SAVi++: Towards End-to-End Object-Centric Learning from Real-World Videos

Gamaleldin F. Elsayed *†, **Aravindh Mahendran** *◇, **Sjoerd van Steenkiste** *◇,
Klaus Greff, Michael C. Mozer & Thomas Kipf *
Google Research

Abstract

The visual world can be parsimoniously characterized in terms of distinct entities with sparse interactions. Discovering this compositional structure in dynamic visual scenes has proven challenging for end-to-end computer vision approaches unless explicit instance-level supervision is provided. Slot-based models leveraging motion cues have recently shown great promise in learning to represent, segment, and track objects without direct supervision, but they still fail to scale to complex real-world multi-object videos. In an effort to bridge this gap, we take inspiration from human development and hypothesize that information about scene geometry in the form of depth signals can facilitate object-centric learning. We introduce SAVi++, an object-centric video model which is trained to predict depth signals from a slot-based video representation. By further leveraging best practices for model scaling, we are able to train SAVi++ to segment complex dynamic scenes recorded with moving cameras, containing both static and moving objects of diverse appearance on naturalistic backgrounds, without the need for segmentation supervision. Finally, we demonstrate that by using sparse depth signals obtained from LiDAR, SAVi++ is able to learn emergent object segmentation and tracking from videos in the real-world Waymo Open dataset.

Project page: https://slot-attention-video.github.io/savi++/

1 Introduction

The natural world consists of distinct entities—people, dogs, cars, trees, etc.—and its complexity emerges from the combined, mostly independent, actions of the entities. This compositional structure must be appreciated to predict future states of the world and to effect particular outcomes. People have an intrinsic understanding of objects: objects have spatiotemporal coherence, they interact when in close proximity, and they

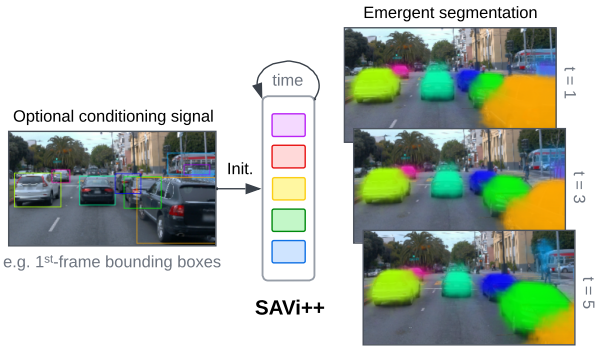


Figure 1: Emergent segmentation and tracking in SAVi++.

*Equal technical contribution. ◇Alphabetical order. †Correspondence to: gamaleldin@google.com
Author contributions: GFE, TK initiated and led the project. GFE, AM, SVS, TK developed the main model. GFE, AM, TK developed real-world driving data and model infrastructure. AM implemented data augmentation. GFE led ablation study. SVS led target signals analyses and metric design. GFE, AM, SVS, TK ran experiments. AM, SVS, KG, TK worked on baselines. KG, MCM provided advice at all stages and helped with project scoping. TK developed visualizations. GFE, MCM, TK worked on figure design. All authors wrote the paper.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

possess persistent, latent characteristics that determine their behavior over extended periods of time [31, 55]. Just as object-centric representations are critical to human understanding, they have the potential in machine learning to greatly improve sample efficiency, robustness, visual reasoning, and interpretability of learning algorithms [19, 41]. For example, consider the challenge faced by an autonomous vehicle operating in diverse surroundings (Figure 1). Generalization across situations requires learning about recurring entities like cars, traffic lights, and pedestrians, and the rules that govern interactions among these entities.

In human brains, the ability to organize edges and surfaces into unitary, bounded, and persisting object representations develops through experience and/or maturation from infancy and without explicit instruction via a 'core system of object representation' [55], i.e., a form of cognitive inductive bias. In deep learning, such an inductive bias has been proposed in *slot-based* architectures which segregate knowledge about individual objects into nonoverlapping but interchangeable pools of neurons. The resulting representational modularity can facilitate causal reasoning and prediction for downstream tasks [19, 53].

A grand challenge in computer vision has been to discover the compositional structure of real-world dynamic visual scenes in an unsupervised fashion. By unsupervised, we mean no segmentation information is provided that specifies which pixels belong together as part of a single object. Initial efforts focused on single-frame, synthetic RGB images [17, 18, 42, 60], but extending this work to video and more complex scenes proved challenging. A key insight to further progress was the realization that a color-intensity pixel array is not the only source of visual information readily available, at least not to human perceptual systems. The human perceptual system extracts motion and depth cues early in the processing stream [11–13, 25, 47]. These cues are correlated with object identities, and can therefore bootstrap the formation of object-centric representations [54].

The recently introduced *Slot Attention for Video* (SAVi) model [37] leveraged *optical flow* (frame-to-frame motion) as a prediction target to obtain object-centric representations of dynamic scenes involving complex 3D scanned objects and real-world backgrounds. However, motion prediction alone is insufficient to learn about the distinction between static objects and the background. Further, in real-world application domains such as self-driving cars, cameras themselves are subject to movement, which globally affects frame-to-frame motion as a prediction signal in non-trivial ways.

In the present work, we describe an enhanced slot-based model for video, referred to as *SAVi++* (Figure 2), which obtains qualitative improvements in object-centric representations by exploiting *depth* signals readily available from RGB-D cameras and LiDAR sensors. SAVi++ is the first slot-based, end-to-end trained model that successfully segments complex objects in naturalistic, real-world video sequences without using direct segmentation or tracking supervision.

A summary of our contributions is as follows:

- We introduce SAVi++: an object-centric slot-based video model that makes several key improvements to SAVi [37] by utilizing *depth prediction* and by adopting best practices for model scaling in terms of *architecture design* and *data augmentation*.
- On the multi-object video (MOVi) benchmark containing synthetic videos of high visual and dynamic complexity [20], we find that SAVi++ is able to handle videos containing complex shapes and backgrounds, and a large number of objects per scene. Improving on SAVi, our approach accommodates both static and dynamic objects and both static and moving cameras.
- Finally, we demonstrate that SAVi++ trained with sparse depth signals obtained from LiDAR enables emergent object decomposition and tracking in real-world driving videos from the Waymo Open dataset [56].

2 Related work

Object-centric learning A growing body of research is addressing the problem of end-to-end learning of object-centric representations from raw perceptual data without direct supervision. Slot-based neural networks such as IODINE [18], MONet [5], and Slot Attention [42] rely on a factorized latent space and independent per-object decoders as inductive bias to enable object discovery in a simple auto-encoding setup. Architectures with stronger inductive biases using fixed object size, presence, or propagation priors have been explored in works such as SQAIR [39] and SCALOR [29], but generally these methods have faced challenges scaling to more complex real-world data when

relying on auto-encoding alone. Our work primarily builds on recent advances in object-centric generative models for video sequences [30, 37, 60, 62, 70]. Different from our approach, these methods have so far been unable to scale to complex real-world multi-object video data. An alternative class of methods using contrastive learning for object discovery [24, 36, 44, 67], most notably GroupViT [67] and ODIN [24], has recently achieved some success in discovering semantic groupings in real-world images. However, neither GroupViT nor ODIN model dynamics and typically fail to separate semantically similar object instances in close proximity. In our work, we follow a generative approach, but instead of tasking the decoder to generate complex visual RGB pixel data, we utilize depth information to bootstrap object-centric learning without direct supervision.

Object discovery in driving scenes A range of recent methods [1, 21, 58, 63] use a multi-stage pipeline of 1) obtaining *pseudo ground truth* (PGT) segmentation or detection labels via some heuristic, and 2) training a model in a supervised fashion on PGT labels. While this class of methods achieves some success in discovering and tracking objects in real-world driving scenes, it crucially hinges on the quality of the PGT labels, requiring carefully engineered task-specific heuristics to extract objects. Earlier methods solely use clustering heuristics to extract approximate segmentation masks directly from motion trajectories for moving objects [4, 48]. In our work, we instead demonstrate that object segmentation and tracking can emerge in an end-to-end setting on complex real-world data without relying on PGT label generation.

Cross-modal learning For self-supervised object-centric learning from visual data, a range of target modalities and training signals have been explored in the literature. By using motion cues from optical flow as prediction targets, several recent methods [37, 68] were able to overcome limitations of purely RGB pixel-level generative models, which frequently failed in the presence of complicated textures [33]. However, this advantage is primarily limited to discovery of moving objects. Utilizing depth targets from a simulator [2] or from sparse LiDAR [21, 58, 63], has been explored in an effort to overcome these limitations. Different from prior works utilizing multi-stage pipelines and hand-crafted heuristics for extraction of pseudo-labels from LiDAR [21, 58, 63], we directly utilize the (sparse) depth signal as target and demonstrate that this can enable emergent object segmentation and tracking on real-world driving data without any additional regularizers or pseudo-labeling techniques.

