MetaSlot: Break Through the Fixed Number of Slots in Object-Centric Learning

Hongjia Liu[1] **Rongzhen Zhao**[1]* **Haohan Chen**[2] **Joni Pajarinen**[1]
[1]Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland
[2]Department of Computer Science, Sichuan University, Chengdu, China
{hongjia.liu, rongzhen.zhao, joni.pajarinen}@aalto.fi
sajel@stu.scu.edu.cn

Abstract

Learning object-level, structured representations is widely regarded as a key to better generalization in vision and underpins the design of next-generation Pre-trained Vision Models (PVMs). Mainstream Object-Centric Learning (OCL) methods adopt Slot Attention or its variants to iteratively aggregate objects' super-pixels into a fixed set of query feature vectors, termed slots. However, their reliance on a static slot count leads to an object being represented as multiple parts when the number of objects varies. We introduce MetaSlot, a plug-and-play Slot Attention variant that adapts to variable object counts. MetaSlot (i) maintains a codebook that holds prototypes of objects in a dataset by vector-quantizing the resulting slot representations; (ii) removes duplicate slots from the traditionally aggregated slots by quantizing them with the codebook; and (iii) injects progressively weaker noise into the Slot Attention iterations to accelerate and stabilize the aggregation. MetaSlot is a general Slot Attention variant that can be seamlessly integrated into existing OCL architectures. Across multiple public datasets and tasks–including object discovery and recognition–models equipped with MetaSlot achieve significant performance gains and markedly interpretable slot representations, compared with existing Slot Attention variants. The code is available at https://github.com/lhj-lhj/MetaSlot.

1 Introduction

Human intelligence is rooted in limited perceptual experience–especially visual information–which enables it to demonstrate outstanding transfer and generalization abilities in entirely new task scenarios [1, 2]. In recent years, major breakthroughs in embodied intelligence have further underscored the inevitable trend of artificial intelligence moving into the physical world [3–5]. However, the key challenge in achieving high-level cognitive reasoning [6, 7] and compositional generalization [8, 9] lies in transforming visual inputs into structured, discrete, and independent object-level representations [10–12], thereby granting agents a deep understanding of physical objects and their dynamic relations [13, 14].

Object-Centric Learning (OCL) has rapidly developed against this backdrop. Its goal is to extract object-level structured representations in an unsupervised manner, rather than relying on attribute-level features or global scene features. Among numerous methods [15–20], Slot Attention (SA) [11] is currently the most influential and widely adopted. Through a competition mechanism among slots, it iteratively clusters distributed scene representations into several object-oriented feature vectors, named slots; each slot can then be decoded separately [11, 21], or all slots can be decoded jointly in an autoregressive fashion [22, 23] to produce semantically consistent segmentation masks. This

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

strategy not only efficiently captures object-level information but also lays a solid foundation for subsequent physical reasoning and relational modeling [24–26].

Nevertheless, classic Slot Attention [11] still suffers from two key limitations: (1) the number of slots must be preset as a fixed hyper-parameter; (2) slot initialization relies on random sampling. The former conflicts with the dynamic variability of object counts in real visual scenes, easily leading to under-segmentation or over-segmentation and harming the identifiability of the representations [27, 28]; the latter often results in object-centric representations that lack a clear correspondence with true object concepts [29]. Overall, a fixed slot count and random initialization are equivalent to imposing inappropriate prior assumptions on the latent space, making the model more prone to sub-optimal solutions and limiting its generalization capability.

Vector quantization (VQ) [30] offers a viable pathway: It has recently shown great value in generative modeling by enabling models to extract and reuse semantic structural patterns [31–33]. Inspired by this insight, we incorporate a VQ codebook that supplies globally shared object prototypes, guiding slot initialization structurally; meanwhile, we prune duplicate slots to provide explicit semantic cues about "objects" from the very start of the aggregation process. In particular, this idea of "object prototypes" echoes Plato's "world of forms" [34]: every concrete object in the perceptual world is a projection of some eternal and perfect ideal form. Analogously, we regard each prototype vector in the VQ codebook as an idealized object concept, whereas the input features are concrete mappings of these forms. Based on this intuition, we propose MetaSlot, a novel object-centric learning framework that employs a unified prior of global prototype slots and a dynamically adaptive two-stage aggregation method to flexibly match the slot count to the objects present in a scene. Specifically, our study introduces two important technical innovations:

Dynamic slot allocation. To address the above limitations, we design the MetaSlot framework to adaptively adjust the number of slots through two-stage aggregation. First, to match input features with the prototype codebook, we perform initial aggregation using Slot Attention in first stage. The resulting slot vectors are then matched to the global discrete codebook, producing semantically consistent discrete slot indices. Next, to allow the model to adjust the effective slot count according to scene complexity, we apply a de-duplication operation to slot indices that correspond to the same prototype, retaining only distinct prototype slots as the initialization for second stage. The object-aware initial slots are then fed into a mask slot attention module for a second aggregation stage, enabling fine-grained object-level assignment. Throughout this stage, the aggregator applies an attention mask to redundant slots and shares the same weights as the first-stage Slot Attention.

Consistent prototypes and stable optimization. To ensure that all parts of the same object converge to a consistent prototype, we use the final slot representations obtained from the second stage to update the codebook. Simultaneously, we employ a k-means-based exponential moving average (EMA) strategy to update the codebook stably and suppress high variance in the early training phase. In addition, we introduce a progressive noise-injection mechanism during training as implicit simulated annealing [35], further reinforcing efficient alignment between the prototype prior and the posterior over targets in latent space.

In short, MetaSlot leverages a global VQ codebook of object prototypes. Slot Attention clusters features, aligns them with their nearest prototypes, prunes duplicates, and passes the resulting semantically rich slots to a second aggregation stage. Moreover, injecting progressive noise during this stage helps stabilize convergence, yielding robust and accurate object representations. Our work makes three primary contributions: (i) MetaSlot module: We devise a two-stage aggregation framework that couples a global vector-quantized codebook with a slot-masking mechanism, enabling dynamic slot allocation for arbitrary numbers of objects. (ii) Progressive noise injection: By injecting gradually diminishing Gaussian noise throughout the Slot Attention iterations, MetaSlot both accelerates convergence and stabilizes the aggregation. (iii) Large-scale validation: Extensive experiments across diverse vision tasks and datasets show that MetaSlot yields substantial improvements on key metrics and exhibits strong adaptability to a wide range of scenes.

2 Method

In this section, we present **MetaSlot** with (i) First-stage Aggregation for Prototype-guided Pruning; (ii) Second-stage Aggregation with Mask-guided Refinement; and (iii) Prototype update via mini-

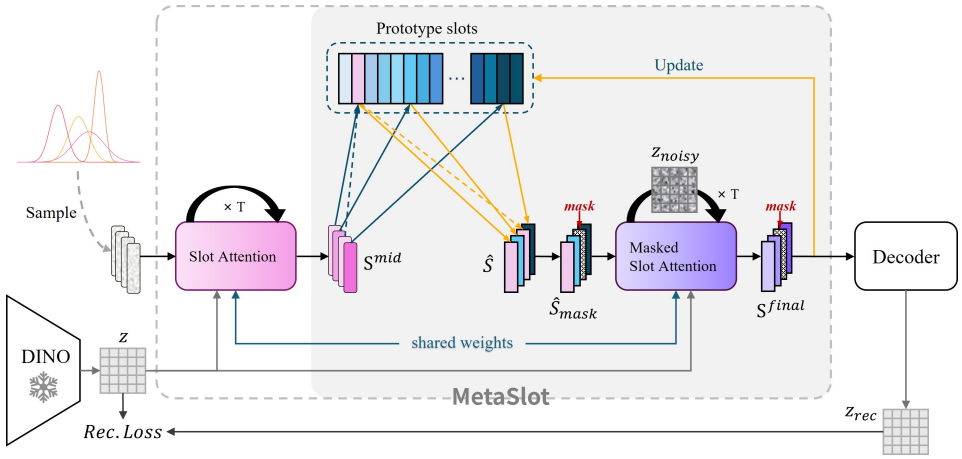


Figure 1: Overview of the MetaSlot framework (depicted on the DINOSAUR backbone for clarity; agnostic to the underlying object-centric architecture). (i) We build and continually update a codebook of "prototype slots" by vector-quantizing slots sampled across the dataset. (ii) Input features $\boldsymbol{Z}$ are first aggregated via Slot Attention to produce an intermediate slot set $\boldsymbol{S}^{\text{mid}}$; we then remove duplicate slots in $\boldsymbol{S}^{\text{mid}}$ by matching them against the prototype slots, yielding the masked subset $\hat{\boldsymbol{S}}_{\text{mask}}$. (iii) Finally, $\hat{\boldsymbol{S}}_{\text{mask}}$ is passed through Masked Slot Attention with progressively attenuated noise to generate the refined slots $\boldsymbol{S}^{\text{final}}$, which are then decoded to reconstruct the original input.

batch K-means. Notably, the two aggregation modules share weights and are jointly trained. In addition, we include pseudocode in Appendix A to provide additional implementation details.

