Object-Centric Multiple Object Tracking

Zixu Zhao[1] Jiaze Wang[2*] Max Horn[1] Yizhuo Ding[3*] Tong He[1] Zechen Bai[1]
Dominik Zietlow[1] Carl-Johann Simon-Gabriel[1] Bing Shuai[1] Zhuowen Tu[1] Thomas Brox[1]
Bernt Schiele[1] Yanwei Fu[3] Francesco Locatello[1] Zheng Zhang[1†] Tianjun Xiao[1]
[1] Amazon Web Services [2] The Chinese University of Hong Kong [3] Fudan University
{zhaozixu, jiazew, yizhuodi, htong, baizeche, bshuai, ztu, zhaz, tianjux}@amazon.com
{hornmax, zietld, cjsg, brox, bschiel, locatelf}@amazon.de, yanweifu@fudan.edu.cn

Abstract

Unsupervised object-centric learning methods allow the partitioning of scenes into entities without additional localization information and are excellent candidates for reducing the annotation burden of multiple-object tracking (MOT) pipelines. Unfortunately, they lack two key properties: objects are often split into parts and are not consistently tracked over time. In fact, state-of-the-art models achieve pixel-level accuracy and temporal consistency by relying on supervised object detection with additional ID labels for the association through time. This paper proposes a video object-centric model for MOT. It consists of an index-merge module that adapts the object-centric slots into detection outputs and an object memory module that builds complete object prototypes to handle occlusions. Benefited from object-centric learning, we only require sparse detection labels (0%-6.25%) for object localization and feature binding. Relying on our self-supervised Expectation-Maximization-inspired loss for object association, our approach requires no ID labels. Our experiments significantly narrow the gap between the existing object-centric model and the fully supervised state-of-the-art and outperform several unsupervised trackers. Code is available at https://github.com/amazon-science/object-centric-multiple-object-tracking.

1. Introduction

Visual indexing theory [45] proposes a psychological mechanism that includes a set of indexes that can be associated with an object in the environment. Each index retains its association with an object, even when that object moves, interacts with other objects, or becomes partially occluded. This theory was originally developed in the cognitive sciences, however, a very similar principle lies at the heart of object-centric representation learning. By learning object-level representations, we can develop models inferring object relations [39, 60, 62] and even their causal structure [36, 40]. Additionally, object-centric representations have shown to be more robust [13], allow for combinatorial generalization [36], and are beneficial for various downstream applications [60]. Since causal relations often unfold in time, it is only logical to combine object-centric learning (OCL) with temporal dynamics modeling, where consistent object representations are necessary.

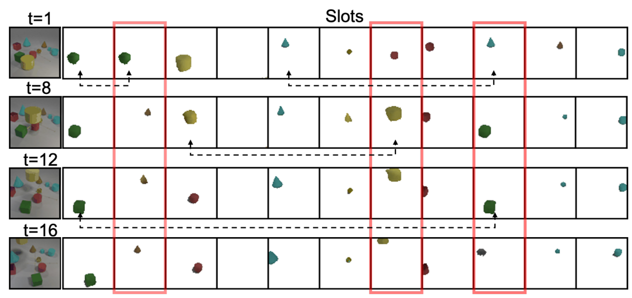

Figure 1. **Temporal inconsistency and part-whole split of object-centric representations.** We visualize a video object-centric model SAVi [32] that groups objects into a set of slots without labels. Common issues include that there exist many duplicate slots that capture the same object or its parts (dashed arrows), and the slots fail to track objects consistently over time (red boxes).

Multiple object tracking (MOT) is a computer-vision problem that resembles visual indexing theory. MOT aims at localizing a set of objects while following their trajectories over time so that the same object keeps the same identity in the entire video stream. The dominant MOT methods follow the detect-to-track paradigm: First, employ an object detector to localize objects in each frame, then perform association on detected objects between adjacent frames to get tracklets. The development of state-of-the-art

*Work completed during internship at AWS Shanghai AI Lab.
†Corresponding author.

MOT pipelines usually requires large amounts of detection labels for the objects we are interested in, as well as video datasets with object ID labels to train the association module. Consequently, such approaches are label intense and do not generalize well in open-world scenarios.

Unsupervised object-centric representation learning tackles the object discovery and binding problem in visual data without additional supervision [50]. Recent work, such as SAVi [32] and STEVE [52], extended such models to the video domain, which hints at possible applications to MOT. However, existing approaches are primarily evaluated without heavy punishment if slots exchange "ownerships" of pixels and rather rely on clustering similarity metrics such as FG-ARI [32]. An object may appear in different slots across time (a.k.a ID switch issue), which hinders downstream applications of OCL models, especially when *directional relationship* among objects and their dynamics must be reasoned upon (e.g., who acts upon whom). Additionally, the part-whole issues are not fully explored, allowing slots to only track parts of an object. Figure 1 visualizes the two problems of OCL models that are developed orthogonally with respect to MOT downstream tasks, leading to a significant gap with the state-of-the-art fully supervised MOT methods. Scalability challenges of unsupervised OCL methods only accentuate this gap.

In this work, we take steps to bridge the gap between object-centric learning and fully-supervised multiple object tracking pipelines. Our design focuses on improving OCL framework on two key issues: 1) track objects as a whole, and 2) track objects consistently over time. For these, we insert a memory model to consolidate slots into memory buffers (to solve the part-whole problem) and roll past representations of the memory forward (to improve temporal consistency). Overall, our model provides a label-efficient alternative to the otherwise costly MOT pipelines that rely on detection and ID labels. Our contributions can be summarized as follows:

(1) We develop a video object-centric model that can be applied to MOT task with very few detection labels ($0\,\%$-$6.25\,\%$) and no ID labels.

(2) OC-MOT leverages an unsupervised memory to predict completed future object states even if occlusion happens. Besides, the index-merge module can tackle the part-whole and duplication issues specific to OC models. The two cross-attention design is simple but nontrivial, serving as the "index" and "merge" functions with their key and query being bi-directional.

