Supplementary Material for: Temporally Consistent Object-Centric Learning by Contrasting Slots

Supplementary Material

A. Training Details

The general hyperparameters utilized during training SLOT CONTRAST are outlined in Table S1, ensuring clarity and reproducibility. Furthermore, the task-specific hyperparameters used for object dynamics prediction are detailed separately in Table S5.

B. Effect of Learned Initialization

To determine the optimal approach for first-frame slot initialization, we compared two techniques: sampling from a random distribution and learning fixed query vectors. Our experimental results show that learned initialization consistently yields superior performance. We hypothesize that this improvement arises from the emergence of contrastive slots during learning, a desirable property that promotes slot specialization. To illustrate this point, we visualized slot similarities for models initialized using both random and learned methods on the MOVi-C and YTVIS datasets (see the first row of Fig. S1). The plots demonstrate a clear pattern: learned slot initializations produce more contrastive representations, highlighting their advantage over random initialization. In addition, using slot-slot contrastive loss, we maintain the constructiveness of the slots (see the second row of Fig. S1), thus allowing for similar initialization for successive frame processing.

Next, we further analyze possible slot initializations that are more flexible than fixed initialization but are still contrastive. In particular, we propose an additional adaptive initialization method using k-means clustering. In particular, we use k-means clustering on dense object-centric features h_0 obtained by adapting original patch DINO features with a simple MLP module g_ψ. The cluster centroids (that are naturally not similar to each other) serve as slot initialization for the initial frame in the video. SLOT CONTRAST trained with such adaptive initialization achieves an FG-ARI score of 73.1 on the MOVi-C dataset ($+2.8$ FG-ARI improvement from fixed initialization). This result highlights the importance of flexible and contrastive first-frame slot initialization on model performance. However, the adaptive initialization is not scalable due to the significant computational overhead of running k-means for each initialization. Despite this limitation, the proof of concept demonstrates the promise of advanced initialization strategies, inviting further research in this direction.

C. Implementation of Slot-Slot Contrastive Loss

In this section, we provide details on the practical implementation of the slot-slot contrastive loss. Given the slot representations s_t and s_{t+1} at time steps t and $t+1$, we compute the similarity matrix $\mathbf{A}$:

$$A^{ij}_{t,t+1} = \frac{s^i_t \cdot s^j_{t+1}}{\|s^i_t\| \|s^j_{t+1}\|} \tag{S1}$$

where each element $A^{ij}_{t,t+1}$ represents cosine similarity between the i-th slot at time t and the j-th slot at time $t+1$.

Next, we apply the cross-entropy loss $\mathcal{L}_{\text{CE}}(\mathbf{P}, \mathbf{I})$ between the computed softmax normalized slot similarities $\mathbf{P} = \text{softmax}(\mathbf{A})$ and the identity matrix $\mathbf{I}$.

Batch Contrastive Loss We modify the similarity matrix $\mathbf{A}$ to include not only the slots for the current frame at time step t and the subsequent frame at time step $t+1$, but also the slots from all frames within the batch of videos that are processed together. Let B, T, K, and D denote the batch size, sequence length, number of slots, and the dimension of the slots, respectively. Initially, the similarity matrix $\mathbf{A}$ has shape $\mathbf{A} \in \mathbb{R}^{B\times(T-1)\times K\times K}$. After modifying it for batch comparison, its shape becomes $\mathbf{A}' \in \mathbb{R}^{(T-1)\times(KB)\times(KB)}$.

D. Feature Reconstruction Loss as Regularizer

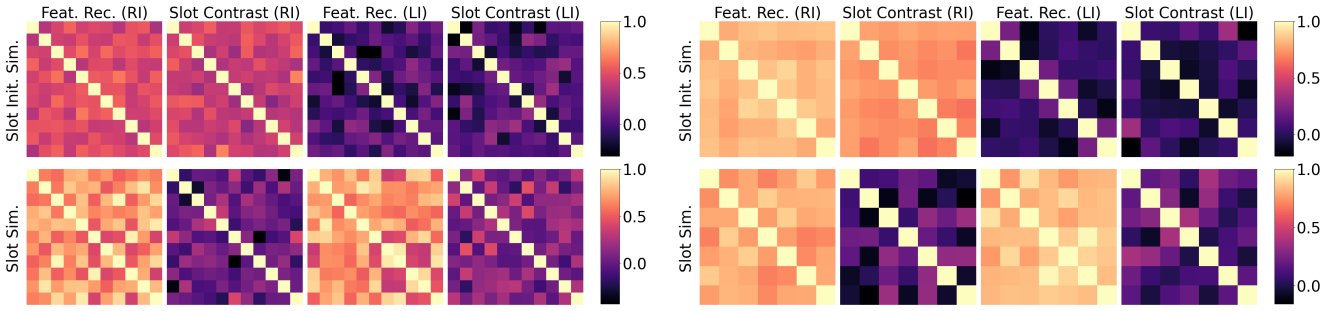
To promote better object discovery we also use feature reconstruction loss. Feature reconstruction loss, $\mathcal{L}_{\text{rec}}$, measures the discrepancy between the predicted features $\hat{h}_t$ and the true features h_t at each time step t. In our case the features correspond to self-supervised DINOv2 features. The loss could be computed using a common distance metric such as Mean Squared Error (MSE):

$$\mathcal{L}_{\text{rec}} = \sum_{t=1}^{T-1} ||h_t - \hat{h}_t||^2 \tag{S2}$$

The loss also serves as an effective regularizer, mitigating undesired behaviors that can arise from the contrastive nature of slot-slot contrastive loss. For example, slot-slot contrastiveloss can't pull slots representing different objects together because it is minimized alongside the feature reconstruction loss $\mathcal{L}_{\text{rec}}$. This way, we maximize slot-slot similarity while still requiring each slot to be informative about original inputs. So *region-wise reconstruction* with

