TSA: TEMPORAL SLOT ACTIVATION FOR PERSISTENT OBJECT-CENTRIC VIDEO REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised video object-centric learning aims to decompose dynamic scenes into temporally persistent entity representations. Existing recurrent video slot-attention methods propagate a fixed set of slots across frames, but typically assume *unconditional slot propagation*: every slot is updated and decoded at every frame, regardless of whether its corresponding object is visible. We show that this design violates a basic lifecycle requirement for persistent slots: when an object is absent or fully occluded, its slot should preserve its previous state and avoid explaining unrelated visible content. Instead, unconditional propagation creates *update-induced state drift*, where current-frame evidence overwrites the absent object's representation, and *decoder-induced reconstruction interference*, where the inactive slot remains coupled to reconstruction through decoder attention. We propose **Temporal Slot Activation (TSA)**, a lightweight mechanism that learns a per-slot, per-frame activation score $\alpha_{k,t} \in (0, 1)$ without visibility supervision. TSA uses this activation as a shared latent control variable for slot lifecycle modeling. When a slot is inactive, TSA anchors its state to the previous slot through activation-gated updating and suppresses its decoder participation through an activation-dependent additive bias on attention logits before softmax normalization. This jointly reduces state drift and reconstruction-driven interference. To improve decisions under partial occlusion and gradual reappearance, TSA further conditions activation prediction on a per-slot temporal memory produced by a Temporal Context Encoder. We evaluate TSA on MOVi-C, MOVi-E, YouTube-VIS HQ, and the occlusion-heavy OVIS benchmarks, using both standard metrics (FG-ARI, mBO) and tracking-based metrics (IDF1, HOTA). TSA improves object decomposition and temporal identity preservation, with large gains on long, heavily occluded videos. Source code will be released upon publication.

1 INTRODUCTION

Humans perceive visual scenes as collections of persistent objects that remain identifiable through motion, occlusion, and reappearance (Spelke & Kinzler, 2007; Kahneman et al., 1992). Object-centric learning (OCL) aims to recover such structure without supervision by decomposing visual inputs into entity-level representations (Greff et al., 2019; Eslami et al., 2016; Burgess et al., 2019; Engelcke et al., 2019; Lin et al., 2020). Slot Attention (SA) (Locatello et al., 2020) has become a standard formulation for OCL and a foundation for compositional reasoning and prediction tasks (Wu et al., 2022; 2023; Kakogeorgiou et al., 2024; Seitzer et al., 2022). Extending SA from images to videos introduces a central requirement, *temporal consistency*: a slot should preserve the same object identity across time, including under partial or full occlusion. To this end, Video Slot Attention (VSA) methods propagate slot states forward and update them using the current frame (Kipf et al., 2021; Elsayed et al., 2022; Zadaianchuk et al., 2023; Manasyan et al., 2025; Zhao et al., 2026).

Despite their effectiveness, these methods share a common structural assumption that we call *unconditional slot propagation*: every slot is updated and decoded at every frame, regardless of whether the corresponding object is currently visible. This assumption conflates object persistence with visual presence, leading to *state drift* under occlusion. Fig. 1 illustrates this failure: when a kayaker becomes fully occluded by a capsized kayak, the competition in Slot Attention forces every slot to align with some visible content, so the kayaker's slot is reassigned to the occluding kayak hull. This overwrites the previously stored object representation with unrelated features, and the slot gradually

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.

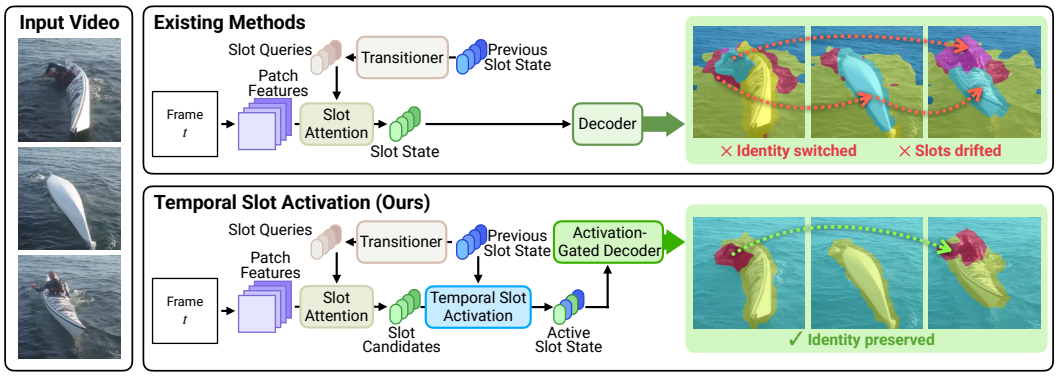

Figure 1: **Unconditional slot propagation vs. TSA under occlusion.** *Top:* Without activation gating, the kayaker's slot drifts toward the occluding hull and triggers an identity switch. *Bottom:* TSA deactivates the absent slot via $\alpha_{k,t}$, preserving its state for consistent reacquisition.

loses the identity of the object it was tracking. When the object reappears, the corrupted slot state can no longer function as a meaningful query for reacquisition, and the object is instead captured by another slot, resulting in an *identity switch*. In this sense, representation drift is the underlying mechanism, while identity switch is its observable consequence. This drift is jointly driven by two coupled mechanisms: *unconditional state update* and *unconditional decoder participation* (Sec. 3).

We address this problem by introducing **Temporal Slot Activation (TSA)**, a lightweight mechanism that assigns each slot k at frame t a learned *activation score* $\alpha_{k,t} \in (0, 1)$, trained without visibility supervision. The activation score serves as a shared latent control variable governing the slot lifecycle. For slot-state evolution, TSA performs an *activation-gated state update*: active slots ($\alpha_{k,t} \to 1$) focus on the current SA candidate, whereas inactive slots ($\alpha_{k,t} \to 0$) remain anchored to their previous states, preventing occlusion-induced overwriting. For decoding, TSA performs *activation-gated decoder participation* by applying an additive log-bias on cross-attention logits before softmax, suppressing inactive slots during decoder competition. Through this dual gating, TSA enforces consistent inactive-but-persistent behavior, with an inactive slot simultaneously protected from current-frame updates and prevented from explaining unrelated visible content.

We evaluate TSA on the standard video OCL benchmarks MOVi-C, MOVi-E (Greff et al., 2022), and YouTube-VIS HQ (Ke et al., 2022). To further examine identity persistence in more challenging settings, we additionally evaluate on **OVIS** (Qi et al., 2022), which features crowded scenes, long videos, severe occlusions, and objects that frequently disappear and reappear. On all benchmarks, we report the conventional grouping metrics FG-ARI and mBO together with the identity-sensitive tracking metrics HOTA (Luiten et al., 2021) and IDF1 (Ristani et al., 2016).

Our contributions are: **(i)** We identify *unconditional slot propagation* as a **fundamental limitation** of recurrent VSA methods. We provide a formal analysis showing how it causes *representation drift* and *identity switch* through two coupled mechanisms: *update-induced state drift* and *decoder-induced reconstruction interference* (Sec. 3). **(ii)** We propose **Temporal Slot Activation (TSA)**, a lightweight mechanism equipping each slot with a learned per-frame *activation score* $\alpha_{k,t}$ that jointly controls slot-state evolution and decoder participation, enabling inactive-but-persistent slot behavior (Sec. 4). **(iii)** We extend the evaluation protocol with an occlusion-centric benchmark and identity-sensitive tracking metrics, making long-term slot persistence explicitly measurable (Sec. 5).

2 Related Work

Object-centric learning (OCL) and Slot Attention (SA). OCL aims to represent a visual scene as a set of discrete object-level entities without supervision. Early approaches achieved this through sequential attention or iterative inference mechanisms that extract objects from an image (Eslami et al., 2016; Burgess et al., 2019; Greff et al., 2019). SA (Locatello et al., 2020) later introduced a scalable alternative based on competitive cross-attention, where a fixed set of slots compete to explain the scene, and has since become the dominant paradigm in OCL. Subsequent work has primarily focused on improving *per-frame decomposition quality*. These improvements come from

stronger pretrained visual features (Oquab et al., 2023; Seitzer et al., 2022), more expressive generative decoders (Wu et al., 2023; Jiang et al., 2023), and flexible slot parameterizations (Fan et al., 2024; Liu et al., 2025). These advances are developed for the single-image setting and form the basis for extensions to video.

Slot utilization. A related direction asks how many slots a scene needs. SOLV (Aydemir et al., 2023) merges slots with high cosine similarity into one. AdaSlot (Fan et al., 2024) predicts which slots to keep and masks the rest in the decoder. MetaSlot (Liu et al., 2025) maps slots to a prototype codebook and removes duplicates. All three judge redundancy at inference from the current frame alone. Others target the same problem through training instead, by regularizing redundant slots or increasing the slot budget over the course of training (Seong et al., 2026; Moon et al., 2026).

Video Slot Attention (VSA). Extending beyond per-frame decomposition, the video setting requires each slot to consistently represent the same object across frames. SAVi (Kipf et al., 2021) addressed this by propagating slot states over time using a learned transition function, followed by refinement with SA at each frame. Subsequent work improves robustness by incorporating additional cues such as depth (Elsayed et al., 2022), discrete tokens (Singh et al., 2022), and stronger pretrained features (Zadaianchuk et al., 2023). Beyond architectural design, another line of work focuses on improving *temporal consistency through training objectives*. For example, VideoSAUR (Zadaianchuk et al., 2023) introduces temporal feature-similarity losses, while SlotContrast (Manasyan et al., 2025) enforces slot identity consistency via contrastive learning. Others refine the propagation itself. RandSF.Q (Zhao et al., 2026) improves temporal prediction by conditioning transitions on sampled slot-feature pairs, and SmoothSA (Zhao et al., 2025) smooths SA recurrences across frames by preheating cold-start first-frame queries and using fewer iterations on subsequent frames.

Despite these advances, all prior recurrent VSA methods share the same regime of *unconditional propagation*: every slot is updated and decoded at every frame. Existing mitigations of temporal inconsistency act on *how* slots are propagated, through transition dynamics (Kipf et al., 2021; Zhao et al., 2026), query initialization and iteration schedules (Zhao et al., 2025), or temporal objectives (Zadaianchuk et al., 2023; Manasyan et al., 2025), but not on *whether* each slot should be updated or decoded. Slot-utilization methods withhold slots, but on a different criterion: they judge redundancy against the visible content, under which an object hidden behind an occluder is indistinguishable from one that is not in the scene, so the slot holding its identity is released rather than preserved for reappearance, as detailed in Appendix A. Thus, slot-level object correspondence remains implicit, without a controlled mechanism for preserving object identity in challenging temporal scenarios such as occlusion. In contrast, TSA introduces a learned *activation score* that explicitly determines whether each slot is updated and decoded at each frame while preserving inactive slots, an axis of control orthogonal to slot propagation and slot-utilization methods.

3 Limitations of Unconditional Slot Propagation

Recurrent VSA Pipeline. Let $\boldsymbol{s}_{k,t} \in \mathbb{R}^d$ denote the state of slot $k \in \{1, \dots, K\}$ at frame $t \in \{1, \dots, T\}$, with slot dimension d, and let $\boldsymbol{F}_t \in \mathbb{R}^{N \times d_f}$ denote the N features of dimension d_f extracted by a frozen visual encoder at time t. Given the previous slot states $\boldsymbol{S}_{t-1} = \{\boldsymbol{s}_{k,t-1}\}_{k=1}^K$, a Temporal Query Transitioner (T_ϕ) predicts a query $\boldsymbol{q}_{k,t} = T_\phi(\boldsymbol{S}_{t-1}, \boldsymbol{F}_t)$ for each slot. Existing VSA methods (Kipf et al., 2021; Elsayed et al., 2022; Singh et al., 2022; Zadaianchuk et al., 2023; Manasyan et al., 2025; Zhao et al., 2026) typically adopt SA, denoted U_θ, to align each query with current-frame evidence via competitive cross-attention:

$$\boldsymbol{s}_{k,t} = \mathrm{SA}(\boldsymbol{F}_t, \boldsymbol{q}_{k,t}) = U_\theta\big(\boldsymbol{F}_t, T_\phi(\boldsymbol{S}_{t-1}, \boldsymbol{F}_t)\big). \tag{1}$$

All slots are then passed to the decoder. Let $\boldsymbol{q}_n^d \in \mathbb{R}^{d_k}$ be the decoder query at position n, and let $\boldsymbol{k}^d(\boldsymbol{s}_{k,t}) \in \mathbb{R}^{d_k}$, $\boldsymbol{v}^d(\boldsymbol{s}_{k,t})$ be the key and value projected from slot k. The decoder attention logits and weights are

$$z_{k,n,t} = \frac{1}{\sqrt{d_k}} \boldsymbol{q}_n^{d\top} \boldsymbol{k}^d(\boldsymbol{s}_{k,t}), \qquad A^d_{k,n,t} = \frac{\exp(z_{k,n,t})}{\sum_{j=1}^K \exp(z_{j,n,t})}. \tag{2}$$

We refer to Eq. 1 as *unconditional state update* and to Eq. 2 as *unconditional decoder participation*, because $A^d_{k,n,t} > 0$ for all k, n, t, a strict positivity that causes every slot to enter the softmax competition of the decoder at every frame. Together they form the *unconditional slot propagation* regime. To assess this regime, we adopt the lens of *object persistence*: each slot is expected to

represent the identity of one object over time. Let $o_{k,t} \in \{0, 1\}$ denote the visibility of the object represented by slot k at frame t, with $o_{k,t} = 1$ when the object is visible and $o_{k,t} = 0$ when it is absent or fully occluded. We say slot k is *active* at frame t if $o_{k,t} = 1$ and *inactive* if $o_{k,t} = 0$. An active slot should update using the current-frame evidence, whereas an inactive slot should remain persistent, preserving identity for future reappearance. Unconditional slot propagation violates this expectation, giving rise to two structurally distinct failure pathways analyzed below.

