RETHINKING OBJECT-CENTRIC REPRESENTATIONS FOR VIDEO DYNAMICS MODELING

Anonymous authors
Paper under double-blind review



Figure 1: UniSlot: Our method disentangles object appearance from spatial pose and enforces temporal alignment, enabling state-of-the-art unsupervised object dynamics learning.

ABSTRACT

Learning to decompose videos into persistent objects is a fundamental challenge in unsupervised object-centric representation learning. Despite recent progress, existing methods struggle to simultaneously achieve accurate object segmentation, consistent identities over time, and reliable foreground–background separation. To address these challenges, we introduce UniSlot (Unified Slots), an unsupervised framework for learning robust and disentangled object-centric representations from videos. UniSlot explicitly separates object appearance from its 3D-aware geometric pose in the scene, linking object identity to appearance while leveraging depth to better distinguish objects from their surroundings. UniSlot achieves state-of-the-art performance in unsupervised object-centric video decomposition and tracking across synthetic and real-world benchmarks, yielding substantially tighter object masks and reducing background leakage while preserving object identities. Beyond decomposition and tracking, these improved representations translate directly to downstream tasks such as unsupervised object dynamics prediction, enabling more accurate forecasting of future object trajectories.

1 INTRODUCTION

Object-centric representations provide a structured abstraction of visual scenes that has proven useful for video prediction and dynamics modeling (Rezazadeh et al., 2024; Villar-Corrales et al., 2023; Wu et al., 2023), causal reasoning (Tang et al., 2023; Xiao et al., 2024; Chen et al., 2025), and planning (Villar-Corrales & Behnke, 2025; Zou et al., 2025). Videos offer particularly rich supervision for learning such representations without object-level annotations, as objects persist, move, and interact over time. A robust object-centric video representation should therefore accurately localize objects while preserving their identities as the scene evolves.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Despite substantial progress, learning such representations from videos remains challenging. Slot-based methods (Burgess et al., 2019; Locatello et al., 2020) have established a powerful paradigm for unsupervised object discovery, with recent approaches leveraging self-supervised visual features (Caron et al., 2021; Seitzer et al., 2023) to extend object-centric learning to complex real-world scenes. In video, methods such as SAVi (Kipf et al., 2022; Elsayed et al., 2022), VideoSAUR (Zadaianchuk et al., 2023), and SlotContrast (Manasyan et al., 2025) further introduce temporal mechanisms to maintain object correspondence across frames. Yet existing methods still struggle to simultaneously produce tight object masks, preserve identities over time, and cleanly separate foreground objects from their surroundings. We argue that these failures originate from the *entanglement of object appearance and geometry in standard slot representations*. Appearance and geometry inherently evolve differently over time: while an object's position and scale change with motion, its appearance provides a more stable cue for correspondence. Enforcing temporal consistency on a representation that mixes these factors (Zadaianchuk et al., 2023; Manasyan et al., 2025) therefore constrains properties that should remain free to change. Furthermore, appearance cues alone can be insufficient to separate objects from complex backgrounds, particularly when foreground and background regions are visually similar. These limitations highlight the need to model appearance and geometry separately in object-centric video representations.

To this end, we propose UniSlot (UnifiedSlots), an unsupervised representation learning framework that explicitly disentangles object appearance from its 3D-aware geometric pose in the scene. UniSlot models appearance as a stable cue for temporal correspondence, while position and scale capture the object's evolving geometric state. This separation enables temporal consistency to be enforced on appearance while allowing geometry to evolve with object motion. UniSlot further leverages monocular depth as a spatial prior to capture the 3D structure of objects, better distinguish them from their surroundings, and reduce background leakage in complex scenes (Fig. 1). Overall, UniSlot learns more stable object-centric representations, with tight localization and consistent identities.

We evaluate UniSlot on synthetic and real-world video benchmarks (Greff et al., 2022; Yang et al., 2019) where it achieves state-of-the-art performance in unsupervised object-centric video decomposition and tracking. Importantly, these improvements extend to downstream tasks such as unsupervised object dynamics prediction (Wu et al., 2023), where UniSlot representations enable more accurate forecasting of future object trajectories. Our main contributions are summarized as follows:

1. We identify the entanglement of appearance and geometry as a key limitation of existing object-centric video representations, motivating their separate modeling.
2. We propose UniSlot, an unsupervised representation learning framework that disentangles appearance from 3D-aware geometry and leverages monocular depth pseudo-supervision for improved object decomposition and tracking.
3. We demonstrate SOTA performance across synthetic and real-world video benchmarks, validate the contributions of individual components through ablations, and show improved downstream unsupervised object dynamics prediction.

2 Related work

Object-centric representation learning Slot Attention (SA) (Locatello et al., 2020) established a widely used paradigm for unsupervised object-centric learning, grouping visual features into a fixed set of latent representations (called "slots") that are decoded to reconstruct the input. Subsequent work has improved object-centric representations along two main directions. First, methods such as ISA (Biza et al., 2023) and DISA (Majellaro et al., 2025) explore disentangled representations to improve object decomposition, separating spatial attributes from appearance in ISA and texture from contour information in DISA. While effective, these methods primarily target individual images from synthetic scenes. The second line of work focuses on scaling object discovery to more visually complex scenes. More expressive decoders, including mixture and compositional architectures (Singh et al., 2022a; Jiang et al., 2023; Kirilenko et al., 2024), increase the capacity of slots to reconstruct complex visual content, while methods such as DINOSAUR (Seitzer et al., 2023) leverage self-supervised features from pretrained vision models (Caron et al., 2021) to enable object discovery in real-world images. However, these directions remain difficult to reconcile, as expressive decoders weaken the separation encouraged by structured representations. Learning disentangled object representations in real-world videos therefore remains an open challenge.

Temporal object-centric learning Extending object-centric learning to video requires maintaining object correspondence as objects move, interact, and become occluded. SAVi (Kipf et al., 2022) propagates slots recurrently across frames, using previous slot states to initialize subsequent inference, while SAVi++ (Elsayed et al., 2022) additionally exploits sparse ground-truth depth obtained from LiDAR sensors. STEVE (Singh et al., 2022b) and SOLV (Aydemir et al., 2023) extend object-centric learning to more complex and naturalistic videos, while VideoSAUR (Zadaianchuk et al., 2023) leverages self-supervised visual features and learns their temporal evolution to improve object discovery and tracking in real-world videos, whereas SlotContrast (Manasyan et al., 2025) introduces a contrastive objective that promotes temporal consistency of slot representations. Recent methods improve video object decomposition through synergistic encoder–decoder learning (SRL) (Seong et al., 2026) and progressive slot allocation to reduce over-fragmentation (Moon et al., 2026). Despite these advances, existing methods still struggle with accurate object segmentation in complex videos, exhibiting oversegmentation, background leakage, and identity switches over time.

Object-centric dynamics modeling Object-centric representations enable visual dynamics to be modeled at the level of individual entities rather than pixels. SlotGNN (Rezazadeh et al., 2024) models interactions between object slots using graph neural networks, while OCVP (Villar-Corrales et al., 2023) and SlotFormer (Wu et al., 2023) predict future object representations from their temporal evolution. However, these methods rely on the quality and structure of the underlying object representations, motivating better separation between persistent and motion-based object properties.

3 METHOD

We propose **UniSlot**, an unsupervised object-centric representation learning framework for learning spatially precise and temporally consistent object representations from videos. As illustrated in Fig. 2, UniSlot represents each object through separate appearance and geometric components, propagates these representations recurrently across frames, and reconstructs both visual content and monocular depth. This structure allows temporal constraints to act on appearance while geometric attributes evolve with object motion, and uses monocular depth as an additional self-supervised training signal to improve object–background separation.

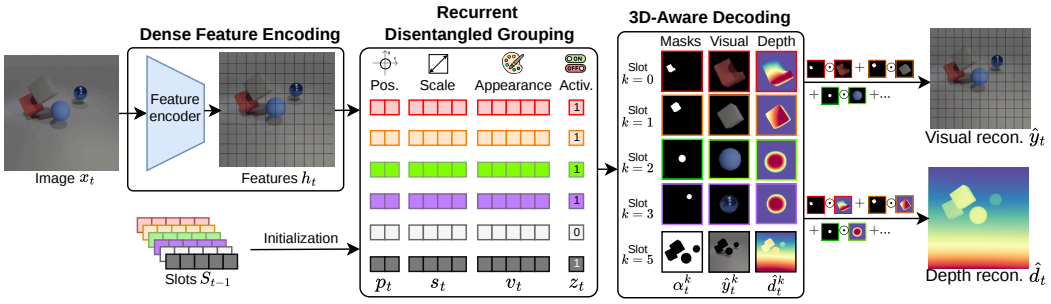


Figure 2: Overview of UniSlot. Given an RGB frame x_t, a visual encoder extracts dense features that are grouped into recurrent, disentangled object slots $S_t^k = (v_t^k, p_t^k, s_t^k)$ representing appearance, position, and scale, with z_t^k controlling slot activation. Active slots are independently decoded into alpha masks, visual content, and depth, which are finally composited to reconstruct the visual and monocular depth targets.

3.1 UNISLOT REPRESENTATION

Given a video $\mathcal{V} = \{x_1, \ldots, x_T\}$, with $x_t \in \mathbb{R}^{H \times W \times 3}$, our goal is to decompose each frame into K object-centric representations $S_t = \{S_t^1, \ldots, S_t^K\}$. Rather than encoding appearance and spatial information within a single latent vector, UniSlot explicitly represents each slot as

$$S_t^k = (v_t^k, p_t^k, s_t^k), \qquad v_t^k \in \mathbb{R}^D, \quad p_t^k, s_t^k \in \mathbb{R}^2, \tag{1}$$

where v_t^k describes object appearance, and p_t^k and s_t^k represent its 2D position and scale. This separation allows appearance to provide a stable cue for correspondence across frames while geometry can evolve with object motion.

Each frame is first encoded into dense visual features using a frozen self-supervised DINO encoder (Caron et al., 2021), followed by a lightweight learned projection:

$$h_t = F_{\text{proj}}(F_{\text{feat}}(x_t)) \,. \tag{2}$$

Architectural details of the encoder and feature projection are provided in Appendix A.

3.2 Recurrent disentangled slot grouping

UniSlot employs a recurrent, disentangled, and adaptive grouping mechanism for learning persistent object representations from videos. Building on the slot-specific spatial reference frames of ISA (Biza et al., 2023), UniSlot augments pose-normalized grouping with temporal propagation and adaptive slot activation (Fan et al., 2024). This allows slots to consistently track the same objects as their geometry changes and the number of active objects varies over time.

At each timestep, the previous slot states are propagated to initialize grouping in the current frame. Appearance is propagated through a learned stochastic predictor G_{pred}, while position and scale are carried forward directly:

$$\tilde{S}_t^k = \begin{cases} \big(G_{\text{pred}}(v_{t-1}^k),\, p_{t-1}^k,\, s_{t-1}^k\big), & t > 1, \\ S_{\text{init}}^k, & t = 1, \end{cases} \tag{3}$$

where S_{init}^k denotes a learned initial slot state. The predicted states $\tilde{S}_t = \{\tilde{S}_t^k\}_{k=1}^K$ are then refined by the grouping module, followed by adaptive slot activation:

$$\big\{(v_t^k, p_t^k, s_t^k)\big\}_{k=1}^K = G_{\text{slot}}(h_t, \tilde{S}_t), \tag{4}$$

$$z_t^k = G_{\text{gate}}(v_t^k), \tag{5}$$

where $z_t^k \in \{0, 1\}$ determines whether slot k is active. Inactive slots are excluded from reconstruction, allowing the active set to adapt to the number of objects in the scene and reducing oversegmentation from redundant slots. Further details of the recurrent predictor G_{pred}, pose-normalized grouping G_{slot}, and adaptive gating G_{gate} are provided in Appendix A.1.