Table S1. Hyperparameters of Slot-Slot Contrast Model for Main Results on MOVi-C, MOVi-E, and YouTube-VIS 2021 Datasets

Hyperparameter	**MOVi-C**	**MOVi-E**	**YouTube-VIS**
Training Steps	100k	300k	100k
Batch Size	64	64	64
Training Segment Length	4	4	4
Learning Rate Warmup Steps	2500	2500	2500
Optimizer	Adam	Adam	Adam
Peak Learning Rate	0.0004	0.0008	0.0008
Exponential Decay	100k	300k	100k
ViT Architecture	DINOv2 Small	DINOv2 Base	DINOv2 Base
Initialization	FixedLearnedInit	FixedLearnedInit	FixedLearnedInit
Patch Size	14	14	14
Feature Dimension (D_{feat})	384	768	768
Gradient Norm Clipping	0.05	0.05	0.05
Image Specifications			
Image / Crop Size	336	336	518
Cropping Strategy	Full	Full	Rand. Center Crop
Augmentations	–	–	Rand. Horizontal Flip
Image Tokens	576	576	1369
Slot Attention			
Slots	11	15	7
Iterations (first / other frames)	3 / 2	3 / 2	3 / 2
Slot Dimension (D_{slots})	64	128	64
Predictor			
Type	Transformer	Transformer	Transformer
Layers	1	1	1
Heads	4	4	4
Decoder			
Type	MLP	MLP	MLP
Loss Parameters			
Softmax Temperature (τ)	0.1	0.1	0.1
Slot-Slot Contrast Weight (α)	0.5	1	0.5



(a) MOVi-C dataset (b) YT-VIS dataset

Figure S1. Similarity matrix between the set of slot initializations, S_0 (first row) and first frame slots, S_1 (second row) for different loss functions (feature reconstruction and slot-slot contrast loss) and different initialization strategies (RI = random initialization; LI = learned initialization).

an MLP decoder decoding slots individually is an *effective regularizer*, preventing "wrong slots pulling" behavior as otherwise pulled slots will not contain the information about the object they are responsible to reconstruct.

Another key scenario is when an object disappears. In this case, it is important to understand what happens to the corresponding slot and how its behavior is governed by the objectives. In that case, we want the corresponding slot to maintain object information. Given the additional reconstruction loss, it is possible by ignoring the disappeared object's slot (thus serving as latent memory until object reappearance). This behavior is evident in the Fig. **??** showing *fewer active slots* compared to baseline that uses all the available slots.

E. Dataset Details

In this section, we provide details about the datasets used in our work. Overall, we use several synthetic datasets (MOVi-C and MOVi-E) and one challenging real-world dataset, YouTube-VIS. For all datasets, annotations are used only during the evaluation of the object discovery, while during training, we use only videos from the datasets.

MOVi Datasets For both MOVi-C and MOVi-E, we utilized the standard train/validation splits. Each dataset contains 9750 training sequences and 250 validation sequences. While the original datasets are provided at a resolution of 256×256, we resized them to 336×336 for our experiments. It is important to note that we did not generate new datasets, but rather modified the resolution of the original data. This way, we make sure that all the methods are comparable in terms of both original input resolution while using a similar or less token during ViT processing (576 for SLOT CONTRAST and VideoSAURv2, and 784 tokens for original VideoSAUR [9]).

Youtube-VIS 2021 The YouTube-VIS dataset is an unconstrained, real-world dataset designed for video instance segmentation. It has two versions: YouTube-VIS 2019 and YouTube-VIS 2021. In our work, we used YouTube-VIS 2021, as it is more complex and challenging compared to the 2019 version. We split the original training set into a new training set and a validation set, comprising 2,775 and 210 videos, respectively. This split was necessary because the original validation set for YouTube-VIS 2021 is not publicly available.

F. Metrics Details

To evaluate our method, we use two metrics: foreground Adjusted Rand Index (FG-ARI) and mean Best Overlap (mBO) to assess the quality of the masks produced by our models. FG-ARI is a variant of the standard ARI metric, computed by excluding the background mask, and is commonly used in the object-centric literature to measure the similarity between predicted object masks and ground truth masks. It primarily evaluates how well objects are segmented.

Mean Best Overlap (mBO), on the other hand, measures the similarity between predicted and ground truth masks using the intersection-over-union (IoU). For each ground truth mask, the predicted mask with the highest IoU is selected, and the average IoU is computed across all matched pairs. mBO also considers background pixels, offering a better measure of how well the masks align with the objects.

To differentiate between per-frame (image-based) and video-wide evaluations, we use "Image" as a prefix for the metrics (e.g., Image FG-ARI and Image mBO) when computed on individual frames. When we do not use an additional prefix, we refer to the "Video" version of the same metric when computed across entire videos. We are particularly interested in video-based metrics, as they additionally consider the consistency of object masks.