Failure Pathway I: Update-Induced State Drift. Consider an interval $\mathbb{I}_{[a,b]} = \{a, \dots, b\}$ during which the object associated with slot k is absent. Object persistence requires $\boldsymbol{s}_{k,t} = \boldsymbol{s}_{k,t-1}$ throughout $\mathbb{I}_{[a,b]}$, since $\boldsymbol{F}_t$ may contain information about other visible objects and the background, but no evidence for object k. Under *unconditional state updating* $\boldsymbol{s}_{k,t} = U_\theta(\boldsymbol{F}_t, T_\phi(\boldsymbol{S}_{t-1}, \boldsymbol{F}_t))$, however, state drift accumulates over the absence interval:

$$\|\boldsymbol{s}_{k,b} - \boldsymbol{s}_{k,a-1}\| \;\le\; \sum_{t=a}^{b} \|\boldsymbol{s}_{k,t} - \boldsymbol{s}_{k,t-1}\| \;=\; \sum_{t=a}^{b} \big\|U_\theta\big(\boldsymbol{F}_t, T_\phi(\boldsymbol{S}_{t-1}, \boldsymbol{F}_t)\big) - \boldsymbol{s}_{k,t-1}\big\|. \tag{3}$$

Because each update is computed from $\boldsymbol{F}_t$ and, through T_ϕ, from the already-drifted state, successive changes tend to point in the same direction rather than cancel, so even small frame-to-frame changes can lead to substantial deviation from the pre-occlusion identity as the absence duration increases.

Failure Pathway II: Decoder-Induced Reconstruction Interference. One might attempt to address update-induced drift by freezing the slot state during absence. However, Eq. 2 shows that $A^d_{k,n,t} > 0$ for all k, n, t, so every slot contributes to the decoded output even when the corresponding object is absent. This creates a training-time reconstruction pathway from the loss to the inactive slots. For a reconstruction loss $\mathcal{L}_{\text{recon}}(\hat{\boldsymbol{y}}_t, \boldsymbol{y}_t)$, where $\hat{\boldsymbol{y}}_t = \{\hat{\boldsymbol{y}}_{n,t}\}_{n=1}^N$ and $\boldsymbol{y}_t = \{\boldsymbol{y}_{n,t}\}_{n=1}^N$ denote the decoded and target features at frame t, the derivative with respect to slot $\boldsymbol{s}_{k,t}$ presents:

$$\frac{\partial \mathcal{L}_{\text{recon}}}{\partial \boldsymbol{s}_{k,t}} = \sum_n \frac{\partial \mathcal{L}_{\text{recon}}}{\partial \hat{\boldsymbol{y}}_{n,t}} \frac{\partial \hat{\boldsymbol{y}}_{n,t}}{\partial \boldsymbol{s}_{k,t}}, \quad \text{where} \quad \frac{\partial \hat{\boldsymbol{y}}_{n,t}}{\partial \boldsymbol{s}_{k,t}} = A^d_{k,n,t} \frac{\partial \boldsymbol{v}^d(\boldsymbol{s}_{k,t})}{\partial \boldsymbol{s}_{k,t}} + \sum_{j=1}^{K} \boldsymbol{v}^d(\boldsymbol{s}_{j,t}) \frac{\partial A^d_{j,n,t}}{\partial \boldsymbol{s}_{k,t}}. \tag{4}$$

Since $A^d_{k,n,t}$ is strictly positive, the inactive slot remains coupled to the reconstruction. Consequently, the decoder can use information from an inactive slot to reduce reconstruction error for unrelated visible content. The training objective thus propagates gradients through inactive slots, optimizing the model parameters in a way that may undermine inactive-but-persistent behavior.

Design Requirement. The above analysis shows that the two failure pathways are structurally distinct. A valid solution must jointly regulate two conditions: (i) Should slot k update from the current frame? and (ii) Should slot k participate in reconstructing the current frame? For an inactive slot, the desired behavior is

$$\underbrace{o_{k,t} = 0 \Rightarrow \boldsymbol{s}_{k,t} \approx \boldsymbol{s}_{k,t-1}}_{\text{(A) Pathway I}} \qquad \underbrace{o_{k,t} = 0 \Rightarrow A^d_{k,n,t} \approx 0 \;\; \forall n}_{\text{(B) Pathway II}}. \tag{5}$$

Condition (A) prevents update-induced state drift, while condition (B) removes the inactive slot from decoder competition and suppresses reconstruction-driven interference. This motivates a shared activation variable $\alpha_{k,t} \in (0, 1)$ that jointly controls both pathways via *activation-gated state update* and *activation-gated decoder participation*: $\alpha_{k,t} \to 0 \Rightarrow \{\boldsymbol{s}_{k,t} \approx \boldsymbol{s}_{k,t-1}$ and $A^d_{k,n,t} \to 0\}$. Using a single activation variable is important for inactive-but-persistent slot behavior, because if state updating and decoder participation were controlled independently, one pathway could remain active while the other is suppressed.

4 Temporal Slot Activation

We instantiate the design requirement in Sec. 3 with **Temporal Slot Activation (TSA)**. Each slot k at frame t is equipped with a learned scalar *activation score* $\alpha_{k,t} \in (0, 1)$, trained without visibility supervision. When the slot is *active* ($\alpha_{k,t} \to 1$), it updates its state and contributes to reconstruction normally. When *inactive* ($\alpha_{k,t} \to 0$), $\alpha_{k,t}$ simultaneously *freezes* the slot state (satisfying Eq. 5(A)) and *silences* the slot in the decoder (satisfying Eq.5(B)). Fig. 2 illustrates the complete forward pass.

4.1 Slot activation estimator

Given the slot query $\boldsymbol{q}_{k,t}$ from the transition module T_ϕ, SA (U_θ) refines it using current-frame features $\boldsymbol{F}_t$ to produce a candidate state $\tilde{\boldsymbol{s}}_{k,t} = U_\theta(\boldsymbol{F}_t, \boldsymbol{q}_{k,t})$. TSA then predicts the activation score

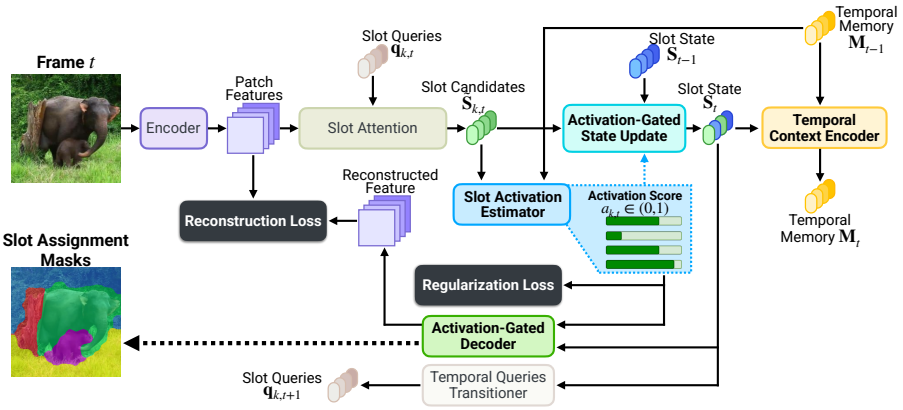

Figure 2: **Overview of Temporal Slot Activation (TSA).** At each frame t, Slot Attention refines slot queries $\boldsymbol{q}_{k,t}$ into slot candidates $\tilde{\boldsymbol{s}}_{k,t}$, from which the Slot Activation Estimator predicts a learned activation score $\alpha_{k,t}$. The score jointly controls state updates (Eq. 7) and decoder attention (Eq. 8), freezing and silencing inactive slots while allowing active ones to track normally.

with a shared Slot Activation Estimator Φ_{act}:

$$\alpha_{k,t} = \sigma\big(\Phi_{\text{act}}\big(\tilde{\boldsymbol{s}}_{k,t},\, \boldsymbol{m}_{k,t-1}\big)\big)\,, \tag{6}$$

where $\boldsymbol{m}_{k,t-1} \in \mathbb{R}^{d_h}$ denotes the *temporal memory* of slot k, a recurrent summary of its trajectory $[\boldsymbol{s}_{k,1}, \ldots, \boldsymbol{s}_{k,t-1}]$ produced by a Temporal Context Encoder Ψ_{tce} (Sec. 4.4), capturing the slot's accumulated behavior over time. We condition Φ_{act} on the slot candidate $\tilde{\boldsymbol{s}}_{k,t}$ rather than the query $\boldsymbol{q}_{k,t}$ because the candidate is directly grounded in current-frame evidence. The query is inherited from $\boldsymbol{s}_{k,t-1}$ and remains object-like even when the object is absent, making it a weak deactivation signal. In contrast, $\tilde{\boldsymbol{s}}_{k,t}$ reflects current-frame evidence directly: when the object is absent, SA fails to align the slot to any region of the object it was tracking, producing a candidate weakly aligned with the slot's own object that serves as a signal for deactivation. The temporal memory $\boldsymbol{m}_{k,t-1}$ supplements this with trajectory context, enabling more robust decisions in ambiguous regimes, such as partial occlusion or gradual reappearance, where $\tilde{\boldsymbol{s}}_{k,t}$ alone may be misleading.

4.2 Activation-Gated State Update

To prevent update-induced state drift, TSA replaces state adoption with an activation-gated update:

$$\boldsymbol{s}_{k,t} = \alpha_{k,t}\, \tilde{\boldsymbol{s}}_{k,t} + (1 - \alpha_{k,t})\, \boldsymbol{s}_{k,t-1}. \tag{7}$$

When the slot is active, $\alpha_{k,t} \to 1$ and the model incorporates the current candidate. When the slot is inactive, $\alpha_{k,t} \to 0$ and the update reduces to $\boldsymbol{s}_{k,t} \to \boldsymbol{s}_{k,t-1}$, satisfying condition (A) of Eq. 5. The candidate is still computed for activation prediction, since it provides the primary deactivation signal to the estimator Φ_{act}, but its ability to overwrite the stored slot state is controlled by $\alpha_{k,t}$.

4.3 Activation-Gated Decoder Participation

The activation-gated state update (Sec. 4.2) alone does not remove an inactive slot from the reconstruction pathway: the slot may still enter the decoder softmax and receive reconstruction-driven gradients. TSA therefore uses the same activation score to modulate decoder attention. Let $z_{k,n,t}$ denote the decoder attention logit between slot k and spatial position n. TSA injects the activation score as a pre-softmax additive bias:

$$A^d_{k,n,t} = \text{softmax}_k(z_{k,n,t} + \log \alpha_{k,t}) = \frac{\alpha_{k,t} \exp(z_{k,n,t})}{\sum_{j=1}^{K} \alpha_{j,t} \exp(z_{j,n,t})}. \tag{8}$$

Placing the log-bias inside the softmax embeds activation directly into slot competition: as $\alpha_{k,t}$ decreases, the contribution of slot k is multiplicatively downweighted before normalization. In the inactive limit ($\alpha_{k,t} \to 0$), the biased logit $z_{k,n,t} + \log \alpha_{k,t} \to -\infty$, yielding $A^d_{k,n,t} \to 0$ for all positions. Thus the same scalar that gates state updates also suppresses decoder participation, aligning decoding with the desired inactive behavior.

This mechanism acts as a learned, continuous analogue of attention masking. Unlike fixed binary masks, TSA uses a dynamic slot-wise gate: intermediate $\alpha_{k,t}$ softly attenuates uncertain slots, while

$\alpha_{k,t} \to 0$ enforces hard exclusion. Because gating occurs before softmax, it removes both pathways through which inactive slots affect reconstruction: their direct contribution vanishes, and their influence on normalization disappears. Consequently, decoder gating not only preserves inactive slots in the state space but also prevents them from explaining unrelated visible content.

4.4 Temporal context encoder

Single-frame evidence is often unreliable for activation, especially under occlusion where residual features can produce convincing but incorrect slot candidates. To mitigate this, each slot maintains a *temporal memory* $\boldsymbol{m}_{k,t} \in \mathbb{R}^{d_h}$ that summarizes its recent trajectory, providing Φ_{act} with historical context to complement the current-frame signal $\tilde{\boldsymbol{s}}_{k,t}$. For this memory to be useful upon reappearance, it must be shielded from current-frame evidence during absence; otherwise drift would corrupt the trajectory context needed for reactivation, the memory-level analogue of the state-level drift analyzed in Sec. 3. We compute $\boldsymbol{m}_{k,t}$ via a Temporal Context Encoder Ψ_{tce} conditioned on the *post-gate* slot state $\boldsymbol{s}_{k,t}$:

$$\boldsymbol{m}_{k,t} = \Psi_{\text{tce}}(\boldsymbol{m}_{k,t-1},\, \boldsymbol{s}_{k,t}), \qquad \boldsymbol{m}_{k,0} = \mathbf{0}. \tag{9}$$

The key design choice is conditioning on $\boldsymbol{s}_{k,t}$ rather than $\tilde{\boldsymbol{s}}_{k,t}$. When the slot is inactive, Eq. 7 ensures $\boldsymbol{s}_{k,t} \approx \boldsymbol{s}_{k,t-1}$, so Ψ_{tce} keeps receiving the preserved pre-absence state rather than current-frame evidence: the activation gate that protects the slot state simultaneously protects the memory from unrelated content, without any additional mechanism. Conditioning on $\tilde{\boldsymbol{s}}_{k,t}$ instead would expose the memory to current-frame evidence during inactivity, reintroducing through the memory pathway the same overwrite problem that Eq. 7 closes at the state level. To prevent unintended interference, $\boldsymbol{m}_{k,t}$ is routed exclusively to Φ_{act}, with no direct connection to the decoder or the transition module T_ϕ.