3.3 3D-aware slot decoding

As UniSlot is trained without object-level annotations, we learn the object decomposition through reconstruction. Each active slot is decoded independently following the spatial broadcasting principle (Watters et al., 2019), intentionally avoiding feature mixing between slots. The position and scale (p_t^k, s_t^k) define a slot-specific coordinate system from which the decoder constructs positional embeddings, while v_t^k provides the appearance representation.

UniSlot jointly reconstructs visual content and monocular depth. Depending on the visual complexity of the dataset, the reconstruction target y_t is either the RGB frame x_t or the dense DINO features h_t. For each active slot, the decoder predicts a visual component $\hat{y}_t^k$, a depth component $\hat{d}_t^k$, and an alpha mask α_t^k. The final reconstructions are obtained by combining the active slots:

$$(\hat{y}_t^k, \hat{d}_t^k, \alpha_t^k) = D_{\text{spatial}}\big(v_t^k, p_t^k, s_t^k\big) \,, \tag{6}$$

$$\hat{y}_t = \sum_{k=1}^K z_t^k \, \alpha_t^k \odot \hat{y}_t^k, \qquad \hat{d}_t = \sum_{k=1}^K z_t^k \, \alpha_t^k \odot \hat{d}_t^k. \tag{7}$$

The pseudo-depth target d_t is generated from each RGB frame using a frozen monocular depth estimator, Depth Anything v3 (Lin et al., 2026). Unlike sensor-based depth supervision (Elsayed et al., 2022), it requires neither LiDAR measurements nor ground-truth 3D annotations. Visual content and depth are reconstructed using the same object-specific position and scale, allowing depth cues to inform the geometric decomposition and improve foreground–background separation. Depth is used only as a training signal; UniSlot requires only an RGB video at inference. Architectural details of D_{spatial} are provided in Appendix A.2.

3.4 LEARNING OBJECTIVES

UniSlot uses mean squared error for the visual and depth reconstruction objectives. We further guide the learning of disentangled slot representations with three regularizations: temporal alignment to maintain object–slot associations across frames, spatial separation to promote slot specialization, and sparsity to reduce redundant slots and over-segmentation. The main training objectives are:

$$\mathcal{L} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{time}}\mathcal{L}_{\text{time}} + \lambda_{\text{sep}}\mathcal{L}_{\text{sep}} + \lambda_{\text{spars}}\mathcal{L}_{\text{spars}}. \tag{8}$$

Unlike SlotContrast (Manasyan et al., 2025), disentanglement allows UniSlot to align only appearance v_t^k across time, avoiding conflicts between temporal consistency and evolving object geometry:

$$\mathcal{L}_{\text{time}} = \frac{\sum_{t,k} z_{t-1}^k z_t^k [1 - \cos(\tilde{v}_t^k, v_t^k)]}{\sum_{t,k} z_{t-1}^k z_t^k}. \tag{9}$$

Here, $\tilde{v}_t^k$ is the appearance predicted from the preceding frame. Finally, $\mathcal{L}_{\text{sep}}$ penalizes positive cosine similarity between active slot appearances, while $\mathcal{L}_{\text{spars}}$ penalizes unnecessary slot activation. Further details about the training objectives and their weights are provided in Appendix A.3.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets We evaluate UniSlot across synthetic and real-world video benchmarks of increasing visual complexity. CLEVRER (Yi et al., 2020) provides a controlled synthetic setting, while MOVi-C and MOVi-E (Greff et al., 2022) feature increasingly complex photorealistic scenes with diverse objects, backgrounds, and motion. For real-world evaluation, we use YouTube-VIS 2019 and 2021 (Yang et al., 2019). We additionally evaluate transfer of the learned representations on DAVIS 2017 (Pont-Tuset et al., 2017b). More details about the datasets are given in Appendix B.1, and results on extra MOVi benchmarks are provided in Appendix E.

Metrics We report complementary metrics for object decomposition and tracking. Adjusted Rand Index (ARI) (Rand, 1971; Hubert & Arabie, 1985) evaluates the complete scene decomposition, including background, while FG-ARI isolates foreground object separation. Mean Best Overlap (mBO) (Pont-Tuset et al., 2017a) measures object-mask overlap. $\mathcal{J}\&\mathcal{F}$ (Perazzi et al., 2016) evaluates video segmentation and tracking quality through region and boundary accuracy.

Baselines We compare UniSlot against established and recent methods: SAVi (Kipf et al., 2022), VideoSAUR (Zadaianchuk et al., 2023), SlotContrast (Manasyan et al., 2025), SRL (Seong et al., 2026) and SlotCurri (Moon et al., 2026), using their official implementations and matching feature encoders where applicable. For completeness, additional comparisons with single-image object-centric methods, including ISA (Biza et al., 2023), Adaptive Slot Attention (Fan et al., 2024), and DINOSAUR (Seitzer et al., 2023), are provided in Appendix E.

Implementation UniSlot uses a frozen DINO ViT-B/16 feature encoder and a frozen Depth Anything v3 (Lin et al., 2026) model (`DA3MONO-LARGE`) to generate monocular pseudo-depth targets during training. Depth is not required at inference. Full architecture, training settings, and dataset-specific hyperparameters are provided in Appendix A.

4.2 OBJECT-CENTRIC VIDEO DECOMPOSITION AND TRACKING

We first evaluate unsupervised object discovery and tracking using sequence-level metrics in Tab. 1. UniSlot achieves the best performance across all metrics and datasets. The improvement is particularly pronounced in terms of ARI, which evaluates the complete scene decomposition including background; on CLEVRER, UniSlot reaches .927 compared with .095 for the strongest baseline. Together with the highest FG-ARI, mBO, and $\mathcal{J}\&\mathcal{F}$, these results indicate that UniSlot improves foreground–background separation without sacrificing object localization or temporal consistency.

Importantly, these gains are not only achieved on synthetic scenes, but also on real-world videos. On YouTube-VIS 2019, UniSlot nearly doubles ARI over SlotContrast (.304 vs. .153), while also improving mBO from .406 to .516 and $\mathcal{J}\&\mathcal{F}$ from .308 to .360. The same trend holds on YouTube-VIS

Table 1: Object-centric video decomposition and tracking. Metrics are computed over video sequences. Best results are shown in bold and second-best are underlined.

Method	CLEVRER				MOVi-C				MOVi-E			
	ARI↑	FG-ARI↑	mBO↑	J&F↑	ARI↑	FG-ARI↑	mBO↑	J&F↑	ARI↑	FG-ARI↑	mBO↑	J&F↑
SAVi	.001	.241	.046	.045	.011	.138	.075	.097	.002	.227	.099	.123
VideoSAUR	.015	.210	.028	.014	.220	.614	.334	.218	.236	.585	.251	.173
SlotContrast	.095	.618	.118	.067	.103	.568	.274	.107	.204	.744	.262	.214
SRL	.080	.451	.071	.052	.036	.195	.057	.030	.143	.754	.246	.198
SlotCurri	.024	.620	.067	.053	.186	.652	.289	.197	.046	.570	.169	.142
UniSlot (Ours)	**.927**	**.781**	**.647**	**.558**	**.505**	**.654**	**.358**	**.280**	**.267**	**.785**	**.282**	**.230**

Method	YouTube-VIS 2019				YouTube-VIS 2021			
	ARI↑	FG-ARI↑	mBO↑	J&F↑	ARI↑	FG-ARI↑	mBO↑	J&F↑
SAVi	.042	.122	.271	.187	.034	.109	.181	.162
VideoSAUR	.150	.145	.398	.298	.133	.161	.368	.256
SlotContrast	.153	.194	.406	.308	.134	.206	.371	.275
SRL	.225	.139	.409	.219	.097	.179	.243	.186
SlotCurri	.083	.112	.247	.206	.058	.154	.250	.193
UniSlot (Ours)	**.304**	**.225**	**.516**	**.360**	**.228**	**.231**	**.455**	**.315**

2021, where UniSlot again outperforms all baselines across all metrics. Consistent improvements from controlled synthetic scenes to complex real-world videos show that UniSlot's disentangled representation supports both foreground–background decomposition and temporal object consistency across substantially different scene conditions.

Qualitative results Figure 3 illustrates the failure modes behind the quantitative differences across methods. VideoSAUR frequently over-segments objects and changes their slot assignments over time, while SlotContrast can merge distinct objects and include large background regions in its foreground masks. These failures occur in both synthetic and real-world videos, degrading decomposition (ARI) as well as region and contour accuracy ($\mathcal{J}\&\mathcal{F}$). In contrast, UniSlot maintains more stable object assignments while producing tighter masks with substantially less background leakage.

Figure 4 examines a more challenging case involving temporary disappearance and re-entry. After the target person leaves the field of view at $t = 8$ and re-enters at $t = 12$, VideoSAUR assigns the person to a different slot, while SlotContrast exhibits fragmentation and reassigns slots. UniSlot instead recovers the target with the same slot after re-entry and maintains this association for the remainder of the sequence. This example shows that UniSlot's temporal consistency extends beyond short-term correspondence between adjacent frames. While Figs. 3 and 4 highlight specific examples, similar failure modes are observed across many sequences.

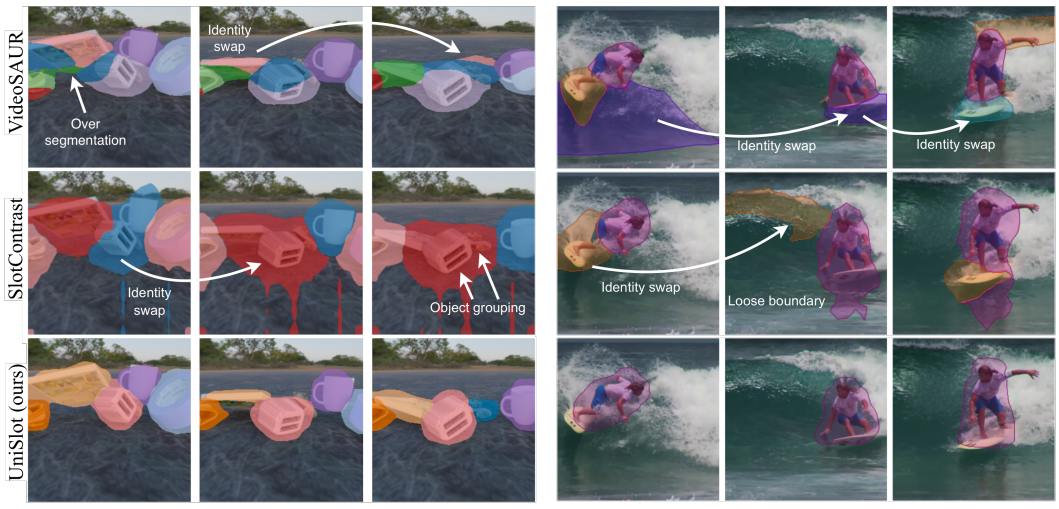


Figure 3: Qualitative comparison of object decomposition and tracking. Colors denote fixed slot indices within each sequence. On MOVi-C (*left*), VideoSAUR exhibits over-segmentation and identity swaps, while SlotContrast merges objects and changes their slot assignments. On YouTube-VIS (*right*), both baselines exhibit identity switches and inaccurate boundaries. UniSlot maintains consistent object–slot associations and tightly localized masks across both settings.