G. Baseline Details

VideoSAUR To compare our method with the state-of-the-art VideoSAUR method [9], we considered two configurations: VideoSAUR trained with DINO features [2] and VideoSAUR trained with DINOv2 [4] features, which we refer to as VideoSAURv2.

For the YouTube-VIS 2021 dataset, the authors of VideoSAUR provided results for both configurations, so we directly used the available checkpoints. However, for the MOVi datasets, results and model for VideoSAUR trained with DINOv2 features were not available. Therefore, we trained VideoSAUR with the default configuration(matching the resolution with SLOT CONTRAST) using DINOv2 features.

While for MOVI-E the default configuration with DINOv2 lead to improved results, MOVi-C results were significantly worse. Thus, we perform an extensive hyperparameter tuning, experimenting with the weight of the temporal similarity loss, temperature parameters, with and without feature reconstruction loss added. We also tested various configurations of keys, values, and output features from the Vision Transformer. Despite these efforts, we could not achieve performance comparable or better to VideoSAUR trained with DINOv1 features. Our best performing VideoSAURv2 configuration (62.1 FG-ARI and 25.5 mBO) on MOVi-C is obtained using temperature $\tau = 0.075$ temporal similarity loss weight $\alpha = 0.1$ combined with feature reconstruction loss. We also used DINOv2 ViT *values* features in contrast to *keys* features used in the original VideoSAUR paper [9] with DINOv1.

This discrepancy raises the question: why does VideoSAURv2 work well on MOVi-E and YouTube-VIS but not on simpler MOVi-C? We hypothesize that the presence of camera motion in MOVi-E might contribute to the success of

Table S2. Temporal consistency on YouTube-VIS 2021.

	Feat. Rec. + SAM2	SLOT CONTRAST + SAM2	VideoSAURv2	SLOT CONTRAST
FG-ARI	43.5	46.3	31.2	38.0
mBO	40.9	43.7	29.7	33.7

DINOv2 features in this context. To test this hypothesis, one can evaluate VideoSAUR on the MOVi-D dataset, which is similar in complexity to MOVi-E, but lacks camera motion.

SAM2 To compare how close current object-centric methods are to supervised methods we compared SLOT CONTRAST with SAM2 as a supervised zero-shot baseline for temporal consistency. As SAM2 is trained on a large dataset with dense video annotations ($190.9K$ masklets), using its tracking can improve segmentation consistency (limited to objects discovered in the first frame). However, while SAM2 can be used only for object tracking, *our method is not limited to tracking*; it jointly does both object discovery in videos and learns consistent object representations with their masks. We evaluate SAM2's tracking capabilities by combining SAM2 with initial frame object discovery using video-based DINOSAUR (i.e, feature reconstruction objective on videos) and SLOT CONTRAST object discovery (see Table S2). We show that SLOT CONTRAST halves the gap between unsupervised object-centric learning and zero-shot SAM2 (5.5 vs 12.3 FG-ARI), while using SLOT CONTRAST object discovery is helpful for overall tracking with SAM2 (+2.8 FG-ARI).

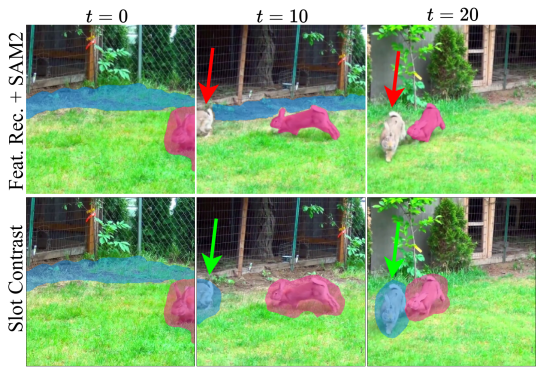


Figure S2. SlotContrast vs SAM2 tracking. SAM2 is limited to track only objects that appeared and discovered in the first frame.

In addition, in Fig. S2, we show limitation of such baseline: detecting and tracking later appearing objects due to missing initial masks. Evaluating SAM2 on YTVIS's first-frame objects gives 46.3 mBO (+6%), while for the later-appearing objects, mBO drops to 7.82 (−34.48%).This highlights SAM2's strength in tracking first-frame objects and its limitation in detecting and tracking later objects due to missing initial masks.

SAVI++ We compared SLOT CONTRAST with weakly supervised method SAVi++. We used improved SAVi similar to VideoSAUR (see App. C.5 VideoSAUR), *reaching* 42.8 *FG-ARI on MOVi-E*. In contrast, unconditioned optical-flow SAVi and depth SAVi++ are only 28.1 and 31.7 as reported by Bao et al. [1]. While adding depth signal in **SAVi++** could be treated as weak supervision, it indeed improves SAVi 16.0 mBO, reaching 22.1 mBO, but *still lagging behind both VideoSAUR and SlotContrast*.

H. Per-frame Scene Decomposition

In this section, we extend our comparison for the scene decomposition task to the MOVi-C dataset. The results are presented in Table S3. Our method outperforms all state-of-the-art approaches by a significant margin, with the sole exception of VideoSAUR, where we observe a minor performance gap of just 0.4 points, indicating comparable results.