(3) We are the first to introduce the object-centric representations to MOT that are versatile enough in a way of supporting all the association, rolling-out, and merging functions, and can be trained with low labeling cost.

2. Related Works

Unsupervised Object-centric Learning. Unsupervised object-centric learning describes approaches which aim at tackling the binding problem of visual input signals to objects without additional supervision [26]. This is often accomplished using architectural inductive biases which force the model to encode input data into a set-structured bottleneck where object representations exert competition [16, 37, 57] or exclusive binding to features [25, 24, 6, 15]. Since their initial development on synthetic image data, these approaches have been extended to more complicated images by adapting the reconstruction objective [51, 50], to the decomposition of 3D scenes [9, 42, 53], to synthetic videos [33, 28, 11, 30, 32, 52] and to real-world videos by exploiting additional modalities and priors [32, 1, 14]. Our work is most closely related to the last group of methods which apply object-centric learning methods to real-world videos, yet in contrast does not focus on the derivation of object-centric representations themselves. Instead we focus on how object-centric representations can be used to perform multiple object tracking via long-term memory. Our work presents the first dedicated memory module, which, independent of the origin of the object-centric representation can match occurrences of objects to previously discovered objects and thus track these over time.

Self-supervised MOT. Most works study MOT in supervised settings, where the models are trained with object-level bounding box labels and ID labels [10, 64, 66, 63, 7]. Tracktor++ [3] uses a ready-made detector[20] to generate object bounding boxes and propagates them to the next frame as region proposals. MOTR [63] simultaneously performs object detection and association by autoregressively feeding a set of track queries into a Transformer decoder at the next timestep. To reduce the hand-label annotations, several recent approaches leverage the self-supervised signals to learn object associations from widely available unlabeled videos. For example, CRW [58] and JSTG [65] learns video correspondences by applying a cycle-consistent loss. Without fine-tuning, these models track objects at inference time by propagating the annotations from the first frame.

Our work is mostly related to the unsupervised detect-to-track approaches that assume a robust detector is available. SORT [4] and IOU [5] associate detections using heuristic cues such as Kalman filters and intersection-of-union of bounding boxes. Such models do not need training but fail to handle scenarios with frequent occlusion and camera motion. A recent related method uses cross-input consistency [2] to train the tracker: given two distinct inputs from the same video sequence, the model is encouraged to produce consistent tracks. Unfortunately, it suffers performance degradation once the detection boxes are not accurate, e.g., the grouping results from the object-centric model. For both supervised and unsupervised track-

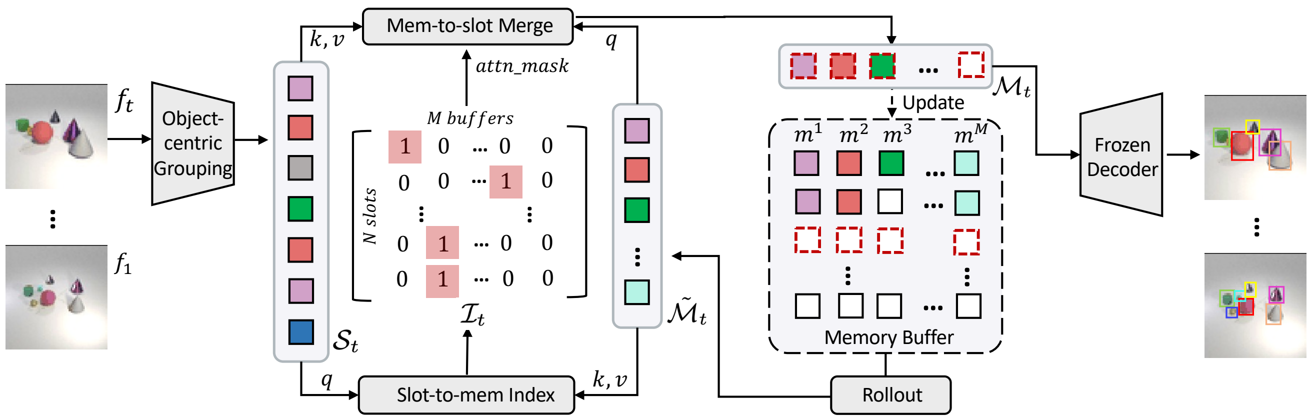

Figure 2. **Overview of OC-MOT**. It consists of two main modules. i) An index-merge module that adapts object-centric slots $\mathcal{S}_t$ into detection results $\mathcal{M}_t$ via two steps. First, index each slot into memory buffers by a learnable index matrix $\mathcal{I}_t$ indicating all the slot-to-memory assignments. Second, merge slots assigned to the same buffer by recalculating the attention weights masked by $\mathcal{I}_t$ backwards. ii) A object memory module that improves temporal consistency by rolling historical state forwards for object association. For MOT evaluation, we decode $\mathcal{M}_t$ to masks or bounding boxes via a frozen decoder in the object-centric grouping module.

ers, large amount of detection labels are required to train a strong detector. Additionally, supervised trackers require ID labels train feature representations. Overall, MOT is a label-heavy pipeline. Our work has the potential reduce the labeling cost. The heavy-lifting part of object localization and feature binding are done in a self-supervised way: on both backbone training and grouping training.

Memory Models. Memory models have been widely used in many video analysis and episodic tasks, such as action recognition [59, 29], video object segmentation [38, 43, 34], video captioning [44], reinforcement learning [22], physical reasoning [21], and code generation [35]. These works utilize an external memory to store features of prolonged sequences, which can be time-indexed for historical information integration. Recently, memory models have also been used to associate objects for video object tracking. MemTrack [61] and SimTrack [17] retrieve useful information from the memory and combine it with the current frame. However, they ignore the inter-object association and only focus on single object tracking. MeMOT [7] builds a large spatial-temporal memory for MOT, which stores all the identity embeddings of tracked objects and aggregates the historical information as needed. As expected, it requires costly object ID labels for training the memory. In this paper, we propose a self-supervised memory that leverages the memory rollout for object association. In contrast to previous learnable memory modules, our approach does not write global information in the memory via gradient descent [55] but rather maintains a running summary of the scene similar to [23] (but with multi-head attention rollout). Different than [23], we explicitly enforce an assignment between objects and memory buffers with subsequent merging steps for MOT.