4.5 Training objectives

TSA is trained with a reconstruction loss $\mathcal{L}_{\text{recon}}$ (Zadaianchuk et al., 2023) and a slot-consistency contrastive loss $\mathcal{L}_{\text{ssc}}$ (Manasyan et al., 2025), augmented with an activation regularizer $\mathcal{L}_{\text{reg}}$ composed of two complementary terms. Without regularization, $\mathcal{L}_{\text{recon}}$ alone admits two degenerate solutions: *full-activation collapse* ($\alpha_{k,t} \equiv 1$), where TSA reduces to unconditional propagation, and *ambiguous gating* ($\alpha_{k,t} \approx 0.5$), where neither pathway is decisively controlled. We address both failure modes with a single regularizer,

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{usage}} + \beta\, \mathcal{L}_{\text{sparse}}, \text{ where } \mathcal{L}_{\text{usage}} = \frac{1}{KT} \sum_{k,t} \alpha_{k,t} \text{ and } \mathcal{L}_{\text{sparse}} = \frac{1}{KT} \sum_{k,t} \alpha_{k,t}(1 - \alpha_{k,t}), \tag{10}$$

where $\mathcal{L}_{\text{usage}}$ penalizes mean activation, creating pressure to deactivate slots that do not improve reconstruction and thereby preventing full-activation collapse; $\mathcal{L}_{\text{sparse}}$ penalizes intermediate activation, sharpening decisions toward near-binary behavior and preventing ambiguous gating; and β controls the relative weight between them. Full-activation collapse ($\alpha_{k,t} \equiv 1$) satisfies $\mathcal{L}_{\text{sparse}} = 0$ but maximizes $\mathcal{L}_{\text{usage}}$, while ambiguous gating ($\alpha_{k,t} \approx 0.5$) maximizes $\mathcal{L}_{\text{sparse}}$ but keeps $\mathcal{L}_{\text{usage}}$ at a moderate level. Neither degenerate mode can minimize both simultaneously, so the combined $\mathcal{L}_{\text{reg}}$ drives activations toward sparse, near-binary behavior. The full training objective is

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{ssc}}\, \mathcal{L}_{\text{ssc}} + \lambda_{\text{reg}}\, \mathcal{L}_{\text{reg}}. \tag{11}$$

5 Experiments

5.1 Experimental setup

Datasets and metrics. Following standard protocols (Zadaianchuk et al., 2023; Manasyan et al., 2025; Zhao et al., 2026; 2025), we evaluate on MOVi-C, MOVi-E, and YouTube-VIS HQ. We additionally include OVIS to stress-test persistence under severe occlusion and crowded scenes. We report FG-ARI ↑ and mBO ↑ as standard object-centric metrics, and further include HOTA ↑ (Luiten et al., 2021) and IDF1 ↑ (Ristani et al., 2016) to directly assess temporal association quality. Dataset details are provided in Appendix B and full metric definitions in Appendix C.

Implementation details. All experiments use a frozen DINOv2 ViT-S/14 (Oquab et al., 2023) encoder. Frames are provided at 256×256 and resized to 224×224 for the encoder, with slot budgets $K \in \{11, 24, 7, 22\}$ for MOVi-C, MOVi-E, YouTube-VIS HQ, and OVIS respectively. Φ_{act}

Table 1: Results on synthetic benchmarks. Mean ± std over 3 seeds. Best in **bold**, second best underlined.

Method	MOVi-C ($K = 11$) *(simple, 24 frames)*				MOVi-E ($K = 24$) *(complex, 24 frames)*				Params (M)↓	FPS↑
	FG-ARI↑	mBO↑	HOTA↑	IDF1↑	FG-ARI↑	mBO↑	HOTA↑	IDF1↑		
VideoSAUR	$53.3_{\pm 2.1}$	$16.1_{\pm 0.4}$	$17.8_{\pm 0.6}$	$8.1_{\pm 1.6}$	$34.6_{\pm 20.7}$	$8.3_{\pm 4.9}$	$9.8_{\pm 3.9}$	$3.2_{\pm 1.5}$	25.1	344
SlotContrast	$59.9_{\pm 5.3}$	$27.7_{\pm 3.0}$	$32.1_{\pm 3.2}$	$29.7_{\pm 6.9}$	$70.6_{\pm 3.8}$	$20.7_{\pm 1.4}$	$22.8_{\pm 1.3}$	$10.7_{\pm 3.5}$	31.4	295
RandSF.Q_{tsim}	$66.3_{\pm 1.7}$	$28.4_{\pm 1.3}$	$32.8_{\pm 1.7}$	$32.4_{\pm 3.2}$	$74.0_{\pm 1.3}$	$22.9_{\pm 0.9}$	$25.9_{\pm 1.9}$	$16.3_{\pm 4.5}$	34.1	411
RandSF.Q_{ssc}	$67.4_{\pm 2.1}$	$29.2_{\pm 3.8}$	$\underline{33.9}_{\pm 3.9}$	$\mathbf{33.1}_{\pm \mathbf{6.5}}$	$\underline{82.1}_{\pm 3.1}$	$23.0_{\pm 1.2}$	$25.9_{\pm 1.2}$	$14.9_{\pm 1.5}$	34.1	411
SmoothSA	$\underline{69.0}_{\pm 0.3}$	$\mathbf{31.7}_{\pm \mathbf{0.8}}$	$26.9_{\pm 0.6}$	$25.4_{\pm 1.3}$	$73.6_{\pm 0.6}$	$\mathbf{28.6}_{\pm \mathbf{0.1}}$	$\underline{27.5}_{\pm 0.2}$	$\mathbf{20.1}_{\pm \mathbf{0.4}}$	35.3	373
TSA (ours)	$\mathbf{75.1}_{\pm \mathbf{0.4}}$	$\underline{29.9}_{\pm 1.1}$	$\mathbf{34.9}_{\pm \mathbf{1.3}}$	$\underline{32.7}_{\pm 1.8}$	$\mathbf{83.6}_{\pm \mathbf{0.9}}$	$\underline{25.1}_{\pm 0.6}$	$\mathbf{27.8}_{\pm \mathbf{0.7}}$	$\underline{16.8}_{\pm 0.9}$	34.2	359

Table 2: Results on real-world benchmarks. Mean ± std over 3 seeds. Best in **bold**, second best underlined.

Method	YouTube-VIS HQ ($K = 7$) *(simple, up to 36 frames)*				OVIS ($K = 22$) *(complex, up to 500 frames)*			
	FG-ARI↑	mBO↑	HOTA↑	IDF1↑	FG-ARI↑	mBO↑	HOTA↑	IDF1↑
VideoSAUR	$49.2_{\pm 0.5}$	$29.9_{\pm 0.4}$	$16.9_{\pm 0.3}$	$6.3_{\pm 0.1}$	$23.4_{\pm 0.4}$	$14.1_{\pm 0.2}$	$5.8_{\pm 0.1}$	$1.4_{\pm 0.1}$
SlotContrast	$49.4_{\pm 1.1}$	$33.0_{\pm 0.2}$	$18.8_{\pm 0.2}$	$8.7_{\pm 0.5}$	$24.3_{\pm 0.6}$	$16.1_{\pm 0.6}$	$6.5_{\pm 0.4}$	$1.5_{\pm 0.1}$
RandSF.Q_{tsim}	$60.4_{\pm 2.3}$	$\underline{39.4}_{\pm 0.3}$	$\underline{23.8}_{\pm 0.4}$	$\underline{19.3}_{\pm 1.3}$	$22.5_{\pm 6.2}$	$16.2_{\pm 3.4}$	$\underline{8.1}_{\pm 1.0}$	$\underline{4.3}_{\pm 0.2}$
RandSF.Q_{ssc}	$58.0_{\pm 1.0}$	$37.6_{\pm 0.4}$	$21.6_{\pm 0.2}$	$15.1_{\pm 0.6}$	$30.4_{\pm 0.9}$	$18.6_{\pm 0.7}$	$7.6_{\pm 0.3}$	$3.0_{\pm 0.2}$
SmoothSA	$\underline{63.0}_{\pm 3.4}$	$38.9_{\pm 0.7}$	$22.7_{\pm 0.4}$	$16.2_{\pm 0.5}$	$\underline{34.4}_{\pm 2.4}$	$\underline{19.7}_{\pm 0.5}$	$8.1_{\pm 0.3}$	$3.9_{\pm 0.2}$
TSA (ours)	$\mathbf{76.3}_{\pm \mathbf{1.2}}$	$\mathbf{54.2}_{\pm \mathbf{1.2}}$	$\mathbf{44.3}_{\pm \mathbf{4.5}}$	$\mathbf{46.7}_{\pm \mathbf{5.9}}$	$\mathbf{61.2}_{\pm \mathbf{0.2}}$	$\mathbf{34.0}_{\pm \mathbf{1.4}}$	$\mathbf{36.2}_{\pm \mathbf{2.6}}$	$\mathbf{36.9}_{\pm \mathbf{2.5}}$

is a two-layer MLP and Ψ_{tce} a single-layer GRU. Full details are in Appendix D, and sensitivity to λ_{reg} and β in Appendix E.

5.2 Main results

Synthetic benchmarks. As shown in Table 1, TSA achieves the best FG-ARI and HOTA on both MOVi-C and MOVi-E, the two metrics that most directly reflect object grouping and temporal association, respectively. On MOVi-C, TSA improves FG-ARI from 69.0 to 75.1 and HOTA from 33.9 to 34.9 over the strongest competing method on each metric, while ranking second on mBO (29.9 vs. 31.7) and IDF1 (32.7 vs. 33.1). The improvement is particularly pronounced in FG-ARI, indicating that activation-aware slot updating improves foreground object decomposition. On the more crowded MOVi-E benchmark, TSA further improves FG-ARI from 82.1 to 83.6 and HOTA from 27.5 to 27.8, again ranking second on mBO and IDF1. Notably, TSA is the only method that ranks in the top two on all eight metrics across both benchmarks, whereas the baselines exhibit clear trade-offs: SmoothSA attains the best mBO but falls behind on MOVi-C tracking (26.9 HOTA, 25.4 IDF1, below even SlotContrast), and RandSF.Q_{ssc} leads MOVi-C IDF1 but drops to 14.9 IDF1 on MOVi-E. These results show that TSA delivers consistently strong performance across both grouping and tracking under synthetic multi-object dynamics, even though short clips of rigid objects offer limited opportunity for the inactive-but-persistent regime that TSA is designed for.

Real-world benchmarks. The benefits of TSA become more pronounced on real-world videos (Table 2), where occlusion, clutter, and long-term dynamics are prevalent. On YouTube-VIS HQ, TSA delivers large gains across both grouping and tracking metrics (e.g. $+27.4$ IDF1), indicating improvements not only in per-frame segmentation but also in temporal consistency. This advantage is further amplified on OVIS, a benchmark characterized by heavy occlusion and long trajectories, where TSA substantially outperforms prior methods (e.g. HOTA improves from 8.1 to 36.2). These results reinforce a key insight: unconditional slot propagation breaks down in realistic settings, whereas the ability of TSA to deactivate and preserve slots enables more reliable object discovery and identity tracking under complex, occlusion-heavy dynamics.

5.3 Further Analysis

Occlusion duration. Table 3 evaluates persistence under varying lengths of disappearance. Performance generally degrades as occlusion becomes longer, reflecting the difficulty of maintaining identity over extended gaps. However, TSA outperforms prior methods by a wide margin in every regime. Even under the most challenging long-duration occlusions ($T_\Delta > 20$), TSA achieves 23.7 HOTA and 22.0 IDF1, higher than what any prior method achieves at any occlusion length, including

Table 3: Temporal persistence under varied invisible intervals (T_Δ) on OVIS. Results from a single seed.

Method	T_Δ=0		1–10		11–20		>20	
	HOTA	IDF1	HOTA	IDF1	HOTA	IDF1	HOTA	IDF1
VideoSAUR	4.6	0.8	3.4	0.5	2.4	0.3	2.9	0.5
SlotContrast	5.1	0.7	3.8	0.7	2.9	0.4	2.8	0.3
RandSF.Q_{ssc}	6.2	1.8	4.3	1.0	3.6	0.7	2.9	0.5
SmoothSA	7.2	2.6	4.7	1.3	4.0	1.1	3.7	0.9
TSA (ours)	**36.6**	**38.9**	**30.3**	**31.6**	**27.5**	**29.8**	**23.7**	**22.0**

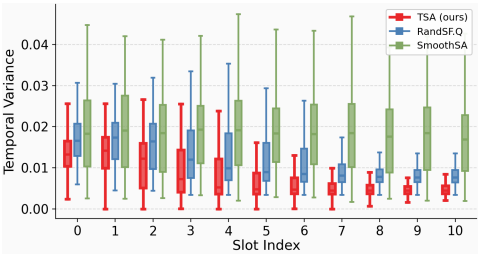

Figure 3: **Temporal variance per slot.**

objects that are never occluded ($T_\Delta = 0$). This highlights the central benefit of TSA: by allowing slots to become inactive while preserving their internal state, it maintains identity through absence rather than forcing erroneous updates. As a result, TSA achieves more robust object persistence and re-identification, especially when objects undergo prolonged occlusion or delayed reappearance.

