Predicting Video Slot Attention Queries from Random Slot-Feature Pairs

Rongzhen Zhao[1], **Jian Li**[2], **Juho Kannala**[3,4], **Joni Pajarinen**[1]
[1]Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland
[2]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[3]Department of Computer Science, Aalto University, Espoo, Finland
[4]Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland
{rongzhen.zhao, juho.kannala, joni.pajarinen}@aalto.fi, lijian2022@ruc.edu.cn

Abstract

Unsupervised video Object-Centric Learning (OCL) is promising as it enables object-level scene representation and understanding as we humans do. Mainstream video OCL methods adopt a recurrent architecture: An aggregator aggregates current video frame into object features, termed slots, under some queries; A transitioner transits current slots to queries for the next frame. This is an effective architecture but all existing implementations both (*i1*) neglect to incorporate next frame features, the most informative source for query prediction, and (*i2*) fail to learn transition dynamics, the knowledge essential for query prediction. To address these issues, we propose Random Slot-Feature pair for learning Query prediction (RandSF.Q): (*t1*) We design a new transitioner to incorporate both slots and features, which provides more information for query prediction; (*t2*) We train the transitioner to predict queries from slot-feature pairs randomly sampled from available recurrences, which drives it to learn transition dynamics. Experiments on scene representation demonstrate that our method surpass existing video OCL methods significantly, e.g., up to 10 points on object discovery, setting new state-of-the-art. Such superiority also benefits downstream tasks like scene understanding.

Source Code, Model Checkpoints, Training Logs — https://github.com/Genera1Z/RandSF.Q

Introduction

By video Object-Centric Learning (OCL) (Singh, Deng, and Ahn 2022; Zadaianchuk, Seitzer, and Martius 2024), objects in the video can be discovered and represented as respective single feature vectors, termed slots, as well as tracked across frames. This is achieved under self-supervision, by simply forcing the slots to reconstruct the input video frames in some format. Representing a visual scene described by a video as object-level features is not only cognitively plausible as we humans perceive the world similarly (Palmeri and Gauthier 2004; Cavanagh 2011), but also practically feasible as a visual scene evolves upon dynamics among objects. This is why OCL is gaining popularity in applications like scene representing (Locatello et al. 2020), dynamics modeling (Wu et al. 2023a), planning and decision-making (Palmeri and Gauthier 2004; Ferraro et al. 2025) lately.

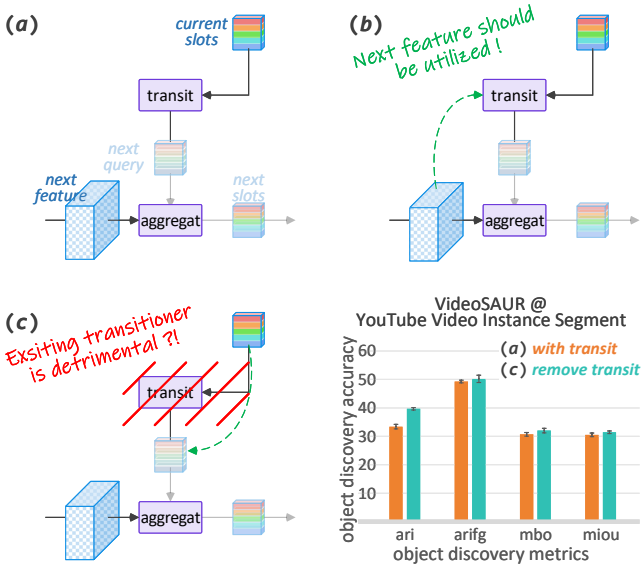

Figure 1: (*a*) Mainstream video OCL adopts a recurrent architecture, where a transitioner transits current slots into the query for next video frame, and an aggregator aggregates the next frame feature into slots under the query. (*b*) Our intuitive observation: To predict next query using the transitioner, next frame feature is already available and very informative thus should also be utilized. (*c*) Our empirical observation: By removing the transitioner and using current slots directly as next query, the algorithm works even better – Existing transitioner is not effectively learned.

Mainstream video OCL methods (Singh, Wu, and Ahn 2022; Kipf et al. 2022) adopt a recurrent architecture, as shown in Figure 1 (*a*). Given some query vectors, the *aggregator*, a Slot Attention (Locatello et al. 2020) module, aggregates current video frame into slots, where each of such feature vectors represents an object. For next frame aggregation, the *transitioner*, typically a Transformer encoder block (Vaswani et al. 2017), transits current slots into the next query to provide temporally consistent representations for objects in dynamically evolving scenes, ensuring coherent

tracking and identity preservation across time. Despite its overwhelmingly wide adoptation, does the query prediction via the transitioner between frames really learn the knowledge to effectively power OCL on videos?

To the best of our knowledge, all existing implementations of this recurrent architecture have not acknowledged or explored the following two issues:

i1 Although visual features of the next frame are already available and are much more informative, existing transitioners predict the next query only based on current (maybe along with previous) slots without leveraging available future information, as shown in Figure 1 (*b*);

i2 Despite the primacy of transition dynamics knowledge in query prediction, existing transitioners lack the inductive bias to learn it, making themselves not only ineffective but even detrimental, as shown in Figure 1 (*c*).

To address the above issues, we propose RandSF.Q, which uses Random Slot-Feature pairs as transition input to enhance the learning of Query prediction. Our method tackles these limitations through two key techniques:

t1 To provide more information for query prediction, we design a new transitioner, which for the first time incorporates both current slots and features of the next video frame as its input, as shown in Figure 2 (*a*);

t2 To enforce the learning of transition dynamics, we train the transitioner to predict query for the next frame from slot-feature pairs that are randomly sampled from available recurrences, as shown in Figure 2 (*b*).

Besides two novel techniques mentioned above, our contributions also include:

- New state-of-the-art on video object discovery tasks, as well as consistent performance boosts on downstream tasks like object recognition and video prediction;
- Straightforward quantification on our transitioner's query prediction capability.

Related Work

In this section, we briefly review existing advances of OCL on video, compare with self-supervised object tracking, and focus on existing query prediction techniques.