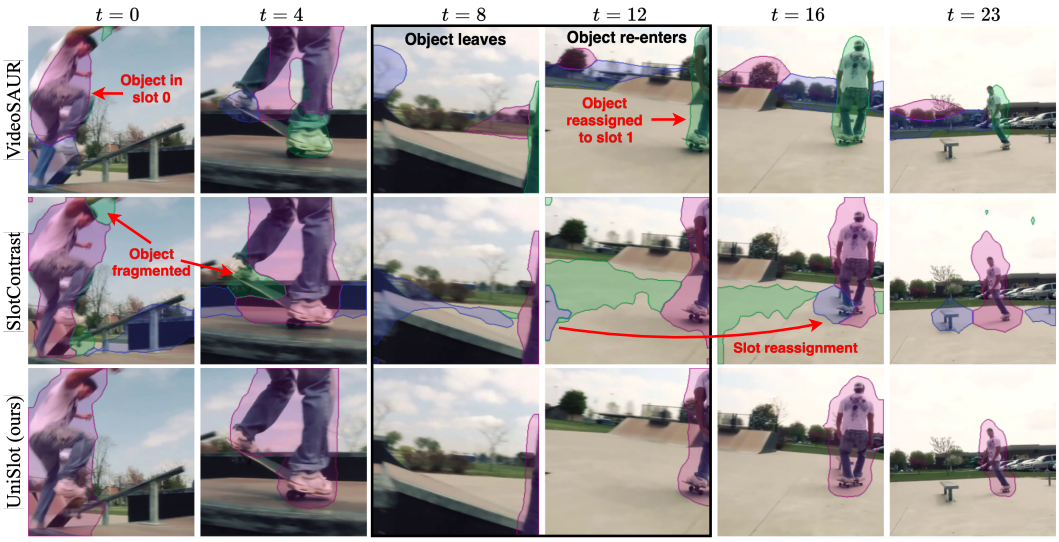


Figure 4: Temporal correspondence under interrupted visibility. Colors denote fixed slot indices across the sequence. The target leaves at t=8 and re-enters at t=12. VideoSAUR assigns the returning person to a different slot, while SlotContrast exhibits fragmentation and slot reassignment. UniSlot preserves the object–slot association after re-entry while maintaining a tight localized mask.

4.3 Object dynamics prediction

To evaluate whether UniSlot representations extend beyond video decomposition and tracking, we consider unsupervised object dynamics prediction as a downstream task. We freeze each object-centric model and train a SlotFormer (Wu et al., 2023) dynamics predictor on the resulting slot trajectories. For UniSlot, the predictor operates on its structured appearance, position, and scale representation, while for the baseline models it operates on their entangled slot embeddings. Predictor architecture and training details are provided in Appendix B.4.

As shown in Tab. 2, UniSlot improves downstream object dynamics prediction across both synthetic and real-world videos, achieving the best performance on 11 of 12 dataset–metric combinations. The improvement is particularly pronounced on MOVi-C, where ARI increases from .123 to .318 over the strongest baseline. These results show that the benefits of UniSlot extend beyond decomposition of observed frames, with its disentangled appearance and geometric state supporting accurate prediction of future object dynamics. Figure 5 illustrates this behavior qualitatively, with UniSlot preserving object identities and localized masks throughout the rollout, while SlotContrast exhibits fragmentation and less accurate predictions.

Table 2: Object dynamics prediction using frozen object-centric representations. Best results are shown in bold and second-best are underlined.

Method	MOVi-C				MOVi-E				YouTube-VIS 2021			
	ARI↑	FG-ARI↑	mBO↑	J&F↑	ARI↑	FG-ARI↑	mBO↑	J&F↑	ARI↑	FG-ARI↑	mBO↑	J&F↑
SAVi	.008	.055	.034	.023	.001	.125	.086	.071	.024	.092	.142	.118
VideoSAUR	.123	.402	.214	.161	.157	.560	.234	.202	.121	.161	.359	.287
SlotContrast	.055	.378	.133	.097	.166	**.692**	.249	.214	.118	.192	.346	.225
UniSlot (Ours)	**.318**	**.504**	**.229**	**.175**	**.202**	.636	**.279**	**.242**	**.194**	**.213**	**.389**	**.297**

4.4 Transfer to DAVIS 2017

We evaluate transfer to unseen real-world videos by applying models trained on YouTube-VIS 2021 directly to DAVIS 2017 (Pont-Tuset et al., 2017b), without retraining or fine-tuning. DAVIS emphasizes challenging temporal conditions, such as occlusion and fast object motion. As shown in Tab. 3, UniSlot achieves the best performance across all metrics, improving over the best baseline for each metric from .158 to .179 ARI, .111 to .148 FG-ARI, .330 to .379 mBO, and .234 to

Table 3: Transfer to DAVIS 2017. All models are trained on YouTube-VIS 2021.

Method	ARI↑	FG-ARI↑	mBO↑	$\mathcal{J}\&\mathcal{F}$ ↑
SAVi	.018	.067	.183	.173
VideoSAUR	.097	.111	.328	.213
SlotContrast	.158	.085	.330	.234
UniSlot (Ours)	**.179**	**.148**	**.379**	**.242**

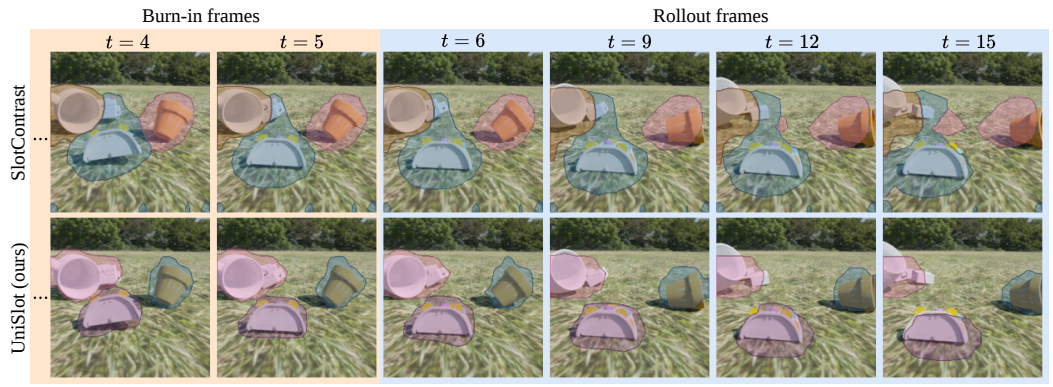


Figure 5: Object dynamics prediction on MOVi-C. After six burn-in frames ($t \leq 5$), future object states are predicted autoregressively. SlotContrast exhibits fragmentation and loose boundaries, while UniSlot maintains coherent object identities and localized masks throughout the rollout.

.242 $\mathcal{J}\&\mathcal{F}$. These results show that the learned object representations transfer to unseen real-world videos without target-domain adaptation.

4.5 Ablation studies

We ablate the principal components of UniSlot to determine their contributions to object decomposition and temporal consistency. We evaluate the performance on two complementary benchmarks: MOVi-C provides photorealistic multi-object scenes with object motion and occlusions, while YouTube-VIS 2021 captures complex real-world videos with camera motion.

Table 4 shows that the components of UniSlot play complementary roles across different scene regimes. Replacing the disentangled representation with standard Slot Attention (SA) substantially degrades performance, supporting the explicit separation of appearance and geometry. On MOVi-C, removing temporal alignment or slot separation causes a substantial degradation, particularly in FG-ARI. Figure 7 illustrates their distinct roles: without $\mathcal{L}_{\text{time}}$, object identities become inconsistent across frames, while without $\mathcal{L}_{\text{sep}}$, distinct objects can be assigned to the same slot. These results support explicitly treating temporal correspondence and separation between object representations as complementary constraints.

Adaptive activation is particularly important on YouTube-VIS, where the number and composition of visible objects vary substantially; removing it reduces ARI from .228 to .136 and mBO from .455 to .378. Monocular depth pseudo-supervision provides consistent improvements across both datasets, including a substantial gain in $\mathcal{J}\&\mathcal{F}$ on MOVi-C (.195 to .280). As illustrated in Fig. 6, depth provides complementary geometric information that improves foreground–background separation and object association. Overall, the results show that disentanglement, regularization, adaptive activation, and depth play complementary roles, with the complete model performing best.

Additional robustness studies across feature encoders, slot configurations, and decoder architectures are provided in Appendix D.

Table 4: Ablation of the principal UniSlot components on MOVi-C and YouTube-VIS 2021.

Variant	MOVi-C				YouTube-VIS 2021			
	ARI↑	FG-ARI↑	mBO↑	J&F↑	ARI↑	FG-ARI↑	mBO↑	J&F↑
w/o $\mathcal{L}_{\text{time}}$	.266	.204	.205	.204	.202	.213	.439	.299
w/o $\mathcal{L}_{\text{sep}}$	.283	.225	.214	.213	.211	<u>.218</u>	.443	<u>.303</u>
w/o adaptive activation	.476	.237	.258	<u>.251</u>	.136	.203	.378	.283
w/o disentanglement (SA)	.111	.233	.161	.164	.165	.181	.411	.296
w/o depth pseudo-supervision	<u>.477</u>	<u>.646</u>	<u>.348</u>	.195	<u>.219</u>	.217	<u>.444</u>	.297
UniSlot	**.505**	**.654**	**.358**	**.280**	**.228**	**.231**	**.455**	**.315**

5 Discussion

Our results support the central assumption of UniSlot: appearance and geometry play different roles in object-centric video representations and benefit from being modeled separately. Applying

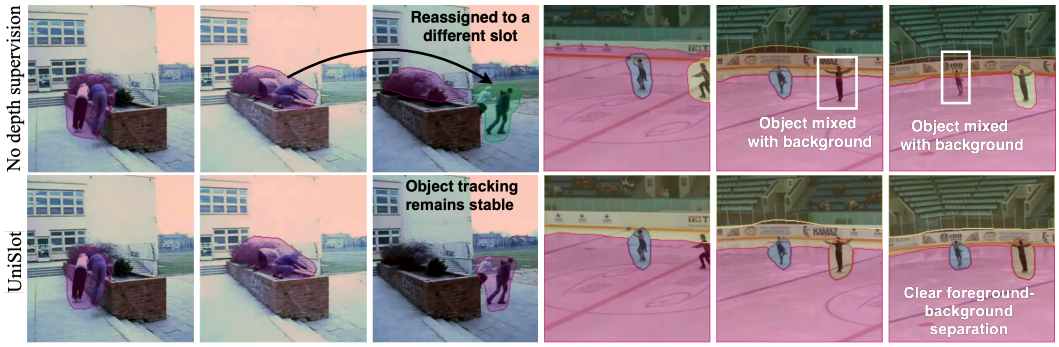


Figure 6: Effect of monocular depth pseudo-supervision. Without depth, object assignments rely only on visual features, sometimes leading to unstable associations after interactions (*left*) and foreground-background mixing (*right*).

