HOLD THE EVIDENCE, THEN LOOK BACK: RESOLVING IDENTITY CONFLICTS IN MEMORY-BASED VISUAL OBJECT TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory-based visual object trackers built on promptable segmentation models still confuse identities when similar objects overlap, occlude one another, or return after long absences. These trackers typically decide such conflicts in the frame where they appear, although the overlap that reveals a conflict usually follows the identity error and the evidence that resolves it arrives during and after the conflict. We present PCIA-Track, a training-free, plug-in framework built on one principle: keep the evidence during a conflict and decide the identity after it. Specifically, *Hold-and-Report* (HaR) holds the memory of a conflicting track but keeps reporting its prediction, and relocates the target among class-agnostic proposals when that prediction duplicates another object. *Post-conflict identity arbitration* (PCIA) waits until the tracks separate, replays the buffered video backward from two fresh, memory-free probes, and hands the identity over only when the candidate, and not the incumbent, reconnects to the incumbent's last stable window. A small-target confidence floor (SCF) removes spurious outputs of vanished small targets. PCIA-Track improves on SAM 3 on LVOSv2, LVOSv1, VOST, DiDi and LaSOT$_{\mathrm{ext}}$, where it is the best of the compared trackers, and reaches a quality score of $Q = 0.735$ on VOTS2026, above the best entry of the challenge. It keeps the model weights and the architecture unchanged.

1 INTRODUCTION

Visual object tracking asks a model to follow the targets given in the first frame through the rest of a video. Recent benchmarks make this harder in two ways. Their scenes contain many small and similar targets that disappear and reappear (Ding et al., 2023; 2025a; Kristan et al., 2026), and their videos last minutes rather than seconds (Hong et al., 2026). Trackers built on the promptable segmentation models SAM 2 and SAM 3 (Ravi et al., 2025; Carion et al., 2026) now lead these benchmarks (Ding et al., 2025b; Videnović et al., 2025; Alansari et al., 2026). They localize each target by attending to a memory of earlier frames and their predicted masks, which makes them robust to appearance change but vulnerable to one specific failure: when two similar targets meet, the memory of one target can absorb the other, and after the targets separate the identity continues on the wrong one.

A tracker therefore needs a protocol for conflicts between its tracks, whether they belong to different targets or to one target and a look-alike. Memory-based trackers typically settle a conflict in the frame where it appears, from the evidence of that frame (Ravi et al., 2025; Kristan et al., 2026). Our analysis of conflicts (Section 3.2) shows that this evidence is incomplete: the overlap that reveals a conflict usually arrives after the identity error has happened, the conflicting track's own prediction can return to the target later in the conflict, and the correct track frequently survives among the tracker's own candidate tracks.

We therefore separate keeping evidence from deciding identity (Figure 2). During a conflict, *Hold-and-Report* (HaR) holds the memory of the lower-scoring track, which keeps the other object out of it, and keeps reporting the track's prediction, which carries the evidence; when that prediction duplicates another target, HaR relocates the target among class-agnostic mask proposals, scored by the track's own memory. After the conflict, *post-conflict identity arbitration* (PCIA) seeds two fresh

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.

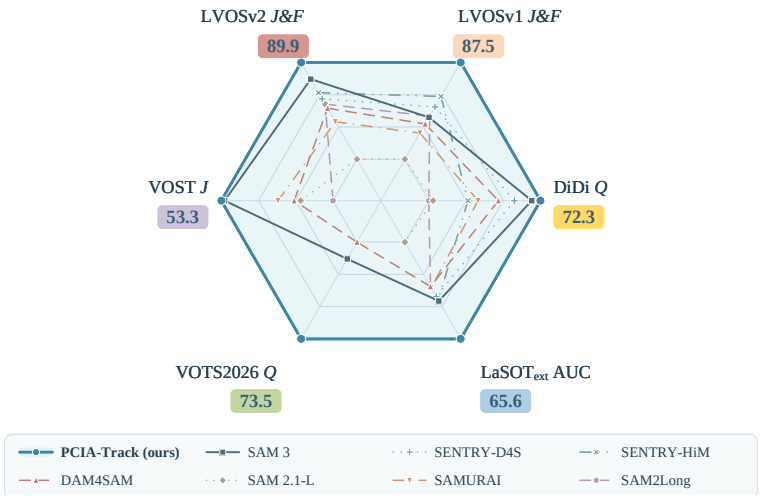

Figure 1: **PCIA-Track and recent trackers on six benchmarks.** Each axis is scaled to the range of the methods shown, from the lowest (inner ring) to the best score (outer ring); the numbers give the score of PCIA-Track.

probes with the masks of the competing tracks and replays the buffered frames backward to the incumbent's last stable window. PCIA hands the identity to the candidate only when the candidate's probe reconnects to that window and the incumbent's does not. Because the probes start from masks alone, they are independent of the memory that caused the confusion.