3. Method

Our OC-MOT improves over traditional OCL frameworks in terms of tracking objects as a whole, and consistently over time. This is achieved by extending the traditional OC framework with a self-supervised memory to: i) Store historical information in the memory to fight against noise and occlusion. This helps improve temporal consistency. ii) Use the complete representation read-out from the memory to consolidate parts captured in different slots, which resolves the part-whole problem. The overall framework of OC-MOT is shown in Figure 2. Given slots $\{\mathcal{S}_t\}_{t=1}^T$ extracted from T video frames by an object-centric grouping module, OC-MOT first uses the memory rollout $\tilde{\mathcal{M}}_t$ to perform slot-to-memory indexing. Then, it merges the slots as $\mathcal{M}_t$ to update the memory.

3.1. Object-centric Grouping

The object-centric grouping module uses Slot Attention[37] to turn the set of encoder features from video frames into a set of slot vectors $\{\mathcal{S}_t\}_{t=1}^T$. The model is trained with a self-supervised reconstruction loss $L_{oc\_rec} = ||y - \text{Dec}(\mathcal{S})||^2$, where y can be the raw frame pixels, or feature representations extracted from the frames. The decoder has a compete-to-explain inductive bias to encourage binding of objects into individual slots.

3.2. Memory Module

We store the historical representations of all tracked objects into memory buffers $\mathcal{M} \in \mathbb{R}^{M \times T \times d}$ where M is the buffer number and d denotes the representation dimension. The memory is implemented with a first-in-first-out data structure and reserves a maximum of T_{max} time steps for each object. At time step t, the detection results are

$\mathcal{M}_t = \{m_t^1, ..., m_t^M\}$ if we denote m_t as the object representation. Intuitively, each buffer is a tracklet.

Memory rollout. At time step t, the memory rolls the past states forward, and predicts the current object representations for all slots to index. The rollout process integrates the multi-view object representations together and handles the part-whole matching in the occlusion scenarios. Without losing generality, we denote all the past representations as $\mathcal{M}_{<t}$. The rollout $\tilde{\mathcal{M}}_t \in \mathbb{R}^{M\times d}$ is obtained by:

$$\tilde{\mathcal{M}}_t = \text{Rollout}(\mathcal{M}_{<t}). \tag{1}$$

We adopt a mini GPT-2 model [46] containing only 1.6M parameters as the rollout module. It performs temporal reasoning via an auto-regressive transformer.

3.3. Index-Merge Module

The index-merge module is used as a discrete interface between memory buffers and slots. To achieve this, we split the object association process into the index step and merge step, as shown in Figure 3, which can be achieved by standard multi-head attention (MHA) [56] blocks.

Slot-to-memory index. The index matrix $\mathcal{I}_t \in \mathbb{R}^{N\times M}$ indicates soft slots-to-buffer assignment. To compute it, we train a MHA block that takes the slots $\mathcal{S}_t \in \mathbb{R}^{N\times d}$ as query, and rollout $\tilde{\mathcal{M}}_t$ as key and values, where N is slot number:

$$\mathcal{I}_t = \text{MHA}(k, v = \tilde{\mathcal{M}}_t, q = \mathcal{S}_t).\text{attn\_weight} \tag{2}$$

Memory-to-slot merge. Our goal is to make sure a buffer represents one object by pooling from the slots that belong to that object, while simultaneously dealing with slots that represent parts of an object or duplicates. Thus, we stack another MHA block to merge the slots, using $\mathcal{I}_t$ as masked attention weights. Specifically, the merging function is defined as below:

$$m_t = \text{MHA}(k, v = \mathcal{S}_t, q = \tilde{\mathcal{M}}_t, attn\_mask = \mathcal{I}_t). \tag{3}$$

Here, the query is the rollout $\tilde{\mathcal{M}}_t$; the key and value are slots $\mathcal{S}_t$. We apply $\mathcal{I}_t$ as the attention mask in MHA such that the re-normalized attention weights can be used for merging. This helps us to deal with wrongly-assigned slots. For example, if there are three slots and two of them are matched to one buffer, the attention weight could be $[0.8, 0.2, 0]$ indicating that the second slot does not belong to this buffer.

3.4. Model Training under EM Paradigm

Losses. The key of training detect-to-track models is to minimize the assignment costs for object associations. Usually, the weights of the pre-trained detector are frozen during training [7, 3]. Therefore, in our scenario, we freeze

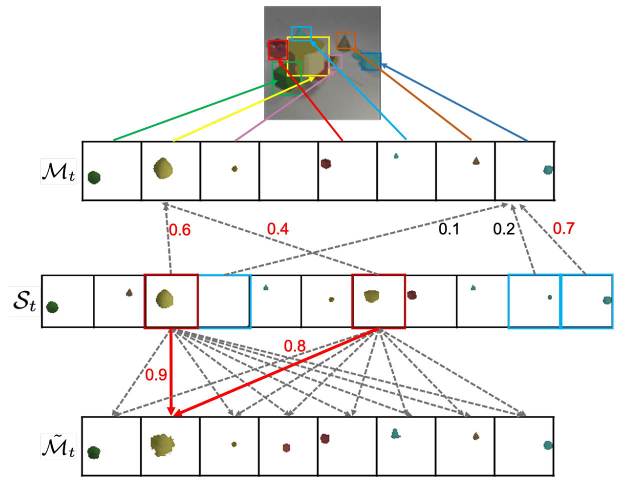

Figure 3. **Visualization of the index-merge module**. Index step: we show how to generate an index matrix from slots $\mathcal{S}_t$ to memory rollout $\tilde{\mathcal{M}}_t$. Note that the duplicate slots (red boxes) or similar slots (blue boxes) may be assigned to the same buffer. Merge step: the model recalculates the attention weights for slot merging, masked by the index matrix. The wrongly assigned slots can be filtered out with very low weights.

