iSEE: Object Permanence through Self-Supervision

Anonymous authors
Paper under double-blind review

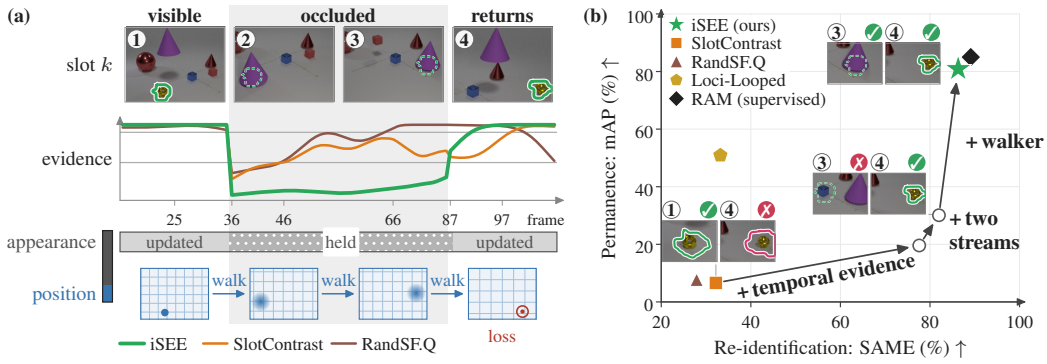

Figure 1: **iSEE achieves self-supervised object permanence, close to a supervised tracker.** (a) Slot k (green) holds the ball; dashed, its mask predicted with iSEE during the occlusion. A slot's evidence is the attention it receives from the frame's patches: iSEE's clearly signals the occlusion and the reappearance. iSEE's slot is separated into appearance and position; during the occlusion the appearance is held and the position keeps walking, learnt only from where it reappears (red ring). (b) On LA-CATER (static), each arrow adds one of our contributions to the existing baseline, SlotContrast (Manasyan et al., 2025).

Abstract

Object permanence, keeping track of an object's identity and position while it is occluded, is central to video representations that track, predict and plan. Trackers that achieve it learn from boxes, track identities and visibility labels. On the other hand, self-supervised object-centric methods discover objects without labels: through slot attention, it represents a video as slots that bind to objects and follow them across frames. However, these slots are lost under occlusion, making the desired permanence impossible. Reasoning permanence is a hard problem because it requires to detect when an object becomes occluded, re-identify when object reappears, and keep the object's hidden position continuous, using reapperance as the only learning cue. To address this, we propose iSEE, a novel framework that offers all three aforementioned requirements, without any labels whatsoever. We built iSEE using the following three proposed components: (i) Object evidence modelling: a slot's attention, compared with its own past, reveals when its object is hidden. (ii) Appearance-position separation: two slot streams let the appearance be held for re-identification while the position keeps changing. (iii) Permanence from reappearance: a walker follows the hidden object's position, trained only on where the object reappears. On LA-CATER static, iSEE returns a reappearing object to its own slot after 86 % of occlusions, against 32 % for SlotContrast (Manasyan et al., 2025), and localises it while hidden within 4.1 mAP of the label-trained SoTA RAM (Tokmakov et al., 2022). The two streams also allow downstream planning, with the position stream as the action of a world model.

1 Introduction

We are interested in occlusion-robustness and object permanence in self-supervised object-centric video representations (Manasyan et al., 2025; Kipf et al., 2022; Singh et al., 2022; Zadaianchuk

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.

et al., 2023) obtained through slot attention (Locatello et al., 2020). These models represent a video as a set of evolving slots, *i.e.*, set of vectors each bound to an object or a region of the frame, and updated from frame to frame. Thus, downstream reasoning and predictions can use object representation with identity, rather than pixels or patches. Specifically, SlotContrast uses contrastive learning on slot evolution, showing improvement in representation over its predecessors. However, slot representations trained thus, without additional labels, are occlusion sensitive (Zadaianchuk et al., 2023; Manasyan et al., 2025). SlotContrast regards *"consistency during long-term full occlusions"* as an open problem (Manasyan et al., 2025).

The problem stems from the implicit assumption that objects are always visible. This justifies the existing methods to update the object slots in every frame. Consequently, under occlusions, the hidden object's slot can simply change its identity to that of a visible patch. Intuitively, under occlusions, object appearance must not be updated, while its position might still change. This relates directly to the second issue that a single slot vector represents both an object's appearance and position. This can lead to the object representation to be biased by position, disregarding visible evidence or lack thereof, under occlusions. Previous approaches to learning object permanence rely on extra supervision: boxes of visible objects (Tokmakov et al., 2022) or a clean image of the scene background (Traub et al., 2024). We instead propose to solve occlusion robustness and object permanence by: (i) Object evidence modeling and ii) Position-appearance separation.

First, we add an explicit object evidence modeling that can acknowledge occlusions to act accordingly. An *evidence* establishes the object's visibility, and can be obtained from a slot's attention to all patches in that frame. Although, previous methods compute the same attention, it is normalized spatially, thus deleting any occlusion cues. We instead obtain the correct evidence essentially free, by normalizing the attention with its maximum along a duration. This temporal evidence normalization (TEN) is applied only at inference. More importantly, TEN allows us to add an explicit *hold* operation, as shown in Fig. 2(b) to pause the slot update while the object is occluded (Fig. 1(a), green line). We tackle the second sub-problem of position-appearance separation with a two stream (appearance and position) framework, hence the *hold* operation in TEN happens solely on the appearance stream. In contrast, the hidden object's position stream continues its update with a trained *walker* (Tokmakov et al., 2022). The two streams are shown in Fig. 2(a), and consists of an appearance stream that reads DINOv2 features computed without a positional encoding (Pawlowsky et al., 2026), and a position stream that reads only the patch coordinates. Finally in order to keep object permanence, we train the walker from TEN's occlusion and reappearance detections (Fig. 1(a)bottom). We show the contributions of TEN, the two streams and the walker in permanence metrics ($8.6 \rightarrow 80.5$), with the three arrows of Fig. 1(b). The resulting model, iSEE, returns reappearing objects to their slot and tracks them while hidden, with no label used at any stage. Additionally, we show the potential of iSEE as a downstream planner of object slots.

Object permanence is considered as a key attribute of visual intelligence in cognitive science, with many studies in infants, in visual cortex and ideations in early computer vision research. Piaget (Piaget, 1954) made the search for an occluded object the milestone of infancy. LA-CATER Girdhar & Ramanan (2020); Shamsian et al. (2020) formulates a computer vision benchmark on the same task. In early computer vision research, Marr's sketches made occluding contours and amodal completion explicit (Marr, 1982; Nitzberg & Mumford, 1990) with similar reasoning. More relevant to our self-supervised training, Gibson (Gibson, 1979) argued that human perception distinguishes occlusions from going out of existence and that observers learn the difference from reappearances. In this work, we model the requirements to learn object permanence self-supervised, *i.e.*, the evidence modeling with TEN, the two-stream representation's explicit what/where separation, and the hidden location walker trained on reappearance in Gibson's principle and showcase its success in LA-CATER.

We summarize our contributions as:

- **Object evidence modelling.** Temporal evidence normalisation (TEN), which compares a slot's attention with its own past to detect when its object is occluded, and holds the slot until the object reappears.
- **Appearance-position separation.** A slot with separate appearance and position streams, so that the appearance of an occluded object is held while its position keeps changing.
- **Object permanence from reappearance.** A walker that follows the position of an occluded object, trained only on where the object reappears, with no labels at any stage.

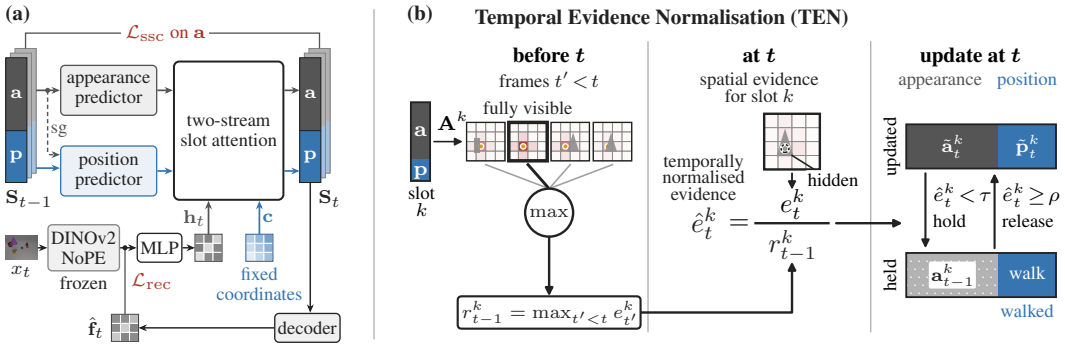

Figure 2: **(a) The two-stream slot in training. (b) TEN measures a slot's evidence on frame t against the largest evidence the slot had before t.** (a) An appearance predictor and a position predictor advance the two streams $\mathbf{a}$ and $\mathbf{p}$ of every slot; the position predictor reads $\mathbf{a}$ through a stop-gradient (sg). The two-stream slot attention updates $\mathbf{a}$ from the appearance grid $\mathbf{h}_t$ and $\mathbf{p}$ from the position grid $\mathbf{c}$ (Eq. 3). The decoder places each object from $\mathbf{p}$ alone, and $\mathcal{L}_{\text{ssc}}$ acts on $\mathbf{a}$ only (Sec. 3.1). (b) For slot k, e_t^k is the mean of its largest attention values on frame t (Eq. 4), and $\hat{e}_t^k = e_t^k / r_{t-1}^k$ (Eq. 5). The slot switches to *held*, its appearance kept and its position walked (Sec. 3.2), when $\hat{e}_t^k$ falls below τ, and back to *updated* when it rises above ρ (Eq. 6).

2 Related Work

Unsupervised object-centric video representation learning. Object-centric models learn, mostly without labels, to represent a video as a set of slots, each bound to image features by slot attention (Locatello et al., 2020) and initialised from the slots of the previous frame (Kipf et al., 2022; Singh et al., 2022). Trained to reconstruct pixels, flow or image tokens, they were shown mostly on synthetic scenes, until self-supervised DINO features (Caron et al., 2021; Oquab et al., 2024) scaled them to real images (Seitzer et al., 2023) and real video (Zadaianchuk et al., 2023; Manasyan et al., 2025). Later methods revise the objective, the slot count or the queries (Seong et al., 2026; Moon & Heo, 2026; Moon et al., 2026; Li et al., 2026; Zhao et al., 2026b;a), and dynamics models predict how the slots evolve (Wu et al., 2023; Villar-Corrales et al., 2023; Li et al., 2025; Jiang et al., 2024; Song et al., 2025). Occlusion, however, remains open: VideoSAUR reports slots reassigned to other objects in longer videos with occlusions.

Separating appearance from position. Generative scene models keep where an object is apart from what it looks like, with an explicit position per object (Eslami et al., 2016; Crawford & Pineau, 2019; Lin et al., 2020b; Jiang et al., 2020; Lin et al., 2020a; Traub et al., 2023) or per particle (Daniel & Tamar, 2022; 2024; Daniel et al., 2026), which DDLP (Daniel & Tamar, 2024) tracks over time and LPWM (Daniel et al., 2026) moves with a latent action per particle. A particle, however, need not follow one object: DDLP reports particles shared by two objects and large objects split over several. LPWM ties each particle to its patch of origin, and all three name camera motion as a limitation. On DINO features, STAITUS (Wei et al., 2026) splits the slot into appearance, position and scale on ISA's per-slot reference frames (Biza et al., 2023), reconstructing RGB from slots placed by their pose. DINO features, however, carry position (Pawlowsky et al., 2026), so its appearance part is not free of it.

Object permanence. Object permanence has mostly been learnt from labels. OPNet on LA-CATER (Shamsian et al., 2020), PermaTrack (Tokmakov et al., 2021) and TCOW (Van Hoorick et al., 2023) are trained on the positions of hidden objects, and RAM (Tokmakov et al., 2022) reduces the labels to boxes and track identities on visible frames, localising the hidden object with a random walk along its memory that is supervised where the object reappears, on the identity it had before the occlusion. Within object-centric learning, Loci-Looped (Traub et al., 2024) and OC-MOT (Zhao et al., 2023) handle occlusion with little supervision: a clean background image and detection labels on at most 6.25 % of frames, respectively.

3 METHOD

A video $x_{1:T}$ shows a scene of at most K objects. Each object has an appearance, which changes slowly over the clip, and a position, its location and extent in the image, which can change on every frame. An object can be fully occluded over a contiguous run of frames, behind an occluder, inside a container or carried by another object, and then reappear (Shamsian et al., 2020).

