Supplementary Material for
OCK: Unsupervised Dynamic Video Prediction with Object-Centric Kinematics

1. Dataset details

We evaluate all methods on synthetic and real-world datasets to establish a robust framework for assessing model performance in dynamic video prediction. Emphasizing multi-object interactions within complex scenes, these datasets have been meticulously selected to challenge the model across a spectrum of tasks, ranging from fundamental geometric object tracking to the interpretation of complex interactions under dynamic conditions.

OBJ3D [6] comprises 2920 training and 200 testing sequences featuring rigid objects in synthetic scenes with neutral grey backgrounds. The dataset simulates dynamic environments by animating objects with distinct shapes, textures, and attributes. Each frame is accompanied by its corresponding object-centric annotations, including 3D spatial orientation, geometric properties, and semantic classifications. The simplicity of the scenes facilitates a focused assessment of the model's geometric comprehension and multi-object motion tracking capabilities.

MOVi-A [4] comprises 9703 training and 250 testing sequences, offering a diverse range of visual content. This dataset simulates basic object interactions within a simple visual environment, incorporating variations in lighting conditions, occlusions, and object appearances. Each scene consists of 3 to 10 randomly positioned objects on a grey floor, captured by a stationary camera oriented towards a fixed origin point.

MOVi-B extends the MOVi-A dataset by incorporating eight additional object shapes and introducing varied background colors. It features dynamic camera placement, with the viewpoint directed towards the scene center and exhibiting diverse scale variations across the dataset. The MOVi-B dataset is designed to highlight more complex object motions, interactions, and a broader diversity of object shapes and colors compared to MOVi-A. As such, MOVi-B is specifically designed to evaluate the model's advanced tracking capabilities and its comprehension of intricate object dynamics in varied visual contexts.

MOVi-C features authentic video sequences of moving objects against diverse backgrounds, incorporating varied lighting conditions and contextual arrangements. This dataset contains genuine objects sourced from the Google Scanned Objects (GSO) dataset [2]. For each video, the background and ground surface are randomly generated using Poly Haven. MOVi-C is specifically designed to evaluate model performance under complex, real-world-like scenarios, presenting a substantially more challenging task due to its intricate object and background compositions.

MOVi-D increases scene complexity by incorporating 10 to 20 static objects and 1 to 3 dynamic objects per scene, with most objects initially positioned on the floor. This dataset is designed to evaluate the model's proficiency in handling stationary objects frequently occluded by a few dynamic objects. By simulating real-world scenarios where objects of interest are not isolated but are rather surrounded by numerous distractors, MOVi-D enables a thorough evaluation of the model to distinguish and track objects amidst complex, cluttered environments. This configuration closely mimics real-world scenarios, providing a robust benchmark for assessing advanced object tracking capabilities.

MOVi-E further extends the complexity from MOVi-D by introducing linear camera movement at a constant velocity, while maintaining focus towards the center of the scene. This dataset preserves the continuity of MOVi-D in terms of the number of objects and their specifications. Among the variations of the MOVi datasets, MOVi-E is the most challenging due to the simultaneous motion of both viewpoints and objects. This configuration provides the most rigorous evaluation for handling complex interactions in dynamic environments, closely approximating real-world scenarios. Consequently, MOVi-E serves as an optimal benchmark for assessing advanced model capabilities in object tracking, scene understanding, and motion prediction.

Waymo Open Dataset [8] requires specific preprocessing to align with object-centric model training schemes. Specifically, we utilize the perception version of the dataset, where images are extracted from *camera_image* converted from byte to PNG, cropped, and resized to $[64 \times 64]$. Image masks from the vehicle asset label set are processed in a similar manner. Bounding boxes and center coordinates are extracted from *camera_box* in (x,y) format. We assess performance quantitatively by benchmarking against ground-truth bounding boxes following Elsayed et al. [3].