Video object-centric learning. Given that Slot Attention has been central to video OCL since its inception, we classify existing methods as classical ones (Jiang et al. 2019; Kosiorek et al. 2018; van Steenkiste et al. 2018; Veerapaneni et al. 2020; Burgess et al. 2019) and modern ones (Kabra et al. 2021; Aydemir, Xie, and Guney 2023; Kipf et al. 2022; Elsayed et al. 2022; Safadoust and Güney 2023; Singh, Wu, and Ahn 2022; Traub et al. 2022; Zadaianchuk, Seitzer, and Martius 2024; Zoran et al. 2021; Qian et al. 2023). Modern methods' performance has improved a lot even on real-world videos under self-supervision. Our work follows this line of research and we omit the word "modern" for simplicity. Video OCL is basically image OCL with recurrent module connecting between frames, i.e., current slots are transited to next query. Like image OCL, decoders, which are mixture-based (Locatello et al. 2020; Seitzer et al. 2023), auto-regressive (Singh, Deng, and Ahn 2022; Kakogeorgiou et al. 2024; Zhao et al. 2025d) or denoising-based (Jiang et al. 2023; Wu et al. 2023b), are also applicable to video OCL; Decoding targets, which are quantized with different inductive biases (Zhao et al. 2025a,b,c), are applicable to video OCL too. Specific to video OCL, temporal intrinsics like temporal prediction (Zadaianchuk, Seitzer, and Martius 2024) and consistency (Aydemir, Xie, and Guney 2023; Manasyan et al. 2025) can be utilized to enhance the performance. We focus on temporal inductive biases.

Unsupervised video object tracking. Unsupervised Video Object Tracking (VOT) resembles (Li et al. 2019; Yuan et al. 2020; Lai, Lu, and Xie 2020; Wang et al. 2021; Xu and Wang 2021; Shen et al. 2022; Li et al. 2022) video OCL very much, because both are query-based and the query/slots should be maintained consistently across frames under self-supervision. However, VOT query only covers interested objects, while video OCL query covers all objects and the background. Essentially, unsupervised VOT only cares about temporal consistency of query representations while video OCL also demand slot representations to be informative enough to represent the scene. Video OCL slots, the sub-symbolic representations, can be directly employed in downstream scene representation and understanding, while VOT representations can not. Comparison with unsupervised VOT methods is of out scope.

Query prediction. It is query prediction that adapts image OCL to video OCL (Singh, Wu, and Ahn 2022). Most video OCL methods employ a single Transformer encoder block to transit current slots into next query, i.e., the naive recurrent architecture of aggreagtion-transition-aggregation... (Zhao et al. 2025c). Well-recognized works like STEVE (Singh, Wu, and Ahn 2022), SAVi (Kipf et al. 2022), SAVi++ (Elsayed et al. 2022), VideoSAUR (Zadaianchuk, Seitzer, and Martius 2024) and SlotContrast (Manasyan et al. 2025) all take such design. There is no transitioner in SOLV (Aydemir, Xie, and Guney 2023) because global slots are aggregated for all frames, which however requires all frames to be available once together. A few works STATM (Li et al. 2025b) and SlotPi (Li et al. 2025a) fuse architectures of video OCL (Kipf et al. 2022) and world model (Wu et al. 2023a) and predict next query as auto-regression, which can be taken as an advanced recurrent architecture. However, none of them acknowledge or explore issues (*i1*) and (*i2*), which are directly addressed by our method.

Proposed Method

In this section, we formulate our RandSF.Q – random slot-feature pair for effective query prediction learning and latest pair for informative prediction. This is realized with a novel transitioner and the corresponding novel training and evaluation strategies upon the recurrent aggregation-transition architecture of mainstream video OCL methods.

Overall Model Architecture

As shown in Figure 2 (*left*), we build our whole method upon SlotContrast (Manasyan et al. 2025), the latest state-of-the-art. Our model consists of an encoder ϕ_{e}, an aggregator ϕ_{a}, a decoder ϕ_{d} and a transitioner ϕ_{t}.