Object-centric video model. A frozen self-supervised backbone f (Oquab et al., 2024) and a small MLP g_ψ map frame x_t to N patch features $\mathbf{h}_t = g_\psi(f(x_t)) \in \mathbb{R}^{N\times D}$. K slots $\mathbf{S}_t = \{\mathbf{s}_t^k\}_{k=1}^K$ represent the scene: a predictor P_ω advances them to the next frame, and a corrector C_θ binds them to its features with slot attention (Locatello et al., 2020; Kipf et al., 2022),

$$\begin{aligned}\mathbf{S}_t^p &= P_\omega(\mathbf{S}_{t-1}^c), \qquad \mathbf{S}_t^c = C_\theta(\mathbf{h}_t, \mathbf{S}_t^p) : \\ \mathbf{A}_t &= \mathrm{softmax}_K\Big(\tfrac{1}{\sqrt{D}}\, k(\mathbf{h}_t)\, q(\mathbf{S})^\top\Big), \quad \mathbf{s}^k \leftarrow \mathrm{GRU}\Big(\sum_n \bar{A}_{t,n}^k\, v(\mathbf{h}_{t,n}),\ \mathbf{s}^k\Big),\end{aligned} \tag{1}$$

where the softmax is over slots, $\bar{\mathbf{A}}$ is $\mathbf{A}$ renormalised over patches, and the update is iterated a few times per frame, starting from $\mathbf{S} = \mathbf{S}_t^p$; $\mathbf{S}_t^c$ is its result. A decoder reconstructs the backbone features, $\hat{\mathbf{f}}_t = \mathrm{Dec}(\mathbf{S}_t^c)$, with per-slot alpha masks that serve as the segmentation (Seitzer et al., 2023). Training minimises feature reconstruction loss $\mathcal{L}_{rec}$ and a slot–slot contrastive term $\mathcal{L}_{ssc}$ (Manasyan et al., 2025), which makes a slot most similar to itself on the previous frame among all slots in the batch,

$$\mathcal{L} = \sum_t \mathcal{L}_{\text{rec}}(f(x_t), \hat{\mathbf{f}}_t) + \alpha\, \mathcal{L}_{\text{ssc}}(\mathbf{S}_{t-1}^c, \mathbf{S}_t^c), \qquad \mathcal{L}_{\text{rec}} = \|f(x_t) - \hat{\mathbf{f}}_t\|_2^2. \tag{2}$$

Two properties of Eq. 1 lose an object under occlusion: i) The slot is one vector, so an update reaches the object's appearance and its position together. ii) The renormalisation over patches gives every slot an update of full weight, so the slot of a hidden object is rewritten with the patches at its former location. iSEE splits the slot into two streams (Sec. 3.1), then makes the normalisation aware of temporal evidence, a primary prerequisite for handling occlusions, and moves the position of a held slot with a walker (Sec. 3.2).

3.1 TWO STREAMS

A slot is split into an appearance stream and a position stream (Fig. 2a), $\mathbf{s}^k = [\mathbf{a}^k; \mathbf{p}^k]$ with $\mathbf{p}^k \in \mathbb{R}^4$. Only $\mathbf{p}^k$ is written from the position grid, and the decoder places an object from $\mathbf{p}^k$ alone.

Inputs. The corrector reads two grids over the same N patches (Fig. 2a, bottom). The appearance grid $\mathbf{h}_t$ is computed from foundation-model features without a positional encoding (Sec. 4.1). The position grid $\mathbf{c} \in \mathbb{R}^{N\times 4}$ is a fixed Fourier code of each patch's image coordinate $\mathbf{x}_n$ (Tancik et al., 2020), the same on every frame.

Two-stream slot attention. The attention logit of patch n for slot k is the sum of an appearance term and a position term. There is one softmax over slots and one attention map, and each stream is updated from its own grid by its own GRU:

$$\begin{aligned}M_{t,n}^k &= q_a(\mathbf{a}^k)^\top k_a(\mathbf{h}_{t,n}) + q_p(\mathbf{p}^k)^\top k_p(\mathbf{c}_n), \\ \mathbf{a}^k &\leftarrow \mathrm{GRU}_a\Big(\sum_n \bar{A}_{t,n}^k\, v_a(\mathbf{h}_{t,n}), \mathbf{a}^k\Big), \quad \mathbf{p}^k \leftarrow \mathrm{GRU}_p\Big(\sum_n \bar{A}_{t,n}^k\, v_p(\mathbf{c}_n), \mathbf{p}^k\Big),\end{aligned} \tag{3}$$

with $\mathbf{A}_t = \mathrm{softmax}_K(M_t)$. A slot claims a patch whose content matches its appearance and whose location matches its position. Because $\mathbf{p}^k$ is written from an attention-weighted mean of fixed codes, it is the slot's location and extent on the grid, and the recurrence carries it as a state. ISA (Biza et al., 2023) instead recomputes position and scale from the attention at every iteration.

Predictors and objective. Appearance and position streams have their own predictors, each a transformer over that stream of all slots. The position predictor is also conditioned on an 8-dimensional projection of $\mathbf{a}$ under a stop-gradient (sg in Fig. 2a), since an object's next position depends on its shape. $\mathcal{L}_{\text{ssc}}$ acts on $\mathbf{a}$ only; the objective is otherwise unchanged. Nothing in the architecture mixes the two streams' slot vectors. Appendix A.1 details the decoder.

3.2 TEMPORAL EVIDENCE NORMALISATION AND THE WALKER

An occlusion changes whether an object is seen, not the object. Over an occlusion on frames $t_s \le t < t_e$, (i) the object's appearance persists and (ii) its position follows a continuous path, so a reappearance is the same appearance seen at the end of that path. TEN detects t_s and t_e from the slot's own attention and holds its appearance, for (i) and the walker continues its position along the path, for (ii).

Evidence modelling. Detecting t_s and t_e needs two readings of the slot's attention: whether the object is still where the slot last saw it, to start the hold, and whether it is seen again anywhere, even partly, to release it. The first reading must be local: when its object is hidden, a slot can move onto the occluder or another object, and its attention over the whole frame stays high. How much attention a slot receives also differs between slots, objects and scenes, so a threshold that fits one slot does not fit another; each reading is therefore compared with the same slot's earlier values. Let $A^k_{t,n}$ be the share of patch n that slot k receives on frame t under the softmax over slots (Eq. 1), on the first corrector iteration, before the frame changes the slots. The *evidence*, read by the release, and the *local evidence*, read by the hold, are

$$e^k_t = \frac{1}{m} \sum_{n \in \mathcal{I}^k_t} A^k_{t,n}, \qquad e^{k,\mathrm{loc}}_t = \max_n \frac{1}{|\mathcal{B}(n)|} \sum_{n' \in \mathcal{B}(n)} W^k_{n'}\, A^k_{t,n'}, \tag{4}$$

with $\mathcal{I}^k_t$ the m patches of largest $A^k_{t,n}$; $\mathcal{B}(n)$ the 3×3 block around patch n, which asks for a connected piece of the object; and W^k a window built from the slot's final-iteration attention on its last frame not held (Appendix A). Each is divided by its own largest value on the earlier frames of the clip (Fig. 2b, left),

$$\hat{e}^k_t = \frac{e^k_t}{r^k_{t-1}}, \;\; r^k_{t-1} = \max_{t'<t} e^k_{t'}; \qquad \hat{e}^{k,\mathrm{loc}}_t = \frac{e^{k,\mathrm{loc}}_t}{r^{k,\mathrm{loc}}_{t-1}}, \;\; r^{k,\mathrm{loc}}_{t-1} = \max_{t'<t} e^{k,\mathrm{loc}}_{t'}. \tag{5}$$

A slot is *held* from the frame on which $\hat{e}^{k,\mathrm{loc}}_t$ falls below τ until the frame on which $\hat{e}^k_t$ rises above $\rho > \tau$, so a partial glimpse does not end it (Fig. 2b, right); a hold does not start while $e^{k,\mathrm{loc}}_t$ is rising sharply (Appendix A). With $u^k_t \in \{0,1\}$ the held indicator, $\tilde{\mathbf{a}}^k_t$ and $\tilde{\mathbf{p}}^k_t$ the corrector's output and $\hat{\mathbf{p}}^k_t$ the walker's position,

$$\begin{aligned} u^k_t &= \mathbf{1}\big[\hat{e}^{k,\mathrm{loc}}_t < \tau\big]\,(1 - u^k_{t-1}) + \mathbf{1}\big[\hat{e}^k_t < \rho\big]\,u^k_{t-1}, \\ \mathbf{a}^k_t &= (1-u^k_t)\,\tilde{\mathbf{a}}^k_t + u^k_t\,\mathbf{a}^k_{t-1}, \qquad \mathbf{p}^k_t = (1-u^k_t)\,\tilde{\mathbf{p}}^k_t + u^k_t\,\hat{\mathbf{p}}^k_t. \end{aligned} \tag{6}$$

The walker. The walker follows the content at a held slot's last position and shifts that position by how far the content moves. It follows the content with a random walk over patches on a graph of appearance features (Jabri et al., 2020; Tokmakov et al., 2022). To read a patch in the coordinates of the position stream, let $P^*(n)$ be the position of a slot that attends to patch n alone: the value to which GRU_p (Eq. 3) converges when its input is $v_p(\mathbf{c}_n)$, with $\mathbf{c}_n$ the Fourier code of the patch's coordinate (Sec. 3.1). Let n_a be the patch whose P^* is nearest the slot's position $\mathbf{p}^k_{t_0}$ on an updated frame $t_0 < t_s$ (Appendix A). The walk π_t, a distribution over patches, starts as a point mass at n_a and moves between consecutive appearance grids,

$$\pi_t = \pi_{t-1}\,\mathcal{T}_t, \qquad \mathcal{T}_t(n \to n') = \mathrm{softmax}_{n' \in \mathcal{W}(n)}\big(s\,\langle \mathbf{z}_{t-1,n}, \mathbf{z}_{t,n'} \rangle + \beta_{n'-n}\big), \tag{7}$$

with $\mathcal{W}(n)$ a window around n, $\mathbf{z}_t$ a learned embedding of $\mathbf{h}_t$, s a learned scale and β a learned prior over offsets. The walker's position is $\hat{\mathbf{p}}^k_t = \mathbf{p}^k_{t_0} + P^*(\bar{\mathbf{x}}_t) - P^*(n_a)$, with P^* interpolated at $\bar{\mathbf{x}}_t$, the mean position of π_t near its mode. As the held slot's position (Eq. 6), $\hat{\mathbf{p}}^k_t$ places the slot's next attention, so the kept appearance is looked for where the walker expects the object. The walker is trained on reappearance: its only target is $\mathbf{p}^k_{t_e+1}$, the position the encoder gives the slot just after its release at t_e,

$$\mathcal{L}_{\text{walk}} = -(1+\lambda L)\,\log \sum_n \pi_{t_e-1}(n)\,\mathcal{N}\big(\mathbf{p}^k_{t_e+1};\; \mathbf{p}^k_{t_0} + P^*(n) - P^*(n_a),\; \sigma^2\mathbf{I}\big), \tag{8}$$

with $\mathcal{N}$ a Gaussian of learned width σ, L the length of the occlusion and λ a weight per occluded frame. Because the held appearance returns the object to the slot that held it, the target is the right object's, and no ground-truth position enters the loss.

The encoder is trained once with Eq. 2 and TEN has no parameters; the walker is trained afterwards on the encoder's outputs, and at inference TEN and the walker run together. Appendix A gives the stages (Table S1), our method's inference algorithm (Algorithm 1) and other constants.

4 EXPERIMENTS

4.1 SETUP

Implementation details. The backbone is DINOv2 ViT-S/14 with registers (Oquab et al., 2024; Darcet et al., 2024) without its positional embedding (Pawlowsky et al., 2026), frozen. Slots have width 64, split into 60 appearance and 4 position dimensions; $K = 11$ on LA-CATER and 15 on KITTI. Everything else is the configuration of Manasyan et al. (2025), except the training window, which is 8 frames rather than 4 so that an occlusion and the reappearance after it fall inside one window. The dynamics model is OCVP-Par (Villar-Corrales et al., 2023). Every other value is in Appendices A, B and D.

Datasets. LA-CATER (Shamsian et al., 2020) extends CATER (Girdhar & Ramanan, 2020): rendered scenes in which geometric objects move and occlude one another, and a target object is covered by cones that may then carry it. Every object is labelled on every frame as visible, occluded, contained or carried, with an amodal box while it is occluded. We use its own static camera and the moving camera of Tokmakov et al. (2022), and score 195 and 388 occlusions of the target, 183 and 360 of them with a scorable reappearance (Appendix B.2). MOVi-C (Greff et al., 2022) renders scanned objects falling on photographic backgrounds and carries discovery, separation and dynamics on its 250-video validation split, as in Manasyan et al. (2025). KITTI (Geiger et al., 2012) is real-world driving video, split in time into training and validation halves as in Tokmakov et al. (2022); discovery is scored on the 70 validation clips and re-identification on the occlusions of cars, both from the masks of KITTI-MOTS (Voigtlaender et al., 2019).