the object-centric model and only train the memory module. Assume we use $\mathcal{L}_{assign}$ to measure the assignment costs between slots $\mathcal{S}_t \in \mathbb{R}^{N\times d}$ and memory buffers $\mathcal{M}_t \in \mathbb{R}^{M\times d}$. The training loss can be formulated as:

$$\mathcal{L}_{MOT} = \sum_{t=1}^{T}\sum_{i=1}^{N} \mathbb{1}[Z_t[i] = j]\mathcal{L}_{assign}(\mathcal{S}_t^i, \mathcal{M}_t^j), \tag{4}$$

where $\mathcal{Z}_t \in \mathbb{R}^N$ denotes the assignments and $\mathcal{Z}_t[i] = j$ means the i^{th} slots matches to the j^{th} buffer. Specifically, we have three options to calculate the assignment cost: 1) use a binary cross-entropy loss on the decoded masks to promote the consistency of object attributes such as shape; 2) use a pixel-wise squared reconstruction loss on the object reconstructions (pixel reconstructions multiplied by object masks) to learn the color information; 3) use the same loss as 2) but directly apply on the feature space. The assignment cost could be a combination of the three losses:

$$\begin{aligned}\mathcal{L}_{assign}(\mathcal{S}_t^i, \mathcal{M}_t^j) &= \lambda_1 BCELoss(\text{Dec}(\mathcal{S}_t^i), \text{Dec}(\mathcal{M}_t^j)) \\ &+ \lambda_2||\text{Dec}(\mathcal{S}_t^i) - \text{Dec}(\mathcal{M}_t^j)||^2 + \lambda_3||\mathcal{S}_t^i - \mathcal{M}_t^j||^2,\end{aligned} \tag{5}$$

where λ_1, λ_2, and λ_3 are the balancing weights. We use the frozen decoder from the object-centric model to decode object representations into pixel reconstructions and masks.

Optimization. In contrast to prior supervised trackers [7, 63] that use ID labels to find the assignments, our model learns the index matrix $\mathcal{I}_t$ without any supervision. One naive solution is to convert $\mathcal{I}_t \in \mathbb{R}^{N\times M}$ to $\mathcal{Z}_t \in \mathbb{R}^N$ by performing argmax along the buffer dimension. However, the argmax function is non-differentiable. Even though we apply the straight-through trick [27] to make it trainable, the optimization easily gets stuck in a local minimum because the model has no chance to evaluate other possible assignments. To tackle this problem, we take inspiration from the Expectation-maximization (EM) paradigm which optimizes the assignments from seeing all possible assignments in $\mathcal{I}_t$. We formulate the expectation of $\mathcal{S}_t$ matches to $\mathcal{M}_t$ as:

$$\begin{aligned} Q(\theta^*, \theta) &= \mathbb{E}[\ln p(\mathcal{S}_t, \mathcal{M}_t | \theta^*)] \\ &= \Sigma_i \Sigma_j p(\mathcal{M}_t^j | \mathcal{S}_t^i) \ln p(\mathcal{S}_t^i, \mathcal{M}_t^j | \theta^*) \\ &= -\Sigma_i \Sigma_j \mathcal{I}_t[i,j] \mathcal{L}_{assign}(\mathcal{S}_t^i, \mathcal{M}_t^j). \end{aligned} \tag{6}$$

Here, θ is the learnable parameters in the memory module. $p(\mathcal{M}_t^j | \mathcal{S}_t^i)$ denotes the probability of the i^{th} slot is assigned to the j^{th} buffer, which, in our model, it exactly equals $\mathcal{I}_t[i,j]$. Further, we can use $\mathcal{L}_{assign}$ to represent $\ln p(\mathcal{S}_t^i, \mathcal{M}_t^j | \theta^*)$. We optimize the parameters of the model in order to maximize the expectation via SGD [49], for which we rewrite equation (4) as:

$$\mathcal{L} = \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathcal{I}_t[i,j] \mathcal{L}_{assign}(\mathcal{S}_t^i, \mathcal{M}_t^j). \tag{7}$$

The above loss (7) is applied to both the merged results $\mathcal{M}_t$ and rollout $\tilde{\mathcal{M}}_t$ with each combination weight set as 1.

3.5. Model Inference

During inference, we binarize the indexing matrix $\mathcal{I}_{t,hard} \in \{0,1\}^{N \times M}$ to strictly assign one slot to one buffer. Specifically, $\mathcal{I}_{t,hard}[i,j] = 1$ only if $j = \text{argmax}(\mathcal{I}_t)[i]$ for $i \in [1, N]$; otherwise, $\mathcal{I}_{t,hard}[i,j] = 0$. The discrete index supports the object in-n-out logic by indicating the presence of an object.

Object-in logic. For the first frame, we filter out duplicate slots before using them to initialize memory buffers. Slots with high mask IoU (bigger than τ_{iou}) to other slots will be discarded. For the next frames, we activate new buffers for new objects if slots have no substantial IoU with any masks of the memory rollout from the last timestep $\{\tilde{\mathcal{M}}_{t-1}^1, ..., \tilde{\mathcal{M}}_{t-1}^k\}$, where k is the active buffer number. Note that, for training, we replace the rollout with slots from the last timestep $\{\mathcal{S}_{t-1}^1, ..., \mathcal{S}_{t-1}^N\}$ because the rollout is not reliable at the early training stage.

Object-out logic. To re-track an object, we keep the buffer alive for τ_{out} consecutive frames when the object is occluded or disappears. In other words, if an object disappears for more than τ_{out} frames, the buffer will be terminated.