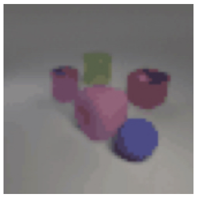
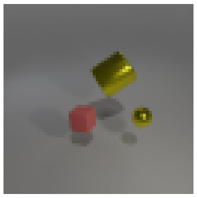
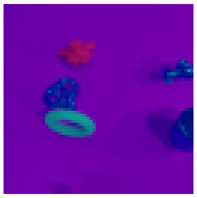
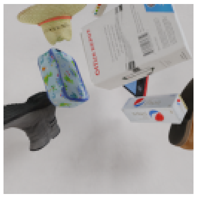
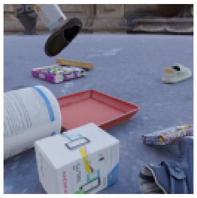
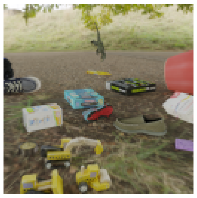
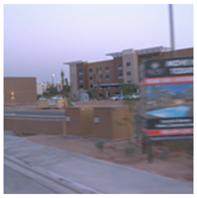
	Sample Frame	Textured Objects	Moving Objects	Moving Camera	Natural	Type
OBJ3D		No	Yes	No	No	Synthetic
MOVi-A		No	Yes	No	No	Synthetic
MOVi-B		No	Yes	No	No	Synthetic
MOVi-C		Yes	Yes	Yes	No	Synthetic
MOVi-D		Yes	Yes	Yes	No	Synthetic
MOVi-E		Yes	Yes	Yes	Yes	Synthetic
Waymo Open Dataset		Yes	Yes	Yes	Yes	Real

Table 1. Overview of datasets comparing attributes such as object texture, motion, camera movement, and natural settings.

		OBJ3D	MOVi-A	MOVi-B	MOVi-C	MOVi-D	MOVi-E	Waymo
Slot Encoder	Number of slots	6	11	11	26	26	26	18
	Slot size	128	128	128	128	128	128	128
	Resolution	[64,64]	[64,64]	[64,64]	[64,64]	[64,64]	[64,64]	[64,64]
	Batch size	16	16	16	16	16	16	16
	Epochs	1200	1500	1500	1500	1500	1500	1000
	Warmup steps	1500	2500	2500	2500	2500	2500	2000
	Iterations	1	2	2	3	3	3	2
Kinematics Encoder	Kins. size	6	11	11	26	26	26	16
	Resolution	[64,64]	[64,64]	[64,64]	[64,64]	[64,64]	[64,64]	[64,64]
	Heads	4	4	4	4	4	4	4
	Acceleration δ_A	0.01	0.01	0.01	0.01	0.01	0.01	0.05
	Trans. layers	2	2	2	2	2	2	2
	Embedding dim	128	128	128	128	128	128	128
	Hidden dim	256	256	256	256	256	256	256
Transformer	Input frame	5	6	6	6	6	6	6
	Predicted frame	5	8	8	8	8	8	8
	Epochs	1500	1500	1500	2500	2500	2500	1500
	Batch size	16	16	16	8	8	8	16
	Learning rate	1e^{-4}	2e^{-4}	2e^{-4}	2e^{-4}	2e^{-4}	2e^{-4}	1e^{-4}
	Trans. layers	2	4	4	4	4	4	4
	Heads	4	8	8	8	8	8	4
	Embedding size	128	256	256	256	256	256	256
	Loss weight α	1	1	1	1	1	1	1

Table 2. List of hyperparameters for the encoders and the transformer modules across six synthetic datasets and one real-world dataset. "Kins" refers to the kinematics encoder and "trans" refers to the transformer.

2. Experimental details

The base object-centric model comprises 400K to 900K parameters, and our model averages approximately 4M parameters. These models can be trained on a single machine equipped with one RTX 3090 GPU, or scaled up to six GPUs for accelerated training. The autoregressive object-centric transformers require approximately 100 hours on a single GPU. Table 2 delineates the specific hyperparameter configurations of the OCK module.