Baselines. We build on top of SlotContrast (Manasyan et al., 2025). We train it under same settings except the training clip window, which is changed from 4 to 8 frames to capture more occlusion-reappearance events. RandSF.Q (Zhao et al., 2026a) predicts the next frame's slot-attention queries from slot-feature pairs sampled from the recurrence. We run the authors' code with a LA-CATER configuration of our own. Loci-Looped (Traub et al., 2024), the permanence model closest to ours, keeps a position and a gestalt code per slot and imagines the position while its percept gate is closed. It also requires a background image of the scene, which its released loader builds from the ground-truth masks, so its rows are not label-free. RAM (Tokmakov et al., 2022) is the supervised reference, trained with boxes, track identities and visibility labels. We report its raw output (Appendix B.3). *Frozen* is iSEE's slot frozen at its last visible position.

Metrics. Permanence is the mAP of OPNet (Shamsian et al., 2020): the share of an occlusion's hidden frames on which the predicted box overlaps the target's amodal box above an IoU threshold, averaged over occlusions. We report *mAP*@$[0.1, 0.3]$ $\uparrow$, its mean over the thresholds 0.10 to 0.30. A slot model's box keeps the size of its mask before the occlusion and is centred on the predicted position, as in Tokmakov et al. (2022) (Appendix B.4).

Re-identification is *same-slot return* (SAME) $\uparrow$: the share of occlusions on which the slot that owns the target at the reappearance is the slot that owned it before the occlusion. A slot owns the target on a frame when its grouping attention carries the largest mass on the target's visible pixels. Before and after are the last and the first frame on which at least half of the target's amodal mask is visible. Appendix C.2 reports HOTA (Luiten et al., 2021) over whole clips.

Discovery uses video FG-ARI (Hubert & Arabie, 1985) and video mBO on the decoder's masks over the whole clip, as in Manasyan et al. (2025).

4.2 OBJECT PERMANENCE AND RE-IDENTIFICATION

In Table 1, we show that iSEE, with no label at any stage, is within 4.1 points of the supervised RAM under the static camera (80.9 vs 85.0 over all hidden frames) and ahead of every other model. Carried frames are the hardest case, because the container moves the target away from its last visible

Table 1: **Position of the object under occlusion in the LA-CATER dataset.** mAP@$[0.1, 0.3]$ ↑ ($\times 100$) on the hidden frames of each label. In parentheses, what a row is given beyond the video (bg: a per-frame background image); *frozen* holds iSEE's slot at its last visible position. Under the static camera iSEE is close to the supervised method RAM. Bold: best row that reads the video alone, which excludes *frozen*, RAM and Loci-Looped.

	static camera			moving camera		
	occluded	contained	carried	occluded	contained	carried
RAM (vis., bbox, id)	73.8	86.7	81.0	83.2	87.4	81.1
Loci-Looped (bg)	56.6	49.9	32.9	49.5	51.8	37.9
frozen	57.8	56.8	41.9	51.4	32.2	13.6
SlotContrast	8.6	4.3	3.5	3.1	1.3	1.2
RandSF.Q	11.5	6.4	5.4	1.3	1.2	1.0
iSEE	**80.5**	**84.2**	**78.7**	**54.4**	**51.6**	**46.3**

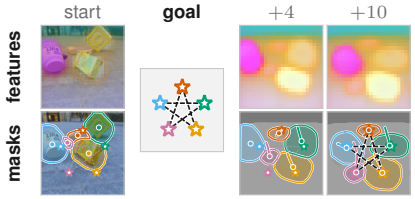

Figure 3: **Planning in imagination on MOVi-C: the goal is a star of the clip's own objects.** *goal*: a pentagon. *features* (top): decoded DINO features, PCA as RGB; *masks* (bottom): decoded slot masks with their tracks. Action optimisation by gradient descent: the error falls from 26.7 to 1.5 px.

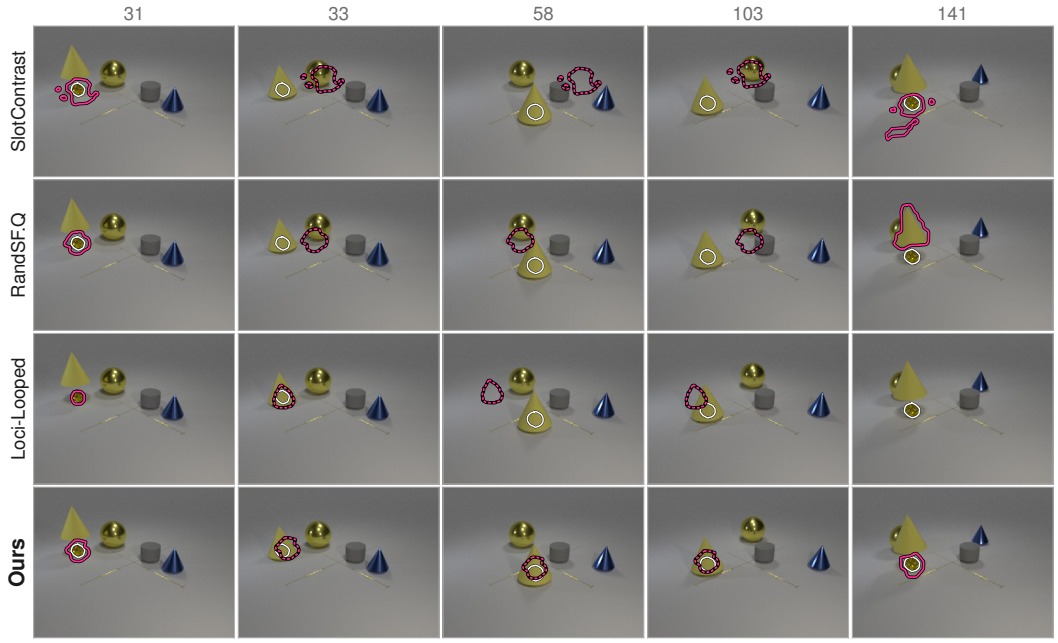

Figure 4: **Object permanence in LA-CATER: a cone is put over the target and carries it across the frame.** The target is visible only in the first and last columns. White: its amodal mask. Red: the followed slot, *solid* for its own mask, *dashed* where moved onto the predicted position; no red: the slot owns no pixel there. Only iSEE tracks the target while it is hidden.

position: *frozen* falls from 57.8 on occluded frames to 41.9 on carried ones, and Loci-Looped from 56.6 to 32.9, while iSEE keeps 78.7. Fig. 4 shows one such occlusion, and more in Appendix E. Under the moving camera, iSEE scores lower than in static camera. The drop starts upstream of the walker, in the encoder, which discovers objects less well under this camera and returns fewer targets to their slot.

Table 2 gives SAME by the length of the occlusion, and Fig. 5 shows one occlusion on KITTI. Figs. S5 and S6 give more examples. Under the static camera, every baseline except the supervised RAM loses the target's slot once the occlusion exceeds a second: its SAME falls by 20 to 36 points and does not recover. iSEE still returns the target after 77.5 % of occlusions longer than four seconds under the static camera and 64.2 % under the moving one, where these baselines reach at most 25.4 %. On KITTI, it returns the car to its slot after 48.4 % of occlusions, against 29.0 % for SlotContrast.

Table 2: **Re-identification after an occlusion, LA-CATER & KITTI.** SAME ↑ (%) by the length of the occlusion in seconds. *requires*: what the row is given beyond the video, at training or at test time. With no label at any stage iSEE is ahead of every row but the supervised RAM in every bin, and ahead of RAM as well on the 0.2–1.0 s and 2.1–4.2 s occlusions under the static camera.

		LA-CATER, static camera				LA-CATER, moving camera				KITTI
	requires	0.2–1.0 s	1.0–2.1 s	2.1–4.2 s	≥4.2 s	0.2–1.0 s	1.0–2.1 s	2.1–4.2 s	≥4.2 s	0.1–2.2 s
RAM	boxes, identities, visibility	90.2	95.5	85.7	88.7	95.0	90.0	77.3	69.5	87.1
Loci-Looped	per-frame background	53.1	33.3	24.5	25.4	39.5	31.1	27.3	21.1	–
RandSF.Q	none	56.1	22.7	16.3	21.1	38.1	23.3	16.7	15.8	–
SlotContrast	none	63.4	27.3	20.4	23.9	30.2	11.7	12.1	13.7	29.0
iSEE	none	**95.1**	**86.4**	**91.8**	**77.5**	**82.0**	**81.7**	**75.8**	**64.2**	**48.4**

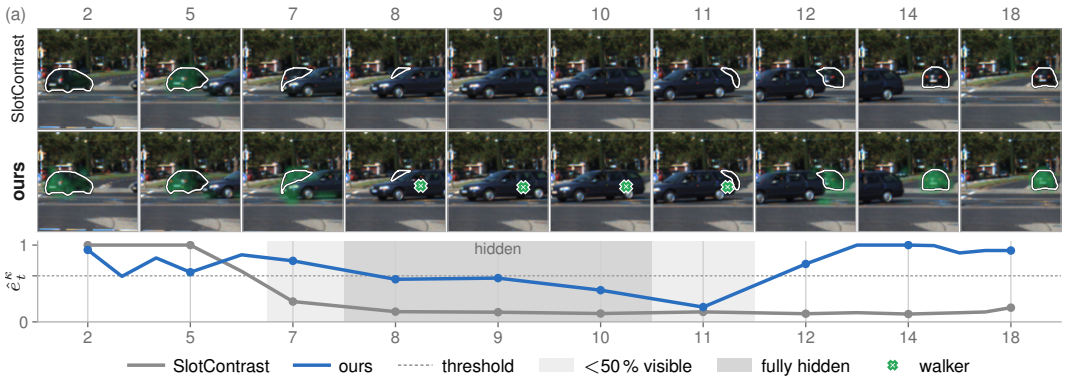

Figure 5: **Occlusions in KITTI: a car passes behind another road user and reappears.** Unevenly spaced frames of one tile, cropped to the car and its occluder. White: the car's visible mask. Green: the slot on the car before the occlusion; cross: the walker's predicted centre while the car is hidden. Both models' evidence falls while the car is hidden; only iSEE's recovers, and its slot returns.

4.3 Object discovery

We report object discovery in Table 3, and show the masks and slot identities on MOVi-C and KITTI in Figs. S3 and S4. iSEE discovers objects better than SlotContrast on every dataset. Separating appearance from position matters more on real video: with TEN on both encoders, iSEE leads by 9.5 FG-ARI and 10.5 mBO on KITTI, against 2.1 to 3.5 FG-ARI on the rendered datasets, while TEN alone does not help SlotContrast there (42.4 to 39.4 FG-ARI).

4.4 Application: planning with a world model

A world model predicts how the world changes under an action. It must learn to separate the action from everything else that changes. Without the action as input, the future is ambiguous, and a predictor trained under a squared error averages over the possible futures (Mathieu et al., 2016). SlotFormer (Wu et al., 2023) and OCVP (Villar-Corrales et al., 2023) do exactly that (Villar-Corrales & Behnke, 2025).

In iSEE, the action comes for free. The position stream is the object's kinematic state, and its change is the action. Given to the predictor through a two-dimensional bottleneck (Fig. S2), it becomes a code of the object's motion. SlotContrast's whole-slot change, given the same way, does not, and its predictor stays 9.4 FG-ARI below its oracle, against 3.9 for iSEE (Table 5).

Because the rollout is differentiable in the action, planning follows by gradient descent. We set the goal to arrange a clip's objects into a star, the planner brings their mask centres from 25.1 to 5.2 px of their vertices at 128 px over 20 validation clips (Fig. 3).

Table 3: **Object discovery on MOVi-C, LA-CATER and KITTI.** Video FG-ARI ↑ / video mBO ↑ ($\times 100$) on decoder masks over whole clips. Splitting the slot costs no discovery.