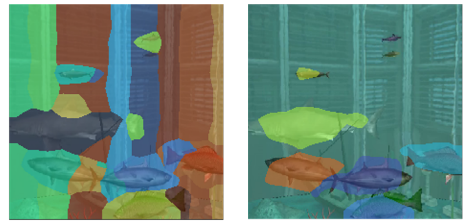

Figure 4. Comparisons of different grouping module settings on FISHBOWL: a) self-supervised DINOSAUR has high object recall but over-segments on both background and big fishes, causing trouble to memory buffer initialization. b) Tuning DINOSAUR with supervised DETR loss and partial mask labels resolves the over-segmentation issue and filters out background slots.

4. Experiments

We show that 1) OC-MOT consolidates "objects" in memory and greatly improves the temporal consistency of object-centric representations; 2) the gap between object-centric learning and MOT can be narrowed down by involving partial labels to improve the grouping performance; 3) the ablation studies demonstrate the effectiveness and feasibility of each module in the framework. Finally, we turn to KITTI [18] to discuss our limitations.

Datasets. CATER [19] is a widely used synthetic video dataset for object-centric learning. It is rendered using a library of 3D objects with various movements. Tracking multiple objects requires temporal reasoning about the long-term occlusions, a common issue in this dataset. FISHBOWL [54] consists of 20,000 training and 1,000 validation and test videos recorded from a publicly available WebGL demo of an aquarium, each with a resolution of 480×320px and 128 frames. Compared to CATER, FISHBOWL records more complicated scenes and has even more severe object occlusions. Besides, we also work on the real-world driving dataset KITTI [18] to analyze the limitation of the proposed object-centric framework.

Metrics. Following the standard MOT evaluation protocols [48, 41], we use Identity F1 score (IDF1), Multiple-Object Tracking Accuracy (MOTA), Mostly Tracked (MT), Mostly Lost (ML), and Identity Switches (IDS) as the metrics. Specifically, IDF1 highlights the tracking consistency, and MOTA measures the object coverage. To weight down the effect of detection accuracy and focus on the association performance, we set the IoU distance threshold as 0.7. We also introduce Track mAP [12], which is more sensitive to identity switches by matching the object bounding boxes to ground-truth through the entire video using 3D IoU.

Implementation details. We train OC-MOT using the Adam optimizer [31] with a learning rate of $2 \cdot 10^{-4}$ and an exponentially decaying learning rate schedule. The mod-

Method	Detection Label	ID Label	IDF1 ↑	MOTA ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓
				CATER [19]					
SAVi [32]			73.2%	52.5%	75.2%	21.2%	305027	130810	20352
IOU [5]			83.0%	77.4%	73.3%	17.4%	**35480**	173595	8259
SORT [4]			84.5%	79.2%	71.8%	24.1%	43097	148068	8219
Visual-Spatial [2]			85.8%	80.3%	76.6%	20.8%	51348	129680	7562
OC-MOT			**88.6%**	**82.4%**	**82.3%**	**13.9%**	57792	**105054**	**5658**
MOTR [63]	100%	100%	89.3%	83.3%	84.8%	4.9%	60647	96746	3366
				FISHBOWL [54]					
SAVi [32]	6.25%		46.9%	32.3%	47.3%	15.1%	122006	**96710**	12504
SORT [4]	6.25%		68.4%	64.3%	42.6%	31.9%	30912	132434	15278
IOU [5]	6.25%		71.3%	66.6%	11.0%	62.7%	31672	135394	10306
Visual-Spatial [2]	6.25%		74.6%	68.1%	48.2%	19.8%	28845	131076	8754
OC-MOT	6.25%		**77.9%**	**70.3%**	**50.2%**	**13.2%**	**14738**	136852	**5898**
MOTR [63]	100%	100%	81.6%	79.8%	58.3%	10.1%	9678	92862	4185

Table 1. **Evaluation results on CATER and FISHBOWL**. For CATER, the object-centric grouping module is pre-trained without any label. For FISHBOWL, the grouping module is pre-trained with 6.25% mask labels to improve the detection accuracy. The supervised MOTR [63] is trained with 100% box labels and ID labels. The best results of unsupervised trackers are marked in bold.

els were trained on 8 NVIDIA GPUs with batchsize of 8. We set τ_{out} as 5 for buffer termination. The IoU threshold τ_{iou} is set as 0.9. For the experiments on CATER, we pretrain a SAVi model for object grouping without any annotation. We set $N = 11$ and $M = 15$. The hyperparameters in the training loss $\lambda_1, \lambda_2, \lambda_3$ are selected as 1, 0.1, 0. For the experiments on FISHBOWL, we used a pretrained image-level DINOSAUR [50] as the grouping module and selected $\lambda_1, \lambda_2, \lambda_3$ as 1, 0, 1. We set λ_2 to 0 due to GPU memory limitations when combining the EM loss computation with the high dimensional DINOSAUR features. We set $N = 24$ and $M = 40$. In complex scenes of FISHBOWL, we noticed a performance drop due to more severe part-whole issues and over-segmentation on the background as illustrated in Figure 4. To avoid tracking background objects and reduce over-segmentation on big objects, we suggest further improving object-centric grouping by utilizing temporal sparse labels. To be more specific, we apply supervised DETR [8]-style loss on the decoded masks of slots. Since the object grouping loss already takes the heavy-lifting of discovering objects and parts, we only require very few mask labels to inject semantics about which objects are interesting and how to merge parts into a whole object. In practice, we utilized $6.25\,\%$ (randomly label 8 frames in 128-frame videos) mask labels for DINOSAUR pre-training, with both DETR loss and self-supervised reconstruction loss.