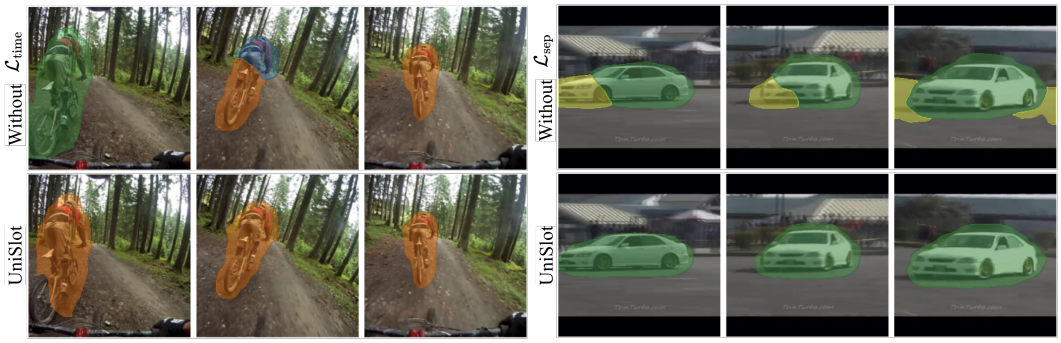


Figure 7: Qualitative effect of UniSlot regularization losses. Without temporal alignment (*left*), the same object switches identities across frames. Without slot separation (*right*), an object can be decomposed in multiple redundant slots.

temporal constraints to appearance while allowing position and scale to evolve enables stable object representations without suppressing motion. The downstream dynamics results further show that the learned representation is useful beyond video decomposition, as its object states can be effectively propagated to predict future dynamics. At the same time, our ablations show that disentanglement alone is not sufficient: temporal alignment, slot separation, adaptive activation, and monocular depth provide complementary mechanisms for learning robust object representations.

UniSlot relies on pretrained visual and monocular depth models to provide appearance and geometric cues during training. While these priors enable learning from videos without object or depth annotations, inaccuracies or domain biases in the pretrained models may propagate to the learned representations. Reducing this dependence, particularly for domains that differ substantially from the pretraining data, remains an interesting direction for future work.

6 Conclusion

We propose UniSlot, an unsupervised framework for learning robust object-centric representations from videos by disentangling appearance from geometric states. By introducing temporal consistency to appearance while allowing position and scale to evolve, and complementing visual features with monocular depth, UniSlot improves object decomposition and temporal consistency across synthetic and real-world videos. The resulting representations provide effective object states for downstream dynamics prediction and also generalize to unseen videos.

Future work could explore higher-level semantic or language-based cues to maintain persistent object identities over longer videos, particularly under prolonged occlusions and substantial appearance changes.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and polishing parts of the manuscript and with writing parts of the software code used to generate the figures. All AI-assisted text was reviewed and edited by the authors, and AI-assisted code was manually reviewed and tested for correctness. We did not use generative AI to generate synthetic datasets or prove mathematical claims. We take responsibility for the final content of this work, including text, code, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

UniSlot studies unsupervised object-centric representation learning from video using standard synthetic and publicly available real-world benchmarks. While video representation and tracking technologies may be misused for surveillance, our work focuses on generic object representations and does not perform biometric identification or person recognition. We encourage responsible use with appropriate consideration of privacy and consent.

REPRODUCIBILITY STATEMENT

All datasets used in this work are publicly available. We provide detailed architectural configurations, training procedures, hyperparameters, and implementation details in the Appendix. The complete source code will be made publicly available upon acceptance.

REFERENCES

Görkay Aydemir, Weidi Xie, and Fatma Güney. Self-supervised object-centric learning for videos. In *NeurIPS*, volume 36, pp. 32879–32899, 2023.

Ondrej Biza, Sjoerd Van Steenkiste, Mehdi S. M. Sajjadi, Gamaleldin F. Elsayed, Aravindh Mahendran, and Thomas Kipf. Invariant slot attention: Object discovery with slot-centric reference frames. In *ICML*, pp. 2507–2527, 2023.

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.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pp. 9650–9660, October 2021.

Zhenfang Chen, Shilong Dong, Kexin Yi, Yunzhu Li, Mingyu Ding, Antonio Torralba, Joshua B. Tenenbaum, and Chuang Gan. Compositional physical reasoning of objects and events from videos. *IEEE TPAMI*, 47(9):7689–7703, 2025.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024.

Gamaleldin Fathy Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael Curtis Mozer, and Thomas Kipf. SAVi++: Towards end-to-end object-centric learning from real-world videos. In *NeurIPS*, volume 35, pp. 28940–28954, 2022.

Ke Fan, Zechen Bai, Tianjun Xiao, Tong He, Max Horn, Yanwei Fu, Francesco Locatello, and Zheng Zhang. Adaptive slot attention: Object discovery with dynamic slot number. In *CVPR*, pp. 23062–23071, 2024.

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 *CVPR*, pp. 3749–3761, 2022.

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

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *ICLR*, 2017.

Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. In *NeurIPS*, volume 36, pp. 8563–8601, 2023.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

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 *ICLR*, 2022.

Daniil Kirilenko, Vitaliy Vorobyov, Alexey Kovalev, and Aleksandr Panov. Object-Centric Learning with Slot Mixture Module. In *ICLR*, 2024.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *ICLR*, volume 2026, pp. 141261–141285, 2026.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *NeurIPS*, volume 33, pp. 11525–11538, 2020.

Riccardo Majellaro, Jonathan Collu, Aske Plaat, and Thomas M. Moerland. Explicitly disentangled representations in object-centric learning. *TMLR*, 2025. ISSN 2835-8856.

Anna Manasyan, Maximilian Seitzer, Filip Radovic, Georg Martius, and Andrii Zadaianchuk. Temporally consistent object-centric learning by contrasting slots. In *CVPR*, pp. 5401–5411, 2025.

WonJun Moon, Hyun Seok Seong, and Jae-Pil Heo. Reconstruction-guided slot curriculum: Addressing object over-fragmentation in video object-centric learning. In *CVPR*, pp. 25001–25010, 2026.

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 *CVPR*, pp. 724–732, June 2016.

Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):128–140, 2017a.

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

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

Alireza Rezazadeh, Athreyi Badithela, Karthik Desingh, and Changhyun Choi. SlotGNN: Unsupervised discovery of multi-object representations and visual dynamics. In *ICRA*, pp. 17508–17514, 2024.

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.

Hyun Seok Seong, WonJun Moon, and Jae-Pil Heo. From vicious to virtuous cycles: Synergistic representation learning for unsupervised video object-centric learning. In *ICLR*, 2026.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *TMLR*, 2026.

Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate DALL-E learns to compose. In *ICLR*, 2022a.

Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. In *NeurIPS*, volume 35, pp. 18181–18196, 2022b.

Qu Tang, Xiangyu Zhu, Zhen Lei, and Zhaoxiang Zhang. Intrinsic physical concepts discovery with object-centric predictive models. In *CVPR*, pp. 23252–23261, 2023.

Angel Villar-Corrales and Sven Behnke. PlaySlot: Learning inverse latent dynamics for controllable object-centric video prediction and planning. In *ICML*, volume 267, pp. 61534–61563, 2025.

Angel Villar-Corrales, Ismail Wahdan, and Sven Behnke. Object-centric video prediction via decoupling of object dynamics and interactions. In *ICIP*, pp. 570–574. IEEE, 2023.

Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in VAEs. *arXiv preprint arXiv:1901.07017*, 2019.

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

Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can I trust your answer? Visually grounded video question answering. In *CVPR*, pp. 13204–13214, June 2024.

Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, pp. 5188–5197, 2019.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. CLEVRER: Collision events for video representation and reasoning. In *ICLR*, 2020.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-Centric Learning for Real-World Videos by Predicting Temporal Feature Similarities. In *NeurIPS*, volume 36, pp. 61514–61545, 2023.

Junhong Zou, Xiangyu Zhu, Zhaoxiang Zhang, and Zhen Lei. Top-down guidance for learning object-centric representations. In *IJCAI*, pp. 2548–2556, 2025. doi: 10.24963/ijcai.2025/284.

A Implementation details

This section provides additional implementation details for UniSlot. Given an RGB frame $x_t \in \mathbb{R}^{H\times W\times 3}$, a frozen self-supervised encoder F_{feat} (Caron et al., 2021) extracts N dense patch features, which are projected to the slot-grouping space using a two-layer MLP:

$$g_t = F_{\text{feat}}(x_t), \qquad g_t \in \mathbb{R}^{N\times D_{\text{feat}}}, \tag{10}$$

$$h_t = F_{\text{proj}}(g_t), \qquad h_t \in \mathbb{R}^{N\times D}. \tag{11}$$

Each slot is represented as $S_t^k = (v_t^k, p_t^k, s_t^k)$, with appearance $v_t^k \in \mathbb{R}^D$ and 2D position and scale $p_t^k, s_t^k \in \mathbb{R}^2$. Architecture and training hyperparameters are reported in Tab. 5.

A.1 Recurrent disentangled slot grouping

At timestep t, the grouping module receives projected features $h_t \in \mathbb{R}^{N\times D}$ and predicted slot states $\tilde{S}_t^k = (\tilde{v}_t^k, \tilde{p}_t^k, \tilde{s}_t^k)$. Building on ISA (Biza et al., 2023), these states are refined through T_{slot} iterations of pose-normalized attention.

Pose-normalized attention Each slot defines a relative coordinate system from its current position and scale:

$$\text{rel_grid}_t^k = \frac{\text{abs_grid} - p_t^k}{s_t^k}, \qquad \text{abs_grid} \in \mathbb{R}^{N\times 2}. \tag{12}$$

The relative coordinates are injected into slot-specific keys and values,

$$\text{keys}_t^k = f_K(h_t) + g(\text{rel_grid}_t^k), \qquad \text{values}_t^k = f_V(h_t) + g(\text{rel_grid}_t^k), \tag{13}$$

while queries are computed from the slot appearance embeddings. Attention is first normalized across slots and then, after adding ϵ, normalized across spatial locations:

$$\bar{a}_{t,n}^k = \text{softmax}_k(a_{t,n}^k), \tag{14}$$

$$\text{attn}_{t,n}^k = \frac{\bar{a}_{t,n}^k + \epsilon}{\sum_m (\bar{a}_{t,m}^k + \epsilon)}. \tag{15}$$

Position and scale are updated from the center and spatial spread of the resulting attention maps:

$$p_t^k = \sum_n \text{attn}_{t,n}^k \, \text{abs_grid}_n, \tag{16}$$

$$s_t^k = \sqrt{\sum_n \text{attn}_{t,n}^k (\text{abs_grid}_n - p_t^k)^2}. \tag{17}$$

Appearance v_t^k is updated using the Slot Attention GRU and MLP. Spatial information is therefore represented explicitly by (p_t^k, s_t^k) rather than being absorbed entirely into the appearance embedding.

Recurrent initialization Previous slot states initialize grouping in the next frame. Appearance is propagated through a stochastic predictor G_{pred}:

$$(\mu_t^k, \log \sigma_t^k) = G_{\text{pred}}(v_{t-1}^k), \qquad \tilde{v}_t^k \sim \mathcal{N}(\mu_t^k, \sigma_t^k), \tag{18}$$

using the reparameterization trick (Kingma & Welling, 2014). Geometry is propagated directly:

$$(\tilde{p}_t^k, \tilde{s}_t^k) = (p_{t-1}^k, s_{t-1}^k). \tag{19}$$

The resulting state is refined as

$$\{(v_t^k, p_t^k, s_t^k)\}_{k=1}^K = G_{\text{slot}}(h_t, \tilde{S}_t). \tag{20}$$

For the first frame, $\tilde{S}_1^k = S_{\text{init}}^k$, where S_{init}^k is a learned initial slot state.

Adaptive gating Following Adaptive Slot Attention (Fan et al., 2024), a lightweight gate predicts whether each refined slot is active:

$$z_t^k = G_{\text{gate}}(v_t^k), \qquad z_t^k \in \{0, 1\}. \tag{21}$$

Only active slots contribute to reconstruction. We use the hard Gumbel–Softmax relaxation (Jang et al., 2017) for end-to-end optimization.