		MOVi-C		LA-CATER static		LA-CATER moving		KITTI	
method	features	FG-ARI ↑	mBO ↑	FG-ARI ↑	mBO ↑	FG-ARI ↑	mBO ↑	FG-ARI ↑	mBO ↑
SlotContrast	DINOv2	66.9	32.0	94.2	21.5	83.1	13.7	42.4	12.4
SlotContrast + TEN	DINOv2	68.1	31.4	94.5	22.3	86.7	14.7	39.4	12.3
iSEE	DINOv2 NoPE	**71.6**	**36.0**	**96.6**	**23.9**	**89.4**	**17.8**	**48.9**	**22.9**

Table 4: **Ablation on LA-CATER (static).** Each row adds one component to the row above, starting from SlotContrast. mAP is the mAP@$[0.1, 0.3]$ of Table 1; *all*: every frame of the clip, the slot fixed before the occlusion. Bold: best per column.

		discovery		mAP ↑	
method	SAME ↑	FG-ARI ↑	mBO ↑	all	hidden
SlotContrast	32.2	94.2	21.5	14.6	6.6
+ TEN	77.6	94.5	22.3	22.9	19.6
+ two streams	71.0	95.3	18.5	20.3	6.9
+ NoPE features	82.0	96.6	23.9	69.0	30.1
+ walker (iSEE)	**86.3**	**96.9**	**24.0**	**88.2**	**80.9**

Table 5: **Dynamics predictor on MOVi-C with action input.** Video FG-ARI ↑ / video mBO ↑ ($\times 100$) over ten rolled-out frames after a 14-frame burn-in; architecture in Fig. S2. *frozen* repeats the burn-in's last slots; *oracle* encodes the true future. iSEE comes close to its oracle; SlotContrast falls further short of its own.

model	FG-ARI ↑	mBO ↑
oracle / frozen, SlotContrast	76.0 / 55.0	31.6 / 21.8
oracle / frozen, iSEE	78.4 / 54.2	37.2 / 23.6
SlotContrast	66.6	26.9
iSEE	**74.5**	**33.1**

4.5 Ablations

TEN improves re-identification already on SlotContrast (Table 4). Splitting appearance and position hurts while the features carry position. On features computed without a positional encoding (DINOv2 NoPE, Pawlowsky et al., 2026), the split improves re-identification and, most notably, mAP. The walker then tracks the object while it is hidden.

5 Conclusion

iSEE identifies two properties that object permanence requires of a slot representation. Building on these properties, iSEE achieves object permanence without any labels: it returns objects to their slots after long occlusions and localises them while hidden. Under a static camera, it is close to a tracker trained with boxes, track identities and visibility labels. The separation of appearance and position gives world models a natural action, through which a planner reaches its goal by gradient descent. Taken together, we expect these properties to carry over to other object-centric models, and to make their representations a basis for world models that plan through occlusion.

Limitations. iSEE is built on its slot model and inherits its binding. When an object is occluded, its slot can latch onto another object that looks alike (Fig. S10a) or be released onto one before the object returns (Fig. S9), and a slot that is not on the target when the occlusion begins cannot follow it (Fig. S11, *wrong slot*). It also inherits a fixed number of slots per dataset. Conversely, any advance in the underlying slot model carries into iSEE for free, and with it into re-identification and permanence.

AI Use Statement

We used AI assistants (Claude, Gemini, ChatGPT) to write the code, to search and cross-check related work, and to edit the text. The ideas, the design of the experiments and the claims are the authors', who checked them against the code and the cited papers.

Reproducibility Statement

The encoder changes only the keys of the released SlotContrast configuration named in Secs. 3 and 4.1. Every other constant, the scored occlusions, the metrics, the baselines' checkpoints and the

dynamics model are in Appendices A, B and D. Code, configurations, the occlusion lists and the trained walkers will be released.

REFERENCES

Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. In *International Conference on Learning Representations (ICLR)*, 2016.

Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008:246309, 2008. doi: 10.1155/2008/246309.

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2507–2527, 2023. arXiv:2302.04973.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

Tal Daniel and Aviv Tamar. Unsupervised image representation learning with deep latent particles. In *International Conference on Machine Learning (ICML)*, 2022.

Tal Daniel and Aviv Tamar. DDLP: Unsupervised object-centric video prediction with deep dynamic latent particles. *Transactions on Machine Learning Research*, 2024. arXiv:2306.05957.

Tal Daniel, Carl Qi, Dan Haramati, Amir Zadeh, Chuan Li, Aviv Tamar, Deepak Pathak, and David Held. Latent particle world models: Self-supervised object-centric stochastic dynamics modeling. In *International Conference on Learning Representations*, 2026. arXiv:2603.04553.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, 1979.

Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for compositional actions and TEmporal reasoning. In *International Conference on Learning Representations (ICLR)*, 2020.

Xinrui Gong, Oliver Hahn, Christoph Reich, Krishnakant Singh, Simone Schaub-Meyer, Daniel Cremers, and Stefan Roth. Motion-refined DINOSAUR for unsupervised multi-object discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2025.

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

dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985. doi: 10.1007/BF01908075.

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

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

Jindong Jiang, Fei Deng, Gautam Singh, Minseung Lee, and Sungjin Ahn. Slot state space models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2406.12272.

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

Jian Li, Pu Ren, Yang Liu, and Hao Sun. Reasoning-enhanced object-centric learning for videos. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2025. doi: 10.1145/3690624.3709168. arXiv:2403.15245.

Zhiyuan Li, Rongzhen Zhao, Wenyan Yang, Wenshuai Zhao, Pekka Marttinen, and Joni Pajarinen. Rethinking temporal consistency in video object-centric learning: From prediction to correspondence. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. In *International Conference on Machine Learning (ICML)*, 2020a.

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. In *International Conference on Learning Representations (ICLR)*, 2020b.

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

Jonathon Luiten, Aljoša Ošep, 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*, 2021.

Anna Manasyan, Maximilian Seitzer, Filip Radovic, Georg Martius, and Andrii Zadaianchuk. Temporally consistent object-centric learning by contrasting slots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, San Francisco, 1982.

Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations (ICLR)*, 2016.

WonJun Moon and Jae-Pil Heo. Selective synergistic learning for video object-centric learning. In *European Conference on Computer Vision (ECCV)*, pp. 21–39, 2026. doi: 10.1007/978-3-032-37383-0_2. arXiv:2606.15527.

WonJun Moon, Hyun Seok Seong, and Jae-Pil Heo. Reconstruction-guided slot curriculum: Addressing object over-fragmentation in video object-centric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Duc Nguyen, Sieu Tran, Hao Vo, Khoa Vo, Duy Minh Ho Nguyen, Nghi D. Q. Bui, Anh Nguyen, Long Mai, and Ngan Le. TSA: Temporal slot activation for persistent object-centric video representation. *arXiv preprint arXiv:2606.13714*, 2026.

Mark Nitzberg and David Mumford. The 2.1-D sketch. In *Proceedings of the Third IEEE International Conference on Computer Vision (ICCV)*, pp. 138–144, 1990. doi: 10.1109/ICCV.1990.139511.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

Moritz Pawlowsky, Antonis Vamvakeros, Alexander Weiss, Anja Bielefeld, Samuel J. Cooper, and Ronan Docherty. What DINO saw: ALiBi positional encoding reduces positional bias in vision transformers. *arXiv preprint arXiv:2603.16840*, 2026.

Jean Piaget. *The Construction of Reality in the Child*. Basic Books, New York, 1954. Translated by M. Cook.

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 Workshops (ECCVW)*, 2016.

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 *International Conference on Learning Representations (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 *International Conference on Learning Representations (ICLR)*, 2026.

Aviv Shamsian, Ofri Kleinfeld, Amir Globerson, and Gal Chechik. Learning object permanence from video. In *European Conference on Computer Vision (ECCV)*, 2020.

Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yeon-Ji Song, Jaein Kim, Suhyung Choi, Jin-Hwa Kim, and Byoung-Tak Zhang. OCK: Unsupervised dynamic video prediction with object-centric kinematics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2404.18423.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Pavel Tokmakov, Allan Jabri, Jie Li, and Adrien Gaidon. Object permanence emerges in a random walk along memory. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Manuel Traub, Sebastian Otte, Tobias Menge, Matthias Karlbauer, Jannik Thümmel, and Martin V. Butz. Learning what and where: Disentangling location and identity tracking without supervision. In *International Conference on Learning Representations (ICLR)*, 2023.

Manuel Traub, Frederic Becker, Sebastian Otte, and Martin V. Butz. Learning object permanence from videos via latent imaginations. In *International Conference on Artificial Neural Networks (ICANN)*, volume 15018 of *Lecture Notes in Computer Science*, pp. 223–240, 2024. doi: 10.1007/978-3-031-72338-4_16. arXiv:2310.10372.

Basile Van Hoorick, Pavel Tokmakov, Simon Stent, Jie Li, and Carl Vondrick. Tracking through containers and occluders in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Angel Villar-Corrales and Sven Behnke. PlaySlot: Learning inverse latent dynamics for controllable object-centric video prediction and planning. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.07600.

Angel Villar-Corrales, Ismail Wahdan, and Sven Behnke. Object-centric video prediction via decoupling of object dynamics and interactions. In *IEEE International Conference on Image Processing (ICIP)*, 2023.

Paul Voigtlaender, Michael Krause, Aljoša Ošep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. MOTS: Multi-object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Amaury Wei, Ismail Nejjar, and Olga Fink. Rethinking object-centric representations for video dynamics modeling. *arXiv preprint arXiv:2606.23436*, 2026.

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

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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 (AAAI)*, 2026a.

Rongzhen Zhao, Wenyan Yang, Juho Kannala, and Joni Pajarinen. Smoothing slot attention iterations and recurrences. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026b.

Zixu Zhao, Jiaze Wang, Max Horn, Yizhuo Ding, Tong He, Zechen Bai, Dominik Zietlow, Carl-Johann Simon-Gabriel, Bing Shuai, Zhuowen Tu, Thomas Brox, Bernt Schiele, Yanwei Fu, Francesco Locatello, Zheng Zhang, and Tianjun Xiao. Object-centric multiple object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

A METHOD

stage	what is trained	signal	what is frozen
1. encoder	g_ψ, corrector, predictors, decoder	Eq. 2	backbone f
2. TEN	nothing	none	everything
3. walker	the walker of Eq. 7	Eq. 8, on the positions and holds of stages 1–2	encoder, P^*

Table S1: The three stages of iSEE. At inference, TEN and the walker run together, and the walker's position is written into the held slots. No box, mask, visibility or occlusion label is used at any stage.

A.1 TWO STREAMS

Slots are 64-dimensional, $[0, 60)$ the appearance stream and $[60, 64)$ the position stream, and are initialised by SlotContrast's learned initialisation over all 64 dimensions. The position grid is $\mathbf{c}_n = [\sin \frac{\pi}{2}\mathbf{x}_n; \cos \frac{\pi}{2}\mathbf{x}_n]$ with $\mathbf{x}_n \in [-1, 1]^2$ the patch centre, carried in its own channels. The position read-in maps start at the identity, so the first position logit is a spotlight on the slot's position and the first position update is the circular mean of the coordinate codes it attends to. The position stream is RMS-normalised, has no MLP, and enters the attention at gain 1. The corrector runs two iterations per frame and three on the first. Each predictor is one transformer block with four heads. The decoder reconstructs $f(x_t)$ in each object's own frame (Biza et al., 2023), reading a position $\boldsymbol{\mu}^k$ and a scale $\boldsymbol{\sigma}^k$ from $\mathbf{p}^k$ alone:

$$(\boldsymbol{\mu}^k, \boldsymbol{\sigma}^k) = h(\mathbf{p}^k), \qquad \mathbf{u}_n^k = \frac{\mathbf{x}_n - \boldsymbol{\mu}^k}{\delta\, \boldsymbol{\sigma}^k},$$
$$(\hat{\mathbf{f}}_n^k, \alpha_n^k) = \mathrm{MLP}\big([\mathbf{a}^k; \gamma(\mathbf{u}_n^k)]\big), \qquad \hat{\mathbf{f}}_n = \sum_k \mathrm{softmax}_k(\alpha_n^k)\, \hat{\mathbf{f}}_n^k,$$

with h a head of 64 hidden units ($\tanh$ for $\boldsymbol{\mu}$, a clamped exponential for $\boldsymbol{\sigma} \in [0.01, 4]$), $\delta = 5$ as in Biza et al. (2023), γ an 8-band Fourier embedding, and an MLP of three hidden layers of 1024. The painting MLP has no access to absolute position.

A.2 TEMPORAL EVIDENCE NORMALISATION

The thresholds are $\tau = 0.5$ and $\rho = 0.9$. The evidence is read from the first corrector iteration and is detached; $m = 4$. The window W^k is the slot's final-iteration attention on the last frame on which it was not held, scaled to a maximum of one and dilated by a 5×5 maximum filter; $W^k = 1$ on the first frame, which is never held. The references r^k and $r^{k,\mathrm{loc}}$ are both updated on held frames as well as on updated ones. A hold does not start while the local evidence is rising: besides $\hat{e}_t^{k,\mathrm{loc}} < \tau$, it requires $\big(e_t^{k,\mathrm{loc}}/(2\,\bar{e}_{t-1}^k)\big)^8 < \tau$, with $\bar{e}^k$ the exponential average of $e^{k,\mathrm{loc}}$ at rate 0.5. A held slot keeps dimensions $[0, 60)$ of the state it committed on the previous frame; the decoder reads the state the corrector produced for the current frame. On a slot that is one vector, the hold keeps the whole vector.