3. Implementation details

3.1. Slot Attention for Video

Slot Attention for Video (SAVi) [5] is an encoder-decoder video model based on Slot Attention [7], which utilizes optical flow as a prediction target to derive object-centric representations of dynamic scenes. The model relies on motion flow prediction as its primary training signal, which limits its applicability to scenarios where all objects in a scene exhibit independent motion. Consequently, SAVi encounters challenges when generalizing to scenes with a moving camera, despite the optical flow field containing information about the static scene geometry in such instances. This limitation is the main reason why the autoregressive object-centric transformer modules for dynamic video prediction do not perform well in complex environments.

3.2. Object Kinematics

Recall that Object Kinematics encompasses the position, velocity, and acceleration states, represented via the center coordinates $[x, y]$ of object features. The total number of Object Kinematics corresponds directly to the number of slots extracted per video frame. The process begins with the initialization of slots using learnable queries, followed by iterative slot updates to generate a sequence of future slots.

3.2.1. Analytical approach

The analytical approach leverages the temporal subsequent kinematics of future frames as guidance by forecasting the subsequent kinematics based on input frames via logical reasoning and concatenating the prediction results with the current kinematics information as follows:

$$\mathbf{x}_{t+1}^{\text{pos}'} = \mathbf{x}_t^{\text{pos}} + \mathbf{x}_t^{\text{vel}} \times \delta_T + \mathbf{x}_t^{\text{accel}} \times \delta_A, \tag{1}$$

where δ_t is the time difference between two consecutive timesteps and δ_A is a parameter that controls the kinematics information. The corresponding velocity and acceleration states at that timestep are calculated accordingly. As the acceleration state is insignificant to the overall position state computation, we ignore it during training.

The analytical approach is particularly effective for several reasons. First, it maintains temporal continuity by simulating the natural flow of objects through space, rather than relying on abrupt changes. Second, it enhances efficiency by focusing on the movement of key objects, rather than analyzing every pixel in video frames. Lastly, the approach prioritizes accuracy by incorporating both the current state and the predicted state, allowing for adjustments in predictions when objects change in motion or direction.

3.2.2. Empirical approach

The empirical approach exclusively utilizes observed motion properties of objects within the current frame. This methodology assesses transformers' capability to learn object motion dynamics in video sequences, with limitations in generalizing to extended temporal sequences. Consequently, this approach serves as a baseline for evaluating the analytical method's performance, offering a frame-specific benchmark for comparison.

3.3. OCK transformers

Following the design principles of BERT [1], we construct our transformer by stacking multiple transformer encoder blocks. Our framework utilizes Adam optimizer with a batch size of 16. The initial learning rate is set to 0.0001 for OBJ3D, 0.0002 for MOVi, and 0.0001 for Waymo Open Dataset, which decays according to a cosine schedule until reaching 0. We incorporate a linear learning rate warmup strategy during the initial 5% of training steps to facilitate smoother convergence. Throughout the training phase, we do not apply gradient clipping or weight decay, maintaining the simplicity and efficiency of our framework.

3.3.1. Joint-OCK

Initially, we concatenate object slots and the Object Kinematics, then linearly project the input sequence of object attributes to a latent space to match the inner dimensionality of the transformer mechanism as $U_t = \text{Linear}([\mathcal{S}_t, \mathbf{K}_t])$. We then add positional encoding to the latent object embedding space. In this way, at each timestep, object slots receive the same positional encoding.

3.3.2. Cross-OCK

This approach follows Joint-OCK to some extent, but the input to the transformers requires additional calculations. The details of the processing are omitted as they are already explained clearly in the method section of the main paper.