Together with an absolute confidence floor for small outputs (SCF), these components form PCIA-Track, a training-free tracker built on SAM 3. It improves on SAM 3, whose checkpoint it uses, on the multi-object benchmarks LVOSv2, LVOSv1 and VOST and on the single-object benchmarks DiDi and LaSOT$_{\text{ext}}$. On VOTS2026 it reaches $Q = 0.735$, above the best challenge entry, without a vision–language model (Figure 1, Table 3).

Our contributions are:

- We analyze identity conflicts in memory-based trackers and show that they are detected after the identity error, that the conflicting track's prediction carries the target during the conflict, and that the correct track often survives among the tracker's own candidate tracks; on this basis we propose Hold-and-Report, which holds the memory of a conflicting track, keeps reporting its prediction, and relocates the target when that prediction duplicates another one.
- We propose post-conflict identity arbitration, which decides between two competing tracks by replaying the video backward from fresh, memory-free probes to the incumbent's last stable window, and changes identity only on one-sided evidence.
- We build PCIA-Track, a training-free tracker that improves on SAM 3 on five multi-object and single-object benchmarks and exceeds the best VOTS2026 entry; we measure what each component contributes by removing it from the full system, and trace the corrections to individual sequences and frames.

2 Related Work

Memory-based visual object tracking. Memory networks maintain temporal coherence under appearance change and occlusion in both video object segmentation and tracking. STM introduced key–value memory reads over earlier frames and masks (Oh et al., 2019). Later work refined how memory is read and organized: joint propagation of several identities with transformers in AOT and DeAOT (Yang et al., 2021; Yang & Yang, 2022), sensory, working and long-term stores in XMem (Cheng & Schwing, 2022), restricted memory banks in RMem (Zhou et al., 2024), object-level queries in Cutie (Cheng et al., 2024), and spatial-semantic features with discriminative queries

in S3 (Li et al., 2025). The same idea underlies memory-augmented trackers such as STMTrack (Fu et al., 2021), RTracker (Huang et al., 2024) and MeMOT (Cai et al., 2022). Across these designs, each frame's prediction and memory update are decided from the evidence of that frame; when two targets compete, the winning hypothesis enters memory and the decision stands. Our work acts at this point: it keeps the evidence of a conflict and revisits the identity once the conflict is over.

SAM 2 and SAM 3 based trackers. SAM 2 turned promptable segmentation into a streaming tracker with a memory of recent frames (Ravi et al., 2025), and SAM 3 adds concept prompts and a stronger detector–tracker pair (Carion et al., 2026). Training-free extensions change how this memory is written or read: SAMURAI adds motion-aware memory selection (Yang et al., 2026), DAM4SAM keeps distractor-resolving frames next to recent appearance (Videnović et al., 2025; 2026), SAM2Long keeps a tree of memory pathways (Ding et al., 2025b), SAMITE calibrates memory with positional prompts (Xu et al., 2025), and HiM2SAM combines hierarchical motion estimation with memory optimization (Chen et al., 2026). SeC adds object-level concepts built by a vision–language model (Zhang et al., 2026), and SENTRY verifies candidate masks before they are written to memory (Alansari et al., 2026). These methods improve the prediction chosen in each frame and the frames that enter memory, and most of them settle conflicts between tracks frame by frame. PCIA-Track leaves the memory design unchanged and adds a protocol for conflicts between tracks, without retraining or changing the architecture.