Scaling strategies for vision models It is common practice to scale architectural capacity with dataset complexity and size, while making use of strong data augmentation when addressing various supervised computer vision tasks [8, 10, 22, 38]. Nonetheless, self-supervised methods for end-to-end object discovery have primarily been relying on overly simplistic and low-capacity backbone architectures [18, 42, 70], likely due to the simplicity of datasets and tasks considered in prior work. By scaling object-centric methods to larger, visually more complex datasets, we find that utilizing stronger visual backbone architectures—in combination with data augmentation—can provide substantial benefits. For simplistic datasets with lower visual complexity (and same number of examples), we found anecdotal evidence in preliminary experiments for the opposite effect: both architecture scaling and data augmentation can negatively affect object discovery performance, likely explaining why prior works have not explored these strategies.

Depth estimation Recent advances in supervised monocular depth estimation (see Ming et al. [46] for a review) could be combined with our method in future work, for instance using ordinal regression losses [14], transformer architectures [52], or more complex instance-wise decoder architectures [64].

3 Methods

We begin by providing a brief introduction to Slot Attention for Video (SAVi), which is the starting point for our exploration. With SAVi++, we introduce several simple yet crucial improvements, which allow us to bridge the gap to complex real-world data. Our framework is summarized in Figure 2.

3.1 Background

Slot Attention for Video (or SAVi) is a recent state-of-the-art architecture for learning object-centric representations from video with minimal supervision. We briefly highlight some of its key components below and refer the reader for complete details to Kipf et al. [37].

SAVi can be viewed as an autoregressive encoder-decoder video model with a structured latent state composed of K object slots. At a given time-step, an *encoder* first encodes the observed video frame

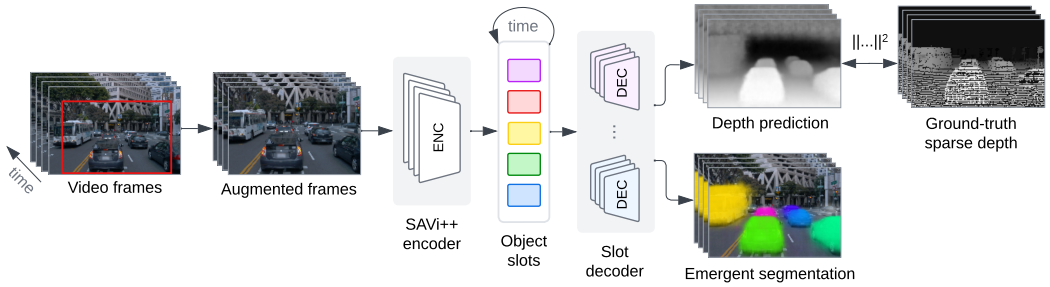


Figure 2: SAVi++ is an object-centric video model based on *Slot Attention for Video* [37], which encodes a video into a set of temporally-consistent latent variables (*object slots*). Input frames and prediction targets are augmented using random crop augmentations. Augmented frames are passed through the improved SAVi++ encoder and mapped onto object slots using an attention mechanism [42]. Slots are updated recurrently for each frame and subsequently decoded independently into a depth map and per-slot alpha masks. SAVi++ is trained using (sparse) depth targets, leading to emergence of temporally-consistent object segmentation in the decoded alpha masks.

to yield high-level image features that are useful for learning about objects. This is followed by Slot Attention [42] (the *'corrector'*), which updates the slots using these features and encourages individual slots to specialize to different parts of the observation. The content of each slot is decoded separately using a *decoder*, which additionally outputs a pixel-level alpha mask to indicate how the decoded values for each slot should be combined. Together, the mask and decoded slots determine the output of the model at the current time-step from which a loss is computed, e.g., to train the model to predict frame-to-frame motion (optical flow) for this frame. Slots for the next time-step (for the corrector to update) are obtained by applying a *predictor*, which can model interactions between slots and learn about object dynamics to predict their future state.

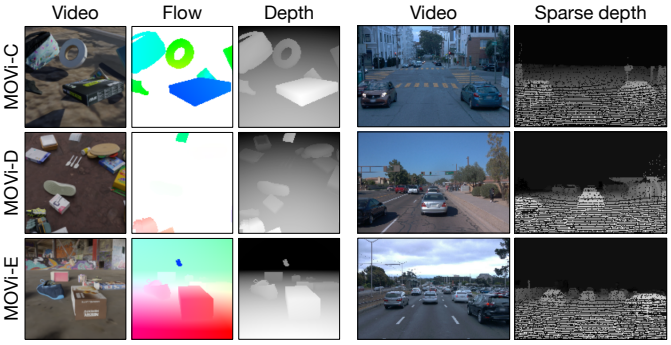
In addition to optical flow prediction, SAVi introduces conditioning that helps reduce uncertainty about the part-whole division into objects by pointing the model to specific locations. Indeed, in the absence of a specific downstream task, scene decomposition can be ambiguous and providing additional information as a conditioning signal may help alleviate this. The conditioning takes place via the *slot initializer*, which initializes the slots used in the initial video frame. The initialization may be learned in an unconditional setting (i.e., learn the initial slot states) or obtained by conditioning the initial state on high-level cues such as bounding boxes of objects of interest in the first video frame. This direction of attention or *input conditioning* helped SAVi to succeed in decomposing more complex visual scenes.

3.2 SAVi++

As SAVi relies on optical flow prediction as its main training signal for object discovery, its application is primarily limited to settings where all objects in a scene have independent motion. In addition, SAVi struggled to generalize to scenes with a moving camera, even though the optical flow field encodes information about (static) scene geometry in this case.

Here, we identify two key directions for improving SAVi and bridging its capabilities to real-world video data, while preserving its core foundation for learning object representations from video: (1) exploiting depth as a prediction signal, which is readily available in many real-world settings, and (2) utilizing model scaling strategies in terms of encoder improvements and data augmentation, which, despite being commonly used for classic vision problems, are generally underutilized for object-centric learning. Our improved approach, called SAVi++, successfully segments complex objects in naturalistic, real-world video sequences without using direct segmentation or tracking supervision.

Exploiting depth information Training object-centric models solely using RGB image or video frame reconstruction proves challenging in the presence of complex visual textures, frequently leading to failure modes such as clustering by color or into object-agnostic spatial regions [18, 21]. In SAVi, optical flow was proposed as a prediction signal to mitigate this issue, while still operating on visual RGB inputs [37]. However, relying solely on optical flow as a prediction target for learning about objects has a clear disadvantage: static objects, which make up the vast majority of visual entities we encounter on a daily basis, are not captured in this signal unless the observer or the entire scene is in



(a) MOVi datasets. (b) Waymo Open dataset.

	Complex texture	Moving objects	Static objects	Moving camera	#frames (train)
MOVi-C	✓	✓	✗	✗	234k
MOVi-D	✓	✓	✓	✗	234k
MOVi-E	✓	✓	✓	✓	234k
Waymo Open	✓	✓	✓	✓	159.6k

(c) Dataset details and statistics.

Figure 3: We consider three synthetic Multi-Object Video (MOVi) datasets [20] and the large-scale real-world driving dataset Waymo Open [56]. All datasets contain complex textures and moving objects. The MOVi datasets increase in complexity from MOVi-C (moving objects only) over MOVi-D (+static objects) to MOVi-E (+ moving cameras). Waymo Open contains all these characterisics.

motion. As a consequence, SAVi fails to represent objects that are at rest, and similarly struggles with scenes observed from a moving camera, as optical flow can prove challenging to model in this case.

Here, we explore depth as a target signal, used in conjunction with flow or even in isolation. Depth estimation has received little attention in slot-based models, yet does not suffer from the limitation of optical flow in datasets with static objects and camera movement. We thus hypothesize that depth may greatly benefit obtaining emergent object decompositions of complex videos. In terms of practical applicability, we note how depth is a readily-available signal in many real-world settings thanks to the prevalence of RGB-D cameras and LiDAR in settings like self-driving cars [56]. Even in the absence of depth sensing capabilities, this signal can be cheaply estimated from multi-camera systems [40].

In our implementation, we represent the depth signal in image space, which we encode using a log transformation $\log(1 + d)$, where d is the distance of a pixel to the camera (see Figure 3a). This log-transform puts a stronger emphasis on close-by objects and—in early experiments—we found this form of normalization crucial for reliably training object-centric models using depth targets. SAVi++ is then trained to minimize the squared difference between the decoder output and this target signal. In case of multiple available targets, such as depth and flow, we concatenate the target images along the channel dimension and predict them using an otherwise unchanged model.

For sparse targets such as depth obtained from LiDAR, we ignore any points in the image space for which no signal is present in the computation of the loss. For LiDAR specifically, we obtain the x, y, z coordinates of all the LiDAR points in the self-driving car (SDC) world and compute the distance of each of the points from the LiDAR sensor. We then use the camera and LiDAR calibration parameters to project the LiDAR point distances from the SDC domain to the camera frame. This projection represents a very sparse approximation of the ground-truth depth signal (Figure 3b).