Representation stability. Fig. 3 reports the per-slot frame-to-frame variation of slot states in MOVi-C. TSA yields consistently lower medians than both RandSF.Q and SmoothSA across all slots, indicating that its slot states evolve more smoothly over time. This confirms that the activation-gated state update reduces update-induced slot drift by anchoring inactive slots to their previous states, with lower frame-to-frame variation reflecting more stable identity preservation. Appendix G.2 provides further evidence: when an object is occluded and later reappears, the slot of TSA changes less between these two moments than that of RandSF.Q, on all four datasets.

5.4 Ablation Studies

We conduct ablation studies on the OVIS benchmark. Additional ablation studies on YouTube-VIS HQ are provided in Appendix F; further analysis and downstream-task evaluation in Appendix G.

A. Effect of activation-gated state update and decoder participation. Table 4 (left) studies the two pathways controlled by $\alpha_{k,t}$. The baseline (Exp. #1), corresponding to unconditional slot propagation, obtains 30.5 FG-ARI, 18.4 mBO, and 7.4 HOTA. Activation-gated decoding alone (Exp. #2) leaves all three metrics nearly unchanged (30.0 FG-ARI, 19.1 mBO, 7.9 HOTA), indicating that suppressing inactive slots in the decoder is insufficient when their states are overwritten by current-frame evidence. Activation-gated state update alone (Exp. #3) improves FG-ARI to 61.5, mBO to 23.3, and HOTA to 21.2, confirming that gating the state update is essential to prevent occlusion-induced slot drift. The full model (Exp. #4) attains comparable FG-ARI while further improving mBO to 34.4 and HOTA to 39.4, supporting our design that state evolution and reconstruction should be *jointly* controlled by a shared activation score.

B. Effect of regularization. Table 4 (middle) shows that reconstruction alone (Exp. #1) cannot learn meaningful slot lifecycles, as the model collapses toward unconditional propagation. $\mathcal{L}_{\text{sparse}}$ alone (Exp. #2) remains at this level, since sharpening activation decisions has little effect when no slot is encouraged to deactivate. $\mathcal{L}_{\text{usage}}$ (Exp. #3) provides the main gain, raising FG-ARI to 61.0, mBO to 31.7, and HOTA to 26.9 by discouraging unnecessary slot activation and enabling inactive-but-persistent slots. Combining both terms (Exp. #4) yields the best performance, improving HOTA further to 39.4 and suggesting complementary roles: $\mathcal{L}_{\text{usage}}$ determines *when* slots should be active, while $\mathcal{L}_{\text{sparse}}$ makes these decisions more decisive.

C. Effect of temporal memory. Table 4 (right) evaluates temporal context. Removing temporal context (Exp. #1) reduces HOTA to 25.4, showing that current-frame evidence alone is unreliable. Conditioning on the previous slot state $\boldsymbol{s}_{k,t-1}$ (Exp. #2) recovers part of the loss, reaching 60.8 FG-ARI and 33.1 HOTA. Using the accumulated memory $\boldsymbol{m}_{k,t-1}$ (Exp. #3) performs best, improving HOTA to 39.4. This indicates that activation decisions benefit from longer trajectory context, especially during gradual reappearance and prolonged partial occlusion.

5.5 Qualitative Results

Fig. 4 presents representative sequences from YouTube-VIS HQ and OVIS. SlotContrast, RandSF.Q, and SmoothSA, all employing unconditional slot propagation, exhibit state drift and identity

Table 4: Ablation studies on OVIS. *Left:* activation-gated pathways. *Middle:* regularization terms in $\mathcal{L}_{\text{reg}}$. *Right:* temporal memory. Results from a single seed. Best per column in **bold**.

Exp.	State update	Decoder participate	Metrics FG-ARI↑	mBO↑	HOTA↑
#1	✗	✗	30.5	18.4	7.4
#2	✗	✓	30.0	19.1	7.9
#3	✓	✗	**61.5**	23.3	21.2
#4	✓	✓	61.1	**34.4**	**39.4**

Exp.	Loss $\mathcal{L}_{\text{sparse}}$	Loss $\mathcal{L}_{\text{usage}}$	Metrics FG-ARI↑	mBO↑	HOTA↑
#1	✗	✗	31.6	19.5	7.9
#2	✓	✗	28.8	18.0	7.5
#3	✗	✓	61.0	31.7	26.9
#4	✓	✓	**61.1**	**34.4**	**39.4**

Exp.	Temporal memory	Metrics FG-ARI↑	mBO↑	HOTA↑
#1	✗	57.5	32.3	25.4
#2	$s_{k,t-1}$	60.8	29.3	33.1
#3	$m_{k,t-1}$	**61.1**	**34.4**	**39.4**

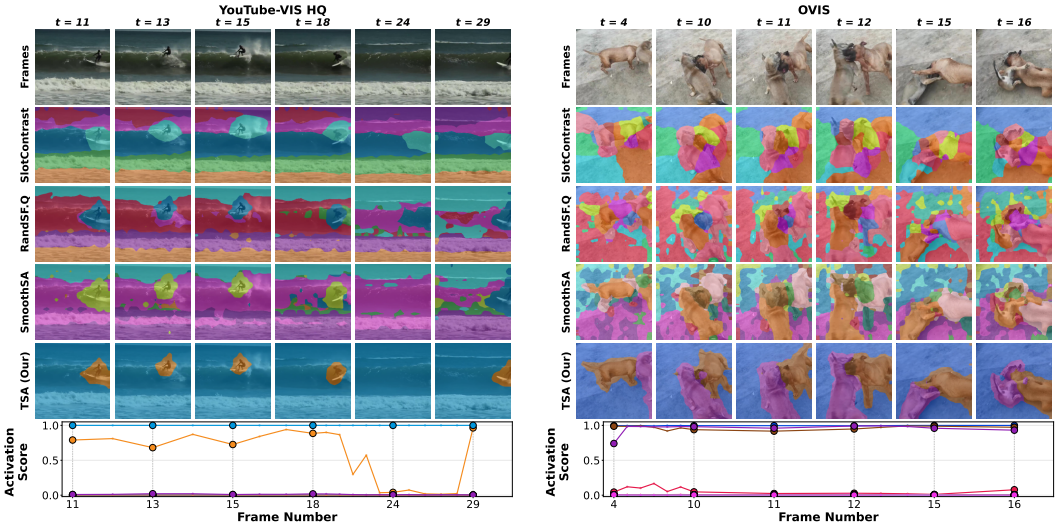

Figure 4: **Qualitative comparison on YouTube-VIS HQ and OVIS.** Colors denote slot identity. The bottom row shows the activation score of each slot predicted by our TSA across frames, with line colors matching the corresponding slot masks.

switches consistent with our analysis in Sec. 3. In contrast, TSA maintains consistent slot identity throughout each sequence. The per-slot activation curves plotted below confirm this behavior: when an object disappears, the activation score of its slot drops to near zero, then rises again upon reappearance, while slots of visible objects remain active. This is consistent with the object lifecycle defined in Sec. 3. Additional qualitative analysis and comparisons are in Appendix H.

6 Conclusion

We present Temporal Slot Activation (TSA), which shifts unsupervised video object-centric learning from continuous propagation to selective persistence. By identifying *unconditional slot propagation* as a principal cause of state drift and identity switching in recurrent VSA, TSA introduces a simple principle: a slot should update and participate in decoding only when its object is present. A single learned activation score $\alpha_{k,t}$ jointly gates slot state updates and decoder participation, while a Temporal Context Encoder conditions activation decisions on accumulated trajectory memory, enabling slots to act as stable temporal anchors that preserve object identity through long occlusions and gradual reappearance, without any visibility supervision. TSA remains competitive on the synthetic MOVi-C and MOVi-E benchmarks and delivers substantial gains on the real-world YouTube-VIS HQ and OVIS, with the largest improvements on heavily occluded sequences, offering a lightweight and effective approach to modeling object permanence in unsupervised video learning.

Limitations and future work. TSA uses a fixed slot budget K; combining activation with scene-adaptive slot allocation remains an open direction. For fair comparison, TSA also builds on a frozen DINOv2 backbone, whose rich features underpin strong performance, though incorporating modalities such as optical flow or depth could further sharpen slot boundaries in cluttered scenes. Finally, while activation gating effectively preserves slot identity through occlusion, gradual appearance changes arising from deformation or scale variation over long sequences remain an open challenge for future work.

Reproducibility Statement

The TSA architecture and training objectives are specified in Sec. 4. Appendix D and Table 7 provide full implementation details and all hyperparameters for each benchmark. All datasets are publicly available (Appendix B), and evaluation metrics are defined in Appendix C, with HOTA and IDF1 computed using the official TrackEval implementation (Luiten & Hoffhues, 2020). Main results are averaged over 3 random seeds. The source code, trained models, and training logs will be publicly released upon publication.

Ethics Statement

All experiments in this work are conducted on publicly available data. There are no ethical concerns associated with this work.

AI Use Statement

We used LLMs only as assistive tools for language editing and for checking the readability and consistency of the paper. All content was written by us, and we reviewed every suggested edit. All research ideas, methods, experiments, analyses, and conclusions are the authors' own, and the authors reviewed and take full responsibility for all content of this paper.

References

Görkay Aydemir, Weidi Xie, and Fatma Guney. Self-supervised object-centric learning for videos. *Advances in Neural Information Processing Systems*, 36:32879–32899, 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.

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

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019.

SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in neural information processing systems*, 29, 2016.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23062–23071, 2024.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International conference on machine learning*, pp. 2424–2433. PMLR, 2019.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3749–3761, 2022.

Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. *arXiv preprint arXiv:2303.10834*, 2023.

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

Ioannis Kakogeorgiou, Spyros Gidaris, Konstantinos Karantzalos, and Nikos Komodakis. Spot: Self-training with patch-order permutation for object-centric learning with autoregressive transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22776–22786, 2024.

Lei Ke, Henghui Ding, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Video mask transfiner for high-quality video instance segmentation. In *European Conference on Computer Vision*, pp. 731–747. Springer, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 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. *arXiv preprint arXiv:2111.12594*, 2021.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. *arXiv preprint arXiv:2001.02407*, 2020.

Hongjia Liu, Rongzhen Zhao, Haohan Chen, and Joni Pajarinen. Metaslot: Break through the fixed number of slots in object-centric learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

Jonathon Luiten and Arne Hoffhues. Trackeval. `https://github.com/JonathonLuiten/TrackEval`, 2020.

Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129(2):548–578, 2021.

Anna Manasyan, Maximilian Seitzer, Filip Radovic, Georg Martius, and Andrii Zadaianchuk. Temporally consistent object-centric learning by contrasting slots. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 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. *arXiv preprint arXiv:2603.22758*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *International Journal of Computer Vision*, 130(8):2022–2039, 2022.

Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, pp. 17–35. Springer, 2016.

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

Hyun Seok Seong, WonJun Moon, and Jae-Pil Heo. From vicious to virtuous cycles: Synergistic representation learning for unsupervised video object-centric learning. *arXiv preprint arXiv:2602.03390*, 2026.

Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. *Advances in neural information processing systems*, 35:18181–18196, 2022.

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

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

Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. *Advances in Neural Information Processing Systems*, 36:50932–50958, 2023.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. *Advances in neural information processing systems*, 36:61514–61545, 2023.

Rongzhen Zhao, Wenyan Yang, Juho Kannala, and Joni Pajarinen. Smoothing slot attention iterations and recurrences. *arXiv preprint arXiv:2508.05417*, 2025.

Rongzhen Zhao, Jian Li, Juho Kannala, and Joni Pajarinen. Predicting video slot attention queries from random slot-feature pairs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13208–13216, 2026.

APPENDIX

A Conceptual comparison of slot-control mechanisms **14**

B Dataset details **14**

C Evaluation metrics **15**

D Implementation details **16**

E Sensitivity to regularization hyperparameters **17**

F Additional ablation study on YouTube-VIS HQ **17**

G Additional analysis and downstream task evaluation **19**

G.1 Activation distribution under each regularization term 19

G.2 Representation drift across occlusion intervals 19

G.3 Downstream task evaluation 20

H Additional qualitative results **22**

H.1 Additional comparisons with prior methods 22

H.2 Ablation visualizations 23

Table 5: **Conceptual comparison of slot-control mechanisms.** Methods: SAVi (Kipf et al., 2021), VideoSAUR (Zadaianchuk et al., 2023), SlotContrast (Manasyan et al., 2025), RandSF.Q (Zhao et al., 2026), SmoothSA (Zhao et al., 2025), SOLV (Aydemir et al., 2023), AdaSlot (Fan et al., 2024), and MetaSlot (Liu et al., 2025). Pathway I/II: whether an absent object's slot is protected from state updates / removed from decoding (Sec. 3). ✓ yes; ✗ no; – not applicable. †Image-level module; evaluated on video only by plugging it into VideoSAUR.