	Model	Objective	Image FG-ARI
$\mathcal{I}$	LSD [3]	Image Rec.	50.5
	DINOSAUR [6]	Image Rec.	68.6
$\mathcal{V} + \mathcal{M}$	Safadoust et al. [5]	+GT Flow	73.8
$\mathcal{V}$	STEVE [7]	Video Rec.	51.9
	VideoSAUR [9]	Temp. Sim.	**75.5**
	Feat. Rec.	Video Rec.	64.0
	SLOT CONTRAST	Slot Contrast	75.1

Table S3. Quantitative Results on MOVi-C dataset in terms of per-frame Image FG-ARI. The methods are grouped by the target data they train on: only images ($\mathcal{I}$), videos with motion segmentation annotations ($\mathcal{V} + \mathcal{M}$), and only videos ($\mathcal{V}$).

Finally, on the YTVIS dataset for the image decomposition task, our method achieves a FG-ARI of 45.1 outperforming both VideoSAUR (40.1 FG-ARI) and VideoSAURv2 (40.5 FG-ARI).

I. Instance-Awareness of Dense Features

In this section, we emphasize the need to adapt self-supervised DINOv2 ViT features for consistent object discovery. While DINOv2 features are primarily semantic, they need refinement to identify specific instances effectively. To facilitate this, we project the frozen features through a multi-layer perceptron (MLP). This transformation maps the features into a new latent space, enhancing their instance-awareness and simplifying the Slot Attention task.

To show the effect of this adaptation on dense features, we visualize the first Principal Component Analysis (PCA) of both the frozen DINOv2 features and the newly learned

Table S4. Comparison of consistent object discovery evaluated by Video FG-ARI. We compare SLOT CONTRAST with frozen DINOv2 features and SLOT CONTRAST based on additionally adapted with MLP dense features.

	MOVi-C	MOVi-E	YouTube-VIS
Frozen DINOv2 Features	68.4	75.3	33.7
MLP Adapted Features	69.3	82.9	38.0

adapted dense features (see the results in Fig. S3). The PCA plots clearly show that while DINO features cluster similarly across different instances, the learned features are more distinct, effectively capturing instance-specific details.

Further, we evaluate the effectiveness of these instance-aware features by conducting experiments with both frozen and learned features. The results, summarized in Table S4. While MOVi-C, where most of the time different objects have different semantic categories, adapting shows minor improvement, the improvements are substantial for MOVi-E and the real-world YouTube-VIS dataset. This demonstrates the clear advantage of learning to adapt DINOv2 features to be instance-aware in challenging real-world scenarios.

J. SlotFormer

To evaluate our model's performance on the object dynamics prediction task, we trained a SlotFormer [8] module on top of our object-centric model. The code for SlotFormer was taken from its official codebase[1]. SlotFormer consists of a transformer encoder with input and output projection, and it adds positional embeddings to the input along the temporal dimension. It takes the slots from T burn-in frames and then predicts the slots for the next K rollout frames in an autoregressive manner. The model is trained by minimizing the mean squared error between the predicted slots and the ground-truth slots provided by the grouper. During training, the entire architecture of the object-centric model is frozen, and only the dynamics predictor module is optimized.

The hyperparameters used for training the models are listed in Table S5. For MOVi-C, we used entire videos for both training and validation, with the first fourteen frames serving as burn-in frames, while the model predicted the slots for the remaining frames. MOVi-E videos are also 24 frames long, but we chose to evaluate performance on the middle segment of the video because most objects remain static in the final frames. To create a more challenging evaluation, we selected the first 5 frames as burn-in and predicted the slots for the next 10 frames. Finally, for YTVIS, we used the first 10 frames as burn-in and had the model predict only the following 5 frames due to the dataset's complexity.

[1] https://github.com/pairlab/SlotFormer

Figure S3. First three Principal Components (combined as RGB channels into one image for convenience) of frozen DINOv2 features and the newly learned dense features. DINOv2 features PCA components are semantic grouping instances of the same category (e.g., people or dogs) and body parts of the different instances (e.g., heads or legs). In contrast, learned dense features have instance-aware components, separating different instances of the same category, thus making object discovery easier.

Table S5. Hyperparameters of SlotFormer for Main Results on MOVi-C, MOVi-E, and YouTube-VIS 2021 Datasets

Hyperparameter	**MOVi-C**	**MOVi-E**	**YouTube-VIS**
Training Steps	100k	100k	100k
Batch Size	128	128	128
Burn-in Steps T	14	5	10
Rollout Steps K	10	10	5
Latent Size D_e	128	256	128
Hidden Size of FFN	512	1024	512
Number of Layers N_τ	1	1	4
Dropout Rate	0.2	0.1	0.1
Peak Learning Rate	2×10^{-4}	2×10^{-5}	10^{-5}

K. Details and Visual Examples on MOVI-C Occluded

We created a targeted subset of the MOVi-C dataset that focuses exclusively on fully occluded object sequences. The MOVi-C dataset provides visibility scores for each object in each frame, indicating the number of pixels the object occupies. Using these scores, we refine the validation set to include only sequences meeting the following conditions: an object initially appears with a visibility score of at least n pixels, then becomes fully occluded (visibility score drops to 0 pixels), and subsequently reappears with a visibility score of at least n pixels. To avoid including very small objects or visual artifacts, we set n to a minimum of 400 pixels (less than 1% of the image pixels). After applying this filtering criterion, we obtain a dataset of 60 sequences where objects undergo complete occlusion and reappearance. Visualizations are presented in Fig. S9.