Scaling strategies Visual complexity present in real-world videos necessitates a different class of encoders than those used for simple synthetic datasets. Inspired by successful visual backbone architectures for set-based supervised object detection models [7, 32], we use a more capable encoder that utilizes the ResNet34 [22] architecture followed by a transformer encoder [61] (with 4 layers, unless otherwise mentioned). To avoid computation of batch and/or temporal statistics, we replace the typical batch normalization in ResNet34 with group normalization [65]. We use a stride 1 convolution and use no max-pooling in the ResNet root block. This results in an overall backbone stride of 8 (as opposed 32), which was found to be important for retaining object decomposition capabilities. Please see the appendix for further architectural details.

Drawing inspiration from training schemes commonly used for real-world vision models [57], we further apply Inception-style cropping as data augmentation. In particular, we randomly crop a region of each frame with aspect ratio $\in [0.75, 1.33]$ such that enough of the frame is retained after cropping. The same crop is applied consistently across all frames and the resulting video is resized to the original resolution. Flow fields and depth maps are adjusted accordingly to keep them accurate and spatially aligned with the video frame.

4 Experiments

The goal of our experimental evaluation is twofold: 1) on synthetic video data of varying complexity we would like to analyze the potential advantages of utilizing a depth signal and model scaling strategies for learning emergent segmentation and tracking, and 2) we would like to investigate whether these improvements enable bridging the gap to complex real-world video data.

Section 4.1 covers both qualitative and quantitative comparisons of SAVi++ against baselines on the synthetic MOVi datasets. In Section 4.2, we perform an ablation study on SAVi++. Finally, in Section 4.3 we demonstrate and analyze results for a SAVi++ model applied to real-world driving videos from the Waymo Open [56] dataset.

Datasets As basis for our experiments, we use videos of different scene and camera complexities (Figure 3c). We use three synthetic Multi-Object Video (MOVi) datasets (Figure 3a) introduced in Kubric [20], which are created by simulating rigid body dynamics. We narrow our investigation to MOVi datasets with complex naturalistic backgrounds and 3D-scanned everyday objects (variants C, D, and E). MOVi-C is generated using a static camera, and all objects (max. 10) are initialized to move independently. MOVi-D introduces more objects, some of which are dynamic (1-3) and the majority rests statically in the scene (10-20). Finally, MOVi-E introduces random, linear camera movement. Each video contains 24 frames sampled at 12 frames per second (fps).

We also train and evaluate SAVi++ in a real-world driving setting using the Waymo Open dataset (Figure 3b). Waymo Open is comprised of high resolution video data of 1280×1920 original resolution from a multi-camera system collected by Waymo vehicles [56]. The dataset consists of 798 train and 202 validation scenes of 20s video each, sampled at 10 fps. We subsample the dataset at 5 fps both for training and validation. The dataset also includes LiDAR signals that we use to compute sparse depth maps as discussed in Section 3.

Training setup For all our experiments, unless stated otherwise, we resize frames to a height of 128 pixels while keeping the aspect ratio fixed, resulting in a 128×128 resolution for MOVi datasets, and a resolution of 128×192 for Waymo Open. We train SAVi++ for 500k steps on Tensor Processing Unit (TPU) accelerators with a batch size of 64 using Adam [35].

We train on randomly sampled sub-sequences of only 6 frames using 24 slots for MOVi and 11 slots for Waymo Open. See appendix for further training details and hyperparameters.

4.1 SAVi++ improves object-centric learning on complex synthetic video data

We investigate whether the key changes introduced to SAVi [37], which constitute our improved SAVi++ model, allow us to overcome limitations of SAVi and address the most challenging synthetic multi-object video (MOVi) benchmarks introduced in Kubric [20].

Setup We train all models independently on each dataset variant. Both SAVi and SAVi++ are trained in a conditional setting where we initialize slots using ground-truth bounding box information in the first frame. We report the same segmentation metrics as in prior work, i.e. Foreground Adjusted Rand Index (FG-ARI) [27, 51] and Mean Intersection over Union (mIoU). FG-ARI is a permutation-invariant clustering similarity metric frequently used for evaluating scene decomposition quality. It compares discovered segmentation masks with ground-truth masks while ignoring any pixels that belong to the background. It is sensitive to temporal consistency of masks, but insensitive to their ordering. The mIoU metric is a standard segmentation metric, here adapted for video as in [6]. We note that this implementation is sensitive to the correct ordering of masks, i.e. it also measures whether models used the conditioning signal (here, first-frame bounding boxes) correctly.

Baselines Besides comparing to SAVi [37], the most representative prior method for the task we are interested in, we compare against a range of baselines aimed at establishing the difficulty of the unsupervised, bounding box-conditioned video object segmentation task: 1) a bounding box copy (*BBox copy*) baseline, which simply repeats the first-frame boxes throughout the video, 2) a learned *BBox propagation baseline* that does not receive visual inputs, to test for easily exploitable biases in the datasets, 3) k-Means clustering baselines, that cluster the flow and/or depth signal across the video sequence (initialized using the ground-truth object centers in the first frame), and 4) a label propagation baseline, that uses visual features to propagate the initial boxes (rendered as rectangular masks) across the video, based on Contrastive Random Walks (CRW) [28]. See appendix for further details.

Table 1: MOVi results in terms of mean score ± standard error (5 seeds) from evaluating SAVi++ and baseline models on validation set video sequences of increased length (24 frames). *: we use the official implementation of CRW [28], which does not report FG-ARI.

	mIoU↑ (%)			**FG-ARI↑ (%)**		
Model	**MOVi-C**	**MOVi-D**	**MOVi-E**	**MOVi-C**	**MOVi-D**	**MOVi-E**
BBox copy	12.3	42.8	32.9	11.8	68.0	54.7
BBox propagation	22.9 ± 0.1	26.7 ± 0.8	24.1 ± 1.1	9.6 ± 0.5	24.9 ± 3.7	18.4 ± 3.9
K-Means (depth)	7.1 ± 0.3	6.0 ± 0.4	5.4 ± 0.3	26.3 ± 1.0	30.9 ± 0.7	32.2 ± 0.6
K-Means (flow)	10.7 ± 0.5	7.4 ± 0.4	6.0 ± 0.3	26.5 ± 1.0	30.9 ± 0.8	33.1 ± 0.7
K-Means (flow+depth)	10.6 ± 0.6	6.7 ± 0.4	5.3 ± 0.3	26.6 ± 1.0	35.9 ± 1.0	34.8 ± 0.7
CRW [28]	27.8 ± 0.2	45.3 ± 0.0	**47.5** ± 0.1	*	*	*
SAVi [37]	43.1 ± 0.7	22.7 ± 7.5	30.7 ± 4.9	77.6 ± 0.7	59.6 ± 6.7	55.3 ± 5.8
SAVi++ (ours)	**45.2** ± 0.1	**48.3** ± 0.5	**47.1** ± 1.3	**81.9** ± 0.2	**86.0** ± 0.3	**84.1** ± 0.9

Results Quantitative results can be seen in Table 1 and qualitative results on MOVi-E in Figure 4a. The *BBox copy* method serves as a trivial baseline which a learning-based approach should outperform. While the original SAVi model does so on MOVi-C, it clearly fails to model the more complex MOVi-D and -E datasets. The BBox copy baselines is—perhaps unsurprisingly—strongest on MOVi-D, where most objects are static. SAVi++ outperforms this baseline on all datasets, indicating that it learns non-trivial segmentation and tracking capabilities. Indeed, this advantage does not solely come from fitting certain biases in the datasets, as a learned BBox propagation baseline (using the same predictor as in SAVi++) that does not receive visual input, fails to generalize to unseen evaluation videos. It is worth noting that neither of the MOVi tasks can easily be solved by simply clustering the target signals, as the results for the k-Means baselines demonstrate.

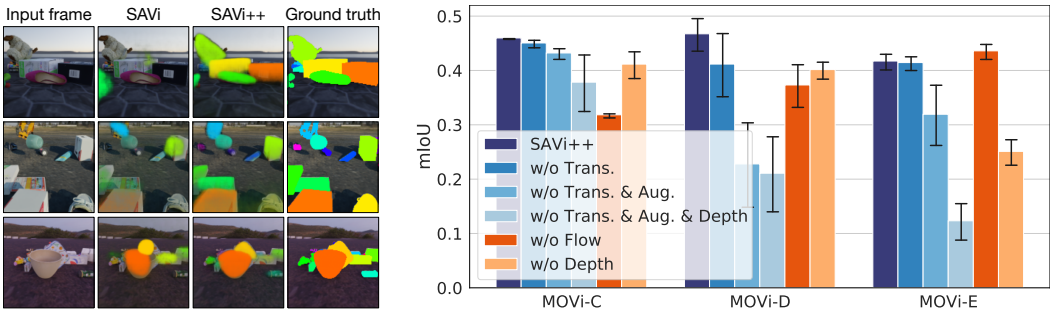
Compared to CRW it can be seen how SAVi++ yields markedly better mIoU on MOVi-C and D, while performance on MOVi-E is similar. Note that, unlike SAVi++, CRW is merely capable of propagating pixel-level annotations across frames in a video and does not by itself produce instance-level object segmentations or corresponding object-representations that could be used for down-stream tasks. Finally, comparing SAVi++ and SAVi directly, we see that SAVi++ overcomes the primary limitations of SAVi on the harder MOVi-D and -E datasets, both quantatively (Table 1) and qualitatively (Figure 4a), while also improving performance on MOVi-C.