Resolving visual ambiguity and identity preservation. Keeping the identity of a target among similar objects is a long-standing problem in tracking. DaSiamRPN makes Siamese tracking distractor-aware in both training and inference (Zhu et al., 2018), KeepTrack associates target candidates across frames to keep track of what not to track (Mayer et al., 2021), and Siam R-CNN re-detects the target and links tracklets to recover it after occlusion (Voigtlaender et al., 2020). In memory-based trackers, DAM4SAM stores frames that resolve distractors (Videnović et al., 2025), and recent challenge trackers add unreported candidate tracks that follow likely look-alikes, together with pairwise checks between overlapping tracks (Kristan et al., 2026). These mechanisms commit to an identity at the moment of ambiguity, from the evidence available then; hypotheses that lose are suppressed, so an early wrong decision is hard to undo. Tracking backward in time is a classical way to validate a prediction: forward–backward error detects tracking failures by tracking back to the start point (Kalal et al., 2010), backward model validation checks appearance-model updates against earlier frames (Yuan et al., 2014), and cycle consistency in time serves as a training signal for correspondence (Wang et al., 2019). SENTRY brings backward verification to SAM-based trackers: it re-tracks each candidate mask over a short backward window and compares the resulting tracklets with the target and its neighbors before admitting the mask to memory (Alansari et al., 2026). Used as checks, these methods accept or reject the prediction or the memory update of a single track over a short window. PCIA uses backward replay for a different decision: rather than validating a single track, it compares two competing tracks after they separate, from fresh probes that carry no memory, against the incumbent's last stable window before the conflict, and changes the identity only on one-sided evidence.

3 METHOD

3.1 BASE TRACKER AND CONFLICT DETECTION

PCIA-Track tracks every initialized target with SAM 3 (Carion et al., 2026) and a DAM4SAM-style memory of recent-appearance and distractor-resolving frames (Videnović et al., 2025); our implementation starts from the released code of SAFEDAM, a VOTS2026 challenge tracker (Kristan et al., 2026). A frame enters memory only if its score passes an adaptive gate, a threshold that follows each track's recent scores. When a target's memory records a distractor-resolving frame, the tracker can spawn a *candidate track* on an alternative region that the SAM 3 detector confirms. A candidate track is tracked like a target but never reported; it is an explicit hypothesis for a nearby look-alike.

PCIA-Track detects conflicts between tracks with a pairwise *conflict check*. For two tracks i and j with masks M_i and M_j, let C_i be the largest connected component of M_i. The overlap ratio

$$\rho_{i|j} = \frac{|C_i \cap C_j|}{|C_i|} \tag{1}$$

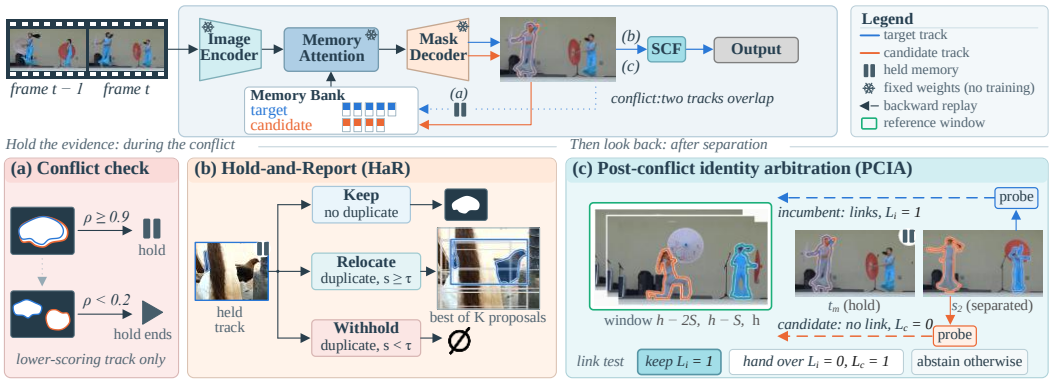

Figure 2: **Overview of PCIA-Track.** Blue: the target track (incumbent); orange: a candidate track; every track has its own memory slots and mask. (a) The conflict check holds the memory of the lower-scoring of two overlapping tracks; (b) HaR keeps reporting that track's prediction and relocates it if the prediction duplicates another target; (c) after the tracks separate, PCIA replays two fresh probes backward to the reference window and hands the identity over only if the candidate's probe alone links ((c) shows a keep; Figure 4 shows a hand-over). Frames are from LVOSv2 val with our output masks; (a) and the kept output in (b) are schematic.

measures how much of track i lies inside track j. Whenever two targets, or two tracks of the same target, overlap, the conflict check considers the one with the lower object score. When its largest ratio reaches 0.9, the check holds that track's memory, which is not updated until the ratio falls below 0.2. The rule comes from the released challenge code, which also empties the outputs of held tracks and has no post-conflict arbitration; PCIA-Track keeps reporting them (Section 3.3) and arbitrates after each conflict (Section 3.4).