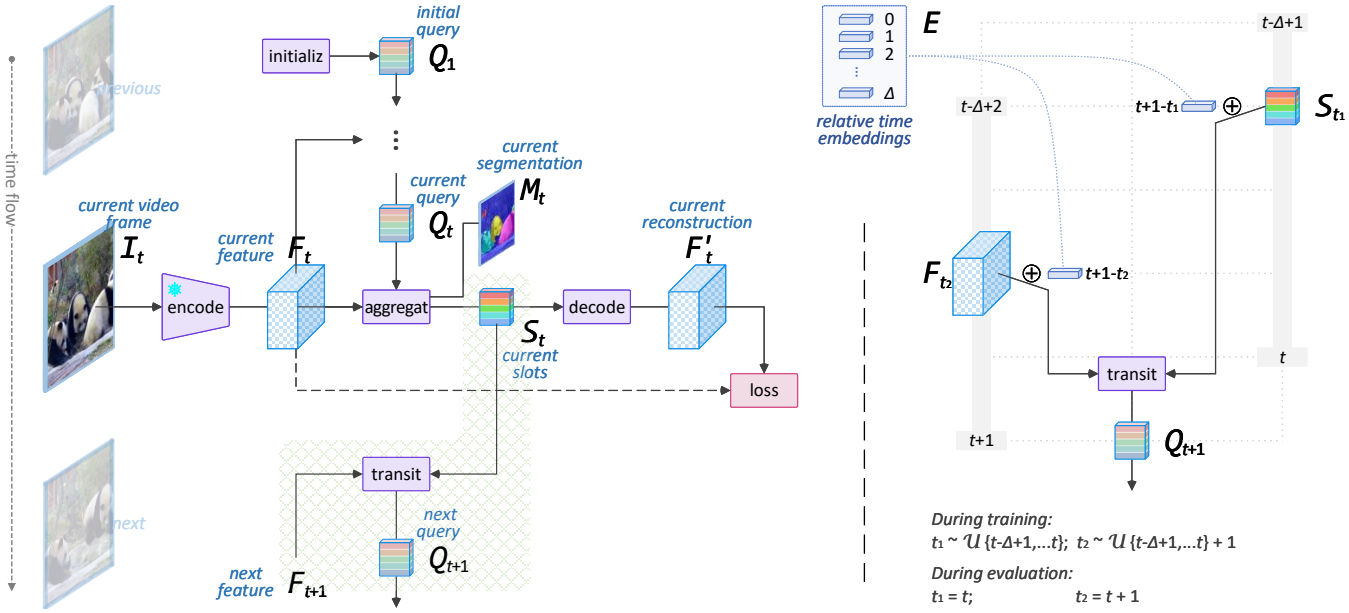

Figure 2: Our model architecture. (*left*) **Our method is built upon SlotContrast** (Manasyan et al. 2025). A frozen DINO2 (Oquab et al. 2023) model *encodes* current video frame $\boldsymbol{I}_t$ into current feature $\boldsymbol{F}_t$; A Slot Attention (Locatello et al. 2020) module *aggregates* $\boldsymbol{F}_t$ into current object-level vectors, slots $\boldsymbol{S}_t$, under current query $\boldsymbol{Q}_t$; A Transformer decoder block (Vaswani et al. 2017) *transits* $\boldsymbol{S}_t$ conditioned on next feature $\boldsymbol{F}_{t+1}$ to next query $\boldsymbol{Q}_{t+1}$; A random Transformer decoder (Zhao et al. 2025d) *decodes* $\boldsymbol{S}_t$ into current reconstruction $\boldsymbol{F}'_t$. The *objective* is minimizing difference between $\boldsymbol{F}_t$ and $\boldsymbol{F}'_t$. (*right*) **How our transitioner works**. To effectively learn transition dynamics, our transitioner explores slots $\boldsymbol{S}_{t_1}$ and feature $\boldsymbol{F}_{t_2}$ at any past time point within window size Δ to predict next query $\boldsymbol{Q}_{t+1}$ during *training*. Relative time embeddings are added to $\boldsymbol{S}_{t_1}$ and $\boldsymbol{F}_{t_2}$ to indicate their offset from $t+1$. To maximize prediction accuracy, our trainsitioner exploits only the most recent slots $\boldsymbol{S}_t$ and feature $\boldsymbol{F}_{t+1}$ to predict $\boldsymbol{Q}_{t+1}$ during *evaluation*. The left sub-figure is adapted from (Zhao et al. 2025c).

At time step t, the **encoder** ϕ_{e} encodes current video frame $\boldsymbol{I}_t \in \mathbb{R}^{h_0 \times w_0 \times c_0}$ into current feature $\boldsymbol{F}_t \in \mathbb{R}^{h \times w \times c}$:

$$\phi_{\mathrm{e}} : \boldsymbol{I}_t \to \boldsymbol{F}_t \tag{1}$$

where ϕ_{e} is parameterized as a DINO2 (Oquab et al. 2023) model, which is a pretrained vision foundation model and is always frozen; h_0, w_0 and c_0 denote the input resolution, i.e., height, width and channel, while h, w and c are that after encoding.

Then the **aggregator** ϕ_{a} aggregates objects and background information scattered in the super-pixels of current feature $\boldsymbol{F}_t$ respectively into corresponding object-level feature vectors, i.e., slots $\boldsymbol{S}_t \in \mathbb{R}^{s \times c}$, by iteratively refining current query vectors $\boldsymbol{Q}_t \in \mathbb{R}^{s \times c}$ and current segmentation masks $\boldsymbol{M}_t \in \mathbb{R}^{h \times w}$:

$$\phi_{\mathrm{a}} : \boldsymbol{Q}_t, \boldsymbol{F}_t \to \boldsymbol{S}_t, \boldsymbol{M}_t \tag{2}$$

where ϕ_{a} is parameterized as typical Slot Attention (Locatello et al. 2020); s is the number of query vectors; $\boldsymbol{M}_t$ is binarized from $\boldsymbol{S}_t$'s attention maps along the slot dimension. Because Slot Attention is a kind of attention, $\boldsymbol{Q}_t$ and $\boldsymbol{S}_t$ always have identical tensor shapes. $\boldsymbol{M}_t$ can be used for object discovery, while $\boldsymbol{S}_t$ can be used for downstream scene representation and understanding.

Besides, the **transitioner** ϕ_{r} transits current slots $\boldsymbol{S}_t$ into the query for next video frame $\boldsymbol{Q}_{t+1} \in \mathbb{R}^{s \times c}$ conditioned on next frame feature $\boldsymbol{F}_{t+1} \in \mathbb{R}^{h \times w \times c}$:

$$\phi_{\mathrm{r}} : \boldsymbol{S}_t, \boldsymbol{F}_{t+1} \to \boldsymbol{Q}_{t+1} \tag{3}$$

where ϕ_{r} is parameterized as a single Transformer decoder block (Vaswani et al. 2017). Aggregator ϕ_{a} and transitioner ϕ_{r} form the aggregation-transition recurrence, which powers OCL through video frames. Please note that **extra specific processes** on $\boldsymbol{S}_t$ and $\boldsymbol{F}_{t+1}$ here are needed during training and evaluation, but are omitted for simplicity. We will detail them in the following two subsections.

Last, the **decoder** ϕ_{d} decodes current slots $\boldsymbol{S}_t$ into the reconstruction $\boldsymbol{F}'_t \in \mathbb{R}^{h \times w \times c}$ of current feature $\boldsymbol{F}_t$:

$$\phi_{\mathrm{d}} : \boldsymbol{S}_t \to \boldsymbol{F}'_t \tag{4}$$

where ϕ_{d} is parameterized as a random auto-regressive Transformer decoder (Zhao et al. 2025d). The input clue needed in such decoding is omitted for simplicity.

The **objective** is to minimize the difference between current prediction $\boldsymbol{F}'_t$ and $\boldsymbol{F}_t$:

$$\arg \min_{\phi_{\mathrm{a}}, \phi_{\mathrm{r}}, \phi_{\mathrm{d}}} \mathrm{MSE}(\{\boldsymbol{F}'_t\}_{t=1}^{T}, \mathrm{sg}(\{\boldsymbol{F}_t\}_{t=1}^{T})) \tag{5}$$

where $\mathrm{MSE}(\cdot, \cdot)$ is mean squared error; $\mathrm{sg}(\cdot)$ is stopping gradient; T is the total number frames in a video. The auxiliary loss (Manasyan et al. 2025) is omitted for simplicity.