2.1 Background

Slot Attention (SA). Slot Attention (SA) [11] transforms a set of input features $\boldsymbol{Z} \in \mathbb{R}^{N \times D}$ into K object-centric representations $\boldsymbol{S} \in \mathbb{R}^{K \times D}$ through iterative cross-attention. Each iteration, slots compete by applying a softmax over themselves to claim parts of the visual input, and each slot's incremental information $\tilde{\boldsymbol{S}}$ is computed as the attention-weighted sum of visual feature vectors.

$$\tilde{\boldsymbol{S}} = f_{\phi_{\text{attn}}}(\boldsymbol{S}, \boldsymbol{Z}) = \left(\frac{A_{i,j}}{\sum_{l=1}^{N} A_{l,j}} \right)^{\top} \cdot v(\boldsymbol{Z}), \quad \text{where } \boldsymbol{A} = \text{softmax}\left(\frac{k(\boldsymbol{Z}) \cdot q(\boldsymbol{S})^{\top}}{\sqrt{D}} \right) \in \mathbb{R}^{N \times K}. \tag{1}$$

In the slot update stage, the incremental information $\tilde{\boldsymbol{S}}$ and the previous slot state $\boldsymbol{S}^{(t)}$ are fed into a Gated Recurrent Unit (GRU) [36]. The GRU output is then refined by a small MLP, yielding the updated slot state:

$$\boldsymbol{S}^{(t+1)} = \text{MLP}\big(\text{GRU}(\boldsymbol{S}^{(t)}, \tilde{\boldsymbol{S}})\big). \tag{2}$$

After T iterations, the final slots $\boldsymbol{S}^{T}$ are employed as the object-centric representation passed to downstream modules. Crucially, these slots are randomly initialized from a learnable Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}))$.

2.2 First-Stage Aggregation for Prototype-guided Pruning

In the first-stage aggregation, MetaSlot performs global prototype alignment, merging all features of an object into one slot, pruning duplicate slots, and yielding a compact, semantically coherent basis for later fine-grained aggregation.

To obtain prototype slot representations from the codebook that faithfully capture the input features, we first perform a preliminary aggregation of the feature maps, producing a set of softly assigned intermediate slots $\boldsymbol{S}^{\text{mid}}$. Each intermediate slot is then matched to the global discrete codebook $\mathcal{E}$ via a nearest-neighbour search, yielding the semantically aligned discrete slots $\hat{\boldsymbol{S}}$.

Given the intermediate slots $\boldsymbol{S}^{\text{mid}}$ produced by the original Slot Attention [11], the fixed number of slots can lead to a single object's features being scattered across multiple slots. To resolve this, prototype matching maps every slot encoding the same object onto a shared prototype, thereby eliminating redundancy. As a result, among slots with identical indices only the unique prototypes $\hat{\boldsymbol{S}}_{\text{mask}}$ are retained, and this compact set is used to initialize the next stage.

Intermediate slots. As in the original formulation, we sample the initial slots $\boldsymbol{S}^{(0)} \in \mathbb{R}^{N\times D}$ from a learnable Gaussian $\mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}))$, and then perform $\boldsymbol{T}$ iterative slot updates on the input feature set $\boldsymbol{Z} \in \mathbb{R}^{F\times D}$.

$$\boldsymbol{S}^{\text{mid}} = \text{SlotAttn}(\boldsymbol{Z}, \boldsymbol{S}^{(0)}, \boldsymbol{T}). \tag{3}$$

Nearest-neighbour quantisation. Let $\mathcal{E} = \{\boldsymbol{e}_k \in \mathbb{R}^D\}_{k=1}^K$ denote a global codebook of K prototypes. For each intermediate slot vector $\boldsymbol{s}_i^{\text{mid}}$, we find the index of its nearest prototype and replace it accordingly:

$$idx_i \;=\; \arg\min_k \left\|\boldsymbol{s}_i^{\text{mid}} - \boldsymbol{e}_k\right\|_2, \qquad \hat{\boldsymbol{s}}_i \;=\; \boldsymbol{e}_{idx_i}, \tag{4}$$

where $\hat{\boldsymbol{s}}_i$ is the quantised slot vector, set to the prototype $\boldsymbol{e}_{idx_i}$.

Duplicate-removal mask. Slots that pick the same prototype are treated as redundant. We mark the first occurrence and mask out the rest:

$$smask_i \;=\; \begin{cases} 1, & \text{if } idx_i \text{ is the first hit,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The surviving slot set is $\hat{\boldsymbol{S}}_{\text{mask}} = \{\hat{\boldsymbol{s}}_i \mid smask_i = 1\} \subseteq \mathbb{R}^{\hat{N}\times d}$ with $\hat{N} \le N$.

2.3 Second-Stage Aggregation with Mask-guided Refinement

In the second-stage aggregation, we refine the pruned prototypes via masked attention with annealed noise injection, producing the final semantically coherent slot set $\boldsymbol{S}^{\text{final}}$.

We re-initialize the slot states with the pruned prototype set $\hat{\boldsymbol{S}}_{\text{mask}}$ and perform T iterations of attention-based updates. At each step, the raw attention logits are masked by the binary slot mask $smask$, so that only retained prototypes participate in computing the attention-weighted slot increments. To alleviate the "cold-start" misalignment between prior prototypes and current inputs, we also inject progressively isotropic Gaussian noise into the features before each iteration, with variance α_t^2 linearly annealed from σ_{noise}^2 to zero. This implicit simulated annealing encourages early exploration and late-stage convergence, yielding the final semantically coherent slot set $\boldsymbol{S}^{\text{final}}$.

Implicit Simulated Annealing via Noise Injection In the classic Slot Attention module, slots aggregate visual features through iterative attention and GRU updates. However, when prototype slots are generated by offline vector quantization (VQ), initial misalignment often occurs between the prior slots and posterior slots derived from the current input in the latent space. To reduce this "cold-start" distance, we explicitly inject decreasing noise into the features before each iteration. This strategy can be interpreted as a form of implicit simulated annealing: injecting large-magnitude noise at early stages relaxes the entropy constraints of soft matching, encouraging exploration among slots; gradually reducing noise at later stages facilitates convergence to precise alignment. At any iteration step t, we add isotropic Gaussian noise to the features $\boldsymbol{Z} \in \mathbb{R}^{N\times D}$:

$$\boldsymbol{Z}_{\text{noise}}^{(t)} \;=\; \boldsymbol{Z}^{(t)} + \boldsymbol{\xi}_{\text{iso}}^{(t)}, \qquad \boldsymbol{\xi}_{\text{iso}}^{(t)} \sim \mathcal{N}\big(\mathbf{0},\, \alpha_t^2 \mathbf{I}_C\big), \tag{6}$$

where

$$\alpha_t \;=\; \sigma_{\text{noise}}\Big(1 - \tfrac{t}{T-1}\Big). \tag{7}$$

Eq. (7) corresponds to a gradual decrease in temperature $\tau \propto \alpha_t^{-2}$, and σ_{noise} is a tunable hyperparameter that specifies the initial standard deviation of the injected isotropic Gaussian noise (i.e. the noise amplitude at $t = 0$) before annealing.

Masked Slot Attention (MSA). To ensure that only surviving slots steer the refinement, we introduce Masked Slot Attention (MSA). At each iteration, a binary mask $smask$ zeros out rows corresponding to duplicate slots, so the attention-weighted update is computed solely from the retained, semantically meaningful prototype slots. This prevents any duplicate-induced interference. It is worth noting that the MSA shares the same set of weights with the SA used in the first-stage aggregation. For each iteration $t = 0, \ldots, T-1$ we compute the MSA:

$$\tilde{\boldsymbol{S}}^{(t)} = f_{\phi_{\text{attn}}}\big(\boldsymbol{S}^{(t)}, \boldsymbol{Z}^{(t)}_{\text{noise}}\big) = \Big(\tfrac{\tilde{A}^{(t)}_{i,j}}{\sum_{l=1}^{N} \tilde{A}^{(t)}_{l,j}}\Big)^{\top} \cdot v(\boldsymbol{Z}^{(t)}_{\text{noise}}), \qquad \tilde{A}^{(t)}_{i,j} = smask_i \, A^{(t)}_{i,j}, \tag{8}$$

with

$$\boldsymbol{A}^{(t)} = \text{softmax}\Big(\tfrac{k(\boldsymbol{Z}^{(t)}_{\text{noise}}) \cdot q(\boldsymbol{S}^{(t)})^{\top}}{\sqrt{D}}\Big) \in \mathbb{R}^{N \times K}, \tag{9}$$

where $q, k, v \in \mathbb{R}^D$ are the linearly projected queries, keys and values.

Finally, the slot states are updated as in Eq. (2), yielding $\boldsymbol{S}^{(t+1)}$.

Gradient Truncation and Bi-level Optimization. Because the vector-quantization (VQ) mechanism truncates gradients–producing instability between Two-stage Slot Aggregation iterations, we stop the gradient flow at the first Slot-aggregation stage. In addition, inspired by the bi-level optimization strategy [29], we further detach gradients during the first $T-1$ iterations of the second Slot-aggregation stage.