3.2 Observations on Identity Conflicts

We ask what evidence a conflict leaves and when the identity can be decided. Three observations shape our protocol.

A conflict is detected after the identity error. When the conflict check holds a track, the track has already moved onto the overlapping object: the overlap that reveals the conflict is the error itself, so the frame of detection cannot tell which track is on the target (Figure 3a).

The held track's prediction carries the target. Holding the memory at detection keeps the other object out of the target's memory. With the memory held at a clean state, a held track's prediction can return to the target: keeping the other object out of the memory gives the held prediction a chance to recover. Reporting these predictions instead of nothing reduces the loss, and most of the reduction comes from the frames after the hold onset, once the prediction has had time to recover (Figure 3b).

The correct track often exists. In many failure segments, one of the tracker's own candidate tracks covers the true target while the target track is elsewhere. The identity can therefore be recovered after the conflict, by deciding which of the two tracks deserves it.

Hence we hold and report during the conflict (HaR) and decide the identity after it (PCIA).

3.3 Hold-and-Report

During a conflict, HaR decides in three steps what the held track f reports.

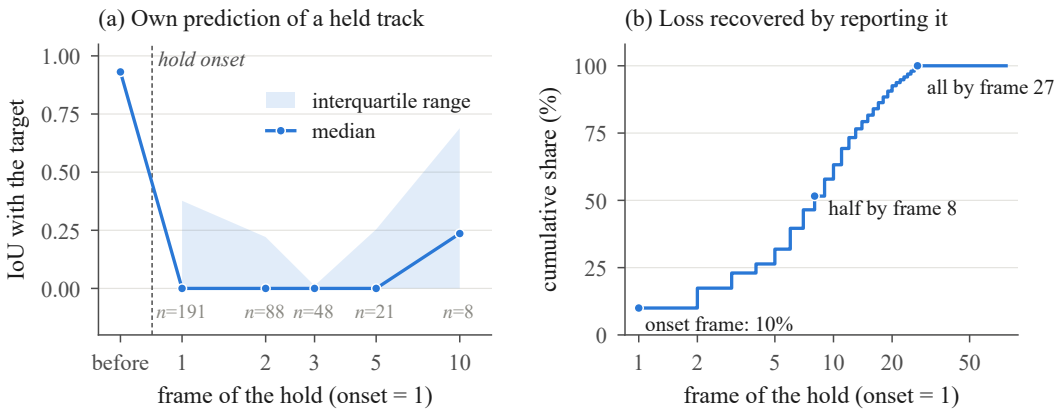

Figure 3: **Conflicts on LVOSv2.** (a) IoU between a held track's own prediction and the target in the 191 hold episodes starting on it, by frame of the hold; shaded band: interquartile range (25th–75th percentile); n: episodes lasting at least that many frames. (b) Cumulative share of the frame-level loss removed by reporting the predictions of held tracks (duplicates withheld) instead of nothing: half within the first eight frames of a hold, all within 27.

Keep. HaR reports f's own prediction M_f unless it duplicates another initialized target p (candidate tracks are excluded), that is, unless $\max_p \rho_{f|p} \geq 0.9$ with the ratio of Eq. (1), the threshold at which the conflict check holds a track. All overlaps use the predictions before any output is withheld.

Relocate. A duplicate means that f's prediction sits on another target, while f's own target may still be visible elsewhere. If f's object-score logit is at least τ, HaR searches for the target among class-agnostic mask proposals. It prompts the tracker's own SAM 3 image model with a regular 32×32 grid of points, as in automatic mask generation (Kirillov et al., 2023), keeps the K proposals with the highest predicted IoU among those nearest to f's last non-empty output, scores each of them with f's memory, and reports the highest-scoring proposal.

Withhold. Only when f's object-score logit is below τ does HaR report an empty mask.

HaR never writes to memory: a kept or relocated mask is an output, and f's memory stays held until the hold ends. We use $K = 30$ and $\tau = 2.1094$ on all benchmarks; both were selected on LVOSv2 by a rule fixed before the evaluation (Appendix A).