A.2 3D-Aware spatial broadcast decoding

Each active slot is decoded independently following the spatial broadcasting principle (Watters et al., 2019). At the initial decoder resolution $r \times r$, position and scale define a slot-specific coordinate grid:

$$\text{rel_grid}_t^k = \frac{\text{abs_grid} - p_t^k}{s_t^k}, \qquad \text{rel_grid}_t^k \in \mathbb{R}^{r\times r\times 2}. \tag{22}$$

The appearance embedding is spatially broadcast and combined with a learned projection of this grid:

$$U_t^k = \text{SB}(v_t^k) + D_{\text{pos}}(\text{rel_grid}_t^k). \tag{23}$$

A shared CNN decoder independently predicts a visual component $\hat{y}_t^k$, depth component $\hat{d}_t^k$, and alpha mask α_t^k:

$$(\hat{y}_t^k, \hat{d}_t^k, \alpha_t^k) = D_{\text{spatial}}(U_t^k). \tag{24}$$

Algorithm 1 Recurrent Disentangled Slot Grouping

Require: Patch features h_t, previous slots $\{S_{t-1}^k\}_{k=1}^K$
if $t = 1$ **then**
  $\tilde{S}_t^k \leftarrow S_{\text{init}}^k$
else
  $(\mu_t^k, \log \sigma_t^k) \leftarrow G_{\text{pred}}(v_{t-1}^k)$
  $\tilde{v}_t^k \sim \mathcal{N}(\mu_t^k, \sigma_t^k)$
  $(\tilde{p}_t^k, \tilde{s}_t^k) \leftarrow (p_{t-1}^k, s_{t-1}^k)$
end if
$(v_t^k, p_t^k, s_t^k) \leftarrow (\tilde{v}_t^k, \tilde{p}_t^k, \tilde{s}_t^k)$
for $i = 1, \ldots, T_{\text{slot}}$ **do**
  for each slot k **do**
    $\text{rel_grid}_t^k \leftarrow (\text{abs_grid} - p_t^k)/s_t^k$
    $\text{keys}_t^k \leftarrow f_K(h_t) + g(\text{rel_grid}_t^k)$
    $\text{values}_t^k \leftarrow f_V(h_t) + g(\text{rel_grid}_t^k)$
    $\text{queries}_t^k \leftarrow f_Q(v_t^k)$
  end for
  Normalize attention across slots and spatial locations
  for each slot k **do**
    $p_t^k \leftarrow$ attention-weighted center
    $s_t^k \leftarrow$ attention-weighted spatial spread
    $v_t^k \leftarrow$ Slot Attention GRU + MLP update
  end for
end for
for each slot k **do**
  $z_t^k \leftarrow G_{\text{gate}}(v_t^k)$
end for
return $\{(v_t^k, p_t^k, s_t^k, z_t^k)\}_{k=1}^K$

The visual and depth predictions are composited over active slots:

$$\hat{y}_t = \sum_{k=1}^{K} z_t^k \, \alpha_t^k \odot \hat{y}_t^k, \qquad \hat{d}_t = \sum_{k=1}^{K} z_t^k \, \alpha_t^k \odot \hat{d}_t^k. \tag{25}$$

The alpha masks are predicted independently and are not normalized across slots.

Visual reconstruction targets For CLEVRER, MOVi-A, and MOVi-B, we reconstruct RGB pixels directly, *i.e.*, $y_t = x_t$. For MOVi-C, MOVi-E, and YouTube-VIS, we reconstruct the projected dense features, *i.e.*, $y_t = h_t$. Decoder output dimensions are adjusted accordingly.

Monocular depth targets For each RGB frame, we generate pseudo-depth using the frozen `DA3MONO-LARGE` model from Depth Anything v3 (Lin et al., 2026). Depth predictions are normalized independently per video by clipping to the 2nd and 98th percentiles, linearly rescaling to $[0, 1]$, and then mapping to $[-1, 1]$ via $d \leftarrow (d - 0.5)/0.5$. They require neither depth sensors nor ground-truth 3D annotations and are used only during training.

Visual content and depth are decoded using the same slot-specific position, scale, and alpha mask. The depth objective therefore supervises the same spatial decomposition as the visual reconstruction, introducing depth information into the learned geometric representation.

A.3 Training Objectives

UniSlot combines visual and depth reconstruction with temporal alignment, slot separation, adaptive-slot sparsity, and stochastic predictor regularization:

$$\mathcal{L} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{time}}\mathcal{L}_{\text{time}} + \lambda_{\text{sep}}\mathcal{L}_{\text{sep}} + \lambda_{\text{spars}}\mathcal{L}_{\text{spars}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}. \tag{26}$$

Reconstruction Visual and pseudo-depth targets are reconstructed using mean squared error:

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_t \left[\|y_t - \hat{y}_t\|_2^2 \right], \qquad \mathcal{L}_{\text{depth}} = \mathbb{E}_t \left[\|d_t - \hat{d}_t\|_2^2 \right]. \tag{27}$$

Temporal alignment Temporal consistency is imposed only on the appearance embeddings of slots active in consecutive frames:

$$\mathcal{L}_{\text{time}} = \frac{\sum_{t,k} z_{t-1}^k z_t^k \left[1 - \cos(\tilde{v}_t^k, v_t^k)\right]}{\sum_{t,k} z_{t-1}^k z_t^k}. \tag{28}$$

The geometric components (p_t^k, s_t^k) remain free to evolve with object motion.

Slot separation and sparsity We penalize positive cosine similarity between different active appearance embeddings:

$$\mathcal{L}_{\text{sep}} = \frac{\sum_t \sum_{i \neq j} z_t^i z_t^j \max\Big(0, \cos(v_t^i, v_t^j)\Big)}{\sum_t \sum_{i \neq j} z_t^i z_t^j}. \tag{29}$$

Adaptive-slot sparsity encourages unnecessary slots to deactivate:

$$\mathcal{L}_{\text{spars}} = \mathbb{E}_{t,k}[z_t^k]. \tag{30}$$

Stochastic predictor regularization The recurrent predictor parameterizes

$$q(\tilde{v}_t^k \mid v_{t-1}^k) = \mathcal{N}\left(\mu_t^k, \text{diag}\big((\sigma_t^k)^2\big)\right). \tag{31}$$

We regularize its variance toward $0.01I$ while leaving the predicted mean unconstrained:

$$\mathcal{L}_{\text{KL}} = \mathbb{E}_{t,k}\left[D_{\text{KL}}\left(\mathcal{N}\big(\mu_t^k, \text{diag}\big((\sigma_t^k)^2\big)\big) \,\big\|\, \mathcal{N}\big(\mu_t^k, 0.01I\big)\big)\right]. \tag{32}$$

This preserves the predicted appearance mean used for recurrent initialization while controlling the sampling variance.

A.4 Optimization and training

Optimization We use Adam with $(\beta_1, \beta_2) = (0.9, 0.999)$ and an exponentially decayed learning rate following linear warmup, with the decay schedule spanning the full training duration. Dataset-specific training configurations are reported in Tab. 5.

Progressive regularization Regularization is introduced progressively after an initial decomposition has formed. $\mathcal{L}_{\text{spars}}$ is enabled after 50k steps on CLEVRER and MOVi-A/B and after 100k steps on MOVi-C/E and YouTube-VIS. $\mathcal{L}_{\text{time}}$ and $\mathcal{L}_{\text{sep}}$ are active from the beginning on CLEVRER and MOVi-A/B and enabled after 50k steps on MOVi-C/E and YouTube-VIS.

Loss balancing Reconstruction provides the primary learning signal, while the remaining objectives shape the slot representation. Exact loss weights are reported in Tab. 5.

B Experimental details

B.1 Datasets

Unless stated otherwise, we use a spatial resolution of 256×256. This is the native resolution of CLEVRER and MOVi used in our experiments, and we adopt the same resolution for YouTube-VIS for consistency. CLEVRER and MOVi sequences contain 24 frames, while YouTube-VIS sequences contain up to 84 frames. During training, UniSlot processes 6-frame clips and is applied recurrently to the full sequence at inference.

CLEVRER CLEVRER (Yi et al., 2020) contains synthetic videos of simple geometric objects undergoing collisions and interactions. Its controlled appearance and dynamics, together with dense instance annotations, provide a clean setting for evaluating object decomposition and temporal identity consistency.

MOVi The MOVi benchmarks (Greff et al., 2022) consist of procedurally generated multi-object videos with increasing visual and geometric complexity. MOVi-A contains simple objects and backgrounds, while MOVi-B introduces more diverse object shapes and appearances. MOVi-C replaces these objects with scanned real-world assets, and MOVi-E further increases scene complexity through more objects, diverse backgrounds, and camera motion. All variants provide dense instance masks and consistent object identities across frames.

YouTube-VIS YouTube-VIS (Yang et al., 2019) contains real-world videos with temporally consistent instance annotations across diverse object categories. Compared with the synthetic benchmarks, its unconstrained appearance, camera motion, occlusions, and background clutter provide a substantially more challenging setting for unsupervised object decomposition and tracking. We independently train and evaluate on YouTube-VIS 2019 and YouTube-VIS 2021, which contain different video collections.

DAVIS 2017 DAVIS 2017 (Pont-Tuset et al., 2017b) is a densely annotated video object segmentation benchmark containing multiple foreground objects under challenging appearance changes, occlusions, and motion. We use DAVIS exclusively for transfer evaluation: models trained on YouTube-VIS 2021 are evaluated on DAVIS without training or fine-tuning on DAVIS.

B.2 Evaluation Metrics

We evaluate object decomposition and tracking using Adjusted Rand Index (ARI), Foreground ARI (FG-ARI), mean Best Overlap (mBO), and $\mathcal{J}\&\mathcal{F}$, reporting each metric where applicable.

Adjusted Rand Index (ARI) ARI (Rand, 1971; Hubert & Arabie, 1985) measures agreement between predicted slot assignments and ground-truth instance segmentation, corrected for chance. Let n_{ij} denote the number of pixels assigned to ground-truth cluster i and predicted cluster j, with $a_i = \sum_j n_{ij}$, $b_j = \sum_i n_{ij}$, and $n = \sum_{ij} n_{ij}$. We compute:

$$\mathrm{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}}. \tag{33}$$

ARI is computed over all pixels, including background, whereas FG-ARI uses the same definition after excluding background pixels. Thus, ARI additionally reflects foreground–background separation, while FG-ARI isolates foreground object grouping.

Mean Best Overlap (mBO) Given ground-truth masks $\mathcal{M}$ and predicted masks $\hat{\mathcal{M}}$, mBO (Pont-Tuset et al., 2017a) matches each ground-truth object to its best-overlapping predicted mask:

$$\mathrm{mBO} = \frac{1}{|\mathcal{M}|} \sum_{M \in \mathcal{M}} \max_{\hat{M} \in \hat{\mathcal{M}}} \frac{|M \cap \hat{M}|}{|M \cup \hat{M}|}. \tag{34}$$

Higher mBO indicates better object-mask overlap.

$\mathcal{J}\&\mathcal{F}$ For video object segmentation and tracking, we additionally report the standard DAVIS $\mathcal{J}\&\mathcal{F}$ metric (Pont-Tuset et al., 2017b). Region similarity $\mathcal{J}$ measures mask intersection over union, while contour accuracy $\mathcal{F}$ is the harmonic mean of boundary precision P_c and recall R_c:

$$\mathcal{J} = \frac{|M \cap \hat{M}|}{|M \cup \hat{M}|}, \qquad \mathcal{F} = \frac{2P_c R_c}{P_c + R_c}, \qquad \mathcal{J}\&\mathcal{F} = \frac{\mathcal{J} + \mathcal{F}}{2}. \tag{35}$$

Boundary precision and recall are computed by matching predicted and ground-truth contours within the standard boundary tolerance. We report the sequence-averaged score.