		Participation decision		Gated pathways			Slot lifecycle		
Method	**Video**	Per-slot, per-frame	Conditioned on	Pathway I (state)	Pathway II (decoder)	Shared control	Persistent when inactive	Reversible	Slot under full occlusion
Slot propagation									
SAVi	✓	✗	–	✗	✗	✗	✗	–	Updated & decoded
VideoSAUR	✓	✗	–	✗	✗	✗	✗	–	Updated & decoded
SlotContrast	✓	✗	–	✗	✗	✗	✗	–	Updated & decoded
RandSF.Q	✓	✗	–	✗	✗	✗	✗	–	Updated & decoded
SmoothSA	✓	✗	–	✗	✗	✗	✗	–	Updated & decoded
Slot pruning / merging									
SOLV	✓	✓	Inter-slot similarity	✗	✗	✗	✗	✗	Merged
AdaSlot	✗	✓	Frame content	–	✓	–	✗	–	Dropped
MetaSlot	✗†	✓	Frame content	–	✓	–	✗	–	Dropped
Slot lifecycle									
TSA (ours)	✓	✓	**Candidate + memory**	✓	✓	✓	✓	✓	**Frozen & silenced**

A Conceptual comparison of slot-control mechanisms

Sec. 1 characterizes recurrent VSA methods by unconditional slot propagation. A separate line of work does assess slot utility and remove slots judged unnecessary, so we situate TSA among all mechanisms that decide whether a slot participates in a frame. Table 5 classifies them by whether a per-slot, per-frame participation decision is made and the evidence it is conditioned on, which of the two failure pathways of Sec. 3 it gates, and the behavior of a slot whose object is fully occluded.

Occlusion requires separating object persistence from slot participation: the slot should stop explaining the current evidence, yet remain available as a query when the object returns. In propagation methods (SAVi (Kipf et al., 2021), VideoSAUR (Zadaianchuk et al., 2023), SlotContrast (Manasyan et al., 2025), RandSF.Q (Zhao et al., 2026), SmoothSA (Zhao et al., 2025)) the question never arises, since participation is fixed by construction and every slot competes for the frame whether or not its object is there, leaving both pathways open. Merging and pruning methods do raise it, but resolve it from evidence that occlusion makes ambiguous. SOLV (Aydemir et al., 2023) judges a slot by its similarity to other slots, so a slot that has drifted onto the occluder is merged into the slot already explaining it. AdaSlot (Fan et al., 2024) learns which slots to keep for the current frame, and MetaSlot (Liu et al., 2025) prunes slots that collapse onto the same learned prototype; both judge a slot by what the current frame supports, a criterion an absent object and a hidden one satisfy equally poorly. The verdict is also final, with the slot absorbed into another, or withheld from the decoder with no record of the object instance it held and no criterion for restoring it.

The two failures are opposite and leave a lifecycle gap: a slot is either kept fully active or released, with nothing in between. TSA fills this gap by conditioning the decision on the slot candidate, obtained by querying the current frame with the propagated slot state, together with the slot's temporal memory, rather than on frame content alone, and by making the decision reversible. TSA may also deactivate slots that are simply not needed; what distinguishes it is what deactivation means. A slot with $\alpha_{k,t} \to 0$ is anchored to its previous state and suppressed in the decoder, so it neither absorbs the occluder nor explains unrelated content, while its representation stays available for reactivation. This inactive-but-persistent regime is the one prior work lacks.

B Dataset details

We evaluate our approach on four complementary video benchmarks that span synthetic and real-world domains, ranging from controlled multi-object scenes to crowded videos with severe occlusion. Table 6 summarizes the key characteristics of each benchmark, while Fig. 5 shows qualitative examples illustrating the visual diversity and difficulty of each dataset.

Table 6: Comparison of the four video benchmarks used in our evaluation.

Dataset	Domain	Object type	Main challenge
MOVi-C	Synthetic	Rigid	Moderate clutter
MOVi-E	Synthetic	Rigid	Dense scenes, camera motion
YouTube-VIS HQ	Real	Non-rigid	Natural appearance variation
OVIS	Real	Non-rigid	Severe occlusion, long trajectories

MOVi-C and MOVi-E. MOVi-C and MOVi-E (Greff et al., 2022) are synthetic multi-object video benchmarks generated with the Kubric simulator. They serve as controlled settings for evaluating object-centric video grouping under known object dynamics. Both datasets consist of rigid objects with stable appearance, but differ substantially in scene complexity: MOVi-C features moderately cluttered scenes on textured backgrounds, whereas MOVi-E exhibits denser object layouts, stronger camera motion, and more frequent inter-object occlusions.

YouTube-VIS HQ. YouTube-VIS HQ (Ke et al., 2022) is a real-world video instance segmentation benchmark with high-quality, manually refined object mask annotations. It contains natural videos featuring non-rigid objects with substantial appearance variation, diverse motion patterns, and cluttered backgrounds, making it well suited for evaluating grouping performance in unconstrained settings.

OVIS. OVIS (Qi et al., 2022) is a real-world video instance segmentation benchmark explicitly designed around heavy occlusion. It contains crowded scenes with non-rigid objects, long object trajectories, and frequent partial or full visibility changes, providing a rigorous stress test for temporal persistence and object re-identification under severe occlusion.

C Evaluation metrics

Foreground adjusted Rand index (FG-ARI). ARI measures the agreement between two clustering assignments over a set of elements, corrected for chance agreement. Following the standard object-centric protocol (Locatello et al., 2020; Kipf et al., 2021), we restrict the computation to ground-truth foreground pixels by excluding the background label, which isolates the signal of the metric to how well distinct objects are separated from one another rather than from the scene background. We compute FG-ARI at the video level by treating all pixels across frames of a sequence as a single clustering problem, so that the metric reflects not only segmentation quality within frames but also identity consistency of slot assignments over time.

Mean best overlap (mBO). mBO quantifies per-object mask coverage by selecting, for each ground-truth instance, the predicted slot mask with the highest intersection-over-union (IoU), and averaging these best-match IoUs across all instances and sequences. Unlike FG-ARI, mBO retains background pixels in the IoU computation, which makes it more sensitive to mask boundary precision and to spurious slot activations on non-object regions.

Higher order tracking accuracy (HOTA). HOTA (Luiten et al., 2021) jointly measures detection accuracy and association quality, avoiding the bias toward either detection or tracking that arises in single-metric evaluations. We compute HOTA with the official TrackEval implementation (Luiten & Hoffhues, 2020), treating each slot index as a tracker identity, without access to identity labels at training time.

IDF1. IDF1 (Ristani et al., 2016) measures the ratio of correctly identified detections over the mean of ground-truth and computed detections, using identity-consistent matching. Each predicted slot is matched to a ground-truth instance by a single sequence-level assignment, following the official TrackEval implementation.



Figure 5: **Qualitative samples from the four benchmarks.** Each row shows four representative videos from a single dataset, illustrating the visual diversity within each benchmark. MOVi-C and MOVi-E provide controlled synthetic scenes with known dynamics; YouTube-VIS HQ contributes natural appearance variation and object motion; OVIS contains crowded scenes with severe occlusion and long object trajectories.

Seeding and averaging. Results on the main benchmarks and the downstream evaluations are reported as mean $\pm$ std over 3 independent random seeds controlling model initialization and data ordering. Ablation and sensitivity studies are run with a single seed, shared across all conditions, so that the conditions are compared under the same random seed and data ordering.

D Implementation Details

Model. We use a frozen DINOv2 ViT-S/14 (Oquab et al., 2023) as the visual encoder. Each frame is resized from 256×256 to 224×224, producing $N = 256$ patch tokens with feature dimension $d_f = 384$. The patch features are projected by a two-layer MLP before being used as keys and values in Slot Attention. Slot Attention uses K slots (dataset-specific, see Table 7) with slot dimension $d = 256$, and runs for 3 iterations on the first frame of each clip and 1 iteration on subsequent frames (Zhao et al., 2025). The temporal query transitioner T_ϕ follows the RandSF.Q transition module (Zhao et al., 2026), instantiated as a single Transformer decoder layer with 4 attention heads. The Slot Activation Estimator Φ_{act} is a two-layer MLP with ReLU activations that maps the concatenation of the current slot candidate ($d = 256$) and the previous temporal memory $\boldsymbol{m}_{k,t-1}$ ($d_h = 64$) to a scalar activation logit. The Temporal Context Encoder Ψ_{tce} is a

single-layer GRU shared across slots, with hidden dimension $d_h = 64$, that produces $\boldsymbol{m}_{k,t}$ from the post-gate slot state at each step and is reset at the start of each video. The temporal memory $\boldsymbol{m}_{k,t}$ is consumed only by Φ_{act} and does not feed back into the slot representation directly. The decoder is an autoregressive Transformer with model dimension $d_f = 384$ that reconstructs DINOv2 features rather than RGB pixels (Zadaianchuk et al., 2023; Manasyan et al., 2025). The activation log-bias from Eq. 8 is added to the decoder cross-attention logits at each layer.

Training. We train the model with Adam (Kingma & Ba, 2014) for $50{,}000$ steps using a batch size of 8 video clips. Each training sample is a contiguous segment of length T frames (dataset-specific; see Table 7). The learning rate is initialized to 5×10^{-5}, linearly warmed up over the first $2{,}500$ steps, and the gradient norm is clipped at 0.05. The total loss is $\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{ssc}}\mathcal{L}_{\text{ssc}} + \lambda_{\text{reg}}(t)\,\mathcal{L}_{\text{reg}}$, where $\lambda_{\text{ssc}} = 0.5$ throughout and the regularization weight $\lambda_{\text{reg}}(t)$ follows a two-stage schedule: it is held at zero for the first T_{warmup} steps, then linearly increased to its target value over the next T_{ramp} steps.

Inference. At test time, slot activations $\alpha_{k,t} \in (0, 1)$ remain continuous and are not thresholded; the same activation-gated state update and activation log-bias used during training are applied unchanged. Full videos are processed sequentially, with slot states and per-slot temporal memories $\boldsymbol{m}_{k,t}$ propagated across frames without resetting.

Hardware. All experiments are run on a single NVIDIA RTX A6000 GPU.

Table 7 summarizes the dataset-specific implementation details, including the number of slots K, training segment length T, loss coefficients, and regularization schedules.

E Sensitivity to regularization hyperparameters

All components inherited from prior work are kept at their published settings (Table 7). TSA introduces two new loss weights, both in $\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{usage}} + \beta\,\mathcal{L}_{\text{sparse}}$ of Eq. 10: λ_{reg}, which sets how strongly the regularizer competes with reconstruction, and β, which weights $\mathcal{L}_{\text{sparse}}$ against $\mathcal{L}_{\text{usage}}$; the warm-up schedule of λ_{reg} is shared across benchmarks. We sweep one weight at a time on YouTube-VIS HQ, keeping the other at its paper value for this benchmark ($\lambda_{\text{reg}} = 0.10$, $\beta = 1.5$; Table 7).

Effect of λ_{reg}. Table 8 (left) shows that, on YouTube-VIS HQ, performance is best at intermediate values and degrades at both ends of the range. A weak regularizer exerts less pressure to deactivate slots, while a strong one increasingly outweighs the reconstruction objective. On this benchmark, performance is stable between 0.10 and 0.15, where every metric stays within 2.3 points.

Effect of β. Table 8 (right) shows that, on YouTube-VIS HQ, performance drops mainly when β is too small, where intermediate activations are penalized less. Performance improves as β increases, and for $\beta \geq 1.5$ every metric stays within 1.8 points of the paper setting.

These ranges are specific to YouTube-VIS HQ: the appropriate values depend on the benchmark, and the settings used for the other benchmarks differ (Table 7).

F Additional ablation study on YouTube-VIS HQ

A. Activation-gated pathways. Table 9 (left): gating the decoder alone (Exp. #2) leaves the metrics nearly unchanged over the baseline, gating the state update alone (Exp. #3) gives a large improvement, and using both (Exp. #4) is best on all three metrics. Decoder gating contributes most when combined with state gating, adding $+9.6$ HOTA over Exp. #3.

B. Regularization. Table 9 (middle): $\mathcal{L}_{\text{sparse}}$ alone (Exp. #2) stays close to reconstruction alone (Exp. #1), $\mathcal{L}_{\text{usage}}$ (Exp. #3) gives a large gain, and using both (Exp. #4) is best on all three metrics, adding a further $+10.0$ HOTA over Exp. #3.

C. Temporal memory. Table 9 (right): the accumulated memory $\boldsymbol{m}_{k,t-1}$ (Exp. #3) is best on mBO and HOTA, followed by the previous slot state $\boldsymbol{s}_{k,t-1}$ (Exp. #2) and no temporal context (Exp. #1);

Table 7: Implementation details across the four benchmarks. Shared hyperparameters are listed once across all columns; dataset-specific values are given per benchmark.

Hyperparameter	**Benchmarks**			
	MOVi-C	**MOVi-E**	**YouTube-VIS HQ**	**OVIS**
Optimization				
Optimizer	Adam			
Training steps	50,000			
Batch size (clips)	8			
Training segment length T	6	6	5	10
Initial learning rate	5×10^{-5}			
LR warm-up steps	2,500			
Gradient norm clip	0.05			
Visual encoder (frozen)				
Backbone	DINOv2 ViT-S/14			
Input resolution	$256 \rightarrow 224$			
Number of image tokens N	256			
Feature dimension d_f	384			
Slot Attention				
Number of slots K	11	24	7	22
Slot dimension d	256			
Key / value dimension	384			
FFN dimension	1,024			
Iterations (first / subsequent frame)	3 / 1			
Temporal Query Transitioner T_ϕ				
Type	Transformer decoder layer			
Heads / FFN dimension	4 / 1,024			
Dropout	0.5			
Slot Activation Estimator Φ_{act}				
Type	Two-layer MLP (ReLU)			
Input dim $(d + d_h)$ / hidden dim	320 / 128			
Temporal Context Encoder Ψ_{tce}				
Type	Single-layer GRU			
Input dim d / hidden dim d_h	256 / 64			
Decoder				
Type	Autoregressive Transformer decoder			
Layers / heads / FFN dim	4 / 4 / 1,536			
Model dimension d_f	384			
Reconstruction target	DINOv2 patch features			
Loss coefficients and schedule				
λ_{ssc}	0.5			
λ_{reg}	0.06	0.02	0.10	0.15
β	1	3	1.5	1
T_{warmup} (steps)	1,000			
T_{ramp} (steps)	6,000			

the three are within 1.3 FG-ARI of each other, with Exp. #2 marginally ahead. The gap between the three is mostly in HOTA (4.2, against 1.3 FG-ARI and 3.5 mBO), since YouTube-VIS HQ has shorter sequences and fewer long occlusions than OVIS.