Discussion It is evident that a small number of critical changes to SAVi [37], namely utilizing depth targets, a stronger architecture, and data augmentation, can have dramatic consequences on the ability of this slot-based model to learn emergent object segmentation and tracking in complex video sequences. The difference between SAVi++ and SAVi is especially evident for the more complex datasets in our study (e.g., improving the mIoU score on MOVi-E from 30.7% to 47.1%; see also Figure 4a). These results demonstrate that SAVi++ is better suited for various data complexities in terms of object dynamics and camera movement, which are likely to exist in real-world data.

4.2 Ablation study

In this section, we report results of an ablation study to gauge the contribution of the different components of SAVi++. The three main ingredients of SAVi++ are 1) the use of depth as training target, 2) the extra capacity added to SAVi by including a transformer encoder, and 3) the use of data augmentation. Figure 4b shows a systematic ablation of each of those components. Removing the transformer encoder reduces object segmentation quality, yet the degradation in performance is relatively limited. While data augmentation only has a mild effect on the simpler MOVi-C dataset, it makes a substantial difference on the more challenging datasets, MOVi-D and E. Finally, removing depth targets reduces performance further and is particularly catastrophic on MOVi-E.

In fact, we find that training *solely* using depth targets without relying on predicting optical flow as well (see *w/o Flow* in Figure 4b) still allows the model to accurately segment and track objects, especially on the more complex MOVi-D and E datasets. This result is particularly strong on MOVi-E where jointly predicting optical flow presents a difficult task for scenes with camera movement. Further, training on depth targets was very crucial to obtain good performance on the most complex



(a) MOVi-E qualitative results.

(b) Ablation study

Figure 4: **Left**: Qualitative results of SAVi++ compared to SAVi [37] on the synthetic MOVi-E dataset with camera motion. **Right**: SAVi++ ablation study on MOVi-C, D, and E. Bars reflect validation set mIoU (mean $\pm$ standard error for 5 seeds). We ablate: 1) the transformer encoder (*w/o Trans.*), 2) data augmentation (*w/o Trans. & Aug.*), and 3) depth targets (*w/o Trans. & Aug. & Depth*). We further report results for training without flow, while only using depth targets (*w/o Flow*).

Figure 5: Qualitative segmentation comparison on the Waymo Open validation set. Naive application of a SIMONe [30] baseline model to this dataset results in failure, while adapting SIMONe to predict (sparse) depth maps yields rough (but frequently misaligned) segmentation masks. SAVi++ generally produces highly accurate segmentation masks, while its unconditional results are promising. Here, we hide masks that occupy more than 1300 pixels on average per frame to ease interpretability.

synthetic data MOVi-E as demonstrated with the large drop in mIoU when ablating depth and relying only on optical flow to train the model.

4.3 SAVi++ enables emergent segmentation on real-world driving data

In the previous section, we found that solely using depth as a training target can be sufficient to learn emergent object segmentation and tracking. This finding provides a strong motivation for scaling this class of methods to real-world data, where the availability of optical flow relies on approximate and potentially inaccurate flow estimation methods, whereas depth can be accurately measured using technologies like LiDAR. To investigate this possibility, we use the Waymo Open dataset [56], which includes videos obtained from cameras mounted on cars in various traffic environments.

Setup To obtain a depth signal, we project 3D LiDAR points into the camera frame, resulting in a very sparse depth image for each time step (see Figure 3b for examples). We exclude pixels that do not have a valid LiDAR point when computing the L2 loss in image space. We train SAVi++ with 11 slots on 6 frames and evaluate the model on sequences of 10 frames. Due to the absence of ground-truth segmentation labels in Waymo Open, we quantitatively measure performance compared to ground-truth bounding boxes using three metrics. The Center-of-Mass (CoM) distance measures the average Euclidean distance between the centroid of the predicted segmentation masks and the centers of the ground-truth bounding boxes. We report the centroid distance normalized by the maximum achievable distance in the video frame. Additionally, we separately measure the fraction

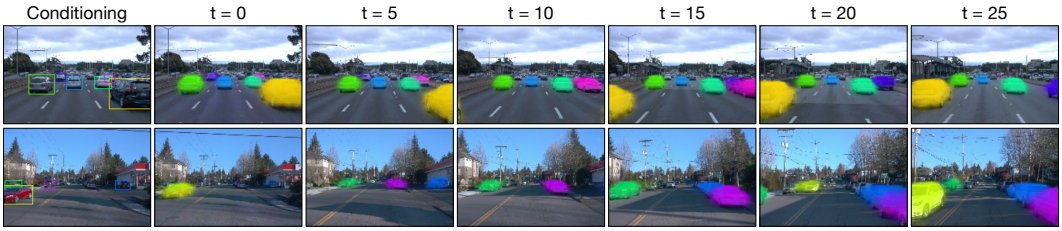


Figure 6: Waymo Open qualitative results of SAVi++ (conditional) over long sequences.

of cases where any sort of segment is predicted when a valid ground-truth box exists, denoted as bounding box recall (B. Recall). The Bounding Box mIoU (B. mIoU) is analog to mIoU using predicted and ground-truth bounding boxes. The former are obtained by training a readout MLP to predict bounding boxes from the slot representations. See appendix for further details.

Baselines We quantitatively compare to the subset of previous baselines that work with (sparse) depth. Further, we report qualitative results for SAVi++ in the unconditional setting, i.e. without providing first-frame bounding boxes to the model to initialize slots, and compare to SIMONe [30] as a representative object-centric video model baseline from the literature.

Results Quantitative results can be seen in Table 2 and qualitative results in Figures 5–6. We find that SAVi++ markedly outperforms the BBox copy and propagation baselines, as well as the clustering baseline in terms of object tracking. Further, the bounding box recall is high indicating that valid objects are rarely ignored. The qualitative results in Figures 5–6 even better reflect the significance of SAVi++'s performance as well as its potential utility for object-centric representation learning from real-world videos (for SAVi++ results divided per object category see Table 4).

Our results using sparse depth targets suggest that SAVi++ does not need complete (i.e. dense) depth supervision. To investigate how *accurate* this signal needs to be, we explored the degree of sensitivity of SAVi++ to noise in the depth signal. We trained SAVi++ with noisy depth targets by applying additive Gaussian noise to the ground-truth sparse LiDAR depth signals with standard deviations of 10cm, 20cm and 40cm. We found that SAVi++ was able to retain its emergent tracking performance even at the highest considered noise scale of 40cm (see Table 5 in Appendix).

We additionally experimented with removing the bounding box conditioning in SAVi++ in the initial frame. Removing this conditioning signal and using a learned initialization together with a simplified encoder also yielded good object decompositions (see SAVi++ (unconditional) in Figure 5). Compared to using plain SIMONe [30], we observe that SAVi++ (unconditional) performs markedly better. Interestingly, modifying the non-autoregressive SIMONe baseline similar to SAVi++ by predicting sparse depth instead of RGB also showed improvement in object emergence. This gives further evidence that using depth is suitable for learning object-centric representations from real videos. Quantitatively, SAVi++ achieves a CoM distance of 6.9 ± 0.5 while SIMONe (with depth loss) achieves 7.4 ± 0.2[1] over a sequence of 12 frames at test time, evaluated using Hungarian matching.

Table 2: Waymo Open results (mean $\pm$ standard error in %, 3 seeds) from evaluating models on sequences of 10 frames. SAVi++ HR is a variant trained on higher-resolution (256×384) video frames.

		(%)	
Model	**CoM↓**	**B. mIoU↑**	**B. Recall ↑**
BBox Copy	5.0	44.3	100
BBox Prop.	5.1 ± 0.1	38.5 ± 0.5	100
K-Means (depth)	13.0 ± 0.1	–	100
SAVi (RGB)	21.5 ± 1.8	7.9 ± 0.9	95.8 ± 2.7
SAVi (depth)	24.7 ± 0.7	10.3 ± 2.4	97.4 ± 0.6
SAVi++	$\mathbf{4.4} \pm 0.2$	$\mathbf{49.7} \pm 0.7$	96.5 ± 0.7
SAVi++ HR	$\mathbf{3.9} \pm 0.1$	$\mathbf{51.9} \pm 0.4$	96.2 ± 0.4
Supervised	1.1 ± 0.0	67.6 ± 0.6	

We show qualitative results for longer sequences in Figure 6 and in video-format in the supplementary material. It is worth noting that SAVi++ was only trained on 6 frames and did not receive any tracking supervision. Interestingly, we find that objects are often consistently tracked until the moment they leave the scene. At this stage, slots are freed up again and tend to bind to previously unexplained or new objects. This behaviour indicates that our reported tracking metrics are an underestimation of the capabilities of the model, as such re-binding is not accounted for. It is, however, conceivable

[1]These baseline results are improved compared to an earlier version of the paper by using exactly the same depth target transformation as for SAVi++.

that re-binding events could be identified post-hoc if one were to use the representations learned by SAVi++ for downstream tasks, which is an interesting avenue for future work.