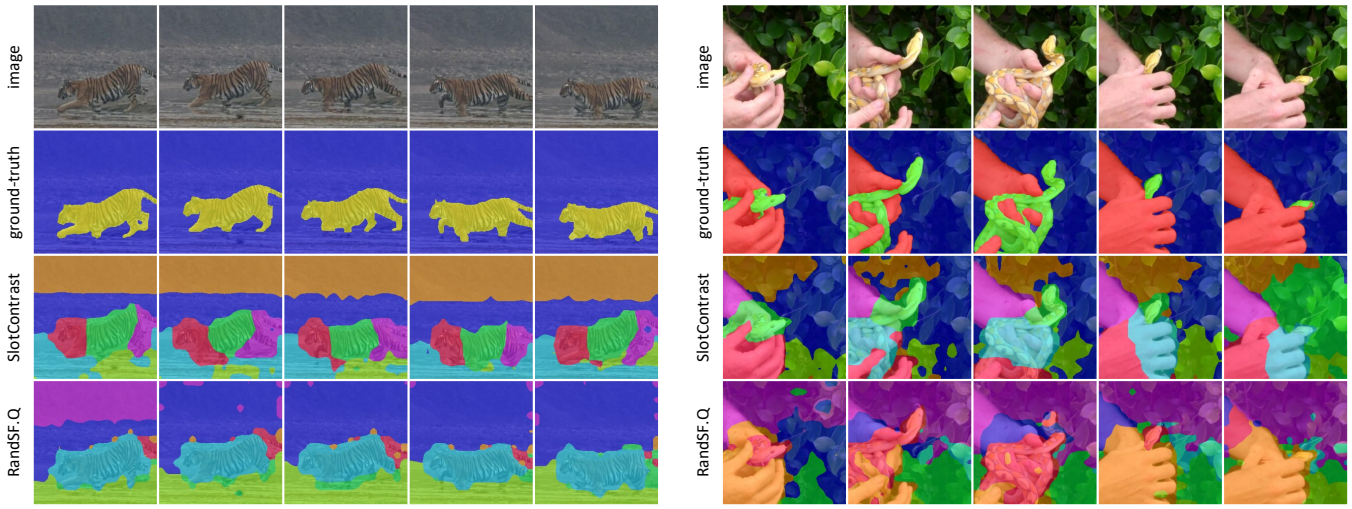

Figure 3: Qualitative results of our RandSF.Q on YTVIS, compared with SotA SlotContrast.

Informative Query Prediction

As shown in Figure 2 (*right*) and Figure 1 (*b*), we introduce a novel technique, which exploits the latest slot-feature pair for more informative query prediction.

During **evaluation**, our simplified description of Equation (3) is actually expanded with relative time embeddings. Our transitioner ϕ_{r} takes current slots $\boldsymbol{S}_t$ plus current relative time embedding $\boldsymbol{E}[1] \in \mathbb{R}^{1\times c}$ as the starting point, and conditions on the already-available next feature $\boldsymbol{F}_{t+1}$ plus the next relative time embedding $\boldsymbol{E}[0] \in \mathbb{R}^{1\times c}$ as the latest incremental information, to predict the next query $\boldsymbol{Q}_{t+1}$:

$$\phi_{\mathrm{r}} : \boldsymbol{S}_t + \boldsymbol{E}[1], \boldsymbol{F}_{t+1} + \boldsymbol{E}[0] \rightarrow \boldsymbol{Q}_{t+1} \tag{6}$$

where $\boldsymbol{E} \in \mathbb{R}^{\Delta\times c}$ is a table of learnable relative time embeddings, with window size $\Delta \in [1, T]$, which we will detail in the following subsection; $\cdot[\cdot]$ is indexing operation.

We assert that next feature $\boldsymbol{F}_{t+1}$ is much more informative than solely current slots $\boldsymbol{S}_t$ or all available slots $\{\boldsymbol{S}_i\}_{i=1}^{t}$ for predicting next query $\boldsymbol{Q}_{t+1}$. This is because, as shown in Equation (2), next feature contains every up-to-date information about next slots and of course next query.

In contrast, all existing transitioners predict the next query only based on available slots. As shown in Figure 1 (*a*) versus (*b*), none of them utilize next feature.

Specifically, most existing video OCL transitioners ϕ_{r}^1 predict next query $\boldsymbol{Q}_{t+1}$ based only on current slots $\boldsymbol{S}_t$:

$$\phi_{\mathrm{r}}^1 : \boldsymbol{S}_t \rightarrow \boldsymbol{Q}_{t+1} \tag{7}$$

where ϕ_{r}^1 is parameterized as a Transformer encoder block (Vaswani et al. 2017; Singh, Wu, and Ahn 2022; Kipf et al. 2022). The transitioner ϕ_{r}^2 of remaining video OCL methods predict from all available slots $\{\boldsymbol{S}_i\}_{i=0}^{t}$:

$$\phi_{\mathrm{r}}^2 : \{\boldsymbol{S}_i\}_{i=1}^{t} \rightarrow \boldsymbol{Q}_{t+1} \tag{8}$$

where ϕ_{r}^2 is parameterized as some multi-layer Transformer encoder variant (Li et al. 2025b,a).

Comment. Compared with existing transitioner ϕ_{r}^1, our slot-feature pair for informative query prediction only introduces a cross attention submodule and a few time embedding parameters. The extra computation overhead in evaluation is negligible relative to the whole model. But compared with existing transitioner ϕ_{r}^2, our design is multi-fold lightweight as we use only one Transformer block while they use a sequential stack of multiple Transformer blocks.

Effective Query Prediction Learning

As shown in Figure 2 (*right*), we introduce another novel technique, which explores random slot-feature pairs to enforce effective query prediction learning.

