 2

[5] Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, Dirk Weissenborn, Jakob Uszkoreit, and Thomas Unterthiner. Differentiable patch selection for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2351–2360, 2021. 3

[6] Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3412–3420, 2019. 2

[7] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranking and sorting using optimal transport. *Advances in neural information processing systems*, 32, 2019. 2

[8] Andrea Dittadi, Samuele Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and robustness implications in object-centric learning. *arXiv preprint arXiv:2107.00637*, 2021. 7

[9] Gamaleldin Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. *Advances in Neural Information Processing Systems*, 35:28940–28954, 2022. 1

[10] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019. 2

[11] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. *Advances in Neural Information Processing Systems*, 34:8085–8094, 2021. 2

[12] SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in neural information processing systems*, 29, 2016. 2

[13] Ke Fan, Zechen Bai, Tianjun Xiao, Dominik Zietlow, Max Horn, Zixu Zhao, Carl-Johann Simon-Gabriel, Mike Zheng Shou, Francesco Locatello, Bernt Schiele, Thomas Brox, Zheng Zhang, Yanwei Fu, and Tong He. Unsupervised open-vocabulary object localization in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13747–13755, 2023. 1

[14] Ke Fan, Jingshi Lei, Xuelin Qian, Miaopeng Yu, Tianjun Xiao, Tong He, Zheng Zhang, and Yanwei Fu. Rethinking amodal video segmentation from learning supervised signals with object-centric representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1272–1281, 2023. 1

[15] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pages 2424–2433. PMLR, 2019. 1, 2

[16] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3761, 2022. 4

[17] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 2, 4

[18] 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. 4

[19] Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021. 1, 2

[20] Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, pages 3499–3508. PMLR, 2019. 2

[21] Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[22] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 8

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 4

[24] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. 1, 2, 3

[25] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete

random variables. *arXiv preprint arXiv:1611.00712*, 2016. 2

[26] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. *arXiv preprint arXiv:2209.14860*, 2022. 2, 4

[27] Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate dall-e learns to compose. In *International Conference on Learning Representations*, 2021. 1, 2

[28] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. *arXiv preprint arXiv:2205.14065*, 2022. 2

[29] Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. *Advances in Neural Information Processing Systems*, 33:20520–20531, 2020. 2

[30] Andrii Zadaianchuk, Matthaeus Kleindessner, Yi Zhu, Francesco Locatello, and Thomas Brox. Unsupervised semantic segmentation with self-supervised object-centric representations. *arXiv preprint arXiv:2207.05027*, 2022. 1