Let $\boldsymbol{S}_2^{(T)}$ be the slots after the t-th refinement step in second-stage aggregation. Thus all paths that reach the encoder features $\boldsymbol{Z}$ through $\boldsymbol{S}_1^{(0)}, \ldots, \boldsymbol{S}_2^{(T-1)}$ are detached, and only the T-th (final) refinement step $\boldsymbol{S}_2^{(T)}$ contributes gradients.

2.4 Prototype update via mini-batch *K*-means.

To encourage the codebook slots to converge toward identifiable slot prototypes, we use the final $\boldsymbol{S}_2^{(T)}$ to update the codebook. To ensure stable updates to the codebook, we adopt a K-means-based exponential moving average (EMA) update strategy. Specifically, at each training step we shift every prototype $\boldsymbol{e}_k$ toward the mini-batch centroid $\boldsymbol{c}_k$ computed over $\boldsymbol{S}_2^{(T)}$:

$$\boldsymbol{e}_k \;\leftarrow\; (1-\eta)\,\boldsymbol{e}_k \;+\; \eta\,\boldsymbol{c}_k, \tag{10}$$

where $\eta \in (0, 1]$ is a small learning rate.

Prototypes that remain unselected for a predefined *timeout* window are marked as dead. For each such code we sample a replacement vector $\tilde{\boldsymbol{c}}$ from the current mini-batch by choosing the slot that is least similar (cosine distance) to all active prototypes, and reset the dead code via

$$\boldsymbol{e}_k \;\leftarrow\; \tilde{\boldsymbol{c}}. \tag{11}$$

3 Related Work

In recent years, unsupervised representation learning has made significant progress, with Slot Attention (SA) [11] playing a pivotal role in advancing this field. SA learns distinct latent representations for each object in an image through an iterative mechanism, and these latent "slots" can subsequently be decoded back into pixel space. Early slot-based methods [11, 21–23, 37] typically employed simple small-scale CNNs [38] or pre-trained ResNet models [39] as feature encoders, and used Spatial Broadcast Decoders [40] or Vision Transformers [41] as decoders, with tests mainly conducted on synthetic datasets. Recent approaches such as SlotDiffusion [42] and LSD [43] integrate SA with diffusion-model decoders [44, 45]. DINOSAUR and its variants [46–48] constructs reconstruction objectives based on DINO [49, 50] features to enhance object discovery in real-world data.

To enhance Slot Attention's intrinsic object-awareness, BO-QSA [29] introduces bi-level mechanisms and slot-level initialization, whereas ISA [51] incorporates pose-equivariance within its attention and

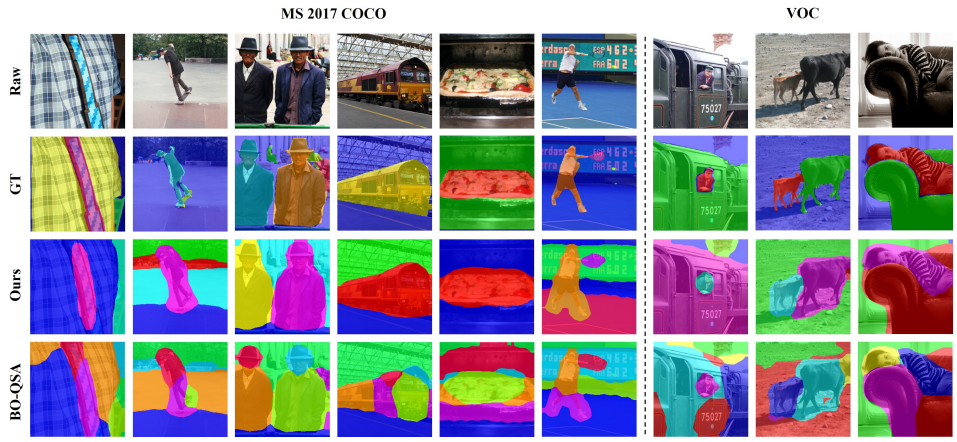


Figure 2: Qualitative results show that MetaSlot's dynamic slot allocation mitigates BO-QSA's over-segmentation, such as splitting a train into unrelated parts, due to its fixed slot count.

generative modules. However, these methods remain constrained by a fixed number of slots, resulting in persistent over-segmentation issues. FT-DINOSAUR [47] mitigates redundancy by selecting the top k most probable slots during the decoding phase, and SOLV [52] clusters aggregated slots to improve semantic consistency. Nonetheless, these approaches rely on heuristic, non-learning-based strategies with explicit thresholding during decoding. While AdaSlot [53] directly predicts the number of slots from feature maps, it does not achieve quantitative improvements over standard Slot Attention on real-world datasets. Furthermore, none of these existing methods incorporate explicit object priors or semantic cues during initialization.

On the theoretical front, a number of studies have offered formal interpretations of object-centric learning (OCL) [54, 27, 55–58]. In terms of evaluation methodologies, recent works have examined the generalization ability of object-centric representations across various downstream tasks, including visual question answering (VQA) [59], world modeling [24, 25, 60, 61], and video generation [59, 62]. Furthermore, several studies have investigated the robustness of object-centric models under out-of-distribution (OOD) conditions [63, 64, 47]. Other works have explored the use of vector quantization mechanisms to improve object disentanglement and interpretability. For instance, methods such as [65, 66] learn hierarchical, compositional discrete representations that align with objects and their attributes.

4 Experiments

Overall, our study targets two goals: first, to show that MetaSlot–when substituted for vanilla slot attention–consistently enhances the performance of object-centric learning (OCL) models; and second, to demonstrate that MetaSlot plugs in naturally to both Transformer-based and diffusion-based OCL frameworks, underscoring its broad applicability. We evaluate its impact on two canonical tasks: object discovery, which demands pixel-accurate masks for every object instance, and set prediction, where classification accuracy reveals how much object information the slots capture and thus reflects the quality of the learned representations.

Datasets We include both synthetic and real-world datasets. ClevrTex [67] comprises synthetic images, each with about 10 geometric objects scattered in complex backgrounds. MS COCO 2017 [68] is a recognized real-world image dataset, and we use its challenging panoptic segmentation and instance-level object annotations. PASCAL VOC 2012 [69] is a real-world image dataset, and we use its instance segmentation. We also report results on the real-world video dataset HQ-YTVIS [70], which contains large-scale short videos from YouTube.

Training Details To eliminate confounding implementation differences, we re-implemented every baseline from scratch rather than reusing published results. All experiments share identical data augmentation pipelines and use the DINOv2 ViT(s/14) [50] as the OCL encoder, with matched

training hyperparameters. Every model—including both the baselines and our variants augmented with MetaSlot–was trained for 50 k steps with the Adam optimizer [71] on a single NVIDIA V100 GPU using 16-bit mixed precision and a batch size of 32; the MetaSlot codebook size was fixed to 512 throughout. As the most advanced publicly available aggregator currently surpassing vanilla Slot Attention, BO-QSA [29] is adopted as the default module in all baselines. This uniform setup ensures fair and reproducible comparisons, enabling precise evaluation of MetaSlot's contribution. All reported results are averaged over three random seeds to mitigate stochastic variance.

4.1 Evaluate on Object Discovery

Models We integrate MetaSlot into object-centric learning (OCL) frameworks and systematically benchmark it against a range of classic models (Table 1) as well as state-of-the-art models (Table 2) to highlight its performance gains. Concretely, SLATE [22] employs a Transformer decoder for autoregressive reconstruction. DINOSAUR [46] uses an MLP-based hybrid decoder to reconstruct directly in DINO feature space, while VideoSAUR [72] adapts this design to video. SlotDiffusion [42] performs decoding with a conditional diffusion model, and SPOT [73] combines nine permutation-based Transformer decoders with a self-training strategy. Evaluating MetaSlot within each of these heterogeneous decoding paradigms enables a comprehensive assessment of its versatility. We exclude IODINE [49], ISA [74], SAVi [21], SAVi++ [37], and MoTok [75] due to outdated performance or reliance on multi-modal priors that hinder fair comparison under our unified setting. Appendix E presents a comparative evaluation of MetaSlot against AdaSlot[53], SlotContrast[76], SysBinder[65], and NLoTM[66] across multiple datasets, further demonstrating the effectiveness of MetaSlot.

Metrics The object-discovery task provides a straightforward view of how effectively individual slots separate distinct objects. Following standard practice in OCL research, we assess representation quality by comparing the mask assigned to each slot with the instance-level ground-truth masks. Concretely, we report the Adjusted Rand Index (ARI) and Foreground Adjusted Rand Index (FG-ARI) [77] to measure clustering similarity, and evaluate mean Intersection-over-Union (mIoU) and mean Best Overlap (mBO) to quantify how well the discovered masks align with the real objects.

Analysis To comprehensively evaluate the effectiveness of the proposed MetaSlot aggregator, we integrate it into several mainstream OCL decoding frameworks ($\text{MetaSlot}_{\text{Mlp}}$, $\text{MetaSlot}_{\text{Tfd}}$, $\text{MetaSlot}_{\text{Dfz}}$) and directly compare it against their original implementations (DINOSAUR, SLATE, SlotDiffusion).