Table 8: Sensitivity to the two regularization hyperparameters on YouTube-VIS HQ. *Left:* regularization strength, varying λ_{reg} with $\beta = 1.5$. *Right:* sparsity weight, varying β with $\lambda_{\text{reg}} = 0.10$. Best per column in **bold**; the setting used in the paper is marked †. Results from a single seed.

λ_{reg}	FG-ARI↑	mBO↑	HOTA↑	IDF1↑
0.05	72.3	49.3	37.3	38.4
$0.10^{\dagger}$	**76.3**	55.8	46.3	**49.2**
0.15	74.0	**56.8**	**46.5**	48.9
0.20	73.6	52.7	40.9	40.8
0.25	68.4	47.4	33.1	30.5

β	FG-ARI↑	mBO↑	HOTA↑	IDF1↑
0.5	74.2	51.5	39.5	40.7
1.0	75.3	53.7	42.3	44.6
$1.5^{\dagger}$	**76.3**	55.8	46.3	49.2
2.0	75.7	**57.6**	**48.0**	**50.9**
2.5	75.8	55.3	45.7	48.6

Table 9: Additional ablation studies on YouTube-VIS HQ. *Left:* activation-gated pathways. *Middle:* regularization terms in $\mathcal{L}_{\text{reg}}$. *Right:* temporal memory. Results from a single seed. Best per column in **bold**.

Exp.	State update	Decoder participate	Metrics FG-ARI↑	mBO↑	HOTA↑
#1	✗	✗	57.1	38.0	21.8
#2	✗	✓	58.8	37.7	21.9
#3	✓	✗	74.8	51.9	36.7
#4	✓	✓	**76.3**	**55.8**	**46.3**

Exp.	Loss $\mathcal{L}_{\text{sparse}}$	Loss $\mathcal{L}_{\text{usage}}$	Metrics FG-ARI↑	mBO↑	HOTA↑
#1	✗	✗	58.6	41.3	24.0
#2	✓	✗	57.6	38.8	22.6
#3	✗	✓	72.2	50.5	36.3
#4	✓	✓	**76.3**	**55.8**	**46.3**

Exp.	Temporal memory	Metrics FG-ARI↑	mBO↑	HOTA↑
#1	✗	76.2	52.3	42.1
#2	$s_{k,t-1}$	**77.5**	54.4	45.1
#3	$m_{k,t-1}$	76.3	**55.8**	**46.3**

G Additional analysis and downstream task evaluation

G.1 Activation distribution under each regularization term

Table 9 (middle) reports what each term of $\mathcal{L}_{\text{reg}}$ contributes in metrics. Here we look at what it does to the activation score itself. For the same four settings, we collect all per-slot, per-frame activations $\alpha_{k,t}$ on YouTube-VIS HQ and bin them into ten intervals (Fig. 6).

Without $\mathcal{L}_{\text{reg}}$, nearly all activations lie above 0.6 and about 80% above 0.8: every slot stays active at every frame, which is the full-activation collapse described in Sec. 4.5 and leaves TSA behaving like unconditional propagation. Adding $\mathcal{L}_{\text{sparse}}$ alone does not break this collapse but sharpens it, moving nearly all activations into the last bin. This is expected, since $\mathcal{L}_{\text{sparse}}$ only pushes activations away from intermediate values and never gives a slot a reason to deactivate, which is why it stays at the level of the unregularized model.

$\mathcal{L}_{\text{usage}}$ alone does move a large fraction of activations below 0.1, so slots now genuinely deactivate. The remaining activations, however, are spread almost evenly over the intermediate bins. A slot in this range is neither properly frozen nor properly silenced: it is partially overwritten by the current frame while still contributing to decoding, so it behaves as a mixture of an active and an inactive slot rather than committing to either. This is the ambiguous-gating regime discussed in Sec. 4.5.

With both terms, the distribution becomes clearly bimodal: the intermediate bins are largely emptied and the activations concentrate at the two ends. Each slot is thus either kept inactive but persistent, or allowed to track and reconstruct normally, with few frames left in between. The two terms are complementary in exactly the way the ablation suggests: $\mathcal{L}_{\text{usage}}$ makes deactivation happen at all, while $\mathcal{L}_{\text{sparse}}$ makes each decision decisive. Because the learned activations are already close to binary, they can be used continuously at inference without any thresholding.

G.2 Representation drift across occlusion intervals

To directly verify that the activation-gated state update mitigates the update-induced state drift discussed in Sec. 3, we measure the *representation drift* of a slot across each occlusion interval as the per-dimension mean squared difference between its states:

$$d_{\text{drift}}(k) = \frac{1}{d} \left\| \boldsymbol{s}_{k,t_{\text{post}}} - \boldsymbol{s}_{k,t_{\text{pre}}} \right\|^2, \tag{12}$$

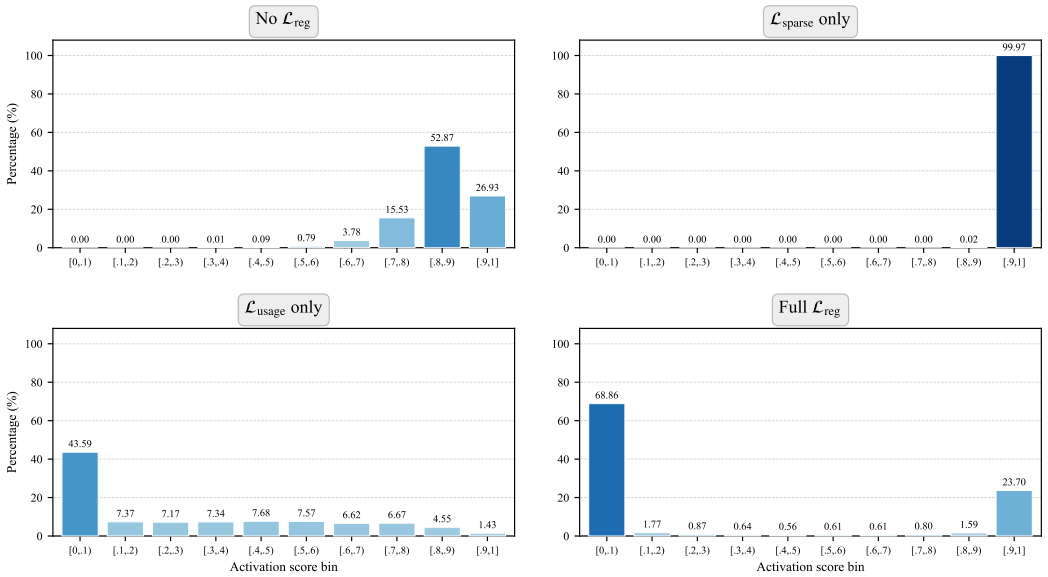

Figure 6: **Distribution of activation scores $\alpha_{k,t}$ on YouTube-VIS HQ** under the four regularization settings of Table 9 (middle). Each panel is a 10-bin histogram over $[0, 1]$, reported as % of all slot–frame pairs.

where t_{pre} denotes the last visible frame before an object becomes fully occluded, and t_{post} denotes the first visible frame after the object reappears. For each object that becomes fully occluded and later reappears, we identify the predicted slot tracking it by selecting the slot with the highest mask IoU at t_{pre}. We then compute d_{drift} between the pre-occlusion and post-reappearance states of that slot. An object is absent in any frame where it has no ground-truth mask, and one occlusion interval is a run of absent frames between two frames where the object is annotated.

We compare RandSF.Q, an unconditional propagation baseline that updates every slot at every step, against TSA. Fig. 7 shows the distribution of d_{drift} stratified by occlusion duration on MOVi-C, MOVi-E, YouTube-VIS HQ, and OVIS.

Two observations stand out. First, representation drift generally increases with occlusion duration, indicating that longer absences make it more difficult to preserve the pre-occlusion slot identity. Second, TSA consistently produces lower drift than RandSF.Q across all datasets and duration bins, with the gap widening at long durations. These results provide direct evidence that the activation-gated state update of Eq. 7 preserves slot identity more effectively across occlusions by anchoring inactive slots to their previous states, consistent with the cumulative-drift analysis in Eq. 3.

G.3 Downstream task evaluation

To further assess the quality of the slot representations learned by TSA, we evaluate them on two downstream tasks on YouTube-VIS HQ. Both tasks operate on *frozen* slot representations, isolating the contribution of the representation itself from any task-specific finetuning. We compare TSA against SlotContrast (Manasyan et al., 2025), RandSF.Q (Zhao et al., 2026), and SmoothSA (Zhao et al., 2025) under identical training and evaluation protocols. The two tasks probe complementary properties of the representation: object recognition is a per-frame appearance test, whereas dynamics prediction is a cross-frame temporal-stability test.

Object recognition. Following RandSF.Q (Zhao et al., 2026), we freeze the object-centric model and train a two-layer MLP to predict the object class and bounding box corresponding to each slot, supervised by the object class labels and bounding box annotations in the dataset. Each predicted slot is matched to a ground-truth instance using a first-visible-frame majority-overlap rule. This task probes whether slot representations preserve discriminative per-frame object information, includ-

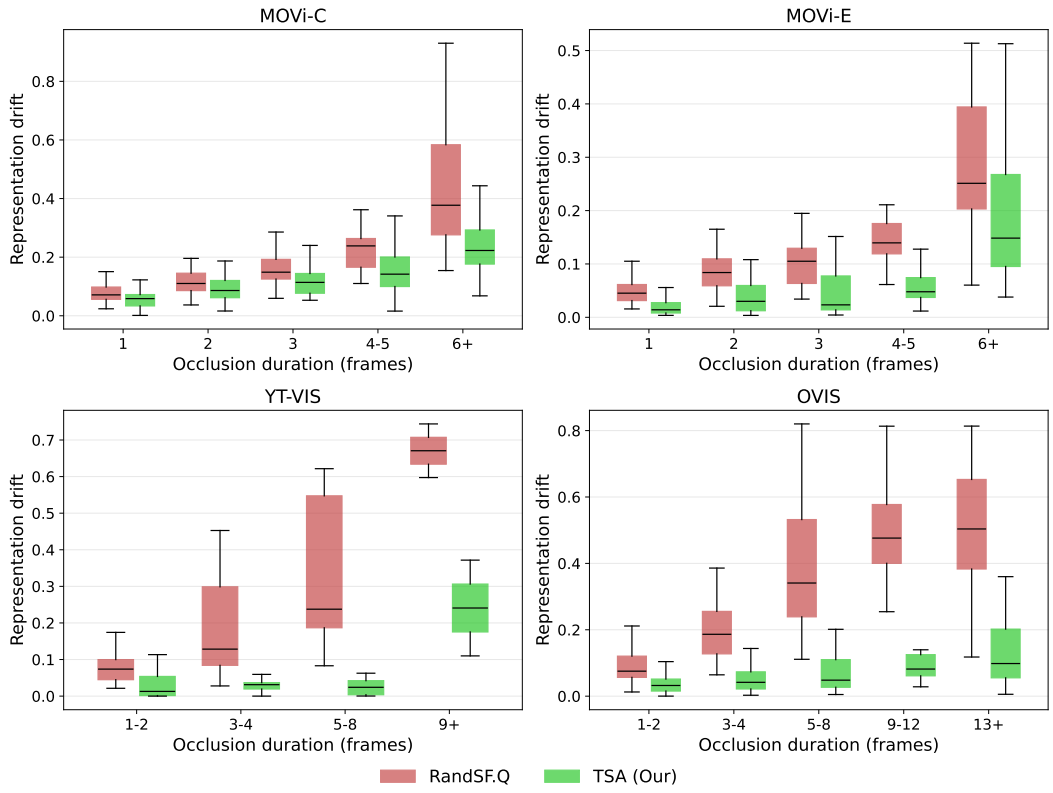

Figure 7: **Representation drift across occlusion intervals.** Box plots show the distribution of the per-dimension representation drift d_{drift} across occlusion-duration bins on MOVi-C, MOVi-E, YouTube-VIS HQ, and OVIS.

Table 10: Object recognition on YouTube-VIS HQ. A two-layer MLP is trained on frozen slot representations. Mean ± std over 3 seeds.

Method	**Top-1↑**	**Top-3↑**	**Bbox IoU↑**	**Matched**
SlotContrast + MLP	$85.8_{\pm 0.3}$	$95.8_{\pm 0.4}$	$\mathbf{51.5_{\pm 0.5}}$	$9249_{\pm 41}$
RandSF.Q + MLP	$90.5_{\pm 0.3}$	$\mathbf{97.9_{\pm 0.3}}$	$50.6_{\pm 0.4}$	$8979_{\pm 123}$
SmoothSA + MLP	$90.4_{\pm 0.2}$	$97.6_{\pm 0.1}$	$42.6_{\pm 1.4}$	$8957_{\pm 34}$
TSA (ours) + MLP	$\mathbf{91.3_{\pm 0.9}}$	$\mathbf{97.9_{\pm 0.3}}$	$\mathbf{51.5_{\pm 0.2}}$	$7908_{\pm 36}$

ing semantic category and spatial localization. We report Top-1 and Top-3 classification accuracy, bounding-box IoU, and the number of matched samples.