3.4. Model training

To construct the OCK transformer modules, we utilize a standard Transformer encoder $\mathcal{T}$ with N_T layers. The input sequence of slots is first linearly projected into a latent space U_t of dimension D_s, ensuring compatibility with $\mathcal{T}$. Then, positional encodings are added to the latent embeddings to encode the temporal order of the input slots. Instead of assigning sinusoidal positional encodings independently to each slot regardless of its timestep, following Wu et al. [9], we apply positional encoding $P_t \in \mathbb{R}^{N \times D_s}$ at the temporal level across all N slots at timestep t, to ensure that all slots at the same timestep share identical positional encodings. In this way, our module maintains permutation equivariance among slots, which is a critical property of object-centric video prediction. Mathematically, the input to the transformer $V \in \mathbb{R}^{(TN) \times D_s}$ is as follows:

$$Z = [U_1, U_2, U_3, ..., U_T] + [P_1, P_2, P_3, ..., P_T]. \tag{2}$$

This temporal encoding strategy improves prediction performance while maintaining efficiency [9]. The transformer $\mathcal{T}$ processes the input sequence of slots Z to model temporal object scene dynamics, generating output features W. We then take the last N features $W_T \in \mathbb{R}^{N \times D_s}$ and apply a linear transformation to obtain the subsequent set of slots at timestep $T+1$ as follows:

$$W = \mathcal{T}(Z) \quad \hat{S}_{T+1} = \text{Linear}(W_T). \tag{3}$$

For future predictions, $\hat{S}_{T+1}$ is treated as ground truth and combined with prior slots $\{S_t\}_{t=2}^{T}$. This iterative process enables the autoregressive generation of future video frames over any desired horizon, $H > 0$.

4. Additional qualitative results

In this section, we provide more qualitative results to complement our quantitative findings as follows:

- Video prediction on MOVi dataset in Fig. 1
- Kinematic trajectories of MOVi dataset in Fig. 2.
- Video prediction on Waymo Open dataset in Fig. 3.
- Scene decomposition on MOVi datasets in Figs. 4 and 5.