As shown in Table1, MetaSlot consistently outperforms its corresponding baseline across all decoding frameworks. In the MLP setting, $\text{MetaSlot}_{\text{Mlp}}$ yields higher decoding accuracy and better reconstruction quality than DINOSAUR. Under the autoregressive setting, $\text{MetaSlot}_{\text{Tfd}}$ achieves substantial performance gains over the original SLATE. Similarly, in the diffusion-based framework, $\text{MetaSlot}_{\text{Dfz}}$ consistently surpasses SlotDiffusion. These results demonstrate MetaSlot's strong compatibility and generalization ability across diverse decoder architectures, highlighting its robustness and superiority in complex visual reconstruction tasks. We also visualize the object-segmentation results of $\text{MetaSlot}_{\text{Mlp}}$ on the COCO and VOC datasets in Fig.2. The examples show that MetaSlot performs dynamic slot allocation effectively, eliminating the over-segmentation problem that afflicts the DINOSAUR baseline, whose BO-QSA [29] aggregator enforces a fixed number of slots.

Furthermore, as shown in Table 2, we compare MetaSlot with recent state-of-the-art methods. Because our goal is not to challenge the entire model architecture but to isolate the impact of the aggregator module. Therefore, in comparing with SPOT [73], $\text{MetaSlot}_{\text{Tfd9}}$ is trained using only the decoder from SPOT in a single training round, without adopting the two-stage self-distillation strategy proposed in the original work. Remarkably, even under this simplified training setup, our method achieves comparable or even superior performance across all evaluation metrics. Similarly, when comparing with VideoSAUR on the YTVIS(HQ) dataset, simply replacing its aggregator with MetaSlot yields substantial performance improvements–particularly in FG-ARI (+18.3) and mBO (+2.9). These results highlight MetaSlot's adaptability, proving effective in both static and video-level object discovery tasks.

4.2 Evaluate on Set Prediction

The set prediction task explicitly reveals the effectiveness of each slot in capturing object information. Following this work [46], images from the MS COCO 2017 dataset are encoded into object-centric

Table 1: Object discovery performance with DINOv2 ViT (s/14) for OCL encoding. The input resolution is 256×256 (224×224). Tfd, MLP and Dfz are Transformer, MLP, and Diffusion [78] for OCL decoding respectively.

	ClevrTex #slot=11				COCO #slot=7				VOC #slot=6			
	ARI	FG-ARI	mBO	mIoU	ARI	FG-ARI	mBO	mIoU	ARI	FG-ARI	mBO	mIoU
SLATE	$17.4_{\pm 2.9}$	$87.4_{\pm 1.7}$	$44.5_{\pm 2.2}$	$43.3_{\pm 2.4}$	$17.5_{\pm 0.6}$	$28.8_{\pm 0.3}$	$26.8_{\pm 0.3}$	$25.4_{\pm 0.3}$	$18.6_{\pm 0.1}$	$26.2_{\pm 0.8}$	$37.2_{\pm 0.5}$	$36.1_{\pm 0.4}$
MetaSlot$_{\text{Tfd}}$	$\mathbf{40.9}_{\pm \mathbf{1.7}}$	$\mathbf{92.4}_{\pm \mathbf{0.7}}$	$\mathbf{49.3}_{\pm \mathbf{0.8}}$	$\mathbf{48.8}_{\pm \mathbf{1.6}}$	$\mathbf{18.6}_{\pm \mathbf{1.2}}$	$\mathbf{33.5}_{\pm \mathbf{1.0}}$	$\mathbf{28.2}_{\pm \mathbf{0.7}}$	$\mathbf{26.7}_{\pm \mathbf{0.6}}$	$\mathbf{20.4}_{\pm \mathbf{0.5}}$	$\mathbf{30.7}_{\pm \mathbf{0.8}}$	$\mathbf{39.0}_{\pm \mathbf{0.3}}$	$\mathbf{37.8}_{\pm \mathbf{0.5}}$
DINOSAUR	$50.7_{\pm 24.1}$	$89.4_{\pm 0.3}$	$53.3_{\pm 5.0}$	$52.8_{\pm 5.2}$	$18.2_{\pm 1.0}$	$35.0_{\pm 1.2}$	$28.3_{\pm 0.5}$	$26.9_{\pm 0.5}$	$21.5_{\pm 0.7}$	$35.2_{\pm 1.3}$	$40.6_{\pm 0.6}$	$39.7_{\pm 0.6}$
MetaSlot$_{\text{Mlp}}$	$\mathbf{64.6}_{\pm \mathbf{0.3}}$	$\mathbf{89.6}_{\pm \mathbf{0.4}}$	$\mathbf{55.2}_{\pm \mathbf{0.5}}$	$\mathbf{54.5}_{\pm \mathbf{0.6}}$	$\mathbf{22.4}_{\pm \mathbf{0.3}}$	$\mathbf{40.3}_{\pm \mathbf{0.5}}$	$\mathbf{29.5}_{\pm \mathbf{0.2}}$	$\mathbf{27.9}_{\pm \mathbf{0.2}}$	$\mathbf{32.2}_{\pm \mathbf{1.2}}$	$\mathbf{43.3}_{\pm \mathbf{0.7}}$	$\mathbf{43.9}_{\pm \mathbf{0.3}}$	$\mathbf{42.1}_{\pm \mathbf{0.2}}$
SlotDiffusion	$66.1_{\pm 1.3}$	$\mathbf{82.7}_{\pm \mathbf{1.6}}$	$54.3_{\pm 0.5}$	$53.4_{\pm 0.8}$	$17.7_{\pm 0.5}$	$28.7_{\pm 0.1}$	$27.0_{\pm 0.4}$	$25.6_{\pm 0.4}$	$17.0_{\pm 1.2}$	$21.7_{\pm 1.8}$	$35.2_{\pm 0.9}$	$34.0_{\pm 1.0}$
MetaSlot$_{\text{Dfz}}$	$\mathbf{81.9}_{\pm \mathbf{0.2}}$	$77.6_{\pm 1.1}$	$\mathbf{64.2}_{\pm \mathbf{0.6}}$	$\mathbf{62.8}_{\pm \mathbf{0.7}}$	$\mathbf{17.7}_{\pm \mathbf{0.2}}$	$\mathbf{32.2}_{\pm \mathbf{0.7}}$	$\mathbf{27.2}_{\pm \mathbf{0.2}}$	$\mathbf{25.8}_{\pm \mathbf{0.1}}$	$\mathbf{19.0}_{\pm \mathbf{0.3}}$	$\mathbf{24.1}_{\pm \mathbf{0.3}}$	$\mathbf{36.5}_{\pm \mathbf{0.1}}$	$\mathbf{35.3}_{\pm \mathbf{0.1}}$

Table 2: Comparison with SOTA methods: SPOT on MS COCO 2017 (images) and VideoSAUR on YTVIS-HQ (videos). All models use a DINOv2 ViT(s/14) backbone. The input resolution is 256×256 (224×224).

	COCO #slot=7			
	ARI	FG-ARI	mBO	mIoU
SPOT	$20.3_{\pm 0.7}$	$41.1_{\pm 0.3}$	$30.4_{\pm 0.1}$	$\mathbf{29.0}_{\pm \mathbf{0.9}}$
MetaSlot$_{\text{Tfd9}}$	$\mathbf{23.1}_{\pm \mathbf{0.2}}$	$\mathbf{41.2}_{\pm \mathbf{0.3}}$	$\mathbf{30.5}_{\pm \mathbf{0.3}}$	$28.6_{\pm 0.8}$

	YTVIS(HQ) #slot=7			
	ARI	FG-ARI	mBO	mIoU
VideoSAUR	$33.0_{\pm 0.6}$	$49.0_{\pm 0.9}$	$30.8_{\pm 0.4}$	$\mathbf{30.1}_{\pm \mathbf{0.6}}$
MetaSlot-VideoSAUR	$\mathbf{60.0}_{\pm \mathbf{2.3}}$	$\mathbf{67.3}_{\pm \mathbf{2.1}}$	$\mathbf{33.7}_{\pm \mathbf{0.8}}$	$28.3_{\pm 0.7}$

representations using OCL. Each slot is tasked with predicting the object category labels and bounding box coordinates via a small MLP. We evaluate classification performance using top-1 accuracy of the category labels and assess regression performance using the R^2 score of the bounding box coordinates.

Table 3 shows that the proposed MetaSlot (i.e., MetaSlot$_{Mlp}$) consistently outperforms the baseline [46] in both object classification and bounding box regression tasks. These results indicate that the object representations captured by MetaSlot are superior, effectively improving the encoding of both categorical and spatial information.

Table 3: Set prediction performance

COCO	class labels	bounding boxes
#slot=7	top1↑	top2↑
DINOSAUR + MLP	0.33	0.54
MetaSlot$_{\text{MLP}}$ + MLP	**0.36**	**0.56**

Table 4: Aggregator comparison.