L. Limitations and Failure Cases

While SLOT CONTRAST demonstrates significant improvements over previous approaches, several limitations remain. One key area for improvement is the sharpness of predicted object masks, which could be tighter and sometimes occupy some background parts (referred to as "bleeding" artifacts). Another major challenge lies in ensuring consistency during long-term full occlusions. Although SLOT CONTRAST often reidentifies objects after such occlusions successfully, some failure cases persist.

Additionally, SLOT CONTRAST lacks control over slot behavior when objects disappear. Ideally, slots corresponding to disappeared objects should remain inactive and not be decoded, but the current implementation leaves this decision to the decoder. Future work could address this by making the behavior more explicit. Lastly, SLOT CONTRAST relies on a predefined, fixed number of slots, which may limit its flexibility. We visualize some of the failure cases in Fig. S11.

M. Additional Examples

In this section we present the following additional visualizations.

- Figure S4: Comparing SLOT CONTRAST to VideoSAUR on YouTube-VIS 2021.
- Figure S5, Figure S6 and Figure S7: ablations of SLOT CONTRAST components.
- Figure S8: Comparing SLOT CONTRAST and Feature Reconstruction on MOVi-C object dynamics prediction.
- Figure S9: Comparing SLOT CONTRAST and Feature Reconstruction on MOVi-C occluded subset.
- Figure S10: Comparing SLOT CONTRAST to VideoSAUR on MOVi-E scene decomposition task.
- Figure S11: SLOT CONTRAST failure cases.

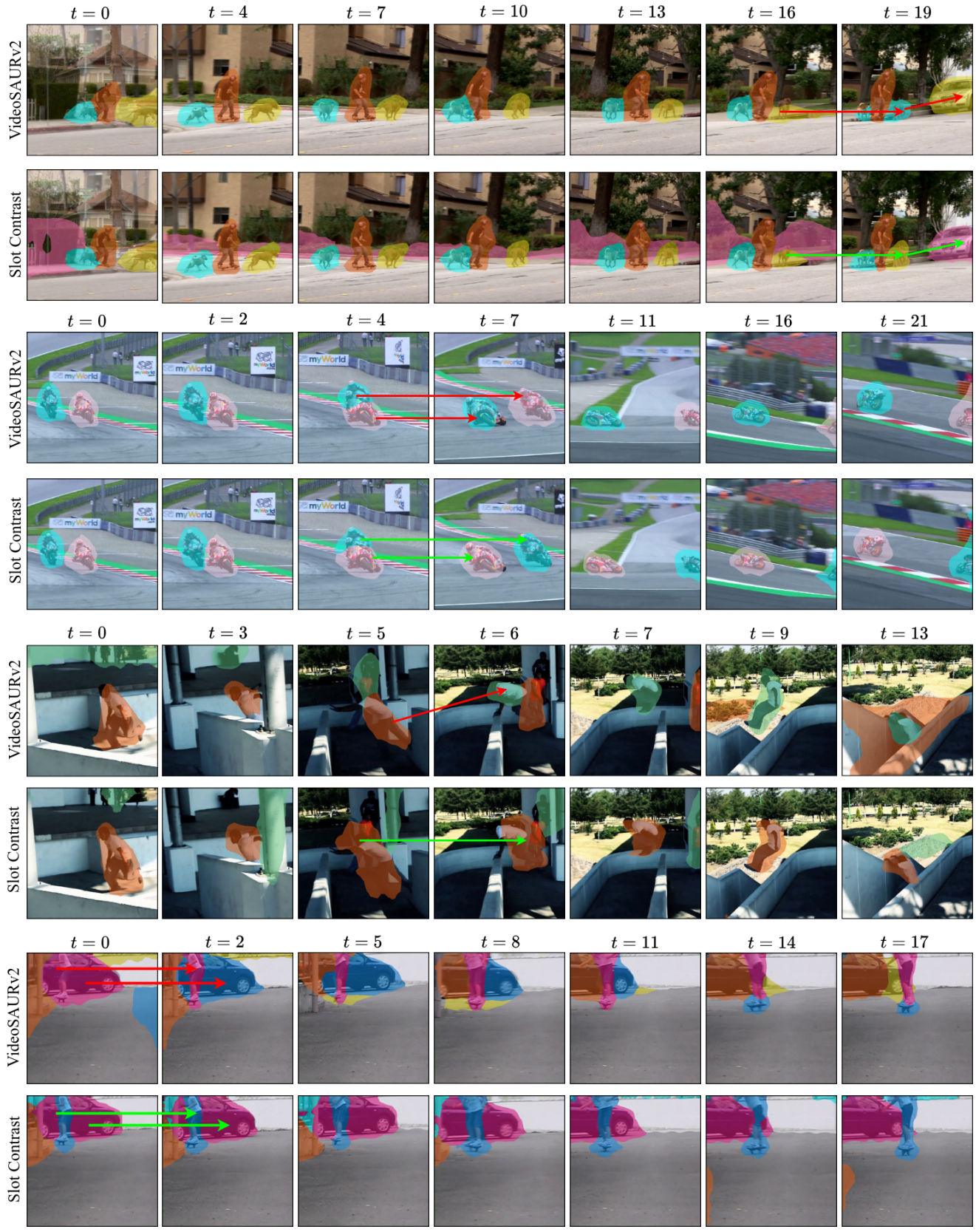


Figure S4. Qualitative comparison of SLOT CONTRAST with VideoSAURv2 on YouTube-VIS 2021 dataset.