During **training**, our simplified formulation of Equation (3) is in fact expanded with random sampling from the available recurrences within a time window. Our transitioner ϕ_{r} takes slots $\boldsymbol{S}_{t_1}$ at random time step $t_1 \in [t-\Delta+1, t]$ plus the corresponding relative time embedding $\boldsymbol{E}[t+1-t_1]$, as the starting point, and conditions on feature $\boldsymbol{F}_{t_2}$ at random time step $t_2 \in [t-\Delta+2, t+1]$ plus the corresponding relative time embedding $\boldsymbol{E}[t+1-t_2]$, as the incremental information, to predict next query $\boldsymbol{Q}_{t+1}$:

$$\phi_{\mathrm{r}} : \boldsymbol{S}_{t_1} + \boldsymbol{E}[t+1-t_1], \boldsymbol{F}_{t_2} + \boldsymbol{E}[t+1-t_2] \rightarrow \boldsymbol{Q}_{t+1} \tag{9}$$

where

$$t_1 \sim \mathcal{U}\{t-\Delta+1, ...t\} \tag{10}$$

$$t_2 \sim \mathcal{U}\{t-\Delta+2, ...t+1\} \tag{11}$$

Here $\boldsymbol{E} \in \mathbb{R}^{\Delta\times c}$ is the same variable in Equation (6). We empirically set Δ to be identical to the widely-adopted video clip length in training, e.g., $\Delta = 6$ given video length $T = 24$ (Kipf et al. 2022; Elsayed et al. 2022) and $\Delta = 5$ given video length $T = 20$ (Zhao et al. 2025d).

We deem that a transitioner should be able to predict the query from any starting point slots $\boldsymbol{S}_{t_1}$ that is not too far away, i.e., within a relative small window size Δ, conditioned on the incremental information provided by feature

	MOVi-C #slot=11, conditional				MOVi-D #slot=21, conditional				YTVIS #slot=7, #step=vid_len			
	ARI	ARI_{fg}	mBO	mIoU	ARI	ARI_{fg}	mBO	mIoU	ARI	ARI_{fg}	mBO	mIoU
STEVE	–	–	–	–	$32.7_{\pm 0.2}$	$66.5_{\pm 0.2}$	$23.0_{\pm 0.3}$	$21.2_{\pm 0.3}$	–	–	–	–
VideoSAUR	$41.9_{\pm 1.1}$	$53.3_{\pm 2.1}$	$16.1_{\pm 0.4}$	$14.8_{\pm 0.4}$	$22.5_{\pm 5.0}$	$40.0_{\pm 20.1}$	$11.6_{\pm 6.6}$	$10.8_{\pm 6.1}$	$33.8_{\pm 0.7}$	$49.2_{\pm 0.5}$	$29.9_{\pm 0.4}$	$29.7_{\pm 0.4}$
SlotContrast	$64.6_{\pm 9.4}$	$59.9_{\pm 5.3}$	$27.7_{\pm 3.0}$	$25.8_{\pm 2.9}$	$45.3_{\pm 4.1}$	$63.9_{\pm 0.2}$	$26.7_{\pm 1.0}$	$25.1_{\pm 1.0}$	$37.2_{\pm 0.6}$	$49.4_{\pm 1.1}$	$33.0_{\pm 0.2}$	$32.8_{\pm 0.1}$
RandSF.Q$_{\texttt{tsim}}$	$64.0_{\pm 2.9}$	$66.3_{\pm 1.7}$	$28.4_{\pm 1.3}$	$26.1_{\pm 1.1}$	$41.2_{\pm 2.2}$	$72.0_{\pm 1.1}$	$27.1_{\pm 0.9}$	$25.4_{\pm 0.9}$	$46.0_{\pm 0.7}$	$60.4_{\pm 2.3}$	$39.4_{\pm 0.3}$	$38.5_{\pm 0.2}$
RandSF.Q$_{\texttt{ssc}}$	$65.4_{\pm 10.7}$	$67.4_{\pm 2.1}$	$29.2_{\pm 3.8}$	$26.8_{\pm 3.7}$	$41.6_{\pm 3.7}$	$77.5_{\pm 1.0}$	$27.4_{\pm 1.0}$	$25.6_{\pm 1.0}$	$40.1_{\pm 0.4}$	$58.0_{\pm 1.0}$	$37.6_{\pm 0.4}$	$37.2_{\pm 0.4}$

Table 1: Object discovery on videos. Input resolution is 256×256 (224×224); DINO2 ViT-S/14 is for encoding. On YTVIS, RandSF.Q surpasses SotA SlotContrast by 10+ points by ARI and ARI_{fg}, and ~7 points by mBO and mIoU. Note that `tsim` means using the time similarity loss from VideoSAUR (Zadaianchuk, Seitzer, and Martius 2024), while `ssc` means using the slot-slot contrast loss from SlotContrast (Manasyan et al. 2025).

	YTVIS #slot=7, #step=vid_len			
	class top1↑	top3↑	bbox IoU↑	#match↑
SlotContrast+MLP	$19.9_{\pm 2.0}$	$49.1_{\pm 3.1}$	$53.5_{\pm 0.2}$	$9259_{\pm 26}$
RandSF.Q +MLP	$26.1_{\pm 1.3}$	$60.9_{\pm 3.2}$	$54.5_{\pm 0.6}$	$7579_{\pm 201}$

Table 2: Object recognition on YTVIS.

$\boldsymbol{F}_{t_2}$. Introducing the above-mentioned randomness in training drives the transitioner to grasp such transition dynamics knowledge for better query prediction.

By comparison, existing video OCL transitioners both ϕ_{r}^1 and ϕ_{r}^2 predict the next query naively from current slots, where there is only one time step of difference. As shown in Figure 1 (*a*) versus (*c*), such naive design cannot drive the transitioner to effectively grasp the dynamics for query prediction. Specifically, ϕ_{r}^1 and ϕ_{r}^2 learn query prediction as in Equation (7) and Equation (8), respectively.

Comment. Same as existing transitioners ϕ_{r}^1 and ϕ_{r}^2, our transitioner ϕ_{r} is trained inside the OCL model in an end-to-end way, requiring no extra loss. Namely, our random slot-feature pair enables effective query prediction learning while maintaining architectural elegance. Our computation overhead in training is mostly same as that in evaluation.

Experiment

We comprehensively evaluate our method on object discovery, as well as downstream tasks of object recognition and visual question answering. We also look into how well our method learns transition knowledge for query prediction. We conduct each experiment using three random seeds whenever it is applicable.