	COCO #slot=7			
	ARI	FG-ARI	mBO	mIoU
Slot Attention	$17.2_{\pm 0.2}$	$38.6_{\pm 0.6}$	$27.7_{\pm 0.4}$	$26.5_{\pm 0.3}$
BO-QSA	$18.2_{\pm 1.0}$	$35.0_{\pm 1.2}$	$28.3_{\pm 0.5}$	$26.9_{\pm 0.5}$
MetaSlot	$\mathbf{22.4}_{\pm \mathbf{0.3}}$	$\mathbf{40.3}_{\pm \mathbf{0.5}}$	$\mathbf{29.5}_{\pm \mathbf{0.2}}$	$\mathbf{27.9}_{\pm \mathbf{0.2}}$

4.3 Interpretability Analysis

Kori et al. [55] interpret Slot Attention (SA) [11] as a Gaussian Mixture Model (GMM), where each slot acts as a Gaussian component explaining a subset of pixels. Building on this view, we shift the focus from concrete objects to abstract prototypes, positing that the real-world distribution can likewise be factorized into a mixture of such prototypes. Concrete objects in the feature map can then be regarded as samples or projections from these prototype distributions. In slot-attention-based object-centric learning, matching these abstract prototypes manifests as slot initialization, while the iterative attention updates project and refine the prototype distributions onto concrete visual appearances.

In SA, all slots are sampled from a shared Gaussian prior, which lacks object-level inductive cues during initialization. BO-QSA [29] mitigates this limitation by assigning independent Gaussian priors to each slot, thereby promoting greater diversity and improving object-attribute binding. However, its fixed slot count limits its adaptability to the diverse objects encountered in real-world scenes. To address this, our proposed MetaSlot introduces a set of adaptive prototype slots that capture abstract representations of real-world entities, further enhancing object binding and improving flexibility in complex scenes.

As shown in Fig. 3, MetaSlot's prototype-based initialization yields slots with pronounced semantic consistency and strong object binding—e.g., all slots from prototype 268 correspond to "keyboard"

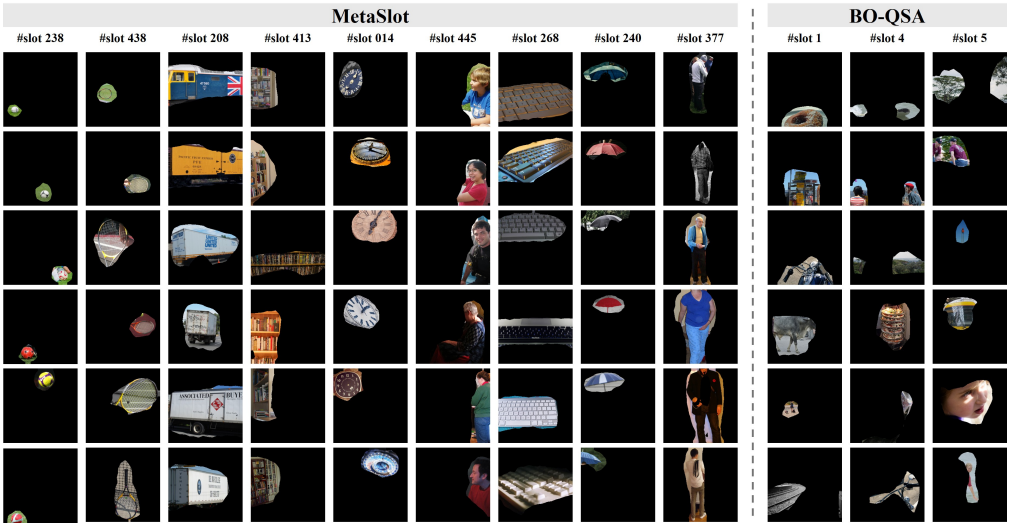


Figure 3: We visualize slot representations initialized from different prototype slots on the COCO dataset, where each column shows a specific initialization slot index–prototype slots in MetaSlot (out of 512) and fixed slots in BO-QSA [29]. MetaSlot's prototype-based initialization yields slots with strong semantic consistency and object binding (e.g., #slot 208 for trucks, #slot 445 for persons). By comparison, the fixed slots in BO-QSA frequently lack such coherent semantic grouping.

objects, while those from prototype 240 correspond to "umbrella" objects. In contrast, BO-QSA, limited by its fixed number of slots, struggles to achieve such fine-grained prototype binding. Additional results on the VOC dataset are reported in Appendix B. Quantitative comparisons under the DINOSAUR [46] decoding framework (Table 4) further confirm that MetaSlot surpasses both SA and BO-QSA on all object discovery metrics, highlighting its superiority in producing disentangled and interpretable slot representations.

4.4 Ablations

To assess the effectiveness of the key architectural components of MetaSlot, we perform a comprehensive ablation study on the MS COCO 2017 dataset. All experiments adopt the DINOSAUR framework with a DINOv2 ViT (s/14) encoder and fix the number of slots to seven. To validate the contribution of individual design choices, we further evaluate two ablated variants: 'MetaSlot w/o noise' omits progressively attenuated noise during slot updates. As shown in Fig.4, injecting noise leads to a lower Adjusted Rand Index (ARI, left) and a higher mean best overlap (mBO, right), implying faster and more stable slot aggregation. 'MetaSlot w/o mask' disables the prototype-based masking strategy. Table 5 summarizes performance across three metrics–Foreground ARI (FG-ARI), mean best overlap (mBO), and mean Intersection-over-Union (mIoU), indicating that each component is pivotal for precise slot-to-object alignment and effective spatial disentanglement. In addition, Appendix E presents an analysis of the codebook prototype size, where we observe that increasing the number of prototypes yields only marginal improvements in performance. We also compare MetaSlot models with varying slot counts, and the empirically optimal number of slots aligns with the long-established consensus within the object-centric learning community. Furthermore, we include additional ablations on architectural components, which provide further evidence for the robustness and effectiveness of our model design.

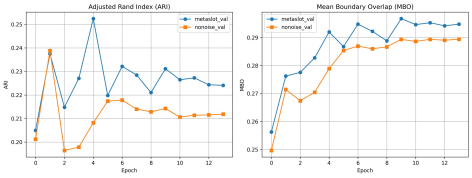
5 Conclusion

This paper introduces MetaSlot, a novel aggregator for object-centric learning (OCL) that addresses two long-standing limitations of conventional Slot Attention models: the fixed number of slots and reliance on random initialization. MetaSlot incorporates a global vector-quantized (VQ) prototype codebook alongside a two-stage aggregate-and-deduplicate framework. This design enables the model to adaptively adjust the number of slots based on scene complexity and to initialize slots

Table 5: Ablation study on architectural components. Backbone: DINOv2 ViT (s/14).

	COCO #slot=7		
	FG-ARI	mBO	mIoU
MetaSlot w/o noise	$39.4_{\pm 0.3}$	$28.9_{\pm 0.4}$	$27.4_{\pm 0.4}$
MetaSlot w/o mask	$38.2_{\pm 0.2}$	$28.5_{\pm 0.1}$	$26.9_{\pm 0.3}$
MetaSlot	$\mathbf{40.3}_{\pm \mathbf{0.5}}$	$\mathbf{29.5}_{\pm \mathbf{0.2}}$	$\mathbf{27.9}_{\pm \mathbf{0.2}}$

Figure 4: Training curves for $\text{MetaSlot}_{\text{Mlp}}$ with/without progressively attenuated noise.



with semantically meaningful object representations. Extensive experiments show that MetaSlot consistently achieves substantial gains across a range of OCL tasks and offers a robust foundation for future research in OCL and its downstream applications.

6 Limitations and Future Directions

Despite its promising results, MetaSlot still faces several limitations that point to fruitful directions for future research. First, the use of absolute positional encoding inherited from the original Slot Attention makes the model non-equivariant to image translations, potentially causing the codebook prototypes to capture position-dependent noise patterns. Future work could leverage translation-equivariant mechanisms such as ISA [74] to promote the emergence of consistent, position-agnostic representations. Second, current object prototypes primarily encode global shape or semantic information while overlooking finer-grained attribute compositions. Enhancing prototype optimization through compositional generalization—that is, constructing compound prototypes integrating multiple attribute-level features—may yield richer and more discriminative object cues for downstream tasks. Third, owing to MetaSlot's two-stage and iterative optimization design, the framework inherently contains self-supervisory signals. Exploring ways to identify the semantically complete object slots emerging in the second stage and using them to provide weak supervision for the aggregation process in the first stage could further advance the development of variable-slot object-centric learning. Finally, MetaSlot has yet to be extensively evaluated under out-of-distribution (OOD) conditions. Systematic studies across diverse domains, coupled with efforts to better align the learned codebook prototype distributions with real-world object distributions, could further improve the model's cross-sample and cross-domain generalization.

Acknowledgments and Disclosure of Funding

We acknowledge the support of Finnish Center for Artificial Intelligence (FCAI), Research Council of Finland flagship program. We thank the Research Council of Finland for funding the projects ADEREHA (grant no. 353198). We also appreciate CSC - IT Center for Science, Finland, for granting access to supercomputers Mahti and Puhti, as well as LUMI, owned by the European High Performance Computing Joint Undertaking (EuroHPC JU) and hosted by CSC Finland in collaboration with the LUMI consortium. Furthermore, we acknowledge the computational resources provided by the Aalto Science-IT project through the Triton cluster.