4.1. Comparison with the State-of-the-art Methods

Baselines. We compare OC-MOT with one object-centric method (SAVi [32]), three unsupervised MOT methods (IOU [5], SORT [4], and Visual-Spatial [2]), and one fully supervised MOT method (MOTR [63]). For the SAVi evaluation, we remove the background slots and treat each slot prediction as a tracklet. When training SAVi on FISHBOWL, we also provide 6.25% temporal sparse mask labels to be comparable to our own setting. For fair comparisons, we use the same pre-trained object-centric model (DINOSAUR with 6.25% detection labels) as the detector for IOU, SORT, and Visual-Spatial. MOTR utilizes a query-based transformer for both object detection and association but requires object-level annotations (both boxes and object ID) for model training. This model and its follow-ups have achieved SOTA results on several MOT benchmarks.

Method	OC Metric	MOT Metric		
	FG-ARI ↑	IDF1 ↑	MOTA ↑	Track mAP ↑
SAVi	90.2	72.3	52.5	42.8
OC-MOT	**93.8**	**88.6**	**82.4**	**66.2**

Table 2. Comparisons with video object-centric models on CATER. Note that FG-ARI [32] is a commonly used OC metric.

Results on CATER. As shown in Table 1, OC-MOT substantially outperforms the video object-centric model and other unsupervised baselines. Our approach is also competitive with supervised MOTR [63] trained on expensive object-level annotations, yielding only slightly lower IDF1 and MOTA. OC-MOT can keep tracking more objects but produces fewer ID switches. For example, it achieves $82.3\,\%$ Mostly Tracked (MT) and $13.9\,\%$ Mostly Lost (ML), and shows only 5658 IDS. Moreover, in Table 2, SAVi achieves 90.2% of FG-ARI but performs bad in terms of other MOT metrics such as 42.8% Track mAP, indicating that the FG-ARI is not a good metric for measuring object-level temporal consistency.

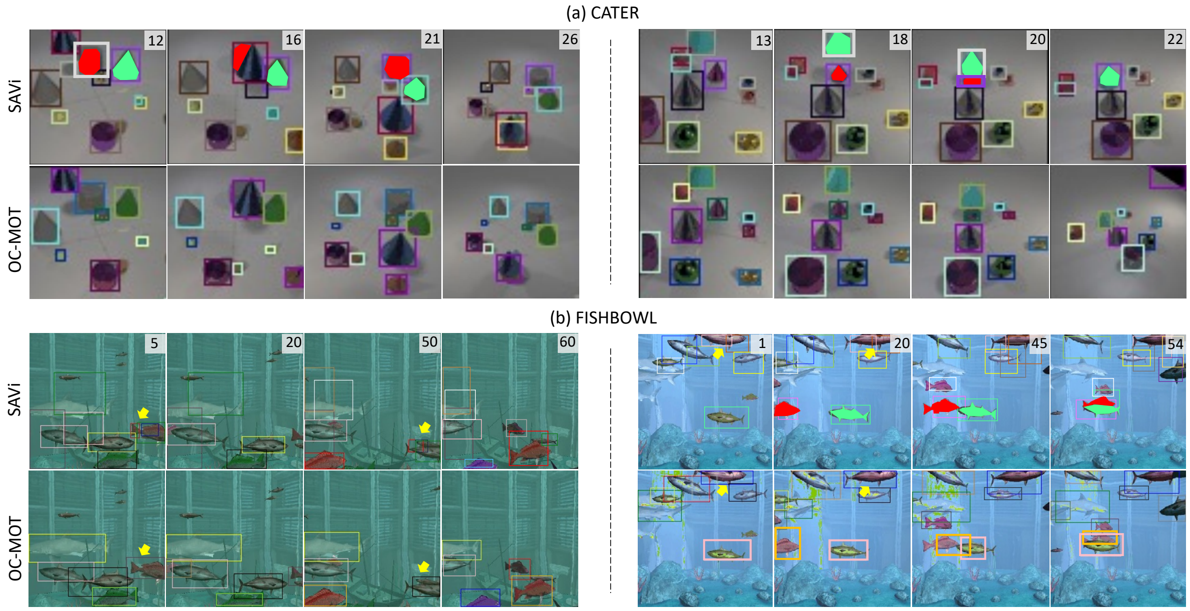

Figure 5. **MOT results on CATER and FISHBOWL.** We highlight the occlusion cases with colored masks. SAVi over-segments the objects (yellow arrows) and has ID switches after occlusions. In contrast, OC-MOT tracks objects more consistently over time.

Results on FISHBOWL. FISHBOWL is a more challenging benchmark with serious occlusions and complicated backgrounds. In this scenario, SAVi [32] tends to split the complicated background into multiple slots, causing the high number of false positive (FP). Table 1 shows that OC-MOT achieves state-of-the-art performance among the unsupervised tracking methods. By getting a much lower IDS number, our approach shows its advantage in solving the occlusion problem. The non-linear transformation in frequent occlusions cannot be handled by IoU-based association (IOU [5]) or Kalman filter (SORT [4]). Compared to supervised MOTR, what would like to highlight is the impressive association capability of OC-MOT. We point out that the lower IDF1 and MOTA are mainly caused by the detection limitation of existing OC models (e.g., DINOSAUR decoder predicts masks at a low feature resolution, making it hard to get pixel-level accuracy).

4.2. Visualization

The MOT results on the occlusion cases are visualized in Figure 5. OC-MOT associates the slots from the object-centric model and generates consistent predictions even when objects frequently interact with each other. Due to the severe occlusions, SAVi [32] fails to track objects even using the track query as input, thereby causing more ID switches. Moreover, SAVi produces more false positives due to over-segmentation.

In Figure 6, we visualize the memory rollout results by decoding the representations to object reconstructions. The memory starts to roll out after the first frame, and, at $t = 1$, we visualize the existing memory features. We can observe that the rollouts achieve good temporal consistency and, even more interestingly, that the memory can predict a complete object even when it has been partially occluded.