Overall Setting

We cover the following baselines and datasets to ensure comprehensive and fair comparisons.

Baselines. STEVE (Singh, Wu, and Ahn 2022) realizes the first Slot Attention-based OCL on videos. VideoSAUR (Zadaianchuk, Seitzer, and Martius 2024) is the first method that utilizes vision foundation model and achieves competitive results in OCL on real-world complex videos. SlotContrast (Manasyan et al. 2025) enables temporal consistency on long videos, setting new state-or-the-art (SotA) recently. Our method RandSF.Q is compared with these methods. Comparing with methods like SOLV (Aydemir, Xie, and Guney 2023), which uses slot pruning, is unfair. So does comparing with SAVi (Kipf et al. 2022) and SAVi++ (Elsayed et al. 2022), which are trained with external stronger supervision like optical flow and depth map.

Datasets. We evaluate these methods on both synthetic and real-world video datasets, following the experiments of SotA method SlotContrast. For synthetic video datasets, we use MOVi-C and MOVi-D of MOVi datasets series[1]. These two subsets contain daily objects with incremental complex textures falling on complex background. These datasets are used for *object discovery*. For real-world video datasets, we use YouTube Video Instance Segmentation[2] (YTVIS) the high quality version[3], which consists of daily-life diverse and complex videos downloaded from YouTube. This dataset is used for both *object discovery* and *object recognition*. We also include the synthetic video-text dataset CLEVRER[4], which consists of geometric objects and paired question texts, testing the multi-modal reasoning capability. This dataset is used for *visual question answering*.

Scene Representation

Video OCL models can be used to represent temporal visual scenes in the object-centric manner. The representation quality can be evaluated by both object discovery, which is achieved by the binarizing slots' attention maps along the slot dimension into object segmentation masks, and object recognition, which is achieved by predicting objects' class and bounding box from corresponding slots. Note that the former is unsupervised while the latter is supervised.

Object discovery. We compare our method RandSF.Q with baselines STEVE, VideoSAUR and SlotContrast in terms of their object discovery performance on datasets MOVi-C, MOVi-D and YTVIS. This is basically an unsupervised object segmentation task. We use multiple recognized metrics for comprehensive measurement: Adjusted Rand In-

[1]https://github.com/google-research/kubric/tree/main/challenges/movi

[2]https://youtube-vos.org/dataset/vis

[3]https://github.com/SysCV/vmt?tab=readme-ov-file#hq-ytvis-high-quality-video-instance-segmentation-dataset

[4]http://clevrer.csail.mit.edu

	CLEVRER #slot=7	
	per option %	per question %
SlotContrast+Aloe	$97.2_{\pm 1.1}$	$95.6_{\pm 0.9}$
RandSF.Q +Aloe	$\underline{98.5}_{\pm 0.8}$	$\underline{96.3}_{\pm 0.7}$

Table 3: Visual question answering on CLEVRER.

dex (ARI)[5] roughly for background segmentation accuracy measurement, ARI foreground (ARI_{fg}) for foreground large objects, mean Best Overlap (mBO) (Uijlings et al. 2013) for best overlapped regions, and mean Intersection over Union (mIoU)[6] as the most strict measurement.

As shown in Table 1, across all datasets and almost all metrics, our RandSF.Q defeats all the baselines. Specifically, on dataset YTVIS, RandSF.Q surpasses the latest SotA SlotContrast by more than 10 points by metrics ARI and ARI_{fg}, and up to 6 points by mBO and mIoU. In contrast, SlotContrast only boost their baseline VideoSAUR limitedly, less than 3 points by metrics mBO and mIoU, or even degrading by ARI and ARI_{fg}.

Object recognition. We compare our method RandSF.Q with baseline SlotContrast in terms of their object recognition performance on dataset YTVIS. This consists of two sub-tasks, i.e., supervised classification and regression from slots. Some literature (Locatello et al. 2020) names this as *set prediction*. We follow (Seitzer et al. 2023) to represent the dataset as slots and train a two-layer MLP to predict the object class and bounding box corresponding to each slot, supervised by the annotations of object class labels and bounding boxes in the dataset. We use top1/top3 accuracy and box IoU score as the class label classification and bounding box regression performance metrics.

As shown in Table 2, our RandSF.Q defeats baseline SlotContrast both in object class label classification and object bounding boxe coordinate regression.

These two aspects of experiments prove our method's superiority in scene especially object representation.

Scene Understanding

Better object representation provides more information for understand and reasoning about the visual scene.

Visual question answering. We compare our method RandSF.Q with baseline SlotContrast in terms of their visual question answering performance on dataset CLEVRER. We follow the data processing and model design of Aloe (Ding et al. 2021; Wu et al. 2023a). We firstly pre-train OCL models on CLEVRER and then freeze them. Each video's frames are all represented as slots, then slots of different frames are added with corresponding time step embeddings; meanwhile, question texts are embedded into vectors and added with corresponding position embeddings. These tokens are fed into Aloe together, appended with a classification token. The output is obtained by projecting the transformed classification token into logits of all answer labels.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html

[6]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.jaccard_score.html

utilizing next feature	✓		✓	✓
sampling slot-feature pair	✓	✓		✓
injecting relative time	✓	✓	✓	
ARI+ARI_{fg}	$108.0_{\pm 1.3}$	$99.7_{\pm 3.2}$	$81.6_{\pm 11.4}$	$64.6_{\pm 4.7}$

sampling window size	2	3	4	5
ARI+ARI_{fg}	$90.0_{\pm 3.3}$	$102.0_{\pm 4.1}$	$107.8_{\pm 0.4}$	$108.0_{\pm 1.3}$

time injecting method	append time emb	sum time emb
ARI+ARIfg	$98.8_{\pm 3.9}$	$108.0_{\pm 1.3}$

Table 4: Ablation study.

As shown in Table 3, our method RandSF.Q defeats baseline SlotContrast in performance measured either by per option accuracy or by per question accuracy. As objects in CLEVRER's synthetic videos are too simple and the baseline performance is already high enough, our method's performance boosts are relatively less significant, compared with those in object discovery and recognition.

Ablation

As shown in Table 4, we ablate the effects of our model designs from the following perspectives.