A.3 WALKER

P^* is computed from the frozen corrector, by 200 iterations of GRU_p from zero driven by $v_p(\mathbf{c}_n)$, and is interpolated bilinearly between patches. The walker moves every second frame, the rate of the encoder's training clips; on the frame in between, the position written into the slot is read one step ahead, from the next frame's appearance grid. It restarts whenever the slot has been updated on three consecutive steps, from the slot's position on the middle one, so a hold is walked from a position the slot had while its object was clearly visible. $\bar{\mathbf{x}}_t$ is the π_t-weighted mean position over the 3×3 patches around the mode of π_t smoothed by a 3×3 box. The embedding $\mathbf{z}_t$ is a layer norm and a 1×1 convolution of $\mathbf{h}_t$ to 32 channels, plus a zero-initialised residual from a 3×3 convolutional GRU (Ballas et al., 2016) run forward over the clip, followed by a 1×1 convolution and ℓ_2 normalisation. $\mathcal{W}(n)$ is 11×11 patches; s starts at 20, β at 5 on the zero offset and 0

elsewhere, and σ at 0.05 with a floor of 0.02. In Eq. 8, $\lambda = 1$ per frame. Anchors and targets are kept only where the slot's position lies within a distance of P^* set per encoder (a point-like slot), 0.35 on the static camera and 0.6 on the moving one. The walker writes its position into a held slot only where $\mathbf{p}^k_{t_0}$ lies within that distance of some $P^*(n)$; elsewhere $\hat{\mathbf{p}}^k_t = \tilde{\mathbf{p}}^k_t$. Each clip adds up to eight pseudo-occlusions of 1 to 8 steps on stretches where the slot is updated throughout; stretches over which it moves two patches or more are drawn five times as often. The walker has about 6×10^4 parameters and is trained on 1,900 clips with Adam at learning rate 10^{-3}, batch 16, for 2,000 steps of a 4,000-step schedule.

Algorithm 1 iSEE inference, one frame t (per slot k; all slots in parallel)

Require: $\mathbf{h}_t$, $\mathbf{c}$, previous state $\mathbf{S}^c_{t-1}$, held indicator u^k_{t-1}, references r^k_{t-1}, $r^{k,\mathrm{loc}}_{t-1}$, window W^k, walker state π

$\mathbf{S}^p_t \leftarrow P_\omega(\mathbf{S}^c_{t-1})$; $\tilde{\mathbf{S}}_t, \mathbf{A}_t \leftarrow C_\theta(\mathbf{h}_t, \mathbf{c}, \mathbf{S}^p_t)$, $\mathbf{A}_t$ of the first iteration ▷ Eqs. 1, 3

$e^k_t \leftarrow$ top-m mean of $\mathbf{A}^k_t$; $e^{k,\mathrm{loc}}_t \leftarrow$ largest 3×3 mean of $\mathbf{A}^k_t \odot W^k$ ▷ Eq. 4

$\hat{e}^k_t \leftarrow e^k_t / r^k_{t-1}$, $\hat{e}^{k,\mathrm{loc}}_t \leftarrow e^{k,\mathrm{loc}}_t / r^{k,\mathrm{loc}}_{t-1}$; update both references by their maximum ▷ Eq. 5

$u^k_t \leftarrow$ held indicator from $\hat{e}^{k,\mathrm{loc}}_t$, $\hat{e}^k_t$, u^k_{t-1} ▷ Eq. 6

if $u^k_t = 1$ **then** ▷ held

  $\mathbf{a}^k_t \leftarrow \mathbf{a}^k_{t-1}$ ▷ appearance kept

  $\pi \leftarrow \pi\,\mathcal{T}_t$; $\hat{\mathbf{p}}^k_t \leftarrow \mathbf{p}^k_{t_0} + P^*(\bar{\mathbf{x}}_t) - P^*(n_a)$ ▷ position walked, Eq. 7

  $\mathbf{p}^k_t \leftarrow \hat{\mathbf{p}}^k_t$ ▷ the next frame's attention, and so the release, looks for $\mathbf{a}^k$ here

else ▷ updated

  $[\mathbf{a}^k_t; \mathbf{p}^k_t] \leftarrow \tilde{\mathbf{s}}^k_t$; restart π at $\mathbf{p}^k_t$

  $W^k \leftarrow$ the slot's final-iteration attention, scaled to a maximum of one and dilated

end if

decode $\hat{\mathbf{f}}_t$ and masks from $\tilde{\mathbf{S}}_t$

return $\mathbf{S}^c_t$, which P_ω reads at $t+1$

B Setup

B.1 Implementation details

Encoder. The backbone is the checkpoint released by Pawlowsky et al. (2026): DINOv2 ViT-S/14 with registers, its learned positional embedding deleted and the network fine-tuned on COCO images to reproduce the original patch features. It is frozen at the resolution of each dataset: 336 px on MOVi-C, giving $24 \times 24 = 576$ patches, as in Manasyan et al. (2025); 518 px on LA-CATER, giving $37 \times 37 = 1369$; and 378 px on KITTI, giving $27 \times 27 = 729$. A two-layer MLP maps its 384-dimensional features to 64, as in Manasyan et al. (2025).

KITTI. Each frame is brought to 384×1280 by the affine map of Tokmakov et al. (2022) and resized uniformly to 378×1260. A training window is 8 consecutive frames at 10 fps, the recording rate, cut to one 378×378 crop starting at $x \in \{0, 294, 588, 882\}$, the crops of Gong et al. (2025); the crop is the same on every frame of the window, and the window is flipped horizontally with probability 0.5. Both encoders are trained to convergence on the 70 clips of the training half only, at batch size 32. Discovery is scored on the four crops of each validation clip, each run over the clip's 50 frames in one pass. Re-identification is scored on a fixed 189×189 crop of the frame around the occluded car, resized to 378×378.

The dataset holds only 39 occlusion–reappearance events of cars, too few to train the walker on, so we extend the pseudo-occlusions of Sec. A.3 to stand in for them entirely: a short stretch of visible frames of a slot is declared hidden, and the target is the slot's own position on the frame after it. On those frames the features of the patches the slot owns are replaced by features drawn uniformly from the patches it does not own in the same frame, so the walker sees other content where the object was, as under a real occluder, and no label enters the features, the anchor or the target.

B.2 DATASETS

LA-CATER. Clips are 300 frames of 320×240 at 24 fps. 9,300 / 3,327 / 1,371 training / validation / test clips under the static camera, and 9,299 / 2,152 / 1,371 under the moving camera of Tokmakov et al. (2022).

The occlusions we score. A scorable occlusion is a maximal run of frames on which the target has no visible pixel, at least five frames long, with three frames on each side on which the target covers at least 20 px and is owned by a resolved slot. For a target that stays in place, holding its last visible position is already the right answer, and such occlusions make up 550 of the 695 under the static camera; we therefore score every occlusion through which the target's amodal centre moves at least 10 px in the image (under the moving camera this includes camera motion), and cap the rest at 50, which still penalise a model that moves a slot whose object stayed put. The 50 are drawn once from the ground truth with a fixed seed, in proportion to the label and length (5–9, 10–29, $\geq$30 frames) of the occlusions they are drawn from, and every row is scored on the same list. Table S2 counts them; Tables 1 and 2 are read on them. The scored occlusions last 5 to 245 frames under the static camera and 5 to 240 under the moving one, with medians of 3.2 and 1.8 s.

	static camera	moving camera
test clips read	1,200	1,200
scorable occlusions	695	546
in clips	526	434
target moves ≥ 10 px while hidden, all scored	145	338
target moves less, sampled	50 of 550	50 of 208
occlusions scored	**195**	**388**
in clips	182	338

Table S2: **From test clips to scored occlusions, LA-CATER.** The first 1,200 test clips of each camera. Displacement is that of the target's amodal centre, between the first hidden frame and the furthest point of the occlusion.

KITTI. We use the 21 labelled tracking videos of KITTI (Geiger et al., 2012), split in time into training and validation halves as in Tokmakov et al. (2022) and cut into clips of 50 frames. For occlusion analysis we choose the car class and find its occlusions from the KITTI-MOTS masks (Voigtlaender et al., 2019): runs of frames on which a car's visible area stays below 15 % of its box and falls below 5 %, with the car visible before and after. We collect 39 such occlusions of 29 cars from the 21 videos, lasting 0.1 to 2.2 s.

B.3 BASELINES

RAM. RAM's released evaluation post-processes its predictions on occluded frames, and its published LA-CATER numbers include this step: for a target that was still before the occlusion, the prediction is replaced by the last visible box or by a box placed on the nearest other detection. Neither rule carries a position through the occlusion, and neither is available to a slot model, so for a fair comparison we report RAM's raw output. RAM has no slots; it returns tracks, boxes that keep an identity across frames. For SAME, the target's track on a frame is the one whose box overlaps the target's ground-truth box most, and SAME asks whether that track has the same identity at the reappearance as before the occlusion. The ground-truth box only selects which track is read and never enters a prediction.

Loci-Looped. Loci-Looped reads a background image of the scene beside the video, which its authors call the only supervised input their model receives. We build it as their released loader does, from the ground-truth instance masks: each pixel is its mean over the frames on which no object covers it. Under the moving camera no single image fits a whole clip, so we rebuild it on every frame by inpainting the ground-truth boxes of the visible objects, leaving the boxes of hidden objects untouched so that no hidden position enters the input. Loci-Looped assigns pixels to slots by a mask rather than by attention, so the slot that owns the target is the one whose mask, after depth competition, carries the largest mass on the target's visible pixels; otherwise it is read as in Sec. 4.1, on its own steps, which advance every second frame.

Checkpoints. iSEE and SlotContrast are read at step 100k, RandSF.Q at convergence (step 50k), and Loci-Looped at convergence under the static camera and, under the moving one, at 170k, the last checkpoint before its training collapsed at 171.5k without recovering. iSEE and *frozen* are read on TEN's states, SlotContrast and RandSF.Q on their own slots.

B.4 PERMANENCE METRIC

The metric is that of OPNet (Shamsian et al., 2020), averaged over IoU thresholds,

$$\mathrm{mAP@}[0.1, 0.3] = \frac{1}{5} \sum_{t \in \{0.10, 0.15, \ldots, 0.30\}} \frac{1}{|\mathcal{O}|} \sum_{o \in \mathcal{O}} \frac{1}{|F_o|} \sum_{f \in F_o} \mathbf{1}\big[\mathrm{IoU}(\hat{b}_f, b_f) > t\big],$$

with $\mathcal{O}$ the scored occlusions, F_o the hidden frames of occlusion o that carry the label being scored, $\hat{b}_f$ the predicted box and b_f the target's amodal ground-truth box. A slot model predicts a position, not a box: as in Tokmakov et al. (2022), its box has the size of the followed slot's grouping mask on the last frame before the occlusion and is centred on the predicted position; the position is read from the slot by a linear map fitted on visible frames of other clips. Loci-Looped is scored on the box of its own amodal mask and RAM on its own box. The mask is cut from the 37×37 patch grid, so on the last frame before the occlusion, where the box is on the target by construction, it overlaps the ground-truth box with IoU > 0.1 on 99.5 % (static) and 96.9 % (moving) of the occlusions but with IoU > 0.5 on only 29.7 % and 17.8 %. Above 0.3 the metric measures the grid's resolution rather than the predicted position (Fig. S1), so we average the thresholds 0.10 to 0.30.

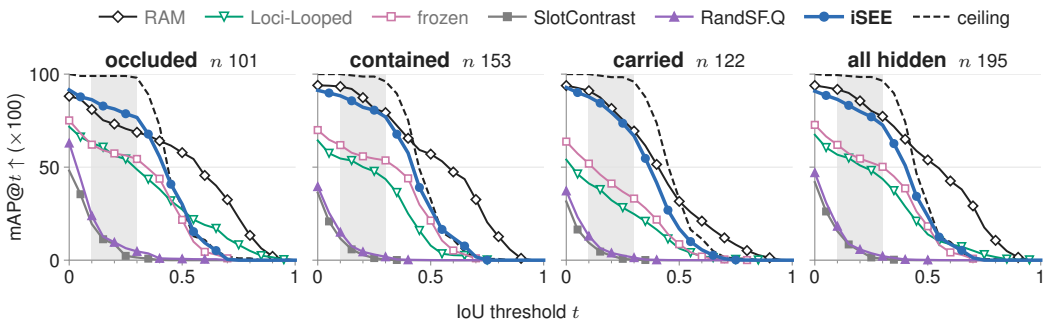

Figure S1: **Permanence against the IoU threshold, LA-CATER static camera.** mAP@t ↑ (×100). *Grey band*: the thresholds 0.10 to 0.30 that Table 1 averages. *Open markers*: rows given more than the video. *ceiling*: the share of occlusions on which iSEE's box overlaps the ground-truth box with IoU $> t$ on the last frame before the occlusion, where the box is on the target by construction. The ceiling falls above $t = 0.3$ because the box is cut from a 37×37 patch-grid mask, so beyond it the metric reads the resolution of the grid, not the predicted position.