References

[1] E. S. Spelke and K. D. Kinzler, "Core knowledge," *Developmental science*, vol. 10, no. 1, pp. 89–96, 2007.

[2] N. Le Roux, N. Heess, J. Shotton, and J. Winn, "Learning a generative model of images by factoring appearance and shape," *Neural Computation*, vol. 23, no. 3, pp. 593–650, 2011.

[3] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π0: A vision-language-action flow model for general robot control, 2024," *URL https://arxiv. org/abs/2410.24164*.

[4] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong *et al.*, "Openvla: An open-source vision-language-action model," in *8th Annual Conference on Robot Learning*.

[5] Y. Zheng, L. Yao, Y. Su, Y. Zhang, Y. Wang, S. Zhao, Y. Zhang, and L.-P. Chau, "A survey of embodied learning for object-centric robotic manipulation," *arXiv preprint arXiv:2408.11537*, 2024.

[6] P. N. Johnson-Laird, "Mental models and human reasoning," *Proceedings of the National Academy of Sciences*, vol. 107, no. 43, pp. 18 243–18 250, 2010.

[7] K. Stenning and M. Van Lambalgen, *Human reasoning and cognitive science*. MIT Press, 2012.

[8] T. Wiedemer, P. Mayilvahanan, M. Bethge, and W. Brendel, "Compositional generalization from first principles," *Advances in Neural Information Processing Systems*, vol. 36, pp. 6941–6960, 2023.

[9] M. Okawa, E. S. Lubana, R. Dick, and H. Tanaka, "Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task," *Advances in Neural Information Processing Systems*, vol. 36, pp. 50 173–50 195, 2023.

[10] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum, "Clevrer: Collision events for video representation and reasoning," *arXiv preprint arXiv:1910.01442*, 2019.

[11] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, "Object-centric learning with slot attention," *Advances in neural information processing systems*, vol. 33, pp. 11 525–11 538, 2020.

[12] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," *arXiv preprint arXiv:1904.12584*, 2019.

[13] S. Lachapelle, P. Rodriguez, Y. Sharma, K. E. Everett, R. Le Priol, A. Lacoste, and S. Lacoste-Julien, "Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ica," in *Conference on Causal Learning and Reasoning*. PMLR, 2022, pp. 428–484.

[14] J. Tenenbaum, "Building machines that learn and think like people," in *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018, p. 5.

[15] M. Engelcke, A. R. Kosiorek, O. P. Jones, and I. Posner, "Genesis: Generative scene inference and sampling with object-centric latent representations," *arXiv preprint arXiv:1907.13052*, 2019.

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

[17] A. Kori, F. Locatello, F. D. S. Ribeiro, F. Toni, and B. Glocker, "Grounded object-centric learning," in *The Twelfth International Conference on Learning Representations*.

[18] M. Chang, T. Griffiths, and S. Levine, "Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 694–32 708, 2022.

[19] S. Löwe, P. Lippe, M. Rudolph, and M. Welling, "Complex-valued autoencoders for object discovery," *arXiv preprint arXiv:2204.02075*, 2022.

[20] S. Löwe, P. Lippe, F. Locatello, and M. Welling, "Rotating features for object discovery," *Advances in Neural Information Processing Systems*, vol. 36, pp. 59 606–59 635, 2023.

[21] 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," in *International Conference on Learning Representations*.

[22] G. Singh, F. Deng, and S. Ahn, "Illiterate dall-e learns to compose," *arXiv preprint arXiv:2110.11405*, 2021.

[23] G. Singh, Y.-F. Wu, and S. Ahn, "Simple unsupervised object-centric learning for complex and naturalistic videos," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 181–18 196, 2022.

[24] A. Villar-Corrales and S. Behnke, "Playslot: Learning inverse latent dynamics for controllable object-centric video prediction and planning," *arXiv preprint arXiv:2502.07600*, 2025.

[25] M. Mosbach, J. N. Ewertz, A. Villar-Corrales, and S. Behnke, "Sold: Reinforcement learning with slot object-centric latent dynamics," *arXiv preprint arXiv:2410.08822*, 2024.

[26] Y.-J. Song, H. Kim, S. Choi, J.-H. Kim, and B.-T. Zhang, "Learning object motion and appearance dynamics with object-centric representations," in *Causal Representation Learning Workshop at NeurIPS 2023*.

[27] J. Brady, R. S. Zimmermann, Y. Sharma, B. Schölkopf, J. Von Kügelgen, and W. Brendel, "Provably learning object-centric representations," in *International Conference on Machine Learning*. PMLR, 2023, pp. 3038–3062.

[28] A. Mansouri, J. Hartford, Y. Zhang, and Y. Bengio, "Object centric architectures enable efficient causal representation learning," in *The Twelfth International Conference on Learning Representations*.

[29] B. Jia, Y. Liu, and S. Huang, "Improving object-centric learning with query optimization," in *ICLR*, 2023.

[30] R. Gray, "Vector quantization," *IEEE Assp Magazine*, vol. 1, no. 2, pp. 4–29, 1984.

[31] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu, "Vector-quantized image modeling with improved vqgan," *arXiv preprint arXiv:2110.04627*, 2021.

[32] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, "Vector quantized diffusion model for text-to-image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 696–10 706.

[33] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[34] C. Plato, "The republic," in *Democracy: A Reader*. Columbia University Press, 2016, pp. 229–233.

[35] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[36] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[37] G. 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," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 940–28 954, 2022.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[40] N. Watters, L. Matthey, C. P. Burgess, and A. Lerchner, "Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes," *arXiv preprint arXiv:1901.07017*, 2019.

[41] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*.

[42] Z. Wu, J. Hu, W. Lu, I. Gilitschenski, and A. Garg, "Slotdiffusion: Object-centric generative modeling with diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 50 932–50 958, 2023.

[43] J. Jiang, F. Deng, G. Singh, and S. Ahn, "Object-centric slot diffusion," in *The Thirty-seventh Conference on Neural Information Processing Systems, NeurIPS 2023*. The Conference on Neural Information Processing Systems, 2023.

[44] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[45] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

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

[47] A. R. Didolkar, A. Zadaianchuk, A. Goyal, M. C. Mozer, Y. Bengio, G. Martius, and M. Seitzer, "On the transfer of object-centric representation learning," in *The Thirteenth International Conference on Learning Representations*, 2025.

[48] R. Zhao, V. Wang, J. Kannala, and J. Pajarinen, "Vector-quantized vision foundation models for object-centric learning," *arXiv preprint arXiv:2502.20263*, 2025.

[49] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner, "Multi-object representation learning with iterative variational inference," in *International conference on machine learning*. PMLR, 2019, pp. 2424–2433.

[50] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[51] O. Biza, S. Van Steenkiste, M. S. Sajjadi, G. F. Elsayed, A. Mahendran, and T. Kipf, "Invariant slot attention: object discovery with slot-centric reference frames," in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 2507–2527.

[52] G. Aydemir, W. Xie, and F. Guney, "Self-supervised object-centric learning for videos," *Advances in Neural Information Processing Systems*, vol. 36, pp. 32 879–32 899, 2023.

[53] K. Fan, Z. Bai, T. Xiao, T. He, M. Horn, Y. Fu, F. Locatello, and Z. Zhang, "Adaptive slot attention: Object discovery with dynamic slot number," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 062–23 071.

[54] A. Mansouri, J. Hartford, Y. Zhang, and Y. Bengio, "Object-centric architectures enable efficient causal representation learning," *arXiv preprint arXiv:2310.19054*, 2023.

[55] A. Kori, F. Locatello, A. Santhirasekaram, F. Toni, B. Glocker, and F. De Sousa Ribeiro, "Identifiable object-centric representation learning via probabilistic slot attention," *Advances in Neural Information Processing Systems*, vol. 37, pp. 93 300–93 335, 2024.

[56] J. Brady, J. von Kügelgen, S. Lachapelle, S. Buchholz, T. Kipf, and W. Brendel, "Towards object-centric learning with general purpose architectures," in *NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward*.

[57] A. R. Didolkar, A. Goyal, and Y. Bengio, "Cycle consistency driven object discovery," in *The Twelfth International Conference on Learning Representations*.

[58] Z. Wang, M. Z. Shou, and M. Zhang, "Object-centric learning with cyclic walks between parts and whole," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023, pp. 9388–9408.

[59] F. Kapl, A. M. K. Mamaghan, M. Horn, C. Marr, S. Bauer, and A. Dittadi, "Object-centric representations generalize better compositionally with less compute," in *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025.

[60] S. Ferraro, P. Mazzaglia, T. Verbelen, and B. Dhoedt, "Focus: Object-centric world models for robotic manipulation," *Frontiers in Neurorobotics*, vol. 19, p. 1585386, 2025.

[61] D. Haramati, T. Daniel, and A. Tamar, "Entity-centric reinforcement learning for object manipulation from pixels," in *The Twelfth International Conference on Learning Representations*.

[62] K. Kahatapitiya, A. Karjauv, D. Abati, F. Porikli, Y. M. Asano, and A. Habibian, "Object-centric diffusion for efficient video editing," in *European Conference on Computer Vision*. Springer, 2024, pp. 91–108.