3.4 Post-conflict identity arbitration

HaR keeps the evidence of a conflict but leaves its outcome to the tracker: after two tracks separate, the identity may continue on the wrong one. PCIA decides this outcome after the fact. It considers a conflict between an *incumbent* i, the track that carries a target identity, and a *candidate* c, one of the candidate tracks, and asks which of the two current masks leads back to the target's last stable state (Figure 2).

Trigger and seed. A conflict starts when the conflict check holds i or c; we denote the first held frame by t_m. PCIA then watches the pair on frames sampled every S frames. The pair has separated once both masks are non-empty and disjoint on two consecutive sampled frames, and the next sampled frame on which they are still disjoint is the seed frame s_2. One conflict yields at most one arbitration; a new conflict of the same pair requires two hold-free sampled steps in between.

Reference window. The reference is the incumbent's last stable period before the conflict. PCIA records *discontinuities* of the incumbent on sampled frames: an empty prediction, an IoU below 0.3 between consecutive sampled masks, a centroid displacement larger than the square root of the mask area, a new distractor-resolving memory frame, or a hold. Scanning backward from t_m, it selects the most recent discontinuity e that is preceded by at least K_{w} discontinuity-free sampled steps and

places the window end at $h = e - S$. The window consists of the incumbent's cached predictions at h, $h - S$ and $h - 2S$. If the incumbent has no discontinuity in its history, $h = t_m - S$; if no window qualifies or it lies more than B sampled steps before s_2, PCIA abstains.

Backward probes and the link test. At s_2, PCIA creates two probes: fresh instances of the same tracker, one initialized with the incumbent's mask and one with the candidate's. Neither probe inherits the memory of either forward track. Both probes process the buffered frames $s_2 - S, s_2 - 2S, \ldots, h - 2S$ in reverse order. A probe $j \in \{i, c\}$ *links* to the reference window if its mask $\overleftarrow{M}_{j,t}$ overlaps the incumbent's cached prediction $M_{i,t}$ on at least two of the three window frames:

$$L_j = \mathbf{1}\Big[\sum_{k=0}^{2} \mathbf{1}\big[\,\mathrm{IoU}(\overleftarrow{M}_{j,h-kS},\, M_{i,h-kS}) \geq 0.5\big] \geq 2\Big]. \tag{2}$$

Decision. If $L_i = 1$, the incumbent traces back to its own stable window, and PCIA keeps it, whatever the candidate does. This covers the frequent case in which the incumbent is still, or again, on the object of its stable window when the tracks separate. If $L_i = 0$ and $L_c = 1$, only the candidate traces back to the target, and PCIA *hands over* the identity: the candidate's complete forward state, including its memory, replaces the incumbent's, and the candidate is removed from the pool of candidate tracks. In all other cases PCIA abstains. The decision is deliberately asymmetric: a hand-over requires one-sided evidence, and ambiguous evidence leaves the trajectory unchanged.

3.5 Small-target confidence floor

When a small target leaves the view, a tracker often keeps reporting a small spurious mask. SAM 3 empties an output only when its object score falls below zero, and the base tracker's gates compare each score with the track's own recent scores, so they follow a declining score downward. The raw score itself remains informative: on LVOSv2, the median object-score logit on such frames is 1.67 for targets smaller than 16 SAM 3 tokens, compared with 7.84 on correctly tracked frames (AUC 0.95). SCF therefore empties an output whose mask covers fewer than 16 tokens and whose object-score logit is below 0.5. It changes only the reported mask, never the memory.

Implementation. PCIA-Track uses the released base code and SAM 3 weights without retraining; all parameters are listed in Appendix A.

4 Experiments

4.1 Setup

Benchmarks. We evaluate on multi-object and single-object benchmarks with similar objects, occlusion and long videos. LVOSv2 (Hong et al., 2026) (140 validation videos) contains long multi-object videos with frequent occlusion; we use its official evaluator and report $\mathcal{J}\&\mathcal{F}$ over seen and unseen categories. Its first version, LVOSv1 (Hong et al., 2023), has 50 validation videos identical to their LVOSv2 counterparts, evaluated with the v1 release of the official evaluator (Appendix A). VOST (Tokmakov et al., 2023) (70 validation videos, 114 objects) contains objects that undergo strong transformations; we report region similarity $\mathcal{J}$ and its value $\mathcal{J}_{\mathrm{tr}}$ on the last quarter of each video, computed with the official code as released. VOTS2026 (Kristan et al., 2026) is scored by its server with quality Q, accuracy (A) and robustness (R). DiDi (Videnović et al., 2025) (180 sequences) collects single-object videos with distractors and is scored with the same Q, A and R in the VOT toolkit. LaSOT$_{\mathrm{ext}}$ (Fan et al., 2021) (150 test videos) contains long single-object videos; we report success AUC, normalized precision and precision.