4.4 Limitations

With SAVi++, we demonstrated the first proof of concept that an emergent object-centric decomposition of real-world complex videos is possible with an end-to-end slot-based approach. Yet, there is still a lot of room for improvement.

Reliance on conditioning We focused our exploration on the conditional setup where we provided cues in the form of bounding boxes of objects in the first frame. Although the use of such "object hints" may share some similarity to how human visual attention (and how humans parse a visual scene) can be directed via external signals (e.g., via gestures such as pointing), it ultimately limits the practical applicability of our approach. Preliminary results with unconditional SAVi++ suggest that this information may not be strictly necessary and could be removed in future research.

Reliance on ground-truth target signals In a similar vein, the reliance of SAVi++ on ground-truth target signals for training is a limitation that may affect its practical applicability. Fortunately, LiDAR sensors for depth estimation are readily available in many application domains (such as in robotics and self-driving), and there is also a rich literature on monocular depth estimation. While estimated, depth (or flow) are expected to be noisier compared to the signals considered in our experiments, our experiment with "noisy depth" offers an initial sign that this may not affect performance much.

Gap to videos recorded in the wild It is also important to point out that although Waymo Open offers a challenging real-world benchmark for learning about objects, its videos are relatively structured compared to real-world videos recorded "in the wild", and especially heavy on cars, roads, traffic signs, pedestrians, etc. Other datasets, such as DAVIS [49] or Kinetics [34] offer greater complexity in that regard and it is foreseeable that further development of SAVi++ will be needed to truly support these. An example of this is that objects in Waymo Open usually do not re-appear, which is an aspect that is currently not explicitly modeled in SAVi++ (e.g. to ensure that the same object is re-captured by the same slot). More generally, there is substantial headroom to improve the modeling of disappearing and reappearing objects in future work, such as by explicitly modeling object presence [39], or by explicitly attending to past latent states [69].

Gap to supervised approaches Finally, we note how both in the conditional and the unconditional setting, the segmentation and tracking performance, though impressive given the minimal amount of supervision the model receives, still qualitatively lags behind supervised approaches. Improving on the temporal consistency of object tracks, especially in the unconditional setting, is another promising direction for future work.

5 Conclusion

We demonstrate that object tracking and segmentation can emerge from utilizing information about scene geometry in the form of depth signals in complex video data with slot-based neural architectures. We utilize a series of synthetic multi-object video benchmarks with increasing complexity to find a simple yet effective set of changes to an existing state-of-the-art object-centric video model (SAVi), allowing us to bridge the gap from synthetic to complex real-world driving videos.

Our work marks a first step towards building end-to-end trainable systems that learn to perceive the world in an object-centric, decomposed fashion without relying on detailed human supervision. While many open challenges remain, this result evidences that object-centric deep neural networks are not inherently limited to simple synthetic environments, and we are excited about the potential for this class of methods to radically reduce the need for human supervision in building scalable perceptual systems for the real world.

6 Acknowledgements

We would like to thank Ben Caine, Alex Bewley and Pei Sun for assistance with self-driving data. We are grateful to Jie Tan, Daniel Keysers, David Fleet, Matthias Minderer, Mehdi Sajjadi and Mario Lučić for general advice and feedback.

References

[1] Zhipeng Bao, Pavel Tokmakov, Allan Jabri, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Discovering objects that can move. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[2] Daniel M Bear, Chaofei Fan, Damian Mrowca, Yunzhu Li, Seth Alter, Aran Nayebi, Jeremy Schwartz, Li Fei-Fei, Jiajun Wu, Joshua B Tenenbaum, et al. Learning physical graph representations from visual scenes. In *Advances in Neural Information Processing Systems*, 2020.

[3] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

[4] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision*, pages 282–295. Springer, 2010.

[5] 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.

[6] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 DAVIS challenge on vos: Unsupervised multi-object segmentation. *arXiv preprint arXiv:1905.00737*, 2019.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[9] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *European conference on computer vision*, pages 436–454. Springer, 2020.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[11] J. Driver, P. McLeod, and Z. Dienes. Motion coherence and conjunction search: Implications for guided search theory. *Perception and Psychophysics*, 51:79–85, 1992.

[12] J. T. Enns and R. A. Rensink. Influence of scene-based properties on visual search. *Science*, 247:721–723, 1990.

[13] J. T. Enns and R. A. Rensink. Sensitivity to three-dimensional orientation in visual search. *Psychological Science*, 1(5):323–326, 1990.

[14] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, June 2018.

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

[16] Google Research. Google scanned objects, 2020. URL `https://app.ignitionrobotics.org/GoogleResearch/fuel/collections/Google%20Scanned%20Objects`.

[17] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, pages 6691–6701, 2017.

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

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

[20] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[21] Adam W Harley, Yiming Zuo, Jing Wen, Ayush Mangal, Shubhankar Potdar, Ritwick Chaudhry, and Katerina Fragkiadaki. Track, check, repeat: An EM approach to unsupervised tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2020. URL `http://github.com/google/flax`.

[24] Olivier J Hénaff, Skanda Koppula, Evan Shelhamer, Daniel Zoran, Andrew Jaegle, Andrew Zisserman, João Carreira, and Relja Arandjelović. Object discovery and representation networks. *arXiv preprint arXiv:2203.08777*, 2022.

[25] Todd S Horowitz, Jeremy M Wolfe, Jennifer S DiMase, and Sarah B Klieger. Visual search for type of motion is based on simple motion primitives. *Perception*, 36(11):1624–1634, 2007.

[26] Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

[27] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[28] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *Advances in Neural Information Processing Systems*, 2020.

[29] Jindong Jiang, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn. SCALOR: Generative world models with scalable object representations. In *International Conference on Learning Representations*, 2020.

[30] Rishabh Kabra, Daniel Zoran, Loic Matthey Goker Erdogan, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P. Burgess. SIMONe: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. In *Advances in Neural Information Processing Systems*, 2021.

[31] Daniel Kahneman, Anne Treisman, and Brian J Gibbs. The reviewing of object files: Object-specific integration of information. *Cognitive psychology*, 24(2):175–219, 1992.

[32] Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. MDETR – Modulated Detection for End-to-End Multi-Modal Understanding. *arXiv preprint arXiv:2104.12763*, 2021.

[33] Laurynas Karazija, Iro Laina, and Christian Rupprecht. ClevrTex: A texture-rich benchmark for unsupervised multi-object segmentation. In *NeurIPS Track on Datasets and Benchmarks*, 2021.

[34] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[36] Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020.

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

[38] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (BiT): General visual representation learning. In *European Conference on Computer Vision*, pages 491–507. Springer, 2020.

[39] 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*, pages 8606–8616, 2018.

[40] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[41] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

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

[43] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[44] Sindy Löwe, Klaus Greff, Rico Jonschkowski, Alexey Dosovitskiy, and Thomas Kipf. Learning object-centric video models by contrasting sets. *arXiv preprint arXiv:2011.10287*, 2020.

[45] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-Object Tracking with Transformers. *arXiv preprint arXiv:2101.02702*, 2021.

[46] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 2021.

[47] K. Nakayama and G. H. Silverman. Serial and parallel processing of visual feature conjunctions. *Nature*, 320:264–265, 1986.

[48] Peter Ochs and Thomas Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[49] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

[50] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[51] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[52] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, October 2021.

[53] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, 2021.

[54] E. S. Spelke. Principles of object perception. *Cognitive Science*, 14:29–56, 1990.

[55] Elizabeth S Spelke and Katherine D Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007.

[56] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[57] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[58] Hao Tian, Yuntao Chen, Jifeng Dai, Zhaoxiang Zhang, and Xizhou Zhu. Unsupervised object detection with lidar clues. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[59] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[60] Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *International Conference on Learning Representations*, 2018.

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[62] Rishi Veerapaneni, John D Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning*, pages 1439–1456, 2020.

[63] Antonin Vobecky, David Hurych, Oriane Siméoni, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, and Josef Sivic. Drive&segment: Unsupervised semantic segmentation of urban scenes via cross-modal distillation. *arXiv preprint arXiv:2203.11160*, 2022.

[64] Lijun Wang, Jianming Zhang, Oliver Wang, Zhe Lin, and Huchuan Lu. SDC-depth: Semantic divide-and-conquer network for monocular depth estimation. In *CVPR*, 2020.

[65] Yuxin Wu and Kaiming He. Group normalization. In *European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[66] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, 2020.

[67] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. *arXiv preprint arXiv:2202.11094*, 2022.