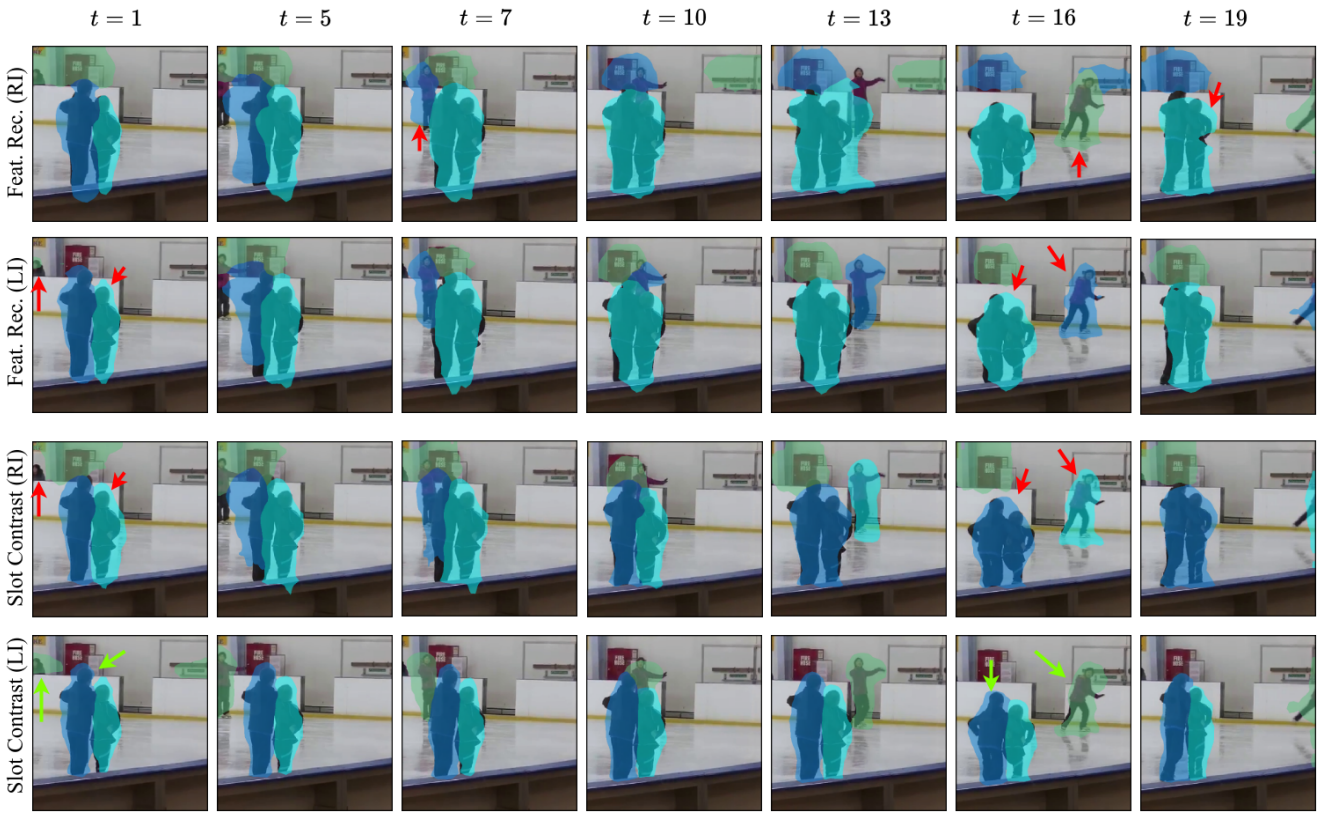


Figure S5. Qualitative results of first frame slot initialization ablations on YouTube-VIS 2021 dataset.