[63] A. Rubinstein, A. Prabhu, M. Bethge, and S. J. Oh, "Are we done with object-centric learning?" in *Workshop on Spurious Correlation and Shortcut Learning: Foundations and Solutions*.

[64] A. Didolkar, A. Zadaianchuk, A. Goyal, M. Mozer, Y. Bengio, G. Martius, and M. Seitzer, "Zero-shot object-centric representation learning," *arXiv preprint arXiv:2408.09162*, 2024.

[65] G. Singh, S. Ahn, and Y. Kim, "Neural systematic binder," in *The Eleventh International Conference on Learning Representations, ICLR2023*. The International Conference on Learning Representations (ICLR), 2023.

[66] Y.-F. Wu, M. Lee, and S. Ahn, "Neural language of thought models," in *The Twelfth International Conference on Learning Representations*. The International Conference on Learning Representations (ICLR), 2024.

[67] L. Karazija, I. Laina, and C. Rupprecht, "Clevrtex: A texture-rich benchmark for unsupervised multi-object segmentation," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[68] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 740–755.

[69] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, pp. 98–136, 2015.

[70] L. Ke, H. Ding, M. Danelljan, Y.-W. Tai, C.-K. Tang, and F. Yu, "Video mask transfiner for high-quality video instance segmentation," in *European Conference on Computer Vision*. Springer, 2022, pp. 731–747.

[71] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[72] A. Zadaianchuk, M. Seitzer, and G. Martius, "Object-centric learning for real-world videos by predicting temporal feature similarities," *Advances in Neural Information Processing Systems*, vol. 36, pp. 61 514–61 545, 2023.

[73] I. Kakogeorgiou, S. Gidaris, K. Karantzalos, and N. Komodakis, "Spot: Self-training with patch-order permutation for object-centric learning with autoregressive transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 22 776–22 786.

[74] O. Biza, S. Van Steenkiste, M. S. Sajjadi, G. F. Elsayed, A. Mahendran, and T. Kipf, "Invariant slot attention: Object discovery with slot-centric reference frames," in *International Conference on Machine Learning*. PMLR, 2023, pp. 2507–2527.

[75] Z. Bao, P. Tokmakov, Y.-X. Wang, A. Gaidon, and M. Hebert, "Object discovery from motion-guided tokens," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 972–22 981.

[76] A. Manasyan, M. Seitzer, F. Radovic, G. Martius, and A. Zadaianchuk, "Temporally consistent object-centric learning by contrasting slots," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5401–5411.

[77] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, pp. 193–218, 1985.

[78] madebyollin, "taesd," https://github.com/madebyollin/taesd, 2025, accessed: 2025-04-27.

A Method

Algorithm 1: MetaSlot.

Input: input features $input$, learnable queries $init$, number of iterations T
Output: object-centric representation S^{final}
Modules: stop gradient module $\mathrm{SG}(\cdot)$, slot attention module $\mathrm{SA}(\cdot,\cdot)$, masked slot attention module $\mathrm{MSA}(\cdot,\cdot,\cdot)$, vector quantization $\mathrm{VQ}(\cdot)$, prune duplicate slots module $\mathrm{Prune}(\cdot)$, update prototype codebook module $\mathrm{Update}(\cdot,\cdot)$, noise injection module $\mathrm{Noisy}(\cdot,\cdot)$

 # First-Stage Aggregation :

$S^{mid} \leftarrow init.detach()$;
for $t = 1$ **to** T **do**
  $S^{mid} \leftarrow \mathrm{SA}(S^{mid}, inputs)$;
$\hat{S}, idx \leftarrow \mathrm{VQ}(S^{mid})$;
$\hat{S}_{mask}, mask \leftarrow \mathrm{Prune}(\hat{S}, idx)$;

 # Second-Stage Aggregation :

$S^{final} \leftarrow \hat{S}_{mask}$;
for $t = 1$ **to** $T - 1$ **do**
  $\boldsymbol{input}_{noisy} \leftarrow \mathrm{Noisy}(input, t)$;
  $S^{final} \leftarrow \mathrm{MSA}(S^{final}, input_{noisy}, mask)$;
$S^{final} \leftarrow \mathrm{SG}(S^{final}) + init - \mathrm{SG}(init)$;
$\boldsymbol{input}_{noisy} \leftarrow \mathrm{Noisy}(input, T)$;
$S^{final} \leftarrow \mathrm{MSA}(S^{final}, \boldsymbol{input}_{noisy}, mask)$;

 # Prototype update :

$\mathrm{Update}(\mathrm{SG}(S^{final}), mask)$;
return S^{final}

B Visualization of prototype slots

As shown in Fig. 5, we visualize the refined slots S^{final} corresponding to the initialization prototype slots on the VOC dataset. The codebook contains 512 object prototypes in total. However, since the VOC-trained model is relatively limited in object diversity and scale, many prototype slots receive few or no refined slots assigned to them, making comprehensive visualization infeasible. Therefore, for practical and technical reasons, we focus on the top 20 most active prototype slots—defined as those associated with the largest number of refined slots in the model trained on the VOC dataset. For each selected prototype slot, we randomly sample six refined slots from its assigned set and visualize their corresponding image patches. As shown in the figure, the results clearly demonstrate that the prototype slots exhibit strong concept binding behavior, with refined slots consistently aligned to semantically coherent object categories. These findings are consistent with our theoretical expectations regarding the semantic consistency induced by prototype-guided initialization.

C Experimental Details

Implementation and Reproducibility. To ensure a fair comparison, we re-implemented all baseline models from scratch rather than relying on publicly reported results. Throughout all experiments, we kept data augmentation strategies, the visual feed-forward module (VFM) in the OCL encoder—based on DINOv2 ViT-s/14 [50]—and all training hyperparameters identical to those reported in the original papers. Furthermore, we replaced each model's original variational autoencoder (VAE) component with a large-scale pre-trained TAESD module [78], which is based on Stable Diffusion.

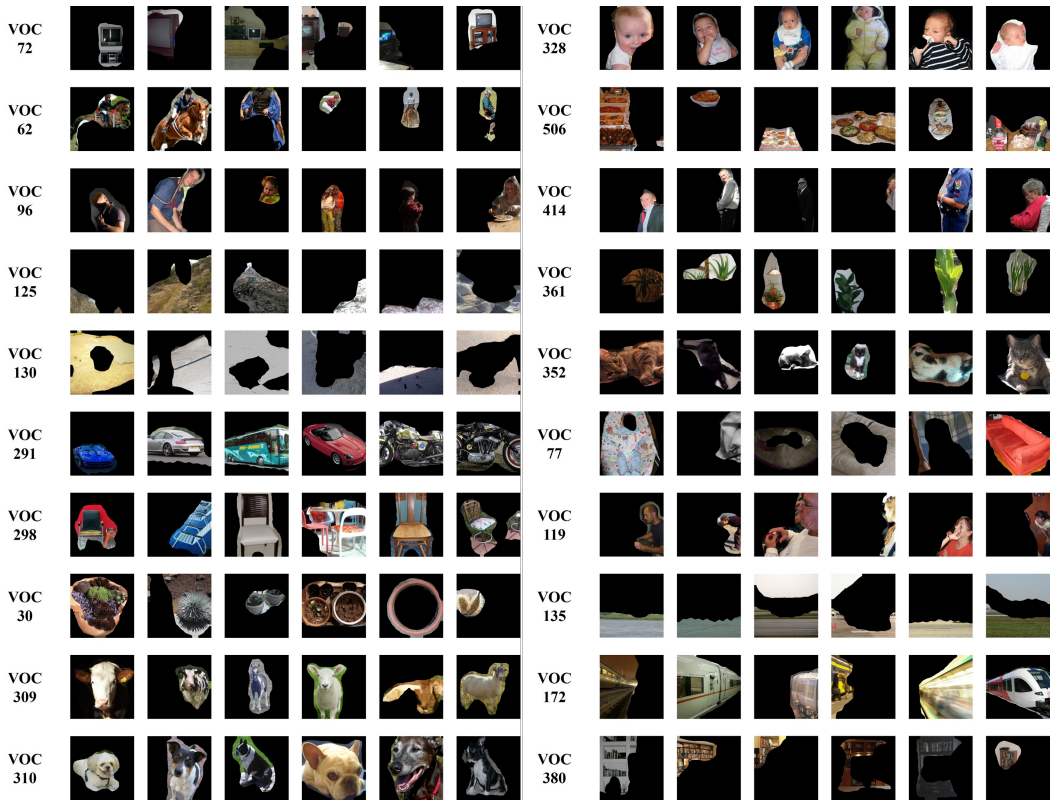


Figure 5: Visualize slot representations initialized from different prototype slots on the VOC dataset.

All models—including both the MetaSlot-augmented variants and their respective baselines—were trained using the Adam optimizer [71] on a single NVIDIA V100 GPU with 16-bit mixed precision. Each run consisted of 50000 training steps, with a batch size of 32 and four data-loading workers. We set the initial learning rate to 2×10^{-4} and maintained it throughout training. For the MetaSlot module, the codebook size was fixed at 512, the feature map resolution at 256×256, and the embedding dimension at 256. The number of slots for each model remained consistent with its original baseline configuration. To reduce the impact of randomness, all experiments were repeated with three different random seeds, and we report the mean results.