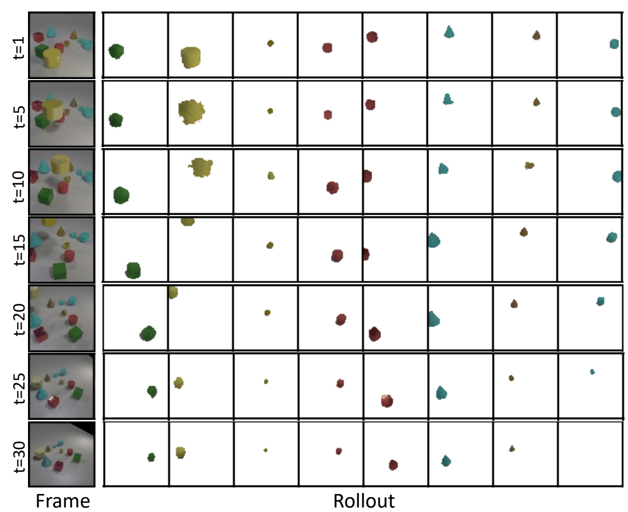

Figure 6. **Visualization of memory rollout.** We show the object reconstructions decoded from the rollout representations. Each column denotes a memory buffer. The rollout predictions are consistent and complete, even when objects are partially occluded

4.3. Ablation Studies

Component analysis. Table 3 compares different design choices for the key components in OC-MOT. For the index-merge module, a naive solution is to use a parameter-free dot-product to calculate the feature similarity, inspired by [47]. As expected, it produces the worst association performance. A further option is to train one single MHA (i.e., two MHAs with shared weights) to cluster slots to buffers as in [23]. To get the discrete index for object in and out logic, we still follow the indexing and merging steps yet only calculating the attention weights once. We observe that this model yields slightly lower IDF1 and MOTA than training two MHA modules. The latter choice is mathematically the similar but with higher module capacity. For the memory module, we compare utilizing the rollout module to only using the last tracks as the index query. Without aggregating the historical memory features, the association performance drops dramatically, indicating the necessity of building a memory to handle the MOT problem.

Index-Merge Module	Memory Module	IDF1 ↑	MOTA ↑	IDS ↓
Dot-product	Rollout	72.0%	61.5%	22050
One MHA	Rollout	86.2%	80.5%	7655
Two MHA	Last Tracks	77.2%	68.8%	16582
Two MHA	Rollout	**88.6%**	**82.4%**	**5658**

Table 3. Ablation on OC-MOT components on CATER.

Effect of memory length. In Table 4, we explore the effect of the memory length T_{max} from 6 to 32. Note that T_{max} equals the length of the training sequence. The tracking performance increases as T_{max} grows. However, for longer videos, we should set a max length of memory due to hardware limitations. To make the model more applicable, we propose to reserve a short-term memory trained with sequences sampled by slow-to-fast pace. The various sampling rates produce both short-term and long-term information and, more importantly, include the occlusion cases during training. Quantitatively, this sampling strategy peaks in performance with $T_{max} = 6$. Unless noted otherwise, we set T_{max} to 6 as default in other experiments.

T_{max}	Sequence Sampling	IDF1 ↑	MOTA ↑	IDS ↓
6	Consecutive	82.9%	76.3%	7601
10	Consecutive	83.2%	76.5%	7524
20	Consecutive	86.9%	78.1%	6230
32	Consecutive	88.4%	82.1%	5763
6	Slow-Fast	**88.6%**	**82.4%**	**5658**

Table 4. Ablation on the memory length on CATER.

4.4. Limitations

There exist some limitations of the proposed OC-MOT.

Inductive bias in the grouping architecture. We apply the same DINOSAUR+DETR grouping module with 6.25% temporally sparse mask labels to KITTI dataset. Figure 7 visualizes the grouping results. The cars can be detected but the predicted masks are not accurate, especially for far-away objects. One reason is that DINOSAUR predicts masks at a feature resolution that is down-scaled 16 times from the original size. The architecture of the grouping module needs to be further improved considering multi-resolution inductive biases that have already been adopted in supervised detection and segmentation pipelines. We encourage researchers to develop stronger OC models with powerful detection performance but low labeling cost.

Figure 7. DINOSAUR grouping results on KIITI. We observe that the masks are imprecise, espcially for objects that are far away.

The model is not trained end-to-end. In this paper, we use the pre-trained OC model as a plug-n-play detector, which is supposed to handle different data flexibly. Potential future work is to extend OC-MOT into an end-to-end framework. The object prototype built in the memory may be useful as a prior for object discovery.

5. Conclusion

In this paper, we build a pipeline for MOT with object-centric backbones. With memory modules, we can address both part-whole issues and consistently track objects over time. Overall, our approach improves over conventional tracking-by-detection pipelines by replacing expensive annotations (especially ID annotations) with self supervision. This work opens many directions for new research. First of all, it allows for active learning. For example, the model could elicit a request for labeling on specific frames, further reducing necessity for costly annotations. Furthermore, incorporating memory information as top-down reasoning prior for the object-centric encoder still remains to be explored. Additionally, we still require few masks and class labels to resolve over-segmentation. Those semantic signals could be distilled from multi-modal foundation models trained with weaker supervision signals (e.g., captioned images). Finally, our results delineate a clear benefit in improving (video) object-centric backbones. As we have demonstrated, improvements in self-supervised object-centric learning can greatly facilitate complex downstream vision tasks like MOT, improving performance by training on unsupervised or weakly-supervised data.

References

[1] Zhipeng Bao, Pavel Tokmakov, Allan Jabri, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Discovering objects that can move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11789–11798, 2022.

[2] Favyen Bastani, Songtao He, and Samuel Madden. Self-supervised multi-object tracking with cross-input consistency. *Advances in Neural Information Processing Systems*, 34:13695–13706, 2021.

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 941–951, 2019.

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[5] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2017.

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

[7] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8090–8100, 2022.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[9] Chang Chen, Fei Deng, and Sungjin Ahn. Roots: Object-centric representation and rendering of 3d scenes. *The Journal of Machine Learning Research*, 22(1):11770–11805, 2021.

[10] Peng Chu and Haibin Ling. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6172–6181, 2019.