Protocol. PCIA-Track receives the target masks of the first annotated frame and tracks these targets jointly, without training; Appendix A gives the single-object protocol, where each video has one target and relocation never runs. We measure the contribution of each component by removing it from the full system. We run SAM 3 with its released checkpoint and the evaluation protocols of Appendix A; this gives 88.3 on LVOSv2 val and 91.5–91.6 on DAVIS 2017 (Pont-Tuset et al., 2017) val, against 88.5 and 92.2 reported.

Table 1: **Comparison on multi-object and single-object tracking benchmarks.** $\mathcal{J}\&\mathcal{F}$ on LVOSv2 val and LVOSv1 val; $\mathcal{J}$ on VOST val; quality Q on DiDi; success AUC on $\text{LaSOT}_{\text{ext}}$; all scores $\times 100$. Values of other methods are copied from the sources in Table 5, except SAM 3, which we run with its released checkpoint (Section 4.1); SAM 2.1-L is the released checkpoint except on VOST (†), where the source does not state the model size, and LVOSv2 and LVOSv1 values of rows marked * are reported by Alansari et al. (2026, arXiv version, Table 25), where SENTRY-HiM and SENTRY-D4S run SENTRY on HiM2SAM and DAM4SAM. †: reported by a third party. VOTS2026 is compared in Table 3; –: not reported or not run.

	Multi-object			Single-object	
	LVOSv2	LVOSv1	VOST	DiDi	$\text{LaSOT}_{\text{ext}}$
Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	Q	AUC
SAM 2.1-L (Ravi et al., 2025)	80.6	80.1	48.4†	64.9†	56.9†
S3 (Li et al., 2025)	83.7	76.0	–	–	–
SAMURAI (Yang et al., 2026)	84.2†	82.1†	49.8†	68.0†	61.0
DAM4SAM (Videnović et al., 2025)	85.5	82.8	48.8†	69.4	60.9
SAMITE* (Xu et al., 2025)	85.8	83.2	–	68.4†	62.1
SAM2Long (Ding et al., 2025b)	85.9	83.4	46.4†	64.6†	60.9
HiM2SAM* (Chen et al., 2026)	86.1	83.5	–	67.0†	62.8
SENTRY-D4S* (Alansari et al., 2026)	86.4	84.1	–	70.5	61.8
SeC (Zhang et al., 2026)	86.5	–	–	–	60.3†
SENTRY-HiM* (Alansari et al., 2026)	87.0	84.9	–	67.3	62.1
SAM 3 (Carion et al., 2026)	88.3	83.3	53.1	71.7	62.2
PCIA-Track (ours)	89.9	87.5	53.3	72.3	65.6

4.2 Main results

Table 1 compares PCIA-Track with recent memory-based trackers; Appendix C gives the LVOSv2 results by category and all measures on the single-object benchmarks. PCIA-Track obtains the best score of the compared trackers on every benchmark of Table 1: it exceeds SAM 3 by 1.6 $\mathcal{J}\&\mathcal{F}$ on LVOSv2, 4.2 on LVOSv1, 0.2 $\mathcal{J}$ on VOST, 0.6 Q on DiDi and 3.4 AUC on $\text{LaSOT}_{\text{ext}}$. On VOST, PCIA-Track is 0.2 above SAM 3 in $\mathcal{J}$ and 2.7 above it in $\mathcal{J}_{\text{tr}}$, which covers the last quarter of each video, after the objects have transformed (Table 8). SAM 3.1 (Meta AI, 2026) replaces the SAM 3 checkpoint that PCIA-Track uses and changes multi-object inference with Object Multiplex; we leave a version of PCIA-Track built on SAM 3.1 to future work.