B.5 SAME-SLOT RETURN (SAME)

Re-identification asks one question per occlusion: once the object reappears, is it back in the slot that held it before? SAME answers exactly this, once per occlusion, from two frames, the last before the occlusion and the first after it. It is the identity switch of CLEAR MOT (Bernardin & Stiefelhagen, 2008), counted across the occlusion only. Whole-clip identity metrics such as HOTA (Luiten et al., 2021) and IDF1 (Ristani et al., 2016) average over every frame of the clip. Most of those frames show the object in plain view, so an identity lost at one occlusion moves them little. They also accept a prediction only above an overlap threshold, which ties the identity score to the quality of the masks. Sec. C.2 reports HOTA.

A slot model makes the test direct:

- *Every slot is already a track.* The recurrence carries each slot from frame to frame, so slot k on frame t and slot k on frame $t + 1$ are the same slot, and no association step is needed.
- *No threshold.* Which slot holds an object is read from the slot attention itself.

- *No label while hidden.* The test needs the object's visible mask only at the two ends of the occlusion, so no box and no label enters while the object is hidden.
- *No ground-truth identities.* The slot indices need not match any ground-truth identity, because the test compares a model's slot with its own slot.

Definition. Let $A_t \in [0,1]^{K\times N}$ be the grouping attention on frame t, from the corrector's last iteration and normalised over the K slots. Let $\phi^j_{t,n} \in [0,1]$ be the fraction of patch n covered by the visible (modal) mask M^j_t of object j. The owner of j on frame t is the slot with the largest attention mass on its visible pixels,

$$\omega_t(j) = \arg\max_k \sum_{n=1}^{N} A_{t,k,n}\, \phi^j_{t,n},$$

defined when M^j_t is not empty. Assigning an object to the slot whose mask overlaps it most is standard in the slot literature. mBO gives each ground-truth mask the slot mask with the largest overlap (Seitzer et al., 2023; Zadaianchuk et al., 2023; Manasyan et al., 2025). The video forms of FG-ARI and mBO score each slot over the whole clip, so a slot that switches objects is penalised (Kipf et al., 2022; Singh et al., 2022; Zadaianchuk et al., 2023; Manasyan et al., 2025). Nguyen et al. (2026) pick the slot that tracks an object before an occlusion in the same way. Our owner is this assignment on a single frame, with the slot's attention mass on the object's visible pixels in place of the overlap of hard masks. The mass is absolute; normalising each slot's attention first would favour a small slot lying inside the object. With $\bar{M}^j_t$ the amodal mask, $v_t(j) = |M^j_t|/|\bar{M}^j_t|$ is the visible fraction. An occlusion o of object j_o is a maximal run of frames $s_o \le t < e_o$ with $|M^{j_o}_t| = 0$, lying inside the clip ($s_o > 0$, $e_o < T$). Its two ends are the last and the first frames around it on which at least half of the object is visible,

$$t^-_o = \max\{t < s_o :\ v_t(j_o) \ge \tfrac{1}{2}\}, \qquad t^+_o = \min\{t \ge e_o :\ v_t(j_o) \ge \tfrac{1}{2}\},$$

and over the set $\mathcal{O}$ of scored occlusions

$$\mathrm{SAME} = \frac{1}{|\mathcal{O}|} \sum_{o\in\mathcal{O}} \mathbf{1}\big[\omega_{t^+_o}(j_o) = \omega_{t^-_o}(j_o)\big].$$

With the same rule at both ends, the owner is read only while the object is mostly in view. A frame showing only a sliver of the object, where the owner is ambiguous, is never an end. An occlusion is scored when both ends exist and $\omega_{t^-_o}$ is defined.

Scored occlusions. On LA-CATER, $\mathcal{O}$ is the target's occlusions of Sec. B.2, each matched to the zero-visible run that overlaps it most. This scores 183 of the 195 static occlusions and 360 of the 388 moving ones. KITTI has no amodal mask, so the car's box stands in for it: t^-_o is the last frame on which the visible mask covers at least half the box, and t^+_o the first on which the visible mask has at least half the car's median visible area.

Models without slots. Loci-Looped's owner is read from its masks after depth competition, and RAM's from its tracks (Sec. B.3).

C More experiments

C.1 Permanence and the length of the occlusion

Frozen loses a quarter of its permanence from the shortest occlusions to the longest and Loci-Looped almost half, while iSEE and RAM lose less than 3 points (Table S3). iSEE's lead over *frozen*, paired per occlusion, therefore grows from $+13.3$ to $+28.3$ points.

C.2 Identity over whole clips

Table S4 scores identity over whole clips with HOTA (Luiten et al., 2021), computed with the official TrackEval implementation.

	mAP@$[0.1, 0.3]$ ↑, occlusion length (frames)			
	5–25 n 50, median 7	25–50 n 24, median 38	50–100 n 50, median 77	≥100 n 71, median 150
RAM	86.0	92.1	82.6	83.5
frozen	63.9	63.3	53.1	48.1
Loci-Looped	69.3	55.7	49.6	37.1
SlotContrast	10.8	9.4	3.2	4.1
RandSF.Q	9.8	10.0	7.1	5.9
iSEE	**77.1**	**91.7**	**81.8**	**76.4**

Table S3: **Permanence against the length of the occlusion, LA-CATER static camera.** mAP@$[0.1, 0.3]$ ↑ ($\times 100$) over all hidden frames of the 195 occlusions, split by length into four disjoint bins, with the rows and boxes of Table 1. *frozen*: the last visible position; RAM is given boxes, track identities and visibility labels, Loci-Looped a per-frame background. iSEE keeps its permanence as the occlusion lengthens, while *frozen* and Loci-Looped fall. Bold: best row that reads the video alone, which excludes *frozen*, RAM and Loci-Looped; n and median length in frames per bin in the header.

- *Tracks.* Each slot is a track; its detection on a frame is the region it wins in the argmax of the decoder's masks, restricted to the foreground as in FG-ARI.
- *Ground truth.* The modal instance masks, matched by mask IoU, so an identity kept across an occlusion is judged by the matches before and after it. *Occluded objects*: objects with no visible pixel for at least five consecutive frames that are visible again, the others joining the ignored background (our gate). *All objects*: every object of the same clips.
- TEN raises association accuracy (AssA) on both encoders and costs one to three points of detection accuracy (DetA). Without it, iSEE is below SlotContrast with the camera static and level with it with the camera moving.
- With the camera static, iSEE and SlotContrast + TEN are level on occluded objects and iSEE leads in AssA over all objects; with the camera moving, iSEE leads in AssA on both object sets.

	occluded objects			all objects		
	HOTA ↑	DetA ↑	AssA ↑	HOTA ↑	DetA ↑	AssA ↑
static camera (150 clips; 224 occluded objects in 127 clips; 1,088 objects)						
SlotContrast	71.5	83.5	61.3	87.8	92.8	83.2
+ TEN	81.3	82.6	80.0	89.5	92.0	87.1
RandSF.Q	42.3	61.8	29.0	35.1	48.1	25.9
Loci-Looped	54.8	52.7	57.2	70.3	68.8	72.2
iSEE without TEN	68.0	82.3	56.2	85.4	92.1	79.3
iSEE	**82.1**	81.2	**82.9**	**90.4**	91.0	**89.9**
moving camera (150 clips; 226 occluded objects in 132 clips; 1,112 objects)						
SlotContrast	63.8	84.5	48.1	77.4	90.9	65.9
+ TEN	68.6	82.5	57.1	80.0	89.7	71.4
RandSF.Q	42.0	56.4	31.4	26.6	28.8	25.0
Loci-Looped	37.5	38.3	37.6	38.5	40.6	37.6
iSEE without TEN	63.6	82.1	49.2	76.0	88.1	65.6
iSEE	**75.3**	79.1	**71.7**	**81.3**	85.0	**77.8**

Table S4: **Identity over whole clips (TrackEval).** HOTA, DetA and AssA ↑ ($\times 100$) on the first 150 test clips of each camera, protocol above. RandSF.Q: its decoder's masks (last-layer cross-attention, head-averaged, softmax over slots). Loci-Looped: the argmax over its decoded masks with its background channel as one more track, at every second frame, at the checkpoints of Sec. B.3.

C.3 Separation of appearance and position

Separation is the *within-object* R^2 ↑ of a two-layer MLP that predicts an object's centre from a block of its slot after the per-object mean is removed, so that the probe has to track motion rather than recognise the object.

In SlotContrast the position is carried by the whole 64-dimensional slot, the block its contrastive term also acts on (Table S5); in iSEE it is carried by the 4-dimensional position stream, which reads more

Table S5: **Position is carried by the position stream, MOVi-C.** Within-object R^2 of an object's modal centre. Feature rows are each model's own frozen backbone pooled over the object's cells, and are the floor for the code that reads them. Position is carried by the 4-dimensional position stream, and the appearance stream reads it below the floor.

code	dim	within-object R^2
DINOv2 (Oquab et al., 2024)	384	0.833
DINOv2 NoPE (Pawlowsky et al., 2026)	384	0.552
SlotContrast slot	64	0.597
iSEE appearance $\mathbf{a}$	60	0.525
iSEE position $\mathbf{p}$	4	0.727

Table S6: **Contrastive loss on the appearance stream or on the whole slot.** SAME, discovery, permanence mAP and the within-object R^2 of the position stream for iSEE trained with the contrastive loss on the appearance stream only (60 dimensions) or on the whole slot (64 dimensions), LA-CATER, static camera.

contrastive loss on	SAME ↑	FG-ARI ↑	mBO ↑	mAP all ↑	mAP hidden ↑	R^2 of $\mathbf{p}$ ↑
appearance stream (60)	82.0	96.6	23.9	69.0	30.1	0.937
whole slot (64)	83.1	94.7	28.5	25.6	13.0	0.693

of it than the model's 384 input features, computed without a positional encoding. The appearance stream's 0.525 is the position already present in the DINOv2 NoPE features it reads. The probe is a read-out, so what it separates is not whether a code holds the position but how easily that position can be used, which is what Sec. 4.4 does with it: given each slot's position delta as an action, the dynamics model on iSEE's slots gains 47 points of headroom and the one on SlotContrast's, given its whole slot delta, five.

C.4 Contrastive loss on the whole slot

With the contrastive loss on the whole slot, the other scores stay similar, but the position encoded in the position stream degrades, and the permanence mAP, which is read from it, suffers as well (Table S6). The loss requires the same position to have a different code across the sequence and across the batch, which a position code cannot satisfy.

C.5 Appearance input of the position predictor

Without $\mathrm{sg}(\mathbf{a})$ the position predictor loses permanence mAP (Table S7), as expected, since how far an object moves depends on its shape.

D Dynamics model

We train one OCVP-Par predictor (Villar-Corrales et al., 2023) per encoder, with $L = 4$ blocks of width 128 and 1.09M parameters including the action path. It reads a 14-frame window of the encoder's frozen slots and predicts the slots of the next frame; in a rollout, each prediction is appended to the window as its newest frame. The predictor is trained on cached slots for 30k steps at seed 0, batch 128, learning rate 2×10^{-4} and dropout 0.2. The action of slot k is its one-step position change $\mathbf{p}^k_{t+1} - \mathbf{p}^k_t$, divided by its r.m.s. over the cache, mapped by a two-layer MLP to a two-dimensional code and injected into every block by adaptive layer normalisation. The loss is a squared error in which the four dimensions of $\mathbf{p}^k$ are weighted by $\lambda = 88.7$ and the sixty of $\mathbf{a}^k$ by 1; for SlotContrast, which has no position stream, it is unweighted.

Table S7: **Appearance input of the position predictor, LA-CATER, static camera.** Permanence mAP@$[0.1, 0.3]$ over all frames of the 195 occlusions, with TEN and without the walker.

position predictor	mAP all ↑
with $\mathrm{sg}(\mathbf{a})$	69.0
without $\mathrm{sg}(\mathbf{a})$	66.4

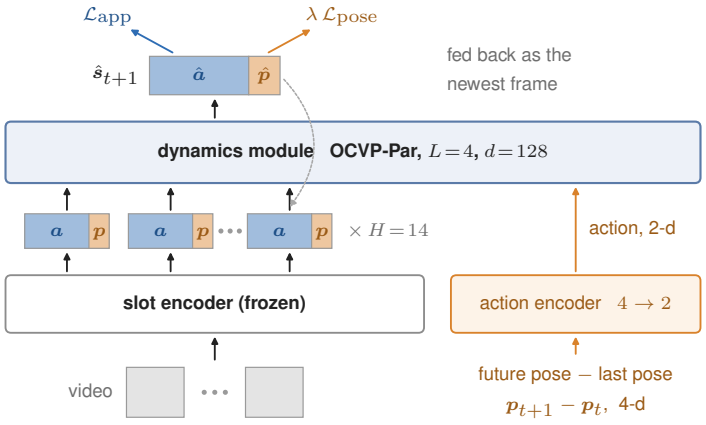

Figure S2: **The action-conditioned dynamics model.** Frozen slots $[\mathbf{a}^k; \mathbf{p}^k]$ enter OCVP-Par (Villar-Corrales et al., 2023); per slot, the position change $\mathbf{p}^k_{t+1} - \mathbf{p}^k_t$ passes a two-dimensional bottleneck and is injected into every block by adaptive layer normalisation. The action reaches the predictor only through the position stream.