[68] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[69] Yi Zhou, Hui Zhang, Hana Lee, Shuyang Sun, Pingjun Li, Yangguang Zhu, ByungIn Yoo, Xiaojuan Qi, and Jae-Joon Han. Slot-vps: Object-centric representation learning for video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3093–3103, 2022.

[70] Daniel Zoran, Rishabh Kabra, Alexander Lerchner, and Danilo J Rezende. Parts: Unsupervised segmentation with slots, attention and independence maximization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10439–10447, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Limitations section.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Yes, see Societal Impact section in appendix.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Instructions to reproduce experiments are included in the main text and appendix sections. For our code release, see our project website at https://slot-attention-video.github.io/savi++/. We only used publicly available datasets.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see experimental section and appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see experimental section and appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We use publicly available data. For Waymo Open all faces and license plates are blurred and made unrecognizable by default.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Societal impact

Our work focuses on extending object-centric representation learning to real-world videos. This class of methods has the potential for enabling systems to more reliably solve down-stream tasks requiring object representations, such as relational reasoning over entities in videos, and providing better interpretation of model decisions. Some applications that may benefit from this approach include perception in autonomous vehicles, robotics and classic computer vision problems such as object detection. While the method demonstrated in this paper is still far from a state where it could be directly employed in computer vision applications, we would like to raise awareness that—as with most methods developed for computer vision—advances in this field might also aid the development applications with potential negative societal impact such as surveillance.

B Additional results

Qualitative results In Figure 7, we show the effect of our mask thresholding heuristic applied for our unsupervised model visualizations. The intention for this simple heuristic is to aid interpretability of the discovered object segmentation masks. We further show qualitative results for emergent tracking on long sequences in the unconditional setting (i.e. without bounding box conditioning) in Figure 8. Compared to the conditional setting (shown for reference), tracking is less consistent and slots explain not only cars, but also environmental objects or part of the background.

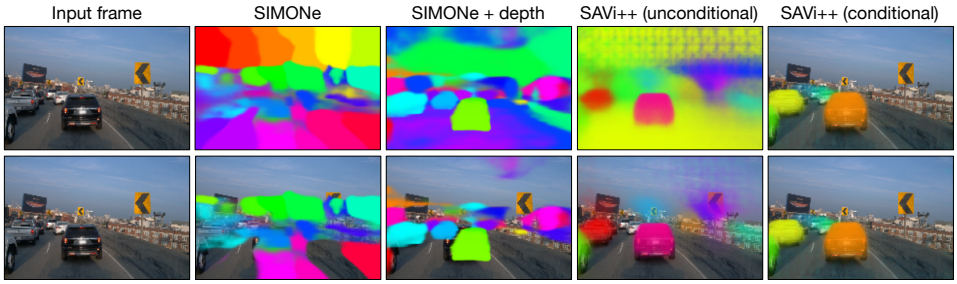


Figure 7: Comparison of qualitative result visualization without (top) and with (bottom) mask thresholding. We apply a simple thresholding heuristic of dropping any masks that (on average across frames) occupy more than 1300px per frame, which aids interpretability. Thresholding does not have an effect on the conditional model.

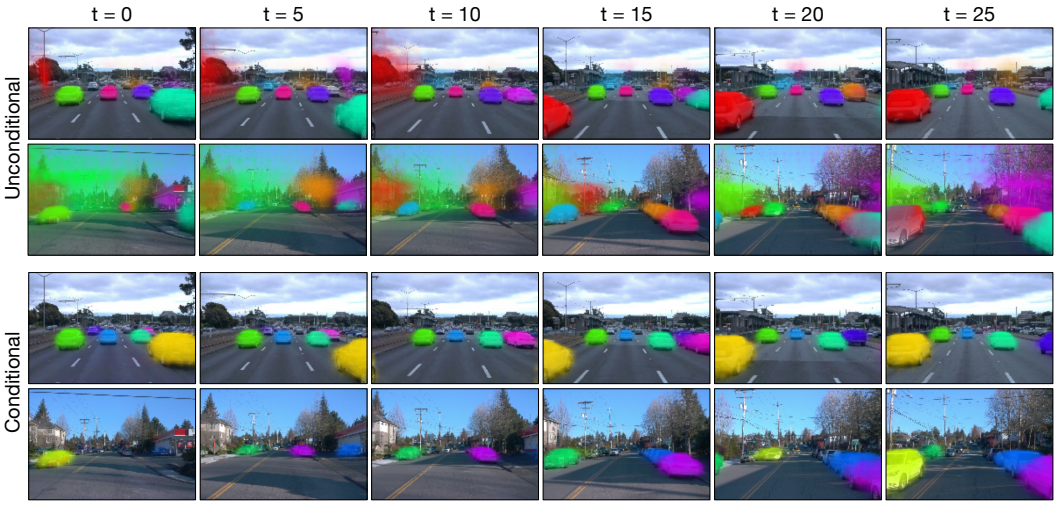


Figure 8: Qualitative results on long Waymo Open validation set videos (for SAVi++ models trained on 6 frames). Top: Results for an unconditional SAVi++ model, trained and evaluated without bounding box conditioning in the first frame. Bottom: Conditional setting shown for reference.

Quantitative results We additionally compare SAVi and SAVi++ on the synthetic MOVi-A and MOVi-B datasets [20]. In MOVi-A, scenes consist of a gray floor, four light sources, a fixed static camera and between 3 and 10 simple geometric objects that vary in terms of their shape (cube, sphere, cylinder), material (rubber, metal), size (small, large), and color (blue, brown, cyan, gray, green, purple, red, yellow). MOVi-B is a straightforward extension, which adds additional variation to the object shape, size, and color; background color; and static camera positions.

In Table 3 it can be seen how SAVi++ performs similar or worse in terms of mIoU and FG-ARI on these datasets. We mainly attribute this to our strategy for scaling in SAVi++, which appears susceptible to overfitting on these much simpler domains. Additionally, the benefit of using depth information is limited, since all objects are in motion for these datasets. The contrast between the results presented in the main paper for MOVi-C to -E and Table 3 (MOVi-A and -B) emphasizes the importance of considering benchmarks that are more representative of the real world for model development.

Table 3: MOVi results in terms of mean score $\pm$ standard error (5 seeds) from evaluating SAVi++ and SAVi models on validation set video sequences of increased length (24 frames).

	mIoU$\uparrow$ (%)		**FG-ARI**$\uparrow$ (%)	
Model	**MOVi-A**	**MOVi-B**	**MOVi-A**	**MOVi-B**
SAVi [37]	82.3 ± 0.3	44.5 ± 9.3	96.8 ± 0.4	73.9 ± 10.7
SAVi++	76.1 ± 0.9	25.8 ± 11.3	98.2 ± 0.2	48.3 ± 15.7

Table 4: Breakdown of SAVi++ results from Table 2 in terms of three classes of objects: car, person, and cyclist.

Metric	Car	Person	Cyclist
Num. objects	15350	2102	275
CoM$\downarrow$	4.2 ± 0.2	6.4 ± 0.1	1.9 ± 0.1
B. mIoU$\uparrow$	52.5 ± 0.8	27.0 ± 0.5	42.2 ± 2.1
B. Recall$\uparrow$	96.7 ± 0.7	95.1 ± 1.1	99.3 ± 0.3

We report per category results for SAVi++ on the WaymoOpen dataset in Table 4. We find that, as expected, this dataset is dominated by cars and performance is very good in this category. SAVi++ also performs very well on cyclists, which are very rare in this dataset, and indicates that SAVi++ has not overfit to blobby car like objects.

We further report results for SAVi++ under the influence of noise in the sparse depth targets on Waymo Open in Table 5. Our results indicate that emergent tracking performance is largely unaffected by noise scales (standard deviations) of up to $\sigma = 40$ cm.

Supplementary videos We provide several video results in the supplementary material for both SAVi++ (conditional, uncondtional, and high-resolution model variants) and the SIMONe [30] baseline (both in its original form and in our adapted depth-prediction variant).

Table 5: Waymo Open results (mean $\pm$ standard error in %, 3 seeds) from evaluating models on sequences of 10 frames. SAVi++ was trained with noisy depth targets with standard deviation (σ) specified below.

		(%)	
Model	**CoM**$\downarrow$	**B. mIoU**$\uparrow$	**B. Recall** $\uparrow$
SAVi++	4.4 ± 0.2	49.7 ± 0.7	96.5 ± 0.7
SAVi++ ($\sigma = 10$ cm)	4.3 ± 0.2	50.1 ± 0.3	96.8 ± 0.3
SAVi++ ($\sigma = 20$ cm)	4.3 ± 0.2	49.7 ± 0.3	97.4 ± 0.5
SAVi++ ($\sigma = 40$ cm)	4.2 ± 0.0	50.1 ± 0.3	96.9 ± 0.5

C Training setup

We train our models for 500k steps (300k steps for the ablation study) on Tensor Processing Unit (TPU) accelerators with a batch size of 64 using Adam [35]. We linearly increase the learning rate for 2500 steps to 0.0002 (starting from 0) and then decay the learning rate with a Cosine schedule [43] back to 0 for the rest of the training steps. We clip the gradients to a global norm value of 0.05 to stabilize training. To create video examples for training, we split each video into sub-sequences of 6 frames each. We use a total of 24 slots on MOVi and 11 slots on Waymo Open for SAVi++ models. We use 1 iteration per frame for the Slot Attention [42] module (as in prior work), unless stated otherwise.