B.3 Baseline implementation

We evaluate all baselines using their official implementations and preserve their original architectures and training objectives wherever possible. For comparisons based on pretrained visual features,

we use the feature encoder specified by each method and otherwise match preprocessing and evaluation protocols to UniSlot. The number of slots is chosen consistently with the maximum scene complexity of each dataset.

SAVi (Kipf et al., 2022), VideoSAUR (Zadaianchuk et al., 2023), SlotContrast (Manasyan et al., 2025), SRL (Seong et al., 2026), and SlotCurri (Moon et al., 2026) form the video-based baselines in the main paper. We additionally evaluate the image-based ISA (Biza et al., 2023), Adaptive Slot Attention (Fan et al., 2024), and DINOSAUR (Seitzer et al., 2023) in the extended results in Appendix E.

SlotContrast, SRL, and SlotCurri rely on DINOv2 features extracted with ViT backbones of patch size 14. We therefore evaluate these methods at 252×252, the closest resolution to our default 256×256 that is divisible by 14. For SRL and SlotCurri, we retrain the models using their official implementations rather than their released checkpoints, which use 336×336 inputs on MOVi and 518×518 inputs on YouTube-VIS. This ensures that all methods are evaluated at comparable input resolutions. Complete baseline configurations are reported in Sec. C.3.

B.4 Object dynamics prediction

To evaluate the learned representations for downstream dynamics modeling, we train SlotFormer (Wu et al., 2023) on frozen slot trajectories extracted by each object-centric model. We condition on H observed frames and autoregressively predict future states using a 4-layer Transformer with latent dimension 128, 8 attention heads, feed-forward dimension 512, and sinusoidal temporal positional embeddings. No slot-index positional encoding is used.

For conventional representations, SlotFormer operates directly on the slot embeddings. For UniSlot, we use the structured state

$$q_t^k = [\, v_t^k;\, p_t^k;\, s_t^k;\, z_t^k \,] \in \mathbb{R}^{D+5}. \tag{36}$$

The model predicts appearance and scale together with a position displacement Δp_t^k, from which $\hat{p}_{t+1}^k = p_t^k + \Delta p_t^k$. Predicted scales are constrained to be positive using Softplus, while slot activation is propagated from the latest state rather than predicted.

We train SlotFormer using a weighted MSE over appearance, position, and scale,

$$\mathcal{L}_{\mathrm{dyn}} = \lambda_v \mathcal{L}_v + \lambda_p \mathcal{L}_p + \lambda_s \mathcal{L}_s, \tag{37}$$

Loss weights and training configurations are reported in Tab. 6. The object-centric models remain frozen throughout dynamics training.

B.5 DAVIS 2017 transfer

To evaluate transfer to unseen real-world videos, we apply models trained on YouTube-VIS 2021 directly to DAVIS 2017, without training or fine-tuning on DAVIS. We use the same inference procedure and slot configuration as on YouTube-VIS and evaluate the resulting object masks over complete video sequences using ARI, FG-ARI, mBO, and $\mathcal{J}\&\mathcal{F}$.

C Model and training configurations

C.1 UniSlot hyperparameters

Unless stated otherwise, training uses 6-frame clips at $H \times W = 256 \times 256$ resolution, while inference is performed recurrently over the full video sequence. The frozen feature encoder F_{feat} is a DINO ViT-B/16 (Caron et al., 2021) with $D_{\mathrm{feat}} = 768$, followed by the projection F_{proj} to the slot dimension $D = 128$. Recurrent disentangled grouping uses $T_{\mathrm{slot}} = 2$ refinement iterations. The stochastic appearance predictor G_{pred} is a two-layer MLP with hidden dimension 256, and the adaptive gate G_{gate} uses the same hidden dimension.

We train with Adam using a peak learning rate of 10^{-4}, 5k warmup steps, batch size 64, and gradient clipping at norm 0.05. Pseudo-depth targets d_t are generated offline using the frozen `DA3MONO-LARGE` model (Lin et al., 2026) and are used only during training. Table 5 reports the dataset-specific configurations and loss weights used in Eq. (26).

Table 5: Dataset-specific UniSlot hyperparameters.

Hyperparameter	CLEVRER	MOVi-A/B	MOVi-C	MOVi-E	YTVIS 2019/21
Training steps	100k	100k	200k	200k	200k
Slots K	7	11	11	15	7
Crop	Center	Full	Full	Full	Center
Augmentation	–	–	–	–	Horiz. flip
Visual target y_t	RGB	RGB	h_t	h_t	h_t
Decoder hidden dim.	64	64	64	64	256
Upsampling stages	4	4	0	0	0
λ_{recon}	1.0	1.0	1.0	0.3	0.3
λ_{depth}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}
λ_{time}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
λ_{sep}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
λ_{spars}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}
λ_{KL}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}

C.2 Object dynamics prediction hyperparameters

We use dataset-specific burn-in and rollout lengths and balance the prediction of appearance, position, and scale through the weights λ_v, λ_p, and λ_s, respectively. The complete training configuration is reported in Tab. 6.

Table 6: Hyperparameters for downstream object dynamics prediction. Dataset-specific settings for the SlotFormer predictor trained on UniSlot representations.

Hyperparameter	MOVi-C	MOVi-E	YTVIS21
Burn-in frames	6	6	10
Rollout length	10	10	5
Appearance weight λ_v	0.1	0.25	0.25
Position weight λ_p	5.0	0.25	0.25
Scale weight λ_s	2.5	0.25	0.25
Batch size	64	64	64
Learning rate	10^{-4}	10^{-4}	10^{-4}
Training steps	100'000	100'000	100'000

C.3 Baseline hyperparameters

We use the official implementations and configurations of SlotContrast, VideoSAUR, and SAVi, with the dataset-specific adaptations described below. Across datasets, SlotContrast and VideoSAUR use 4-frame training segments, $3/2$ Slot Attention iterations for the first/subsequent frames, and a single-layer, 4-head Transformer predictor. SAVi uses 6-frame segments, 2 Slot Attention iterations, 128-dimensional slots, and the same single-layer, 4-head Transformer predictor. All models are trained with Adam and gradient clipping at 0.05.

SlotContrast For SlotContrast (Manasyan et al., 2025), all experiments use 4-frame training segments, batch size 64, 2.5k learning-rate warmup steps, and gradient clipping at 0.05. We use 3 Slot Attention iterations for the first frame and 2 thereafter, followed by a single-layer, 4-head Transformer predictor. Inputs are resized to 252×252 to match the DINOv2 ViT patch size of 14, and we use $\lambda_{\text{contrast}} = 0.5$ throughout. Dataset-specific settings are reported in Tab. 7.

Table 7: Dataset-specific SlotContrast hyperparameters. We use the official configurations of Manasyan et al. (2025) for MOVi-C, MOVi-E, and YouTube-VIS, and the MOVi-C configuration for CLEVRER and MOVi-A/B.

Hyperparameter	CLEVRER	MOVi-A/B	MOVi-C	MOVi-E	YouTube-VIS
Training steps	100k	100k	100k	300k	100k
Peak LR	$4{\cdot}10^{-4}$	$4{\cdot}10^{-4}$	$4{\cdot}10^{-4}$	$8{\cdot}10^{-4}$	$8{\cdot}10^{-4}$
DINOv2 backbone	ViT-S/14	ViT-S/14	ViT-S/14	ViT-S/14	ViT-B/14
Crop	Center	Full	Full	Full	Center
Augmentation	–	–	–	–	H. flip
Slots K	7	11	11	15	7
Slot dimension	128	128	128	128	64

VideoSAUR For VideoSAUR (Zadaianchuk et al., 2023), all experiments use 4-frame training segments, batch size 128, 2.5k learning-rate warmup steps, a peak learning rate of $4 \cdot 10^{-4}$, and gradient clipping at 0.05. We use DINO ViT-B/8 features at 256×256 resolution, $3/2$ Slot Attention iterations for the first/subsequent frames, and a single-layer, 4-head Transformer predictor. Dataset-specific settings are reported in Tab. 8.

Table 8: Dataset-specific VideoSAUR hyperparameters. We use the official configurations of Zadaianchuk et al. (2023) for MOVi-C, MOVi-E, and YouTube-VIS, and the MOVi-C configuration for CLEVRER and MOVi-A/B.

Hyperparameter	CLEVRER	MOVi-A/B	MOVi-C	MOVi-E	YouTube-VIS
Crop	Center	Full	Full	Full	Center
Augmentation	–	–	–	–	H. flip
Slots K	7	11	11	15	7
Slot dimension	128	128	128	128	64
Allocator layers	2	2	2	2	3
Renderer layers	4	4	4	4	3
λ_{recon}	0	0	0	0	0.1
Temperature τ	0.075	0.075	0.075	0.075	0.25

SAVi For SAVi (Kipf et al., 2022), all experiments use 6-frame training segments, batch size 64, 2.5k learning-rate warmup steps, a peak learning rate of $2 \cdot 10^{-4}$, and gradient clipping at 0.05. We use 128-dimensional slots with 2 Slot Attention iterations and a single-layer, 4-head Transformer predictor. For MOVi-C, MOVi-E, and YouTube-VIS, we replace the CNN encoder with frozen DINO ViT-B/16 features. Dataset-specific settings are reported in Tab. 9.

Table 9: Dataset-specific SAVi hyperparameters. We follow the official configuration of Kipf et al. (2022) for MOVi-A/B and CLEVRER, and use DINO features for the more visually complex datasets.

Hyperparameter	CLEVRER	MOVi-A/B	MOVi-C	MOVi-E	YouTube-VIS
Encoder	CNN	CNN	DINO ViT-B/16	DINO ViT-B/16	DINO ViT-B/16
Crop	Center	Full	Full	Full	Center
Augmentation	–	–	–	–	H. flip
Slots K	7	11	11	15	7

SRL For SRL (Seong et al., 2026), we use the official dataset-specific configurations with 4-frame training segments, 2.5k learning-rate warmup steps, Adam optimization, gradient clipping at 0.05, and 2 Slot Attention iterations. We use the prescribed DINOv2 ViT-S/14 features for MOVi-C and DINOv2 ViT-B/14 features for MOVi-E, while retaining the official dataset-specific architecture and loss settings. All experiments are run at 252×252 resolution. For CLEVRER, we use the MOVi-C configuration with $K = 7$ slots; for YouTube-VIS 2019, we use the YouTube-VIS 2021 configuration. Dataset-specific settings are reported in Tab. 10.

Table 10: Dataset-specific SRL hyperparameters. We follow the official configurations for MOVi-C, MOVi-E, and YouTube-VIS 2021. CLEVRER uses the MOVi-C configuration with $K = 7$, and YouTube-VIS 2019 uses the YouTube-VIS 2021 configuration.

Hyperparameter	CLEVRER	MOVi-C	MOVi-E	YTVIS 2019	YTVIS 2021
Slots K	7	11	15	7	7
Slot dimension	64	64	128	64	64
DINOv2 backbone	ViT-S/14	ViT-S/14	ViT-B/14	ViT-B/14	ViT-B/14
Training steps	100k	100k	300k	100k	100k

SlotCurri For SlotCurri (Moon et al., 2026), we follow the official dataset-specific configurations with 4-frame training segments, 2.5k learning-rate warmup steps, Adam optimization, gradient clipping at 0.05, and 2 Slot Attention iterations. The official curriculum, loss functions, and remaining architecture settings are retained, while the input resolution is changed to 252×252. For CLEVRER, we use the MOVi-C configuration with $K = 7$ slots; for YouTube-VIS 2019, we use the YouTube-VIS 2021 configuration. Dataset-specific settings are reported in Tab. 11.