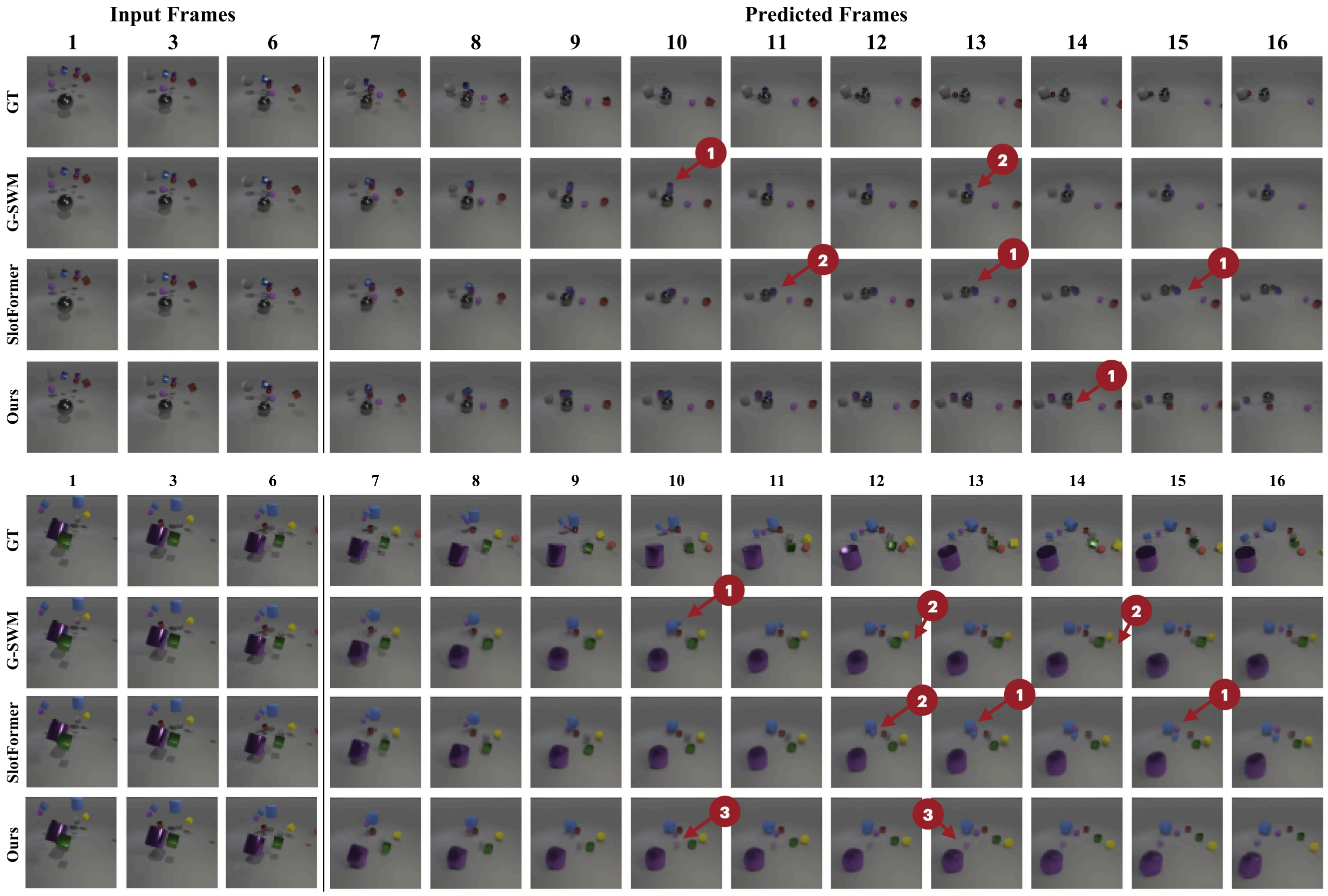


Figure 1. Additional qualitative results on MOVi-A dataset. The objects that all models have failed to detect or track in common have been neglected. 1: Wrong dynamics, 2: Missing object(s) when compared to other models, 3: Blurry object(s).

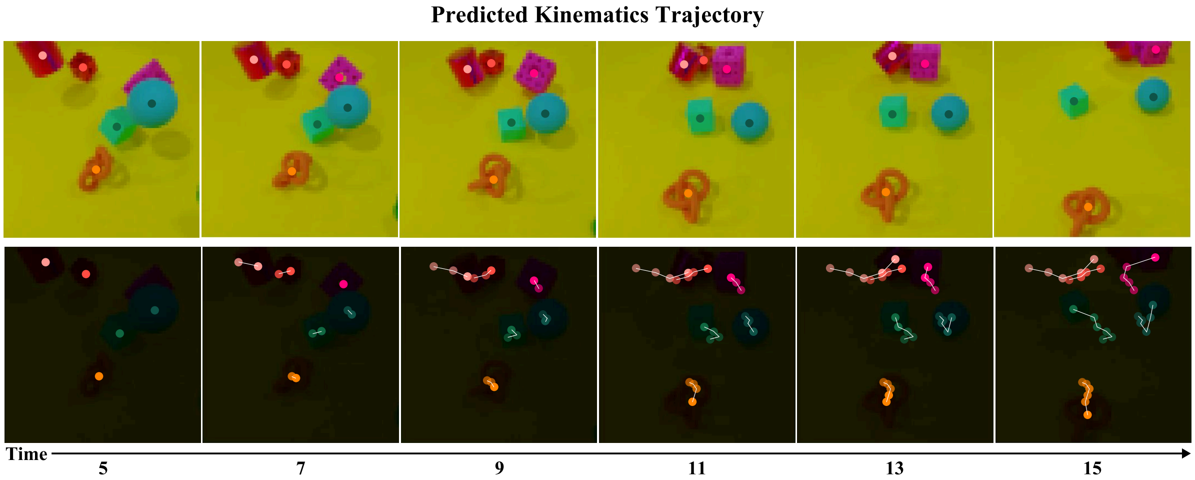


Figure 2. Additional qualitative results on predicted kinematics trajectory on MOVi-B dataset.