Utilizing next feature: Whether to utilize next feature for query prediction? Utilizing next feature as condition, along with the input slots as starting point for our transitioner does provide more information for query prediction.

Random sampling slot-feature pair: Whether to randomly sample slot-feature pairs for query prediction learning? Randomly sampling slot-feature pairs from available recurrences ensures better learning of query prediction.

Window size Δ: If sampling, then what window size to use? 2, 3, 4 or 5? Relatively larger window size is better but the performance saturates when window size is equal to the typical video clip size. By the way, window size larger than the video clip size is not practically possible.

Injecting relative time: Whether to inject relative time into our transitioner? Injecting relative time of slots and features respectively is very important to tell the transitioner where is the starting point and where is the incremental information for query prediction.

Injection method: If injecting, then how to inject relative time? Appending the relative time embeddings to slots and features respectively or broadcast summing them to slots and features respectively? Broadcasting and summing time embeddings is obviously better than appending.

Discussion

Let us go back to our core claim: **Does our transitioner truly grasp the knowledge of transition dynamics for query prediction**?

For analysis, we use non-latest slot-feature pairs for query prediction in evaluation, then we count the video OCL performance. Intuitively, non-latest slot-feature pairs contain less up-to-date information thus leading to inferior query prediction and ultimately inferior video OCL performance.

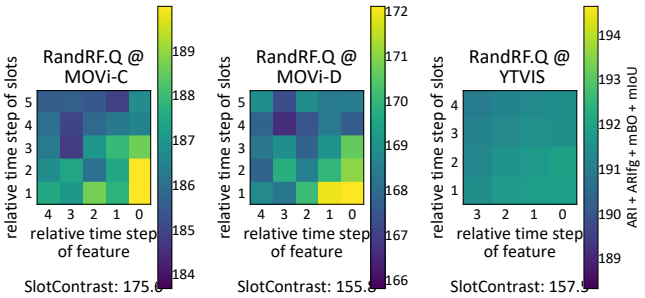

Figure 4: RandSF.Q performance with queries predicted from slot-feature pairs at different relative time steps.

If our transitioner really learns transition dynamics, then it would still predict good enough queries in such cases.

Note that we do not visualize ARI, ARI_{fg}, mBO and mIoU separately. Their values often only reflect some specific aspects of video OCL performance, instead of the overall performance, and also often lack clear regularity. Also note that it is not straightforward to evaluate our query prediction by calculating some distance between the query predicted from non-latest and latest slot-feature pairs, compared with the indirect evaluation scheme described above. In addition, we ensure the color bar is within identical value range sizes for easy comparison. We also put the overall performance of baseline SlotContrast here for simple comparison.

As shown in Figure 4, **the elements of each matrix are roughly brighter on the bottom-right**, compared with those on the top-left. This aligns with our intuition that the more up-to-date the slots and feature being used for the transitioner to predict query, the better the query is and the better our RandSF.Q model performs. Although using non-latest slot-feature pairs for query prediction degrades the performance, **our RandSF.Q still always has significant performance advantage over SlotContrast** even when using the oldest slot-feature pair for query prediction.

Interestingly, the performance matrix of RandSF.Q on dataset YTVIS shows little performance fluctuation compared with those on MOVi-C and D. We explain this as foreground objects in human-shot videos are usually well tracked. This means queries from the near past differ little and thus the query prediction is relatively easier. In contrast, objects in MOVi-C and D drop and bounce chaotically, not as well centered as those in human-shot videos.

Conclusion

Based on our two observations, we propose our method RandSF.Q. We realize informative query prediction by utilizing the next feature and effective query prediction learning by randomly sampling slot-feature pairs from available recurrences as transitioner inputs for query prediction. Our method show significant performance advantages over the latest SotA methods on scene representation and scene understanding. Our core claim is also verified through performance matrix analysis. Our work pushes the video OCL research greatly forward.

Limitations and future work. Our method still cannot solve the long-existing issue of the number of slots often mismatches with the real number of objects in a visual scene. And if introducing those adaptive slot number techniques, our random slot-feature pair sampling technique cannot be directly applied. Sophisticated designs are necessary to combine these two types of techniques.

Acknowledgments

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), BERMUDA (362407) and PROFI7 (352788). 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

Aydemir, G.; Xie, W.; and Guney, F. 2023. Self-supervised object-centric learning for videos. *Advances in Neural Information Processing Systems*, 36: 32879–32899.

Burgess, C. P.; Matthey, L.; Watters, N.; Kabra, R.; Higgins, I.; Botvinick, M.; and Lerchner, A. 2019. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*.

Cavanagh, P. 2011. Visual Cognition. *Vision Research*, 51(13): 1538–1551.

Ding, D.; Hill, F.; Santoro, A.; Reynolds, M.; and Botvinick, M. 2021. Attention over Learned Object Embeddings Enables Complex Visual Reasoning. *Advances in neural information processing systems*, 34: 9112–9124.

Elsayed, G.; Mahendran, A.; Van Steenkiste, S.; et al. 2022. SAVi++: Towards End-to-End Object-Centric Learning from Real-World Videos. *Advances in Neural Information Processing Systems*, 35: 28940–28954.

Ferraro, S.; Mazzaglia, P.; Verbelen, T.; and Dhoedt, B. 2025. FOCUS: Object-Centric World Models for Robotic Manipulation. *Frontiers in Neurorobotics*, 19: 1585386.

Jiang, J.; Deng, F.; Singh, G.; and Ahn, S. 2023. Object-Centric Slot Diffusion. *Advances in Neural Information Processing Systems*.

Jiang, J.; Janghorbani, S.; De Melo, G.; and Ahn, S. 2019. SCALOR: Generative World Models with Scalable Object Representations. In *International Conference on Learning Representations*.

Kabra, R.; Zoran, D.; Erdogan, G.; Matthey, L.; Creswell, A.; Botvinick, M.; Lerchner, A.; and Burgess, C. 2021. Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. *Advances in Neural Information Processing Systems*, 34: 20146–20159.

Kakogeorgiou, I.; Gidaris, S.; Karantzalos, K.; and Komodakis, N. 2024. 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*, 22776–22786.