Pre-trained VQ-VAE Configuration for SLATE and Slot Diffusion. For the SLATE and Slot Diffusion baselines, we adhered to the standard VQ-VAE implementation based on ResNet-18. Specifically, we used a codebook size of 4096 and an embedding dimension of 256. Pre-training was conducted for 30000 steps with a batch size of 64 shared between the text and vision branches, four data-loading workers, and an initial learning rate of 2×10^{-3}. The feature map resolution remained 256×256. As before, each configuration was evaluated over three independent runs, and the reported metrics represent the averaged performance across these trials.

D Supplementary Ablation Experiments

Impact of Codebook Size As shown in table 6, we investigated the impact of different codebook sizes (256, 512, and 1024) on model performance. We observed that codebook size has a limited effect on performance; however, a larger prototype set allows for finer distinctions between slots that share semantic concepts but differ in spatial location. For example, as shown in Fig. 3, slot #445 corresponds to the concept of "person" located on the right side of the image, while slot #337 represents the same concept but appears at the center of the scene.

Impact of Slot Number As shown in table 7, we examined how the number of slots affects final performance. Since the codebook and the aggregator module are jointly optimized, the quality of the

Table 6: Results of MetaSlot with varying codebook sizes on MS COCO.

	DINOSAUR on MS COCO			
	ARI	FG-ARI	mBO	mIoU
MetaSlot, codebook_size=256	**26.7**	38.1	29.4	27.5
MetaSlot, codebook_size=512	22.4	**40.3**	**29.5**	**27.9**
MetaSlot, codebook_size=1024	20.0	40.4	29.2	27.3

Table 7: Results of MetaSlot with varying slot numbers on MS COCO.

	DINOSAUR on MS COCO			
	ARI	FG-ARI	mBO	mIoU
MetaSlot, slot_num=5	**25.5**	37.8	29.3	27.4
MetaSlot, slot_num=7	22.4	**40.3**	**29.5**	**27.9**
MetaSlot, slot_num=11	19.8	38.7	28.4	26.9
MetaSlot, slot_num=15	15.9	36.0	27.2	26.0

aggregator—particularly in early training stages—can significantly influence codebook convergence. We found that the empirically optimal slot count aligns well with long-standing choices commonly adopted in the community.

Impact of Architectural Components As shown in table 8, we conducted additional experiments to gain deeper insight into the working mechanism of MetaSlot. Specifically, we tested two variants: (1) removing the prototype guidance and instead initializing slots in the second stage by sampling from separate Gaussian distributions, as done in the first stage; and (2) disabling the reactivation mechanism for stale (dead) prototypes during the codebook update process. The results confirm the effectiveness of our module design and are consistent with the theoretical assumptions proposed in the paper.

Furthermore, prior studies in object-centric learning have consistently shown that simply increasing the number of Slot Attention iterations does not lead to meaningful performance gains. To validate this observation, we performed an additional ablation study on the BO-QSA model by increasing its iteration count to 6, matching that of MetaSlot. As shown in the results, increasing the iteration count alone does not improve BO-QSA's performance, further highlighting the importance of our design choices beyond iteration depth.

Table 8: Supplementary ablation on architectural components.

	MS COCO #slot=7		
	FG-ARI	mBO	mIoU
MetaSlot w/o proto	40.2	29.1	27.6
MetaSlot w/o prune	40.0	29.1	27.5
MetaSlot	**40.3**	**29.5**	**27.9**
BO-QSA,iter_num=6	37.9	27.7	26.3

Table 9: Comparison with SlotContrast on the MOVi-C dataset.

Method	FG-ARI	mBO
SlotContrast	62.4	30.6
MetaSlot	**63.9**	**35.0**

E Supplementary Comparative Experiments

Comparative Evaluation against SlotContrast As shown in table 10, we further include a comparison with the SlotContrast[76] model on the MOVi-C dataset. In these experiments, MetaSlot does not employ SlotContrast losses or any temporal-specific enhancements. Due to constraints in time and computational resources, we used a batch size of 8 (vs. 64 in SlotContrast), 24 frames per sample, and trained for up to 20,000 steps (vs. 100,000 in SlotContrast). Despite these limitations, MetaSlot still demonstrates notable performance advantages.

It is worth noting that SlotContrast's contrastive loss was originally designed for a fixed number of slots, and extending it to handle variable slot counts would require additional effort. Nevertheless, as discussed earlier, even without incorporating SlotContrast's core contribution—the contrastive

loss—MetaSlot achieves comparable or even superior performance in object discovery tasks. We believe that integrating SlotContrast's contrastive learning objective in the first stage with MetaSlot's dynamic aggregation in the second stage could further enhance performance, particularly for temporal object discovery. However, this would necessitate architectural modifications that are beyond the scope of the current work and are left for future research.

Table 10: Comparison with SlotContrast on the MOVi-C dataset.

Method	FG-ARI	mBO
SlotContrast	62.4	30.6
MetaSlot	**63.9**	**35.0**

Table 11: Comparison with AdaSlot on the MS COCO dataset.

Method	FG-ARI	mBO
AdaSlot	35.6	29.4
BO-QSA	35.0	28.3
MetaSlot	**40.3**	**29.5**

Comparative Evaluation against AdaSlot As shown in Table 11, we further report the results of AdaSlot[53] compared with MetaSlot. All models use a DINOv2 ViT-S/14 backbone, and the input resolution is 256×256 (or 224×224). We speculate that the performance difference is partly due to the fact that, when the number of slots varies, the model can no longer apply separate Gaussian initialization to each slot individually.

Additional Evaluation of VQ-based Object-Centric Methods We further evaluated several object-centric learning methods that employ vector quantization. In SysBinder[65] and NLoTM[66], the quantization mechanisms operate at the attribute level within each slot. However, this design does not necessarily lead to improved representation quality in unsupervised settings. In synthetic datasets, object attributes such as color, size, shape, and material are relatively fixed and can often be described using fewer than ten shared labels. In contrast, real-world datasets such as COCO and VOC contain objects with far more diverse and non-shared attributes. For instance, the object *person* cannot be meaningfully described with the same attribute set as the object *mouse*. This discrepancy likely explains why SysBinder and NLoTM were not originally evaluated on real-world benchmarks.

Furthermore, the initial intuition behind MetaSlot is inspired by Platonic philosophy. In Plato's theory of Forms, sensible particulars are intelligible only insofar as they "participate" (metechē) in a transcendent Form (Phaedo 100c–d). Similarly, Aristotle maintains in the Categories that accidents belong to substances only as predicates of discourse, not as essential ingredients shared by all beings (Categories 2a11–19).

Our empirical evaluation on COCO and ClevrTex shows that MetaSlot substantially outperforms both SysBinder and NLoTM. Although SysBinder achieves a slight advantage in FG-ARI on ClevrTex, it underperforms the DINOSAUR baseline on other key metrics such as ARI, mBO, and mIoU. Moreover, the SysBinder paper only reports FG-ARI and does not include ARI, mBO, or mIoU—metrics widely regarded as standard for evaluating object discovery. All models use a DINOv2 ViT-S/14 backbone, and the input resolution is 256×256 (or 224×224).

Table 12: Comparative results of SysBinder, NLoTM, DINOSAUR, and MetaSlot on MS COCO.

	MS COCO			
	ARI	FG-ARI	mBO	mIoU
SysBinder	16.8	37.7	27.2	26.0
NLoTM	57.8	14.7	23.6	17.8
DINOSAUR	18.2	35.0	28.3	26.9
MetaSlot	**22.4**	**40.3**	**29.5**	**27.9**

Table 13: Comparative results of SysBinder, NLoTM, DINOSAUR, and MetaSlot on ClevrTex.

	ClevrTex			
	ARI	FG-ARI	mBO	mIoU
SysBinder	21.7	**91.4**	46.8	46.5
NLoTM	21.5	88.7	44.1	43.1
DINOSAUR	50.7	89.4	53.3	52.8
MetaSlot	**64.6**	89.6	**55.2**	**54.5**

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: All statements in the abstract and introduction accurately reflect the scope and contributions of this paper, which introduces MetaSlot, a novel variant of Slot Attention.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We provide additional discussion of the limitations of our method in Appendix B.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [NA]

Justification: The innovation of our work lies at the practical level, guided by intuition. More rigorous theoretical justification remains an open direction for future research.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide experimental details in the paper and the Appendix to ensure the reproducibility of our work. And we will make the code publicly available.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We use publicly available datasets and provide anonymous links to our project.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide the key experimental setups in the paper.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All reported results are averaged over three random seeds to mitigate stochastic variance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research conducted in the paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the potential societal impacts of our work in the Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: Our work poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: When using code from a third party, we report the source of the code directly with the block of code used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Our anonymous URL includes these new codes, new results, and related documentation.

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: The paper does not use LLMs as an important, original, or non-standard component of the core methods in this research.

    Guidelines:

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