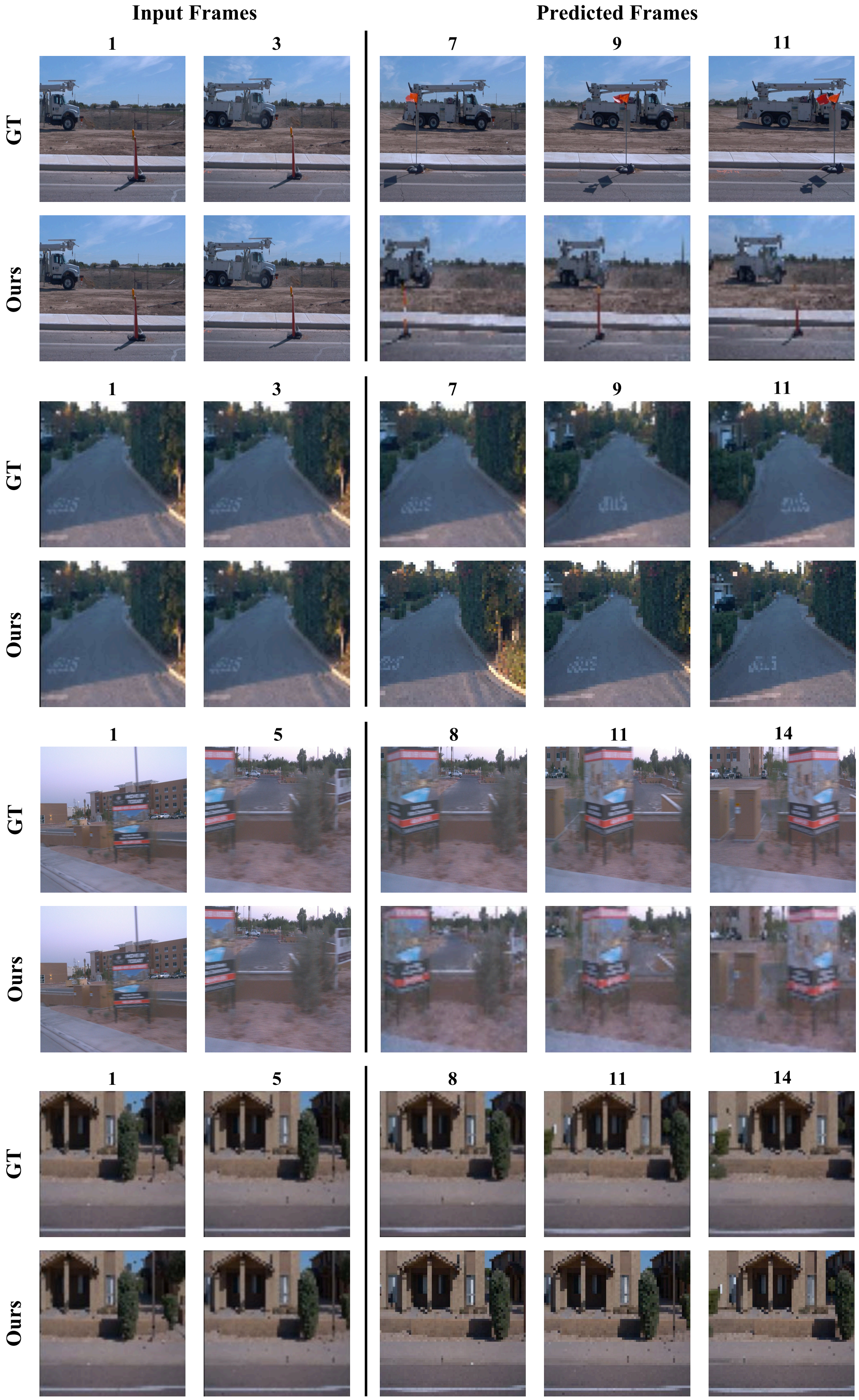


Figure 3. Additional qualitative results on Waymo Open Dataset.