[11] Eric Crawford and Joelle Pineau. Exploiting spatial invariance for scalable unsupervised object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3684–3692, 2020.

[12] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 436–454. Springer, 2020.

[13] Andrea Dittadi, Samuele S Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and robustness implications in object-centric learning. In *International Conference on Machine Learning*, pages 5221–5285. PMLR, 2022.

[14] Gamaleldin F Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. *arXiv preprint arXiv:2206.07764*, 2022.

[15] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations. In *International Conference on Learning Representations*, 2020.

[16] SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in Neural Information Processing Systems*, 2016.

[17] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13774–13783, 2021.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[19] Rohit Girdhar and Deva Ramanan. Cater: A diagnostic dataset for compositional actions and temporal reasoning. *arXiv preprint arXiv:1910.04744*, 2019.

[20] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[21] Anirudh Goyal, Aniket Didolkar, Nan Rosemary Ke, Charles Blundell, Philippe Beaudoin, Nicolas Heess, Michael C Mozer, and Yoshua Bengio. Neural production systems. *Advances in Neural Information Processing Systems*, 34:25673–25687, 2021.

[22] Anirudh Goyal, Abram Friesen, Andrea Banino, Theophane Weber, Nan Rosemary Ke, Adria Puigdomenech Badia, Arthur Guez, Mehdi Mirza, Peter C Humphreys, Ksenia Konyushova, et al. Retrieval-augmented reinforcement learning. In *International Conference on Machine Learning*, pages 7740–7765. PMLR, 2022.

[23] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent Independent Mechanisms. In *International Conference on Learning Representations*, 2021.

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

[25] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. *Advances in Neural Information Processing Systems*, 30, 2017.

[26] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

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

[28] Jindong Jiang, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn. SCALOR: Generative World Models with Scalable Object Representations. In *Proceedings of ICLR 2020*, 2020.

[29] Yueming Jin, Yonghao Long, Cheng Chen, Zixu Zhao, Qi Dou, and Pheng-Ann Heng. Temporal memory relation network for workflow recognition from surgical video. *IEEE Transactions on Medical Imaging*, 40(7):1911–1923, 2021.

[30] Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matt Botvinick, Alexander Lerchner, and Chris Burgess. Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. *Advances in Neural Information Processing Systems*, 34:20146–20159, 2021.

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[32] Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2022.

[33] Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects. In *Advances in Neural Information Processing Systems*, 2018.

[34] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2020.

[35] Shangqing Liu, Yu Chen, Xiaofei Xie, Jing Kai Siow, and Yang Liu. Retrieval-augmented generation for code summarization via hybrid gnn. In *International Conference on Learning Representations*, 2021.

[36] Yuejiang Liu, Alexandre Alahi, Chris Russell, Max Horn, Dominik Zietlow, Bernhard Schölkopf, and Francesco Locatello. Causal triplet: An open challenge for intervention-centric causal representation learning. 2023.

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

[38] Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 661–679. Springer, 2020.

[39] Davide Mambelli, Frederik Träuble, Stefan Bauer, Bernhard Schölkopf, and Francesco Locatello. Compositional multi-object reinforcement learning with linear relation networks. *arXiv preprint arXiv:2201.13388*, 2022.

[40] Amin Mansouri, Jason Hartford, Kartik Ahuja, and Yoshua Bengio. Object-centric causal representation learning. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022.

[41] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[42] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[43] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019.

[44] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8347–8356, 2019.

[45] Zenon Pylyshyn. The role of location indexes in spatial perception: A sketch of the finst spatial-index model. *Cognition*, 32(1):65–97, 1989.

[46] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[47] Nasim Rahaman, Muhammad Waleed Gondal, Shruti Joshi, Peter Gehler, Yoshua Bengio, Francesco Locatello, and Bernhard Schölkopf. Dynamic inference with neural interpreters. *Advances in Neural Information Processing Systems*, 34:10985–10998, 2021.

[48] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*, pages 17–35. Springer, 2016.

[49] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

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

[51] Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate DALL-e learns to compose. In *International Conference on Learning Representations*, 2022.

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

[53] Karl Stelzner, Kristian Kersting, and Adam R Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv preprint arXiv:2104.01148*, 2021.

[54] Matthias Tangemann, Steffen Schneider, Julius Von Kügelgen, Francesco Locatello, Peter Gehler, Thomas

Brox, Matthias Kümmerer, Matthias Bethge, and Bernhard Schölkopf. Unsupervised object learning via common fate. *arXiv preprint arXiv:2110.06562*, 2021.

[55] Frederik Träuble, Anirudh Goyal, Nasim Rahaman, Michael Mozer, Kenji Kawaguchi, Yoshua Bengio, and Bernhard Schölkopf. Discrete key-value bottleneck. *arXiv preprint arXiv:2207.11240*, 2022.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[57] Julius von Kügelgen, Ivan Ustyuzhaninov, Peter Gehler, Matthias Bethge, and Bernhard Schölkopf. Towards causal generative scene models via competition of experts. *arXiv preprint arXiv:2004.12906*, 2020.

[58] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

[59] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.

[60] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. *arXiv preprint arXiv:2210.05861*, 2022.

[61] Tianyu Yang and Antoni B Chan. Learning dynamic memory networks for object tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 152–167, 2018.

[62] Jaesik Yoon, Yi-Fu Wu, Heechul Bae, and Sungjin Ahn. An investigation into pre-training object-centric representations for reinforcement learning. *arXiv preprint arXiv:2302.04419*, 2023.

[63] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 659–675. Springer, 2022.

[64] Wenwei Zhang, Hui Zhou, Shuyang Sun, Zhe Wang, Jianping Shi, and Chen Change Loy. Robust multi-modality multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2365–2374, 2019.

[65] Zixu Zhao, Yueming Jin, and Pheng-Ann Heng. Modelling neighbor relation in joint space-time graph for video correspondence learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9960–9969, 2021.

[66] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, pages 474–490. Springer, 2020.