E QUALITATIVE EXAMPLES

E.1 OBJECT DISCOVERY

Figures S3 and S4 draw the decoder-mask argmax of SlotContrast and iSEE on MOVi-C and on KITTI; Table 3 is the population these two clips are drawn from.

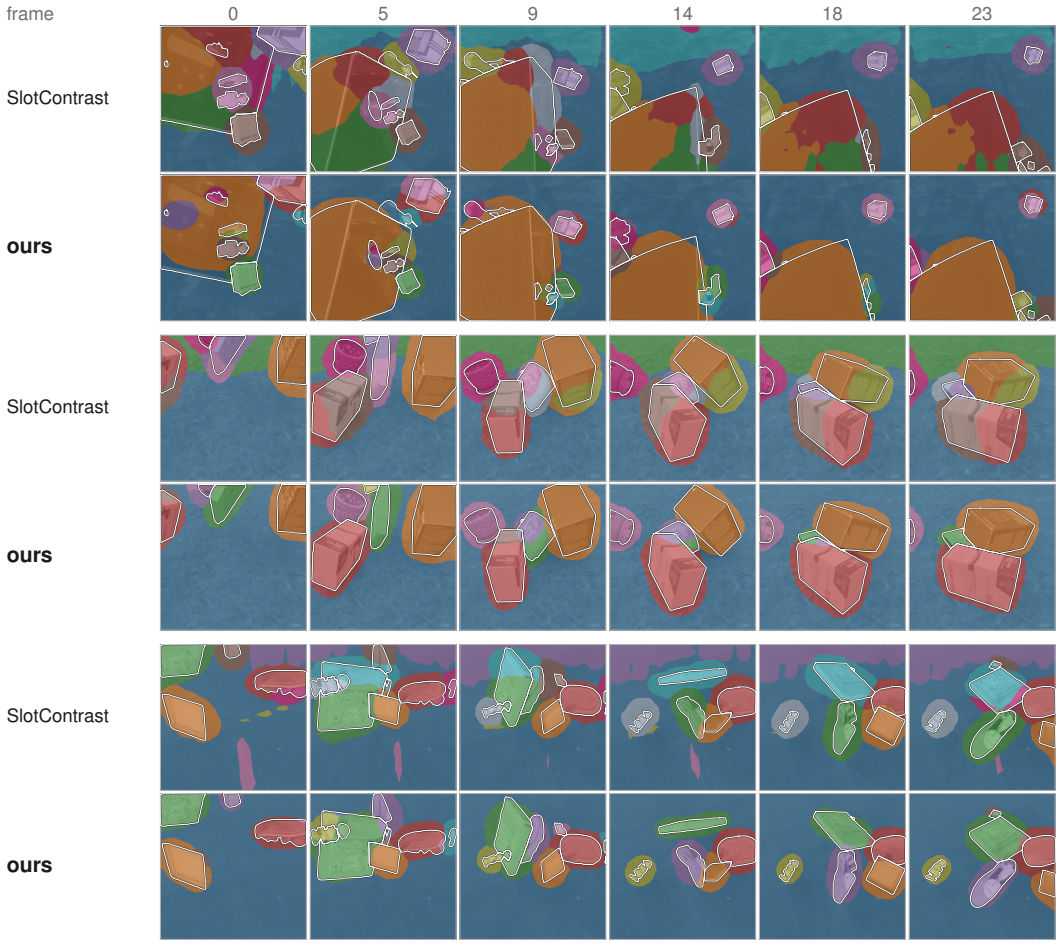

Figure S3: **Object discovery on MOVi-C.**

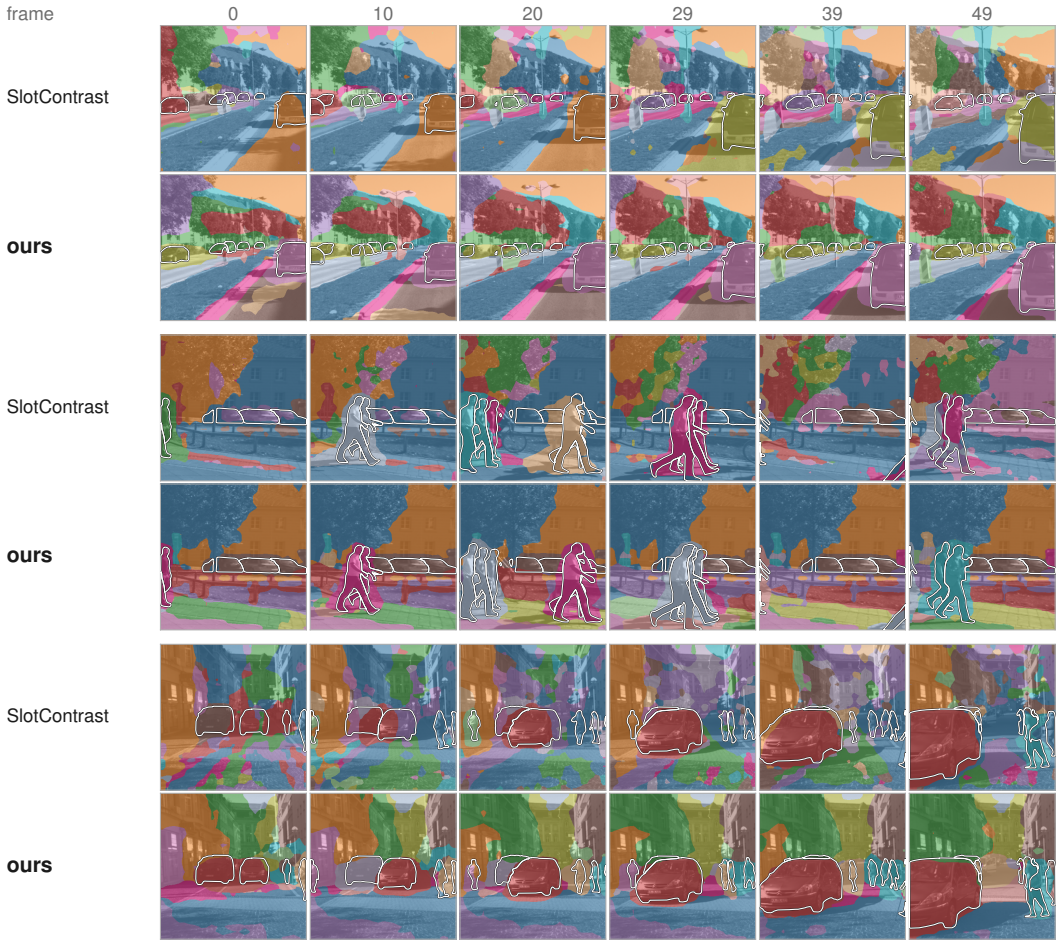

Figure S4: **Object discovery on KITTI.**

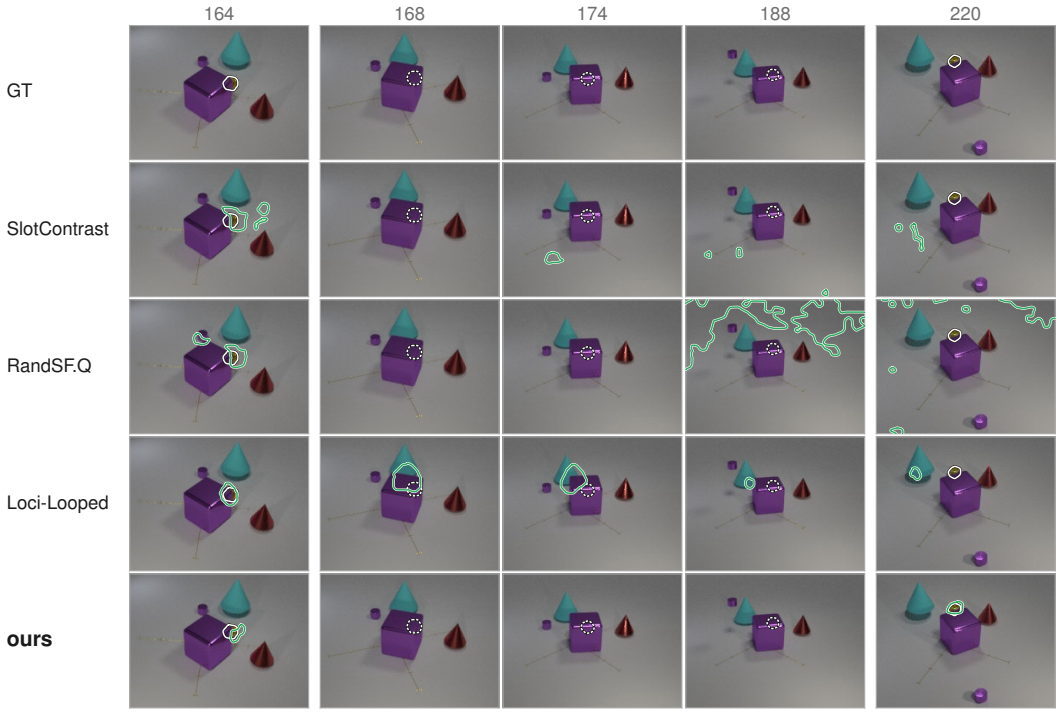

Figure S5: **Re-identification: one occlusion in a LA-CATER scene, the target hidden behind another object and reappearing.** Columns are frames, the first the last one on which the target is visible and the last the frame Table 2 scores, in one crop of the frame that is the same throughout. *GT*: the target's true (amodal) mask, in white on every row, dashed on the frames where none of it is visible. Green: the mask of the slot that owns the target in the first column, empty where that slot claims nothing in view; iSEE's claims nothing anywhere while the target is hidden, because TEN holds it. Only iSEE has that slot back on the target when it reappears.

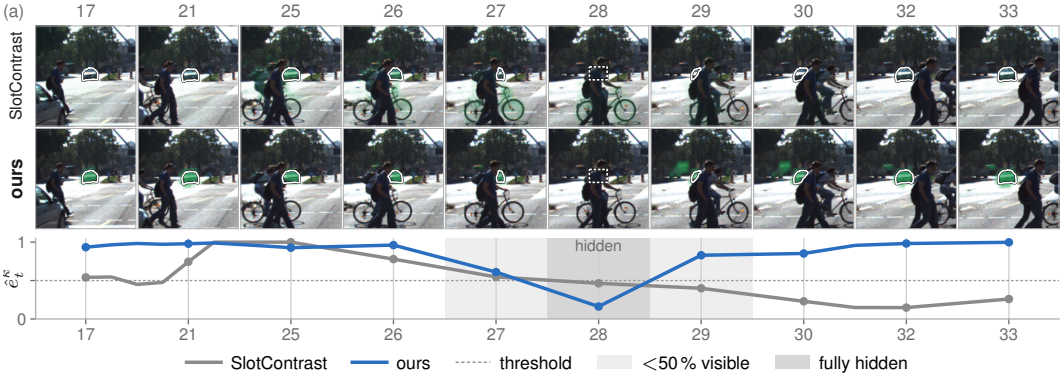

Figure S6: **A qualitative example on KITTI.** Columns are frames of one tile, unevenly spaced; the whole 2× tile the model sees. *GT*: the car's visible mask in white, its KITTI box dashed where none of it is visible. Green: the slot on the car before the occlusion. iSEE's evidence falls while the car is hidden and its slot is back on the car when it reappears; SlotContrast's slot moves onto the occluding cyclist.

E.2 Re-identification on LA-CATER

Figure S5 follows one target through an occlusion and back out, and asks whether the slot that held it before still holds it after; Table 2 is the population.

E.3 OBJECT PERMANENCE ON LA-CATER

Figures S7 and S8 draw one occlusion each in the form of Figure 4, in a second static-camera scene and on the moving-camera split. Each is the occlusion with the largest margin of iSEE over the other label-free rows of Table 1, so both are best cases.