Object dynamics prediction. Following SlotContrast (Manasyan et al., 2025), we train SlotFormer (Wu et al., 2022) on top of the frozen slot representations to predict object dynamics. SlotFormer receives 10 burn-in frames of inferred slots and autoregressively predicts slots for 5 rollout steps. Both the object-centric model and SlotFormer operate entirely in feature space, and SlotFormer is trained using only the slot reconstruction loss. Unlike object recognition, which is evaluated on matched per-frame slots, dynamics prediction depends strongly on whether slot identities and trajectories remain stable over time. We therefore use this task to assess the temporal consistency and predictability of the learned slot representations. We report FG-ARI and mBO on the predicted slot rollouts.

Table 11: Object dynamics prediction on YouTube-VIS HQ. Mean ± std over 3 seeds.

Method	FG-ARI↑	mBO↑
SlotContrast	$29.5_{\pm 0.2}$	$33.2_{\pm 0.1}$
RandSF.Q	$38.2_{\pm 0.5}$	$43.7_{\pm 0.6}$
SmoothSA	$44.8_{\pm 1.6}$	$43.4_{\pm 0.7}$
TSA (ours)	$\mathbf{45.6_{\pm 3.8}}$	$\mathbf{46.1_{\pm 0.6}}$

Discussion. Tables 10 and 11 show that TSA preserves discriminative object information while improving temporal predictability. For object recognition (Table 10), TSA achieves the best classification performance, improving Top-1 accuracy to 91.3, compared with 90.5 for RandSF.Q, 90.4 for SmoothSA, and 85.8 for SlotContrast. It matches RandSF.Q on Top-3 accuracy (97.9 for both) and remains above SmoothSA (97.6) and clearly above SlotContrast (95.8). Since recognition is a per-frame appearance test, these results indicate that the activation mechanism does not trade away per-frame semantic content in exchange for temporal consistency. The bounding-box IoU of TSA (51.5) is on par with the best baseline (51.5 for SlotContrast) and higher than RandSF.Q (50.6) and SmoothSA (42.6). TSA matches fewer ground-truth instances than the baselines (7908 versus 8957, 8979, and 9249), reflecting that fewer slots participate in matching.

The advantage of TSA is more pronounced in object dynamics prediction (Table 11). TSA achieves 45.6 FG-ARI and 46.1 mBO, outperforming SmoothSA by $+0.8$ FG-ARI and $+2.7$ mBO, RandSF.Q by $+7.4$ FG-ARI and $+2.4$ mBO, and SlotContrast by $+16.1$ FG-ARI and $+12.9$ mBO. Since SlotFormer is trained on top of frozen slot representations, these gains reflect the quality of the learned slot trajectories rather than changes in the downstream predictor. The large improvement in rollout FG-ARI over RandSF.Q and SlotContrast indicates that TSA produces slots with more stable object correspondence across time, making future slot states easier to predict. Together, the recognition and dynamics results show that TSA retains strong per-frame object information while providing more temporally consistent representations for prediction, and that these representations transfer effectively to tasks beyond the primary object-centric segmentation setting.

H Additional qualitative results

H.1 Additional comparisons with prior methods

We compare TSA against RandSF.Q (Zhao et al., 2026), SlotContrast (Manasyan et al., 2025), and SmoothSA (Zhao et al., 2025) on YouTube-VIS HQ (Fig. 8), OVIS (Fig. 9), and MOVi-C/E (Fig. 10). Figs. 9 and 10 each isolate one of the behaviors predicted in Sec. 3, and Fig. 8 examines whether the same stability holds without occlusion.

Slot identity is preserved through occlusion and absence. In the two-animal sequence of Fig. 9 (t=24–52), the second animal is first occluded by the leading animal, then hidden by foliage, and finally leaves the field of view between t=34 and t=40. TSA maintains the two animals on two separate slots throughout the sequence, and the same slot reacquires the animal upon its reappearance at t=42. RandSF.Q, SlotContrast, and SmoothSA instead fragment each animal into several slots whose assignment changes from frame to frame. The behavior of TSA is the one predicted by Eq. 7: when the current frame provides no evidence for the object, its slot state is preserved rather than overwritten, and therefore remains a valid query at reappearance.

A slot does not come to explain content other than its own object. Fig. 10 (top, MOVi-C, t=4–22) illustrates the complementary behavior. TSA holds the small object at the bottom of the scene on the same teal slot across the entire sequence, and the thin rod on the same dark purple slot for as long as it remains visible, whereas in the baselines the slots previously bound to these objects spread onto unrelated regions. The same failure is visible at scene level on MOVi-E (t=1–11), where SlotContrast assigns the background together with several objects to a single slot, while TSA separates the individual objects and retains one background slot. Silencing a slot in the decoder (Eq. 8) is what prevents it from accounting for content it does not represent.

The same stability is observed in the absence of occlusion. Fig. 8 shows two sequences in which the objects remain visible throughout. In the sequence at the top (t=3–15), a person, a dog and a small yellow toy move in close contact at very different scales: TSA assigns one slot to each and a single slot to the background across all frames, whereas SlotContrast splits the person into several part-level slots. RandSF.Q and SmoothSA, in turn, fail to separate the toy as its own object and leave the background scattered with noisy slot assignments that change from frame to frame. The deer sequence (bottom) shows the same contrast on a single object: SlotContrast, RandSF.Q, and SmoothSA each split the deer into several slots, while TSA keeps it on one. Activation gating is therefore not confined to occlusion-heavy settings: it also keeps objects whole and the scene partition steady when every object stays in view.

H.2 Ablation visualizations

Fig. 11–13 provide qualitative evidence for the design choices studied quantitatively in Sec. 5.4. All examples are drawn from YouTube-VIS HQ, using the ablation models of Appendix F.

Effect of activation-gated state update and decoder participation. Fig. 11 visualizes the three gated configurations on a deer sequence. With decoder gating alone, the antlers are split across several part-level slots and the background is repartitioned from frame to frame: suppressing a slot in the decoder does not stop its state from being overwritten by current-frame evidence, so each slot no longer stays on the same object. Gating the state update alone keeps the antlers on a single slot across the whole sequence, confirming that anchoring the state is what stabilizes identity, but the background is still scattered with spurious slot activations. Gating both pathways holds the antlers on one slot and the background on one slot throughout, giving the cleanest decomposition and supporting the design that state evolution and reconstruction should be jointly controlled by a shared activation score.

Effect of regularization terms. Fig. 12 illustrates the complementary roles of $\mathcal{L}_{\text{usage}}$ and $\mathcal{L}_{\text{sparse}}$. Without either term, the scene is divided into many slots and the object is not held consistently across frames. With $\mathcal{L}_{\text{usage}}$ alone, slot assignments become temporally consistent across frames; however, the pressure to reduce the active slot count can lead to over-compression, where a single slot absorbs multiple objects. With $\mathcal{L}_{\text{sparse}}$ alone, activations are sharpened toward binary decisions but redundant slots remain active, so the model behaves similarly to unconditional propagation. Combining both losses balances these effects: $\mathcal{L}_{\text{usage}}$ enforces compact slot usage with stable temporal correspondence, while $\mathcal{L}_{\text{sparse}}$ ensures decisive activation transitions without collapsing distinct objects into the same slot.

Effect of temporal memory. Fig. 13 compares different inputs to the Slot Activation Estimator Φ_{act} on a sequence with persistent partial occlusion. Without temporal memory, activation predictions rely solely on the current Slot Attention candidate $\tilde{s}_{k,t}$, which carries no summary of how the slot has behaved over time, so slots still drift onto other objects. Conditioning on the previous slot state $\boldsymbol{s}_{k,t-1}$ provides a short-term temporal prior and stabilizes the overall scene partition, but a single past frame is not enough to keep the assignment consistent and the same drift remains. The full model uses the temporal memory $\boldsymbol{m}_{k,t-1}$ from the Temporal Context Encoder Ψ_{tce}, which summarizes the recent slot trajectory and keeps each slot on its own object throughout the sequence.