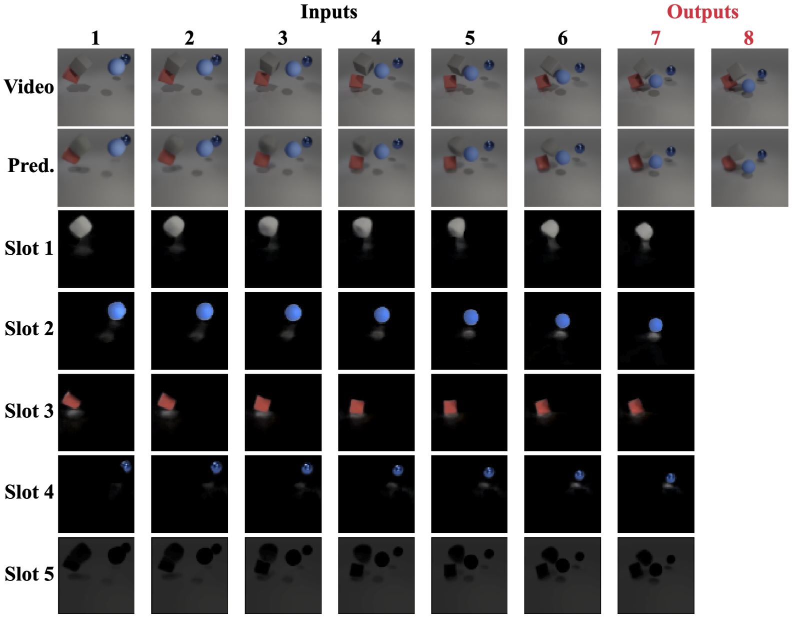


Figure 4. Per-slot decomposition results of MOVi-A dataset on Cross-OCK. We visualize the reconstruction of individual object slots based on the input predicted scene (Pred.), where Slot 5 denotes the background slot.

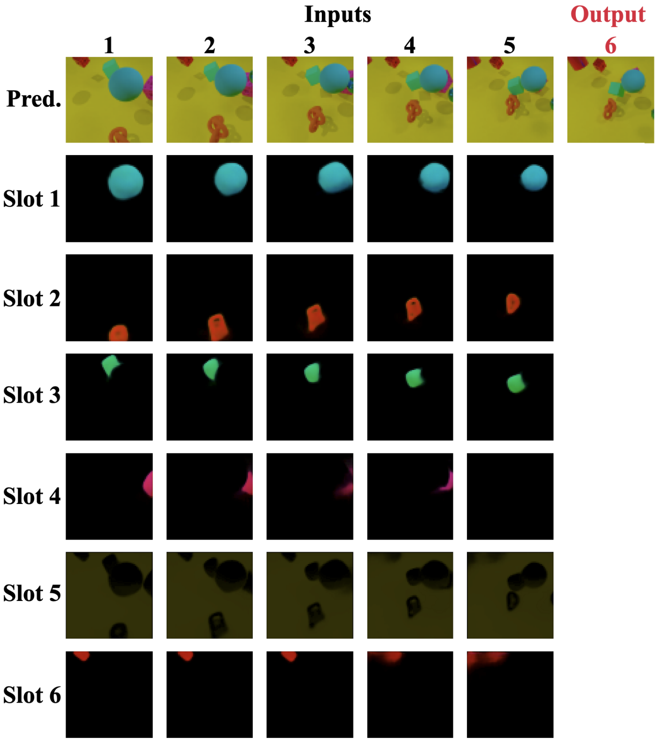


Figure 5. Per-slot decomposition results of MOVi-B datåset on Cross-OCK. Slot 5 denotes the background slot.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4

[2] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 1

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

[4] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3749–3761, 2022. 1

[5] Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021. 3

[6] Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models, 2020. 1

[7] 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. 3

[8] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 1

[9] 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. 4