Table 11: Dataset-specific SlotCurri hyperparameters. We follow the official configurations for MOVi-C, MOVi-E, and YouTube-VIS 2021. CLEVRER uses the MOVi-C configuration with $K = 7$, and YouTube-VIS 2019 uses the YouTube-VIS 2021 configuration.

Hyperparameter	CLEVRER	MOVi-C	MOVi-E	YTVIS 2019	YTVIS 2021
Slots K	7	11	15	7	7
Slot dimension	64	64	128	64	64
DINOv2 backbone	ViT-S/14	ViT-S/14	ViT-B/14	ViT-B/14	ViT-B/14
Training steps	100k	100k	300k	100k	100k

C.4 Compute requirements and inference efficiency

Training compute. All training experiments were conducted on NVIDIA RTX3090 (24GB) and A100 (80GB) GPUs. We use gradient accumulation in PyTorch Lightning to simulate larger batch sizes without multi-GPU training. For UniSlot, the effective batch size is 64 clips of 6 frames. A single training run of UniSlot with DINO ViT-B/16 for 100k steps takes approximately 18 hours on one A100 GPU. All results are averaged over 3 random seeds.

Inference speed. We measure inference efficiency on the YouTube-VIS 2021 dataset using an NVIDIA A100-40GB GPU, PyTorch 2.8.0, and CUDA 12.9. All models are evaluated with batch size 8. We report frames per second (FPS), floating-point operations (GFLOPs), and peak GPU memory usage (VRAM). FPS is measured at the frame level, while GFLOPs and VRAM are reported for the same inference configuration.

As shown in Table 12, UniSlot achieves comparable throughput to SlotContrast while using fewer FLOPs and less memory. Compared with VideoSAUR, UniSlot is faster and more efficient in both computation and memory.

Table 12: Inference efficiency on YouTube-VIS 2021 using an A100-40GB GPU, PyTorch 2.8.0, and batch size 8.

Method	FPS↑	GFLOPs / batch↓	VRAM (GB)↓
VideoSAUR	220	35,900	12.52
SlotContrast	729	13,400	8.44
UniSlot	709	8,900	6.61

D ADDITIONAL ABLATIONS AND ANALYSIS

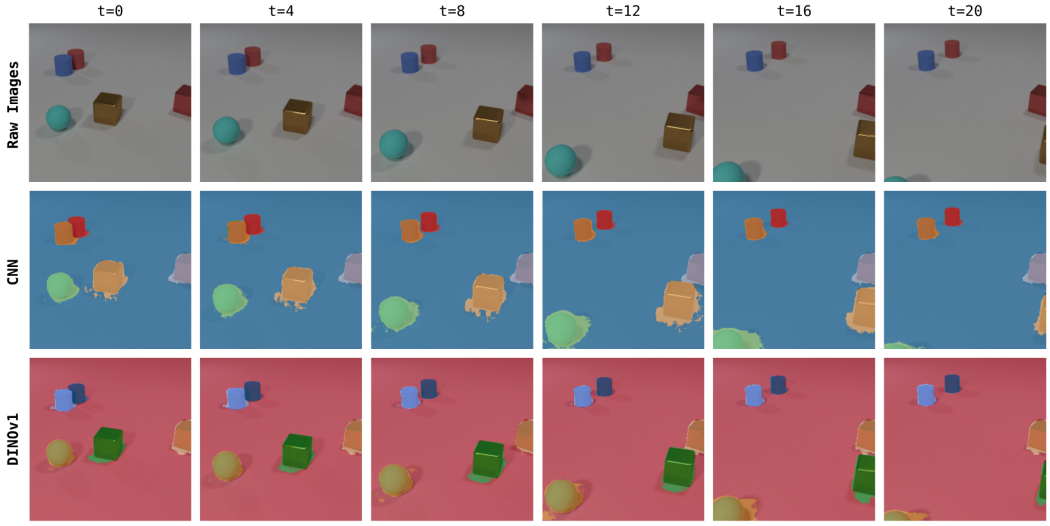
D.1 FEATURE ENCODER

Learnable CNN encoder The main experiments use frozen DINO features as the visual input to UniSlot. We additionally evaluate whether the proposed representation can be learned without a pretrained visual encoder by replacing DINO with a lightweight four-layer CNN trained jointly with UniSlot. We evaluate this setting on CLEVRER and MOVi-A, where appearance is sufficiently simple to learn visual features from scratch. The CNN produces dense features with the same dimensionality as the projected DINO features used by the grouping module.

Table 13: Robustness to a learnable visual encoder. Object decomposition and tracking performance when replacing the frozen DINOv1 encoder with a four-layer CNN trained jointly with UniSlot.

Encoder	CLEVRER			MOVi-A		
	ARI↑	FG-ARI↑	mBO↑	ARI↑	FG-ARI↑	mBO↑
DINOv1	**.927**	**.781**	**.647**	**.882**	.740	**.599**
Learnable CNN	.863	.741	.620	.818	**.747**	.562

Figure 8: Effect of the visual encoder. UniSlot produces similar object decompositions with a lightweight learnable CNN and a frozen DINOv1 encoder on CLEVRER.



As shown in Tab. 13 and Fig. 8, replacing DINOv1 with a learnable CNN leads to only a small decrease on CLEVRER and a larger but still moderate decrease on MOVi-A. This shows that UniSlot can learn object-centric representations from scratch in visually simple settings, while pretrained features provide a stronger visual signal.

Robustness to pretrained visual features We further evaluate whether UniSlot is sensitive to the particular pretrained visual representation by replacing DINOv1 with DINOv2 and DINOv3 while keeping the remaining model unchanged. Quantitative results are reported in Tab. 14, with corresponding qualitative comparisons in Figs. 9 and 10.

Table 14: Robustness to pretrained visual features. Object decomposition and tracking performance of UniSlot with different DINO encoders, evaluated on full video sequences. The DINOv1 configuration used in the main experiments is highlighted in gray.

Encoder	CLEVRER			MOVi-C			YouTube-VIS 2021		
	ARI↑	FG-ARI↑	mBO↑	ARI↑	FG-ARI↑	mBO↑	ARI↑	FG-ARI↑	mBO↑
DINOv1 (Caron et al., 2021)	**.927**	**.781**	.647	**.505**	.654	.358	**.228**	.231	**.455**
DINOv2 (Darcet et al., 2024)	.852	.743	.639	.420	.532	.313	.203	.212	.345
DINOv3 (Siméoni et al., 2026)	.904	.762	**.676**	.451	**.679**	**.365**	.200	**.281**	.432

UniSlot produces consistent object decompositions across all three feature encoders, with relatively small variations across most datasets and metrics. No encoder dominates across all settings: DINOv3 improves FG-ARI and mBO on several benchmarks, while DINOv1 remains strongest in ARI on MOVi-C and YouTube-VIS. The qualitative results show a similar pattern: all three encoders recover comparable object assignments and maintain slot identities over time, while differing in fine-grained segmentation behavior. On YouTube-VIS, DINOv2 occasionally separates object parts into different slots, whereas DINOv3 can produce more semantically coherent groupings, such as separating a tennis racket from the player or a bird from background clutter. On MOVi-C, DINOv1 produces more stable object decompositions, consistent with its quantitative performance.

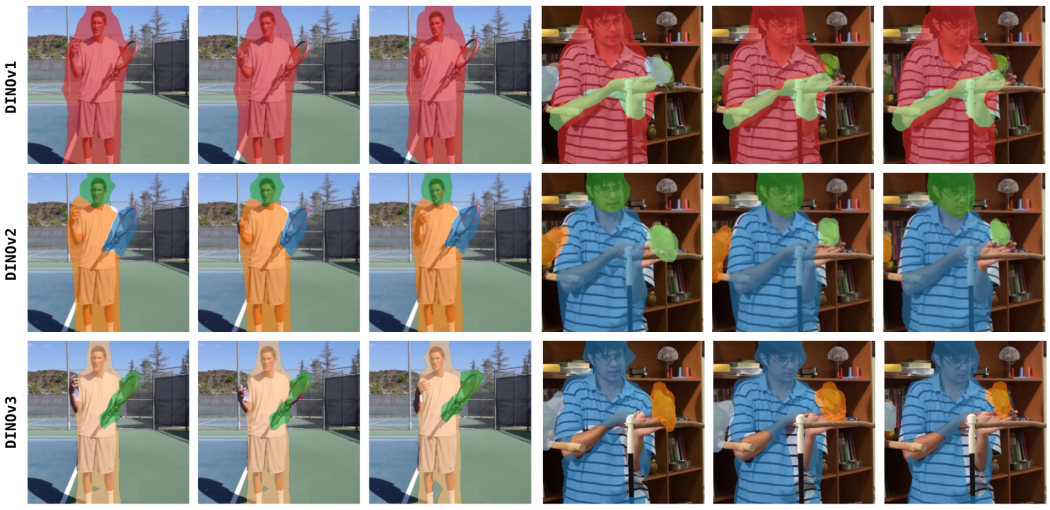


Figure 9: Object tracking on the same YouTube-VIS frames using different pretrained encoders. Despite changes in the distribution of input features, UniSlot produces consistent object masks.

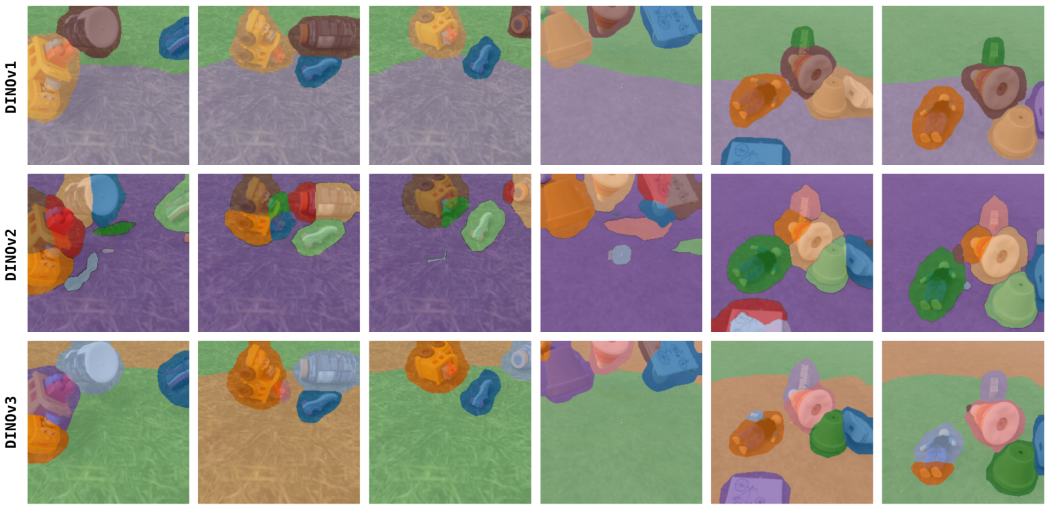


Figure 10: Object tracking on the same MOVi-C frames using different pretrained encoders. Despite changes in the distribution of input features, UniSlot produces consistent object masks.

Overall, these results show that UniSlot remains effective across substantially different pretrained visual features, although the choice of encoder influences the resulting object boundaries and granularity. We use DINOv1 in the main experiments to maintain comparability with prior video object-centric methods.

D.2 Alternative decoder architecture

We additionally examine whether the disentangled slot representation can be used with an alternative decoder architecture. In this experiment, conducted without pseudo-depth supervision, we replace the Spatial Broadcast Decoder (SBD) (Watters et al., 2019) with the per-slot MLP decoder commonly used in feature-based object-centric models (Seitzer et al., 2023; Manasyan et al., 2025).