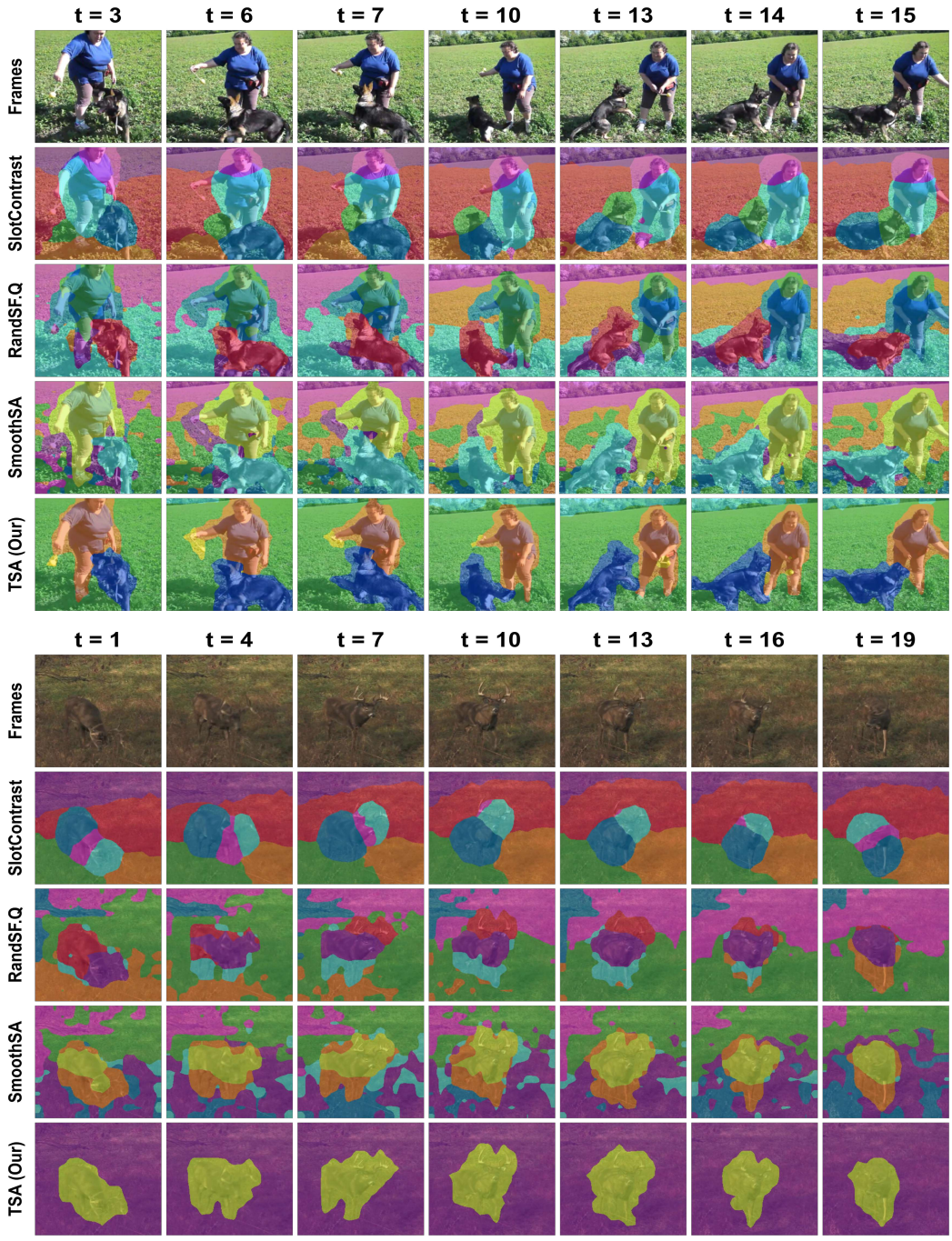

Figure 8: **Additional qualitative results on YouTube-VIS HQ.**

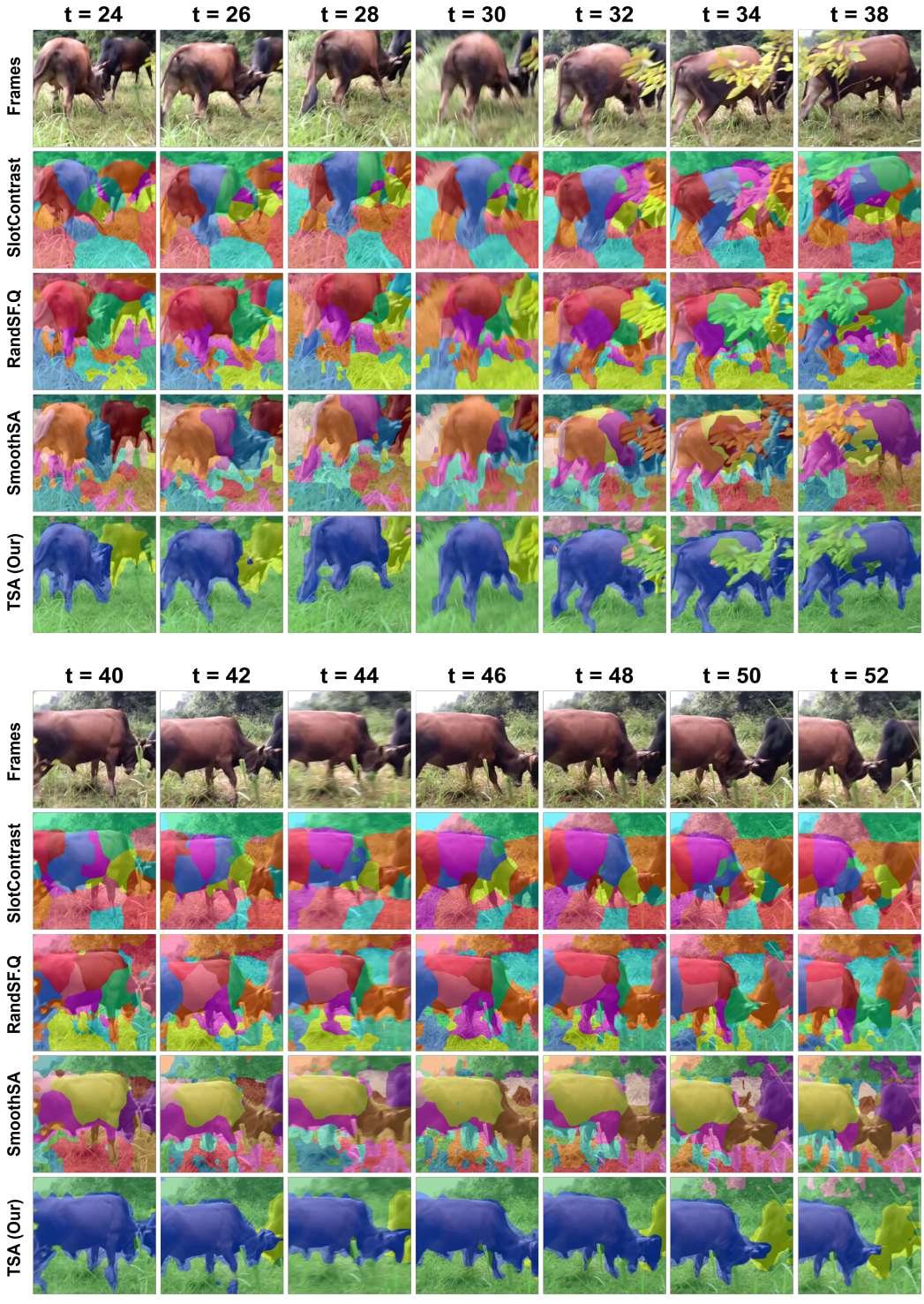

Figure 9: **Qualitative results on OVIS.**

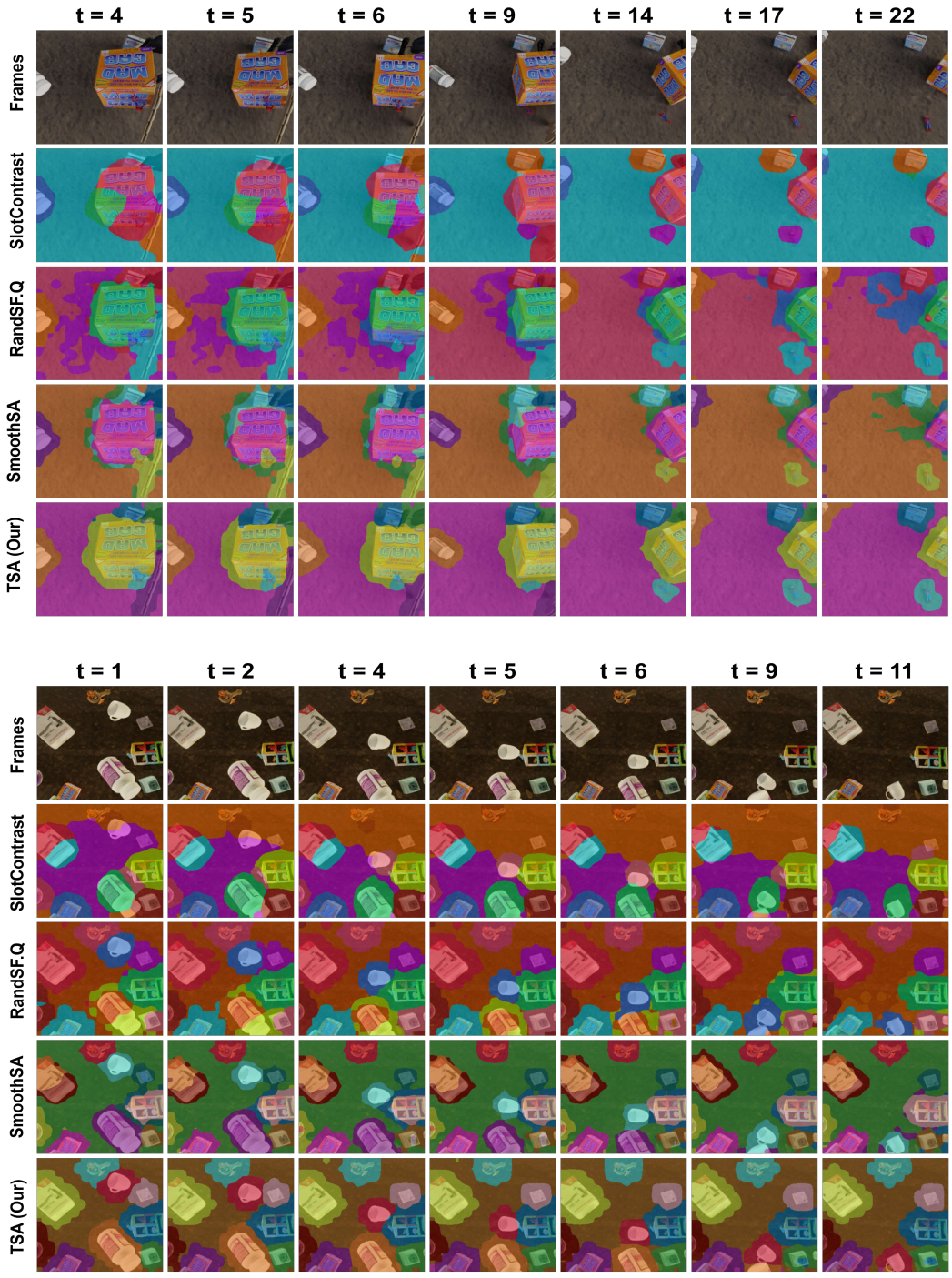

Figure 10: **Qualitative results on MOVi-C and MOVi-E.**

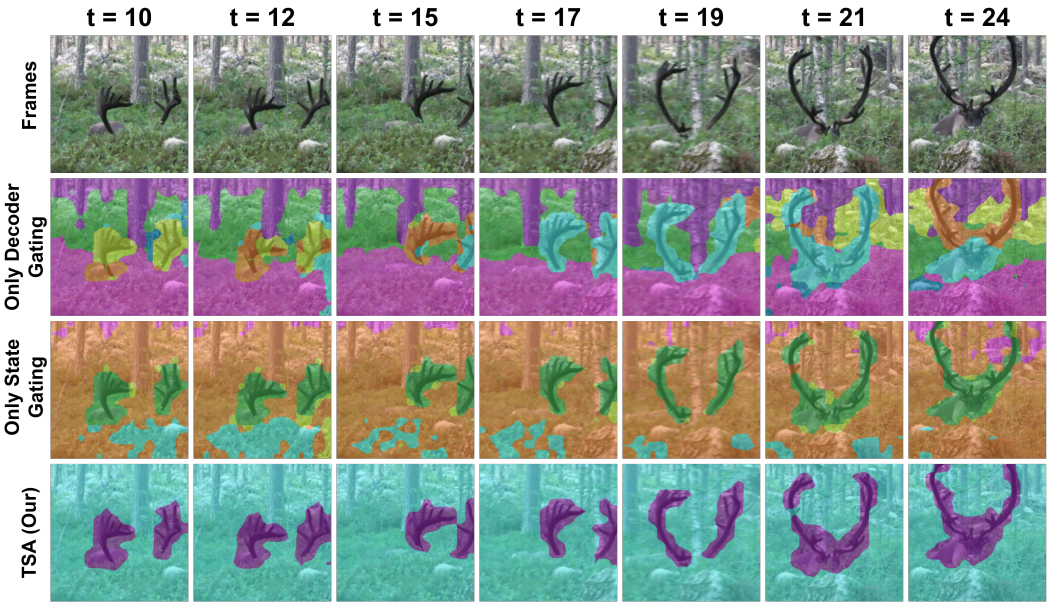

Figure 11: **Activation pathway ablations.** Comparison of TSA with activation-gated decoder participation only, activation-gated state update only, and both pathways jointly gated.

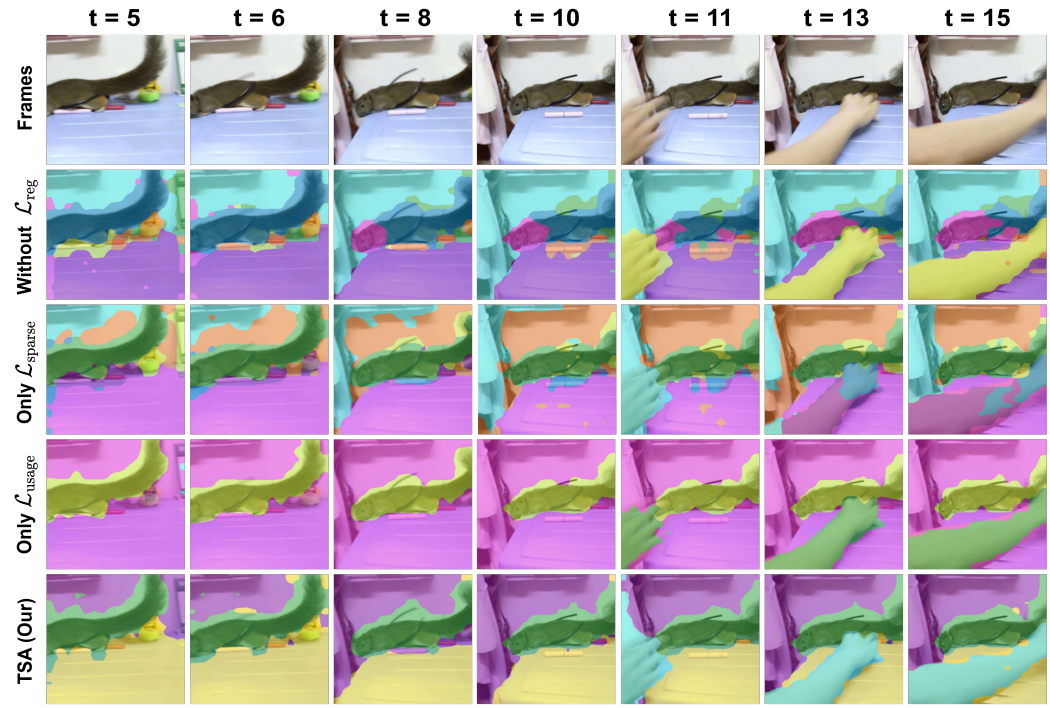

Figure 12: **Activation regularization ablations.** Comparison of TSA trained without $\mathcal{L}_{\text{reg}}$, with $\mathcal{L}_{\text{sparse}}$ only, with $\mathcal{L}_{\text{usage}}$ only, and with both terms combined.

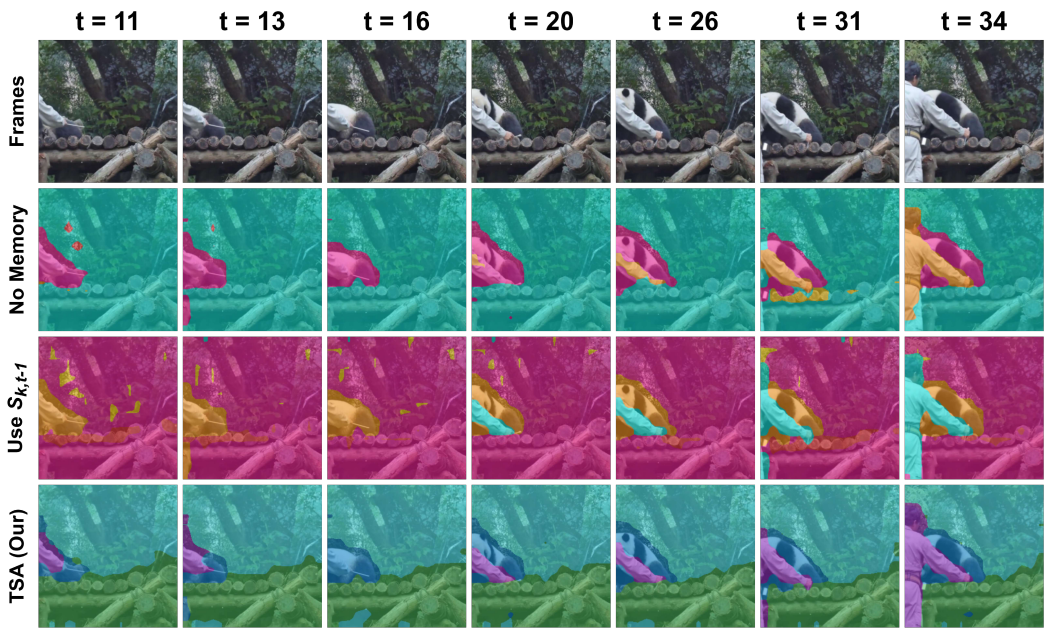

Figure 13: **Temporal memory ablation.** Comparison of different inputs to the Slot Activation Estimator Φ_{act}: no memory, the previous slot state $\boldsymbol{s}_{k,t-1}$, and the temporal memory $\boldsymbol{m}_{k,t-1}$ from the Temporal Context Encoder Ψ_{tce}.