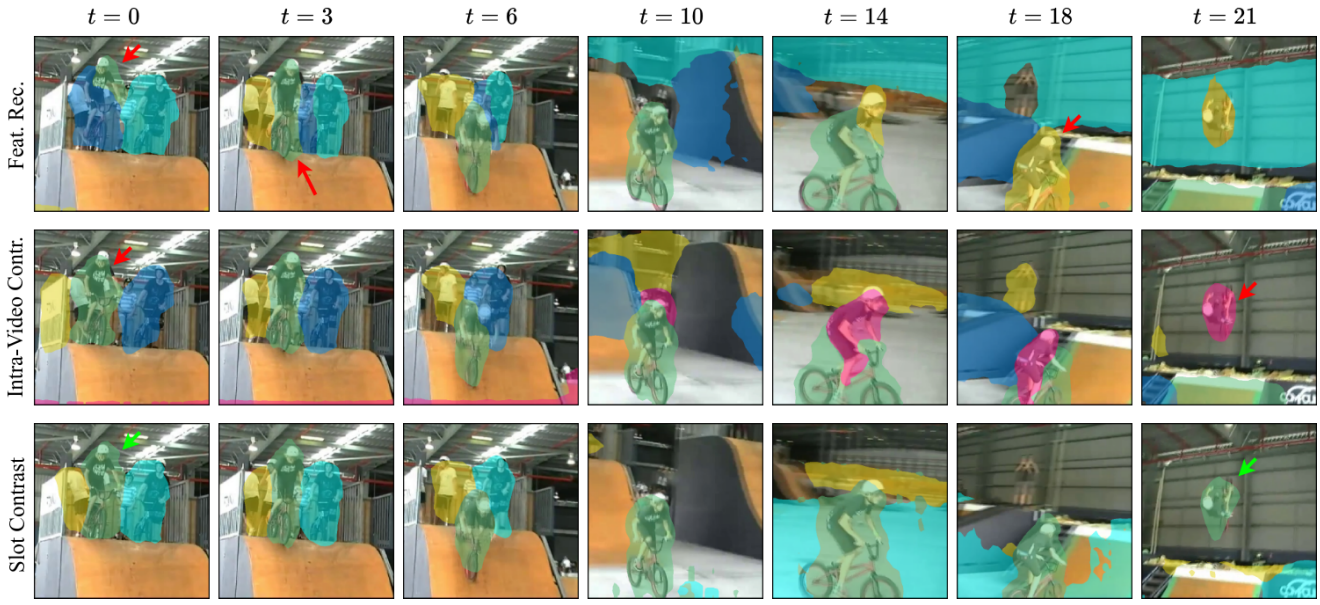


Figure S6. Qualitative results of loss function ablations on YouTube-VIS 2021 dataset.

Figure S7. Qualitative comparison of SLOT CONTRAST with Features Reconstruction baseline with learned initialization on YouTube-VIS 2021 dataset.

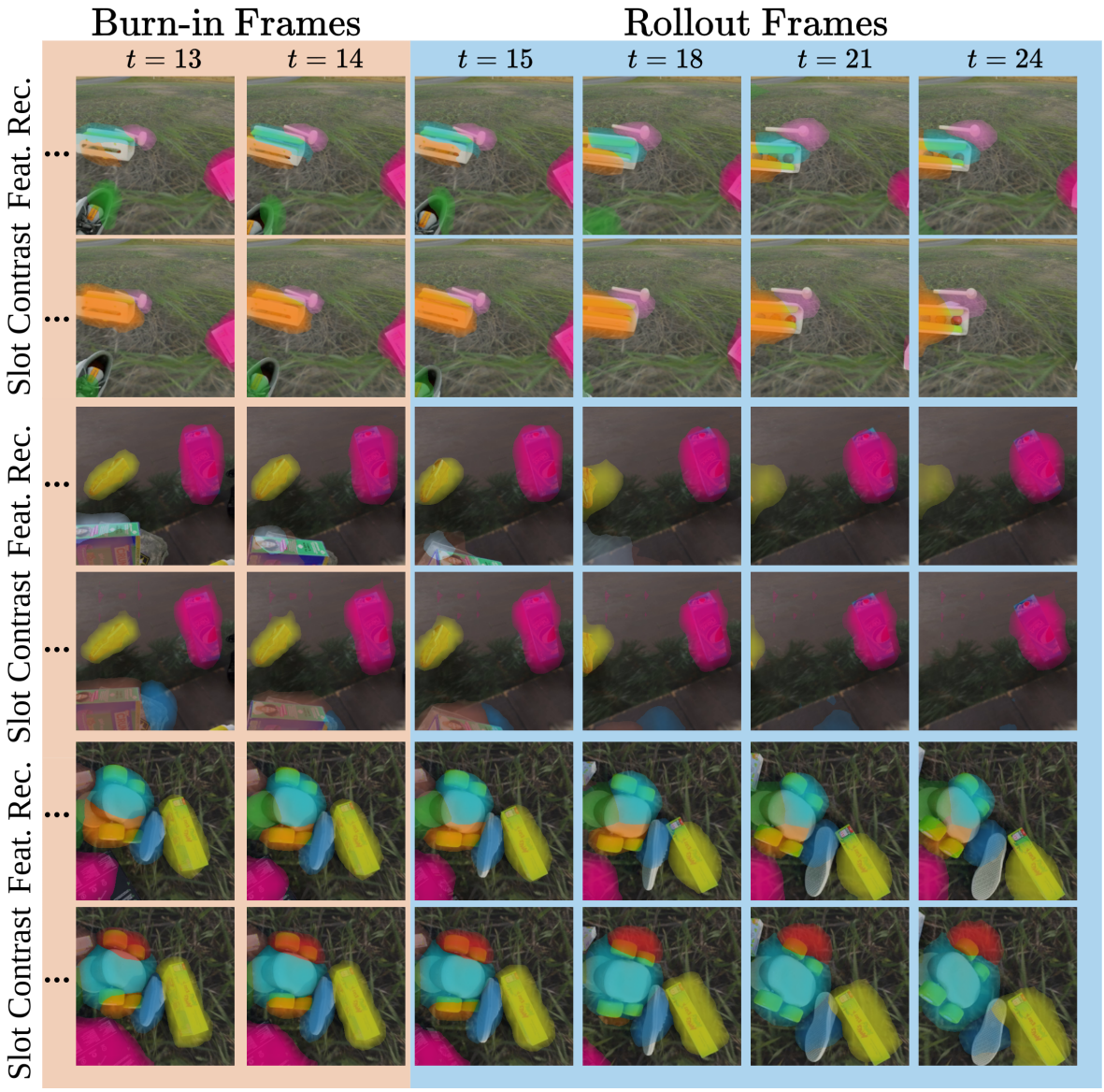


Figure S8. Comparison of masks obtained by decoding the predicted slots from SlotFormer, trained on top of the feature reconstruction baseline, versus SLOT CONTRAST, tested on the MOVi-C dataset.