The MLP independently decodes each slot into the visual target $\hat{y}_t^k$ and alpha mask α_t^k. Instead of shared positional encodings, we use slot-relative coordinates derived from (p_t^k, s_t^k), preserving the separation between appearance and geometry. We use a four-layer MLP with hidden dimensions 768 and 1024 for MOVi-E and YouTube-VIS, respectively.

Table 15: Robustness to decoder architecture. Object tracking performance of UniSlot with Spatial Broadcast and per-slot MLP decoders on MOVi-E and YouTube-VIS 2021.

Decoder	MOVi-E			YouTube-VIS 2021		
	ARI↑	FG-ARI↑	mBO↑	ARI↑	FG-ARI↑	mBO↑
Spatial Broadcast	**.252**	.784	.279	**.219**	.217	**.444**
MLP	.242	**.795**	**.301**	.202	**.230**	.418

Despite the substantial architectural change, the two decoders obtain similar performance: the MLP slightly improves FG-ARI and mBO on MOVi-E, while the SBD performs better on YouTube-VIS. This experiment shows that the disentangled appearance and geometric representation can be used with different per-slot decoders. We use the more efficient SBD in the main model.

D.3 Robustness to the number of slots

We evaluate sensitivity to the maximum number of available slots K. With adaptive activation, K defines an upper bound on the number of represented objects, while unnecessary slots can be deactivated. The binary gate uses hard Gumbel–Softmax with a fixed decision threshold of 0.5.

Table 16 compares UniSlot with VideoSAUR and SlotContrast as K increases beyond the default setting. UniSlot is significantly less sensitive to excess slot capacity, particularly in FG-ARI and mBO. On MOVi-C, these metrics remain stable across $K = 11$–20, whereas SlotContrast collapses at $K = 20$. A similar trend is observed on YouTube-VIS 2021, where the baselines degrade more substantially as K increases.

Table 16: Robustness to the maximum number of slots K. Sequence-level ARI/FG-ARI/mBO are reported for increasing slot capacity. The smallest K corresponds to the default configuration used in the main experiments.

Method	MOVi-C			YTVIS 2021		
	$K = 11$	$K = 15$	$K = 20$	$K = 7$	$K = 10$	$K = 15$
VideoSAUR	.22/.61/.33	.23/.55/.35	.18/.50/.35	.14/.19/.38	.10/.15/.31	.07/.12/.27
SlotContrast	.10/.57/.27	.17/.62/.27	.00/.01/.04	.14/.23/.36	.13/.20/.33	.08/.15/.27
UniSlot	**.51/.65/.36**	**.37/.64/.35**	**.28/.71/.37**	**.23/.23/.46**	**.18/.25/.43**	**.16/.21/.38**

E Additional quantitative results

E.1 Per-image object discovery

The main experiments evaluate object decomposition and tracking at the sequence level, requiring consistent slot assignments across frames. We additionally report per-image results, where ARI, FG-ARI, and mBO are computed independently for each frame and then averaged over the dataset. This evaluation isolates frame-level object decomposition from temporal slot consistency and enables comparison with image-based object-centric methods—DINOSAUR (Seitzer et al., 2023), ISA (Biza et al., 2023), and Adaptive Slot Attention (Fan et al., 2024)—that do not maintain object identities across frames.

Table 17: Per-image object discovery. ARI, FG-ARI, and mBO are computed independently for each frame and then averaged over the dataset. We compare both image-based and video-based object-centric methods. Best and second-best results are shown in bold and underlined, respectively.

Method	CLEVRER			MOVi-C			MOVi-E		
	ARI↑	FG-ARI↑	mBO↑	ARI↑	FG-ARI↑	mBO↑	ARI↑	FG-ARI↑	mBO↑
Image-based methods									
DINOSAUR (Seitzer et al., 2023)	.046	**.971**	.153	.152	.650	.402	.128	.607	**.381**
ISA (Biza et al., 2023)	.032	.851	.164	.145	.612	.385	.152	.701	.314
AdaSlot (Fan et al., 2024)	.020	.344	.064	.018	.209	.044	.073	.202	.094
Video-based methods									
SAVi (Kipf et al., 2022)	.012	.411	.069	.063	.466	.217	.011	.612	.288
VideoSAUR (Zadaianchuk et al., 2023)	.372	.281	.184	.241	.733	**.434**	**.253**	.667	.331
SlotContrast (Manasyan et al., 2025)	.050	.967	.154	.109	.708	.231	.203	.774	.303
SRL (Seong et al., 2026)	.084	.891	.112	.057	.310	.078	.148	.803	.284
SlotCurri (Moon et al., 2026)	.042	.935	.094	.195	.725	.326	.052	.645	.207
UniSlot (Ours)	**.927**	.952	**.741**	**.544**	**.765**	.403	.243	**.831**	.283

Method	YTVIS 2019			YTVIS 2021		
	ARI↑	FG-ARI↑	mBO↑	ARI↑	FG-ARI↑	mBO↑
Image-based methods						
DINOSAUR (Seitzer et al., 2023)	.201	.264	.524	.175	.215	.502
ISA (Biza et al., 2023)	.194	.245	.525	.183	.231	**.528**
AdaSlot (Fan et al., 2024)	.086	.164	.273	.075	.142	.255
Video-based methods						
SAVi (Kipf et al., 2022)	.058	.197	.268	.052	.181	.233
VideoSAUR (Zadaianchuk et al., 2023)	.173	.170	.445	.160	.227	.443
SlotContrast (Manasyan et al., 2025)	.191	.221	.446	.162	.276	.398
SRL (Seong et al., 2026)	.244	.206	.442	.112	.276	.265
SlotCurri (Moon et al., 2026)	.102	.164	.290	.073	.252	.293
UniSlot (Ours)	**.329**	**.303**	**.537**	**.244**	**.310**	.470

E.2 Additional results on MOVi-A and MOVi-B

For completeness, we additionally evaluate UniSlot on MOVi-A and MOVi-B, which provide simpler variants of the MOVi benchmark than MOVi-C and MOVi-E used in the main evaluation. Table 18 reports sequence-level ARI, FG-ARI, and mBO, using the same evaluation protocol as in the main paper.

Table 18: Additional results on MOVi-A and MOVi-B. Sequence-level object decomposition and tracking performance, evaluated using the same protocol as in the main experiments. Best and second-best results are shown in bold and underlined, respectively.

Method	MOVi-A			MOVi-B		
	ARI↑	FG-ARI↑	mBO↑	ARI↑	FG-ARI↑	mBO↑
SAVi (Kipf et al., 2022)	.004	.083	.031	.139	.265	.152
VideoSAUR (Zadaianchuk et al., 2023)	.192	.304	.145	.181	.269	.143
SlotContrast (Manasyan et al., 2025)	.177	**.905**	.194	.148	**.826**	.268
UniSlot (Ours)	**.882**	.740	**.599**	**.732**	.544	**.502**

F Additional qualitative results

We provide additional qualitative comparisons across synthetic and real-world benchmarks to complement the results in the main paper. The examples illustrate object decomposition and tracking over time and compare UniSlot with the main video-based baselines.

Figure 11: Additional qualitative results on YouTube-VIS 2021 (part 1). Colors indicate slot assignments and are kept consistent across frames to visualize temporal object tracking.



Figure 12: Additional qualitative results on YouTube-VIS 2021 (part 2). Colors indicate slot assignments and are kept consistent across frames to visualize temporal object tracking.

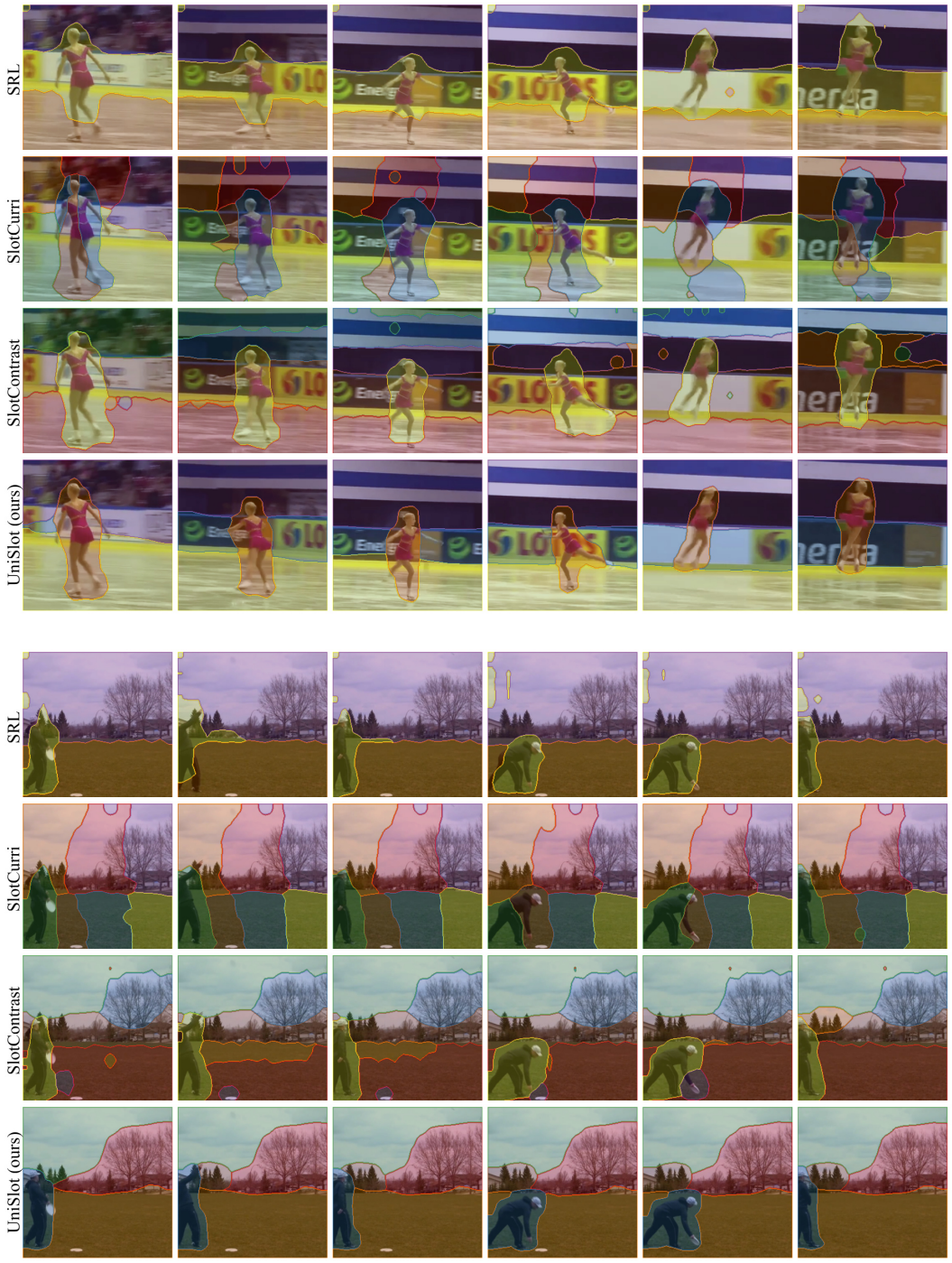


Figure 13: Additional qualitative results on MOVi-C. Colors indicate slot assignments and are kept consistent across frames to visualize temporal object tracking.



Figure 14: Additional qualitative results on CLEVRER. Colors indicate slot assignments and are kept consistent across frames to visualize temporal object tracking.