Kipf, T.; Elsayed, G.; Mahendran, A.; et al. 2022. Conditional Object-Centric Learning from Video. *International Conference on Learning Representations*.

Kosiorek, A.; Kim, H.; Teh, Y. W.; and Posner, I. 2018. Sequential attend, infer, repeat: Generative modelling of moving objects. *Advances in Neural Information Processing Systems*, 31.

Lai, Z.; Lu, E.; and Xie, W. 2020. MAST: A Memory-Augmented Self-Supervised Tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, J.; Han, W.; Lin, N.; Zhan, Y.-L.; Chengze, R.; Wang, H.; Zhang, Y.; Liu, H.; Wang, Z.; Yu, F.; et al. 2025a. SlotPi: Physics-informed Object-centric Reasoning Models. *arXiv preprint arXiv:2506.10778*.

Li, J.; Ren, P.; Liu, Y.; and Sun, H. 2025b. Reasoning-Enhanced Object-Centric Learning for Videos. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 659–670.

Li, L.; Zhou, T.; Wang, W.; Yang, L.; Li, J.; and Yang, Y. 2022. Locality-Aware Inter- and Intra-Video Reconstruction for Self-Supervised Correspondence Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8719–8730.

Li, X.; Liu, S.; De Mello, S.; Wang, X.; Kautz, J.; and Yang, M.-H. 2019. Joint-task self-supervised learning for temporal correspondence. *Advances in Neural Information Processing Systems*, 32.

Locatello, F.; Weissenborn, D.; Unterthiner, T.; et al. 2020. Object-Centric Learning with Slot Attention. *Advances in Neural Information Processing Systems*, 33: 11525–11538.

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

Oquab, M.; Darcet, T.; Moutakanni, T.; et al. 2023. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*.

Palmeri, T.; and Gauthier, I. 2004. Visual Object Understanding. *Nature Reviews Neuroscience*, 5(4): 291–303.

Qian, R.; Ding, S.; Liu, X.; and Lin, D. 2023. Semantics meets temporal correspondence: Self-supervised object-centric learning in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16675–16687.

Safadoust, S.; and Güney, F. 2023. Multi-object discovery by low-dimensional object motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 734–744.

Seitzer, M.; Horn, M.; Zadaianchuk, A.; et al. 2023. Bridging the Gap to Real-World Object-Centric Learning. *International Conference on Learning Representations*.

Shen, Q.; Qiao, L.; Guo, J.; Li, P.; Li, X.; Li, B.; Feng, W.; Gan, W.; Wu, W.; and Ouyang, W. 2022. Unsupervised Learning of Accurate Siamese Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8101–8110.

Singh, G.; Deng, F.; and Ahn, S. 2022. Illiterate DALL-E Learns to Compose. *International Conference on Learning Representations*.

Singh, G.; Wu, Y.-F.; and Ahn, S. 2022. Simple Unsupervised Object-Centric Learning for Complex and Naturalistic Videos. *Advances in Neural Information Processing Systems*, 35: 18181–18196.

Traub, M.; Otte, S.; Menge, T.; Karlbauer, M.; Thuemmel, J.; and Butz, M. V. 2022. Learning What and Where: Disentangling Location and Identity Tracking Without Supervision. In *The Eleventh International Conference on Learning Representations*.

Uijlings, J. R.; Van De Sande, K. E.; Gevers, T.; and Smeulders, A. W. 2013. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104: 154–171.

van Steenkiste, S.; Chang, M.; Greff, K.; and Schmidhuber, J. 2018. Relational Neural Expectation Maximization: Unsupervised Discovery of Objects and their Interactions. In *International Conference on Learning Representations*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Veerapaneni, R.; Co-Reyes, J. D.; Chang, M.; Janner, M.; Finn, C.; Wu, J.; Tenenbaum, J.; and Levine, S. 2020. Entity Abstraction in Visual Model-based Reinforcement Learning. In *Conference on Robot Learning*, 1439–1456. PMLR.

Wang, N.; Zhou, W.; Song, Y.; Ma, C.; Liu, W.; and Li, H. 2021. Unsupervised Deep Representation Learning for Real-Time Tracking. *International Journal of Computer Vision*, 129(2): 400–418.

Wu, Z.; Dvornik, N.; Greff, K.; Kipf, T.; and Garg, A. 2023a. SlotFormer: Unsupervised Visual Dynamics Simulation with Object-Centric Models. *International Conference on Learning Representations*.

Wu, Z.; Hu, J.; Lu, W.; Gilitschenski, I.; and Garg, A. 2023b. SlotDiffusion: Object-Centric Generative Modeling with Diffusion Models. *Advances in Neural Information Processing Systems*, 36: 50932–50958.

Xu, J.; and Wang, X. 2021. Rethinking Self-Supervised Correspondence Learning: A Video Frame-Level Similarity Perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10075–10085.

Yuan, D.; Chang, X.; Huang, P.-Y.; Liu, Q.; and He, Z. 2020. Self-supervised deep correlation tracking. *IEEE Transactions on Image Processing*, 30: 976–985.

Zadaianchuk, A.; Seitzer, M.; and Martius, G. 2024. Object-Centric Learning for Real-World Videos by Predicting Temporal Feature Similarities. *Advances in Neural Information Processing Systems*, 36.

Zhao, R.; Wang, V.; Kannala, J.; and Pajarinen, J. 2025a. Grouped Discrete Representation for Object-Centric Learning. In *ECML-PKDD*.

Zhao, R.; Wang, V.; Kannala, J.; and Pajarinen, J. 2025b. Multi-Scale Fusion for Object Representation. In *ICLR*.

Zhao, R.; Wang, V.; Kannala, J.; and Pajarinen, J. 2025c. Vector-Quantized Vision Foundation Model for Object-Centric Learning. In *ACM Multimedia*.

Zhao, R.; Zhao, Y.; Kannala, J.; and Pajarinen, J. 2025d. Slot Attention with Re-Initialization and Self-Distillation. In *ACM Multimedia*.

Zoran, D.; Kabra, R.; Lerchner, A.; and Rezende, D. J. 2021. Parts: Unsupervised segmentation with slots, attention and independence maximization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10439–10447.