Figure S9. Qualitative comparison of SLOT CONTRAST with Features Reconstruction on MOVi-C occluded subset.

Figure S10. Example frames comparing SLOT CONTRAST and VideoSAUR on the MOVi-E scene decomposition task. VideoSAUR occasionally misses objects or splits one object into multiple slots, while these errors are avoided by SLOT CONTRAST.

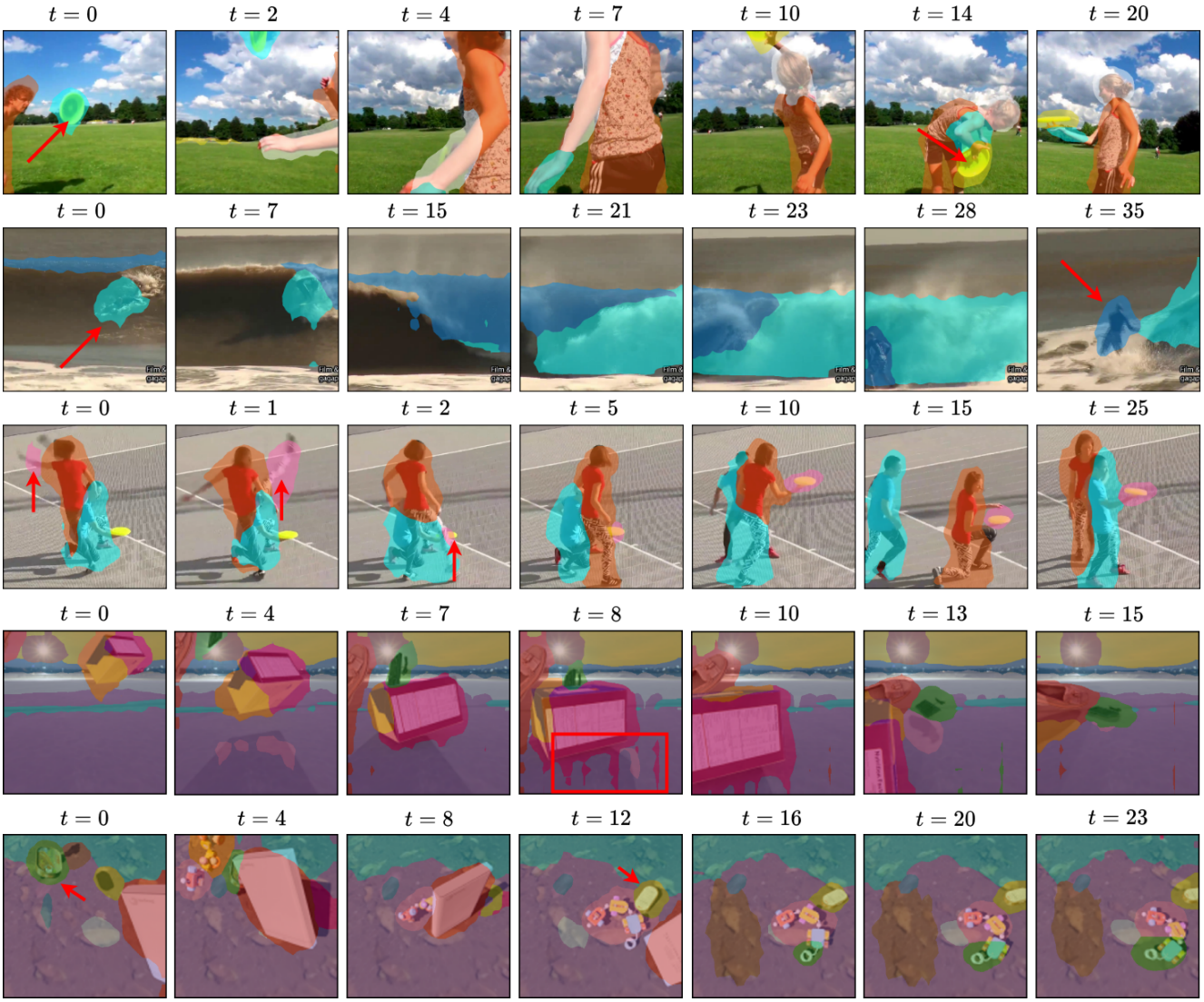


Figure S11. The visualizations depict various failure cases encountered by SLOT CONTRAST. The first three rows illustrate examples from the SLOT CONTRAST model trained on the YouTube-VIS 2021 dataset, while the last two rows are from the MOVi-C dataset. These examples highlight challenges such as failures due to complete occlusions or examples of mask "bleeding" artifacts.

References

[1] Zhipeng Bao, Pavel Tokmakov, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Object discovery from motion-guided tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22972–22981, 2023. 4

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. *ICCV*, 2021. 3

[3] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. In *NeurIPS*, 2023. 4

[4] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2023. 3

[5] Sadra Safadoust and Fatma Güney. Multi-object discovery by low-dimensional object motion. In *ICCV*, 2023. 4

[6] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *ICLR*, 2023. 4

[7] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple Unsupervised Object-Centric Learning for Complex and Naturalistic Videos. In *NeurIPS*, 2022. 4

[8] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. In *ICLR*, 2023. 5

[9] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023. 3, 4