Following the conditional setup in Kipf et al. [37], the initial state of the slots is obtained by encoding bounding boxes corresponding to the objects in the first frame; thus providing the model with rough cues of which objects to bind to initially. For the unconditional experiments, we initialize slots using an equal amount of learnable parameter vectors. In experiments that use optical flow, we convert the 2D flow signal to three RGB channels following prior work [68]. As described above, we apply a log-transform to the (sparse) depth signal (incremented by 1 to avoid underflow). We train our model to minimize the squared error (L2 loss) between the predicted and ground-truth targets in pixel space. We implement SAVi++ in JAX [3] using the Flax [23] neural network library. Training SAVi++ on a single MOVi dataset on 8 TPUv4 chips with 32GiB memory each takes approximately two days for 500k training steps.

D Model details

D.1 SAVi++

Our architecture building blocks are similar to that of the SAVi model from Kipf et al. [37] with all the parameters shared across time steps. SAVi++ uses exactly the same parameters for the slot initializer and decoder as the SAVi model (except for SAVi++ HR where we use an additional 5×5 ConvTranspose layer with stride 2 and 64 channels to account for the higher resolution frame size). Below we list the details and hyper parameters of all the modules of SAVi++:

Encoder We used a ResNet-34 [22] backbone with modified root convolutional layer that has 1×1 stride (except for SAVi++ HR that uses a root stride of 2×2). For all layers, we replaced the batch normalization operation by group normalization [65]. We used a linear positional encoding identical to that used in Slot Attention [42] with horizontal and vertical coordinates normalized to $[-1, 1]$ range. These coordinates were then projected to the same size of the ResNet feature maps using a learnable linear layer. Finally, the ResNet features and positional encoding are combined by an addition operation. Following the backbone, the frame features are projected to 64 embedding dimensions by a linear layer followed by ReLU activation and then fed to a transformer network with 4 transformer blocks, except for SAVi++ unconditional Waymo Open models where we found that ResNet features alone produced clearer segmentations. This is likely due to the stronger CNN image prior being beneficial with less supervision [59]. Each transformer block uses a multi-head dot-product attention from [61] with pre-normalization [66] and 4 attention heads. For each attention head, the query/key/value embedding size was set to 16. The output of each block is then processed by a residual feed-forward block with pre-normalization, using an MLP with a single hidden layer of 1024 hidden units and ReLU activation function.

Corrector/Predictor Similar to SAVi [37], we use 1 iteration for the Slot Attention corrector module. We increase the corrector query/key/value size to 256 compared to 128 embedding size used in SAVi. For the predictor, we similarly increased the query/key/value projection size to 256, and the MLP hidden layer size to 1024. We found that the larger embedding sizes for the corrector and predictor increased SAVi++ mIoU by a few percentage points.

Decoder Our decoder follows that of SAVi [37] with two exceptions: For Waymo Open, we use a larger spatial broadcast grid of 8×12 as video frames have 128×192 resolution. For SAVi++ HR, we use frames of a higher resolution of 256×384 and add an additional 5×5 ConvTranspose layer with stride 2 and 64 channels to account for the higher resolution frame size. The decoder otherwise uses four 5×5 ConvTranspose layers with stride 2 and 64 channels, followed by ReLU activations.

Initializer Similar to SAVi, we consider two initializers *conditional* and *unconditional* to set the initial state of the models' K slots. For the conditional case, the initializer mapped each of K bounding boxes (represented using 4D coordinates) via a trainable MLP to set the D dimensional state of the corresponding slot. For the unconditional case, rather than associating a specific bounding box in a specific video to a slot, we learn K D-dimensional initial slot states to be used in all videos.

Data augmentation As discussed in the main text, an Inception style [57] random crop is used for data augmentation. Crops are afterwards resized to the target resolution (128×128 for MOVi and 128×192 for Waymo Open, unless otherwise mentioned). We ensure that the cropped view covers at least 20% of the original frame in the MOVi datasets and 75% in Waymo Open. We found that more aggressive cropping was helpful for the synthetic MOVi datasets and less aggressive cropping worked best on Waymo Open. We take care to handle depth maps and flow fields during this operation as explained next for each of the datasets. *MOVi:* Say the crop size is $h \times w$, then optical flow fields are cropped, resized, and then rescaled by $\left[\frac{128}{w}, \frac{128}{h}\right]$. Dense depth maps are simply cropped and resized. *Waymo Open:* Sparse LiDAR point clouds are projected to 2D and retained as tuples (2D point, lidar-range) throughout the data augmentation pipeline. The affine transformation equivalent to the crop and resize operation is computed and applied to these 2D points. As a consequence, several points will fall outside the cropped frame. These points are discarded when projecting them into a depth image after data augmentation is complete.

D.2 Baselines

SAVi For the SAVi baseline, we use the best-performing model variant described in Kipf et al. [37], i.e. SAVi trained with a Resnet34 backbone. Different from SAVi++, this baseline does not use depth prediction (it only predicts optical flow), does not use data augmentation, and does not use a transformer encoder after the convolutional backbone. We choose the same hyperparameters as described in SAVi [37].

SIMONe This baseline [30] is a non-autoregressive model for encoding short video clips of fixed lengths into a set of latent object variables (fixed across time) and a per-frame global latent variable. Note that SIMONe cannot be applied auto-regressively and has to be applied to the same sequence length at both training and test time. SIMONe uses a CNN encoder per frame followed by a transformer encoder that is applied across frames to finally obtain object and frame latent variables by pooling transformer tokens across time and space, respectively. The model is trained by reconstructing input frames using a form of a spatial broadcast decoder and additionally uses a KL-based regularizer on the latent variables. We use a JAX [3] reimplementation of the SIMONe model for which we verified that it reproduces results mentioned in the paper on the CATER [15] dataset. We train SIMONe on sequences of 6 frames (same as SAVi++) at a resolution of 128. We subsample reconstruction targets by a factor of 4 (see Kabra et al. [30] for details). Other hyperparameters are chosen as follows: reconstruction loss scale $\alpha = 0.2$, pixel likelihood scale $\sigma_x = 0.08$, object latents KL loss weight $\beta_o = 1\mathrm{e}-5$, and frame latents KL loss weight $\beta_f = 1\mathrm{e}-4$. We encode frames using a 4-layer CNN with 128 channels, $(4, 4)$ kernel size and $(2, 2)$ stride. Each transformer uses 4 layers, 5 heads, a qkv-size of 64 per head, and an MLP hidden layer size of 1024. Latents are of size 32. The decoder uses an MLP with 5 hidden layers of 512 units. To train SIMONe with sparse depth targets, we replace the RGB target signal with the LiDAR-based depth signal and only compute the loss for pixels that have a depth signal.

CRW Contrastive Random Walks (CRW) [28] is a cycle consistency based self-supervised learning method for learning grid structured latent representations. After pre-training, a simple label propagation scheme can be applied on these latent representations to obtain tracking behavior. This typically requires segmentation labels for the objects of interest in the first frame. The method tracks these objects and outputs segmentation masks for them over subsequent frames. In order to use this method with only bounding box conditioning in the first frame, we flood fill boxes into rectangular masks and propagate those instead. Overlap between boxes is resolved based on the box order.

We adopted their training and evaluation best practices. We pre-trained stride-8 ResNet backbones [22] using their publicly available code and propagated labels using the activations output by the second last ResNet stage. For pre-training we tuned edge-dropout, training temperature and for tracking we tuned evaluation temperature independently on each of the three MOVi datasets. We

found that, despite our efforts, a ResNet34 backbone was not able to train using the cycle consistency loss. We obtained much better results using a ResNet18 backbone, which is the model we report results for. Optimal hyper-parameters (dropout, training temperature, evaluation temperature) were as follows: MOVi-C $(0.0, 0.001, 0.5)$, MOVi-D $(0.05, 0.001, 0.5)$, MOVi-E $(0.05, 0.001, 0.5)$. Other relevant hyper-parameters are: training clip length (6), frame-skip (1), batch size (16), learning rate (0.0001), training epochs (125) with a learning rate drop after the 100th epoch.

Bounding box copy We simply repeat the bounding boxes of the objects visible in the initial frame for the rest of the video sequence. To obtain pixel-level segments for computing metrics, such as FG-ARI, we 'render' the entire bounding box as a segment in pixel space. Bounding boxes are rendered in the same order as they were provided in the initial frame, such that later boxes take precedence when multiple of them cover the same pixel location.