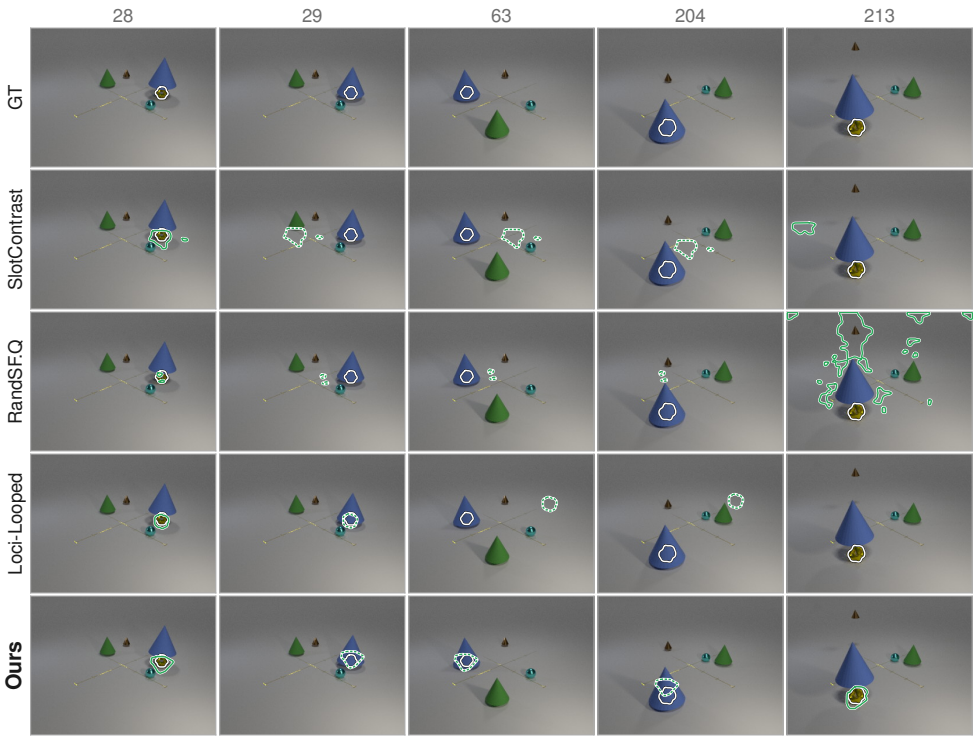

Figure S7: **Qualitative comparative example on LA-CATER, static camera**, against SlotContrast, RandSF.Q and Loci-Looped.

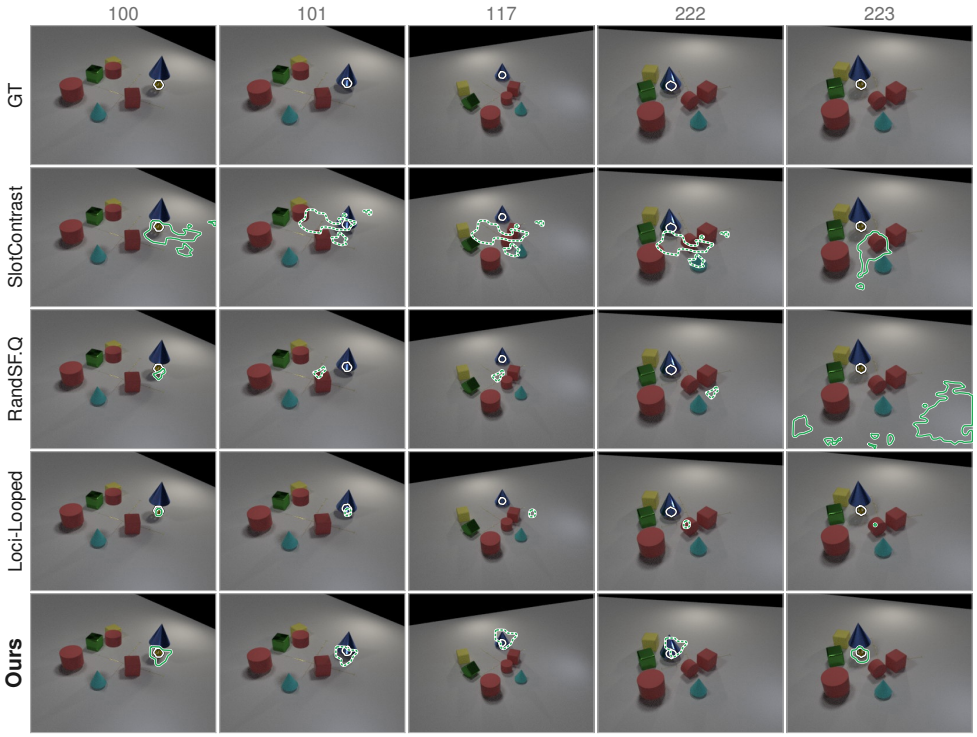

Figure S8: **Qualitative comparative example on LA-CATER, moving camera**, against SlotContrast, RandSF.Q and Loci-Looped.

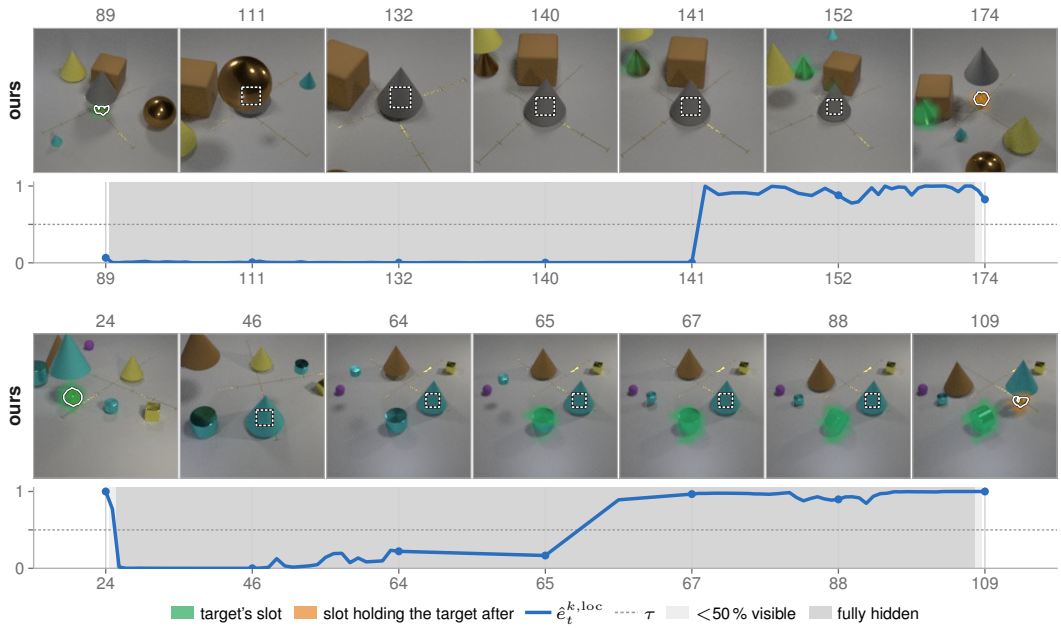

Figure S9: **TEN releases the slot while the target is still hidden.** Columns are frames of one clip, unevenly spaced, in one crop held fixed through the sequence; the first is the last frame on which the target is at least half visible. *GT*: the target's visible mask in white, its amodal box dashed where none of it is visible. Green: the slot on the target before the occlusion; orange: the slot holding the target after.

E.4 Failure cases

E.4.1 Re-identification

Figure S9 draws two moving-camera LA-CATER occlusions on which TEN releases the slot while the target is still hidden, and the slot takes another object. Figure S10 draws two ways iSEE loses a car through a KITTI occlusion.

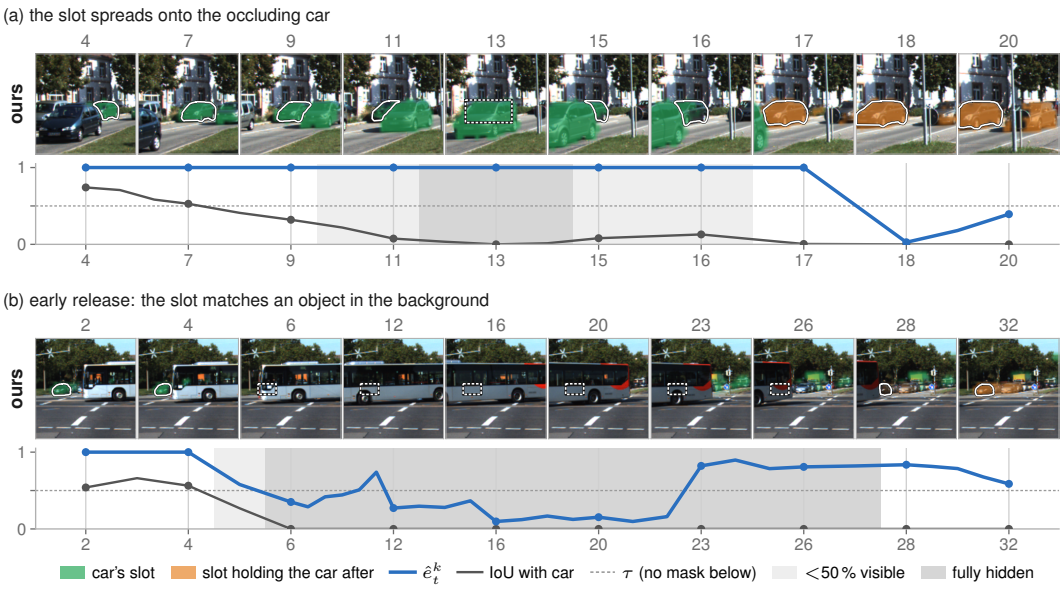

Figure S10: **Two ways iSEE loses a car through a KITTI occlusion.** Columns are frames of one tile, unevenly spaced. *GT*: the car's visible mask in white, its KITTI box dashed where none of it is visible. Green: the slot on the car before the occlusion.

E.4.2 OBJECT PERMANENCE

Figure S11 draws one LA-CATER occlusion for each of five ways iSEE's predicted position fails: it moves to the wrong place (*wrong way*), it moves while the target stays still (*phantom*), the slot's box is too small to reach IoU > 0.1 (*box size*), the slot is not on the target when it goes in (*wrong slot*), or it stays still while the target moves (*frozen*). Each row is the median occlusion of its mode, not the worst.

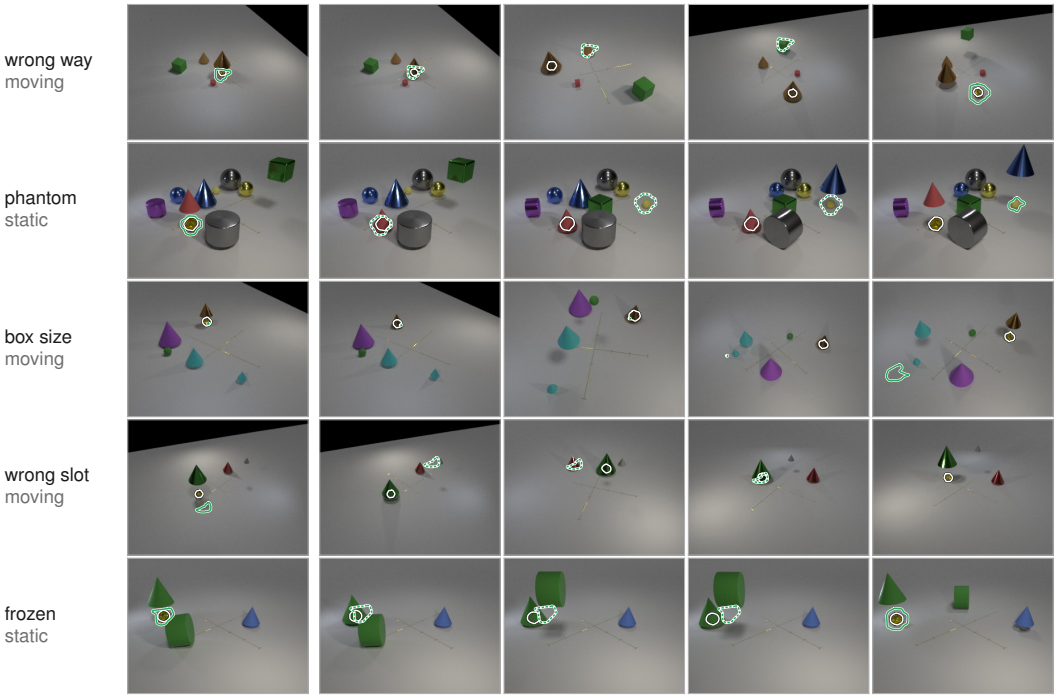

Figure S11: **Five ways iSEE loses the target through a LA-CATER occlusion.** *GT*: the target's true (amodal) mask, in white. Green: the slot iSEE follows, *solid* where it is that slot's own mask and *dashed* where that outline is moved onto the predicted position; no green where the slot owns no pixel.