Learned bounding box propagation In this baseline, we use the SAVi++ initializer and predictor (without encoder, corrector, or decoder) to learn a bounding box propagation model. The model receives (just as in SAVi++) bounding boxes for all objects in the first frame of the video, which are passed to the initializer to learn initial slot representations. Afterwards, the predictor learns a mapping of slots at time step t to slots at time step $t + 1$. We train the model by reading out individual slot representations at each time step using an MLP with a single hidden layer of 256 units that predicts the corner coordinates (top-left and bottom-right) of the bounding box associated with a slot at a particular time step, supervised using ground-truth bounding boxes. We use the Huber [26] loss (L2 loss between $[-1, 1]$ and L1 loss outside of this interval) to train the model. If an object is not visible or present, its bounding box is set to $[0, 0, 0, 0]$ in the ground-truth target.

K-Means clustering of flow/depth To evaluate how much information about instance segmentation can be exctracted directly from the depth and optical flow modalities, we evaluate a k-Means clustering baseline on videos from MOVi and Waymo Open. For that purpose we treat each pixel of a video as a datapoint, each with 7 dimensions: one for log-depth ($\log 1 + d$), three for optical flow converted to RGB (only for MOVi), two for linear position encoding, and one for time. All dimensions are normalized to the range of $[0, 1]$. For Waymo Open we discard all points that do not have an associated depth value. To make it as comparable as possible to the conditional setup of SAVi++, we set k to the ground-truth number of objects plus one for the background, and initialize each cluster-center to the average value of points within the first-frame bounding box of each object. The background cluster is initialized to the average value of all points in the first frame. K-Means is then run until convergence, and we evaluate the resulting cluster-assignments using mIoU and FG-ARI scores for MOVi, and by computing the normalized distance of the center of mass of each segment to the corresponding bounding box for Waymo Open.

Supervised baseline To estimate how much headroom there is in terms of tracking performance given our model architecture, we train a variant of the SAVi++ model where we replace the depth decoder with a bounding-box prediction head. Instead of self-supervised training using depth prediction, this model is trained to directly predict object bounding boxes at every time step given the slots of the model. This is similar to a TrackFormer [45] model, but we instead use the SAVi++ architecture and we train in a conditional setting (i.e. initial first-frame bounding boxes are provided as slot initialization), which means we can train the model without using any form of matching. Similar to the learned bounding box propagation baseline, we train the model by reading out individual slot representations at each time step using an MLP with a single hidden layer of 256 units that predicts the corner coordinates (top-left and bottom-right) of the bounding box associated with a slot at a particular time step, supervised using ground-truth bounding boxes. We apply a Huber [26] loss (L2 loss between $[-1, 1]$ and L1 loss outside of this interval) for training. If an object is not visible or present, its bounding box is set to $[0, 0, 0, 0]$ in the ground-truth target.

E Datasets

We used the synthetic Multi-Object Video (MOVi) datasets introduced in Kubric [20]. The Kubric dataset generation pipeline is available under an Apache 2.0 license. For real-world experiments, we used the Waymo Open dataset [56]. The Waymo Open dataset is licensed under the Waymo Dataset License Agreement for Non-Commercial Use (August 2019): `https://waymo.com/open/terms`.

Dataset details are summarized in the following:

- **MOVi-C**: uses approximately 380 high-resolution HDR photos as backgrounds and three to ten dynamic objects obtained from a set of 1028 3D-scanned everyday objects [16], representing various household objects. The camera in this dataset has a random pose, yet the camera pose is static across the video sequence. Each video is sampled at 12 frames per second (fps). We trained our models on randomly sampled sequences of 6 frames and evaluate on sequences of 24 frames. We trained our models on 9.75k training set videos, and evaluated models on 250 evaluation set videos. Model tuning was performed on a separately generated set of 250 videos.
- **MOVi-D**: has a similar camera setting as MOVi-C, but adds more objects, the majority of which are initialized to be static. This dataset includes one to three dynamic objects and 10 to 20 static objects in each video sequence. We used the same frame rate and train/evaluation sequence lengths as for MOVi-C. We trained our models on 9.75k training set videos, and evaluated models on 250 evaluation set videos. Model tuning was performed on a separately generated set of 250 videos.
- **MOVi-E**: adds additional complexity compared to MOVi-D by introducing random linear camera movement throughout the video sequence. We used the same frame rate and train/evaluation sequence lengths as for MOVi-C. We trained our models on 9.75k training set videos, and evaluated models on 250 evaluation set videos. Model tuning was performed on a separately generated set of 250 videos.
- **Waymo Open**: contains high-resolution video sequences with frame size of 1280×1920. We solely use videos recorded from the front camera of the car. We down-sampled the videos to 128×192 (or 256×384 for SAVi++ HR). The dataset consists of 798 train and 202 validation scenes of 20s video each, sampled at 10 fps. The dataset includes also 2D bounding box annotations, which we used for the conditional experiments and to compute the B. mIoU evlauation metrics. The Waymo Open dataset further includes LiDAR data collected from five LiDARs; one mid-range LiDARs placed on top of the car and four short-range LiDARs placed front, left, right, and rear. The LiDAR data is used to compute sparse depth targets as discussed in the Methods section. To slightly simplify the task as we train on lower-resolution frames, we discard any bounding box labels in Waymo Open which cover an area of 0.5% or less of the first sampled video frame, both during training and testing.

F Metrics

In the following, we give a detailed overview of the metrics used for each of the datasets.

F.1 MOVi

On the MOVi datasets [20] we have access to ground-truth pixel-level segmentations, which lets us directly measure the quality of the learned segmentations using the same segmentation metrics as in prior work. Note that SAVi and SAVi++ are trained in a conditional setting where we initialize slots using ground-truth bounding box information in the first frame. Because of this, we will only measure metrics from the second frame onward.

Foreground Adjusted Rand Index (FG-ARI) A permutation-invariant clustering similarity metric frequently used for evaluating scene decomposition quality [27, 51]. It compares discovered segmentation masks with ground-truth masks while ignoring any pixels that belong to the background. It is sensitive to temporal consistency of masks, but insensitive to their ordering.

Mean Intersection over Union (mIoU) A standard segmentation metric for measuring the quality of predicted segments. Our implementation is identical to the semi-supervised DAVIS challenge Jaccard-Mean metric for video [6, 50]. We note that this implementation is sensitive to the correct ordering of masks, i.e. it also measures whether models used the conditioning signal (here, first-frame bounding boxes) correctly.

Model selection on MOVi was done mainly using mIoU. On the one hand to avoid learned segments bleeding into the background overly much, and on the other hand to ensure that the bounding box initialization was properly utilized.

F.2 Waymo Open

On the Waymo Open dataset we only have ground-truth bounding boxes available, which necessitates an alternative set of metrics for measuring quantitative performance. Similar to before, because of conditioning, we will only measure metrics from the second frame onward.

Center-of-Mass (CoM) distance This tracking metric measures the average Euclidean distance between the centroid of the predicted segmentation masks and the centers of the ground-truth bounding boxes. The former are obtained by computing the geometric mean of the 2D coordinates associated with the pixels belonging to a segment, where we exclude pixels that do not have a valid LiDAR point associated with them (this is similar to how we compute the loss during training). To allow for comparable CoM distance across multiple resolutions, we report the distance normalized by the maximum achievable distance in the video frame (length of the diagonal). Objects that are fully occluded in a frame (or have disappeared) are excluded from the computation. In the conditional case (i.e. with bounding box information provided to the model in the first frame) we use the order of the provided bounding boxes to compute the metric between each slot and ground-truth bounding box. In the unconditional case, we use Hungarian matching to associate entire bounding box tracks with decoded object masks and we assign a penalty of 1 (maximum CoM distance) for all empty segments.

Bounding Box Recall (B. Recall) This metric measures the fraction of cases where any sort of segment is predicted when a valid ground-truth box exists. It serves a complement to CoM distance when no matching is used and empty segments are not considered. In the case of unconditional evaluation using Hungarian matching, we incorporate a matching penalty of the maximum possible CoM distance for empty segments and thus do not separately report segment recall.

Bounding Box mIoU (B. mIoU) This metric is the bounding box analog of the segmentation mIoU discussed above. Given corresponding predicted and ground-truth bounding box tracks, their per-frame intersection-over-union is computed and averaged over time exactly as in the average IoU metric of the TAO benchmark [9]. Predicted bounding box tracks are obtained using a per-slot readout MLP, with one hidden layer of 256 units. This is jointly trained with the SAVi++ model by minimizing the Huber loss [26] between predictions and $[0, 1]$ normalized ground-truth box coordinates. A stop-gradient is used to prevent these loss gradients from propagating back into SAVi++. Objects that are fully occluded across the entire video sequence are excluded from the computation.

We initially conducted model selection on Waymo Open using a combination of B. mIoU and a heuristic metric to measure what fraction of the pixels belonging to a predicted segment are inside the associated ground-truth bounding box. During the final stages of development, we primarily focused on the B. mIoU metric since it is analogous to mIoU on MOVi.