4.3 Component analysis

Table 2 removes one component at a time from the full system, which adds SCF, HaR and PCIA to the SAM 3 base tracker, and an intermediate row replaces the relocation step of HaR by withholding the output. On LVOSv2, every component contributes: removing HaR, PCIA, SCF or only the relocation step lowers $\mathcal{J}\&\mathcal{F}$ by 0.22, 0.11, 0.03 and 0.02. On VOTS2026, removing HaR lowers Q from 73.52 to 73.38 and removing only its relocation step lowers it to 73.45: relocation adds 0.07 points on a benchmark whose scenes often contain overlapping similar targets. Removing PCIA lowers Q to 73.34, entirely through the one sequence in which it hands over. SCF changes Q there by only 0.02: it raises absence-detection quality but also withholds some outputs of visible targets. On DiDi, only HaR contributes ($+0.12$); SCF changes Q by -0.05 and PCIA by -0.37: it hands over twice, once correctly (`dancingshoe`) and once onto another object (`hat-1`). Removing the conflict check altogether, so that no track is ever held, lowers $\mathcal{J}\&\mathcal{F}$ on LVOSv2 from 89.90 to 89.36.

4.4 Hold-and-Report on LVOSv2

Reporting held tracks recovers frames. On LVOSv2, the conflict check holds a target in 31 of the 140 videos (771 object-frames; Appendix F). Table 12 in Appendix D compares output rules for held tracks on 137 paired videos, relative to reporting nothing, with the VOTS measures computed from the ground truth; videos without a held target stay exactly unchanged. Reporting the prediction of every held track improves Q by 0.128 points, but it degrades 12 videos while improving 15, and its

Table 2: **Component analysis** on LVOSv2 ($\mathcal{J}\&\mathcal{F}$), VOTS2026 and DiDi (Q), scores $\times 100$. Each component is removed from the full system; the intermediate row withholds the duplicates that HaR would relocate, which changes nothing on single-object benchmarks, where relocation never runs.

Configuration	LVOSv2	VOTS2026	DiDi
PCIA-Track (full)	**89.90**	**73.52**	72.32
without SCF	89.87	73.50	72.37
without HaR	89.68	73.38	72.20
HaR without relocation	89.88	73.45	= full
without PCIA	89.79	73.34	**72.69**

confidence interval includes zero. Withholding the predictions that duplicate another target removes most of the degradation: false outputs on frames where the target is absent drop from 83 to 14, the number of degraded videos falls from 12 to 5, and the gain becomes $+0.136$ points with an interval of $[+0.025, +0.279]$. Figure 3b splits the recovered frame-level loss: 10% comes from the onset frame, where the prediction has usually just jumped, and 90% from the later frames of the hold, where it returns to the target. Relocating duplicates instead of withholding them adds 0.012 points on LVOSv2 (interval $[-0.004, +0.034]$; 12 videos improved, 3 degraded) at the cost of 13 more false outputs, and 0.07 points of Q on VOTS2026.

4.5 Post-conflict identity arbitration

PCIA is conservative by design. On VOTS2026, PCIA is armed in 37 sequences and arbitrates 109 conflicts (Table 13 in Appendix E): it keeps the incumbent 66 times, abstains 42 times and hands over once. Keeping is the most frequent outcome on LVOSv2 and VOTS2026, and hand-overs are rare; their net effect is positive on LVOSv2 and in both VOTS2026 configurations. On VOTS2026, the single hand-over, in the sequence `goldfish`, changes the outputs of this sequence and adds 0.180 Q-points (Figure 6 in Appendix E). In the LVOSv2 video `aFytsETk` (Figure 4), a person shuffles three identical cups, and the track leaves the annotated cup during the shuffles, with and without PCIA. After the tracks separate, PCIA hands the identity back to it at frame 209; without PCIA the track stays on another cup (Figure 7 in Appendix E), as do SAM 3, SAM 2.1, SAMURAI and DAM4SAM; SAM2Long returns 30 frames after PCIA's hand-over. The object's $\mathcal{J}\&\mathcal{F}$ is 0.917 for PCIA-Track, 0.683 without PCIA, and 0.546–0.684 for the other trackers.

What limits PCIA. The link test compares both probes with the incumbent's own history, so its accuracy is bounded by that history: in all nine wrong keeps and in both wrong hand-overs on LVOSv2, the reference window itself already lay off the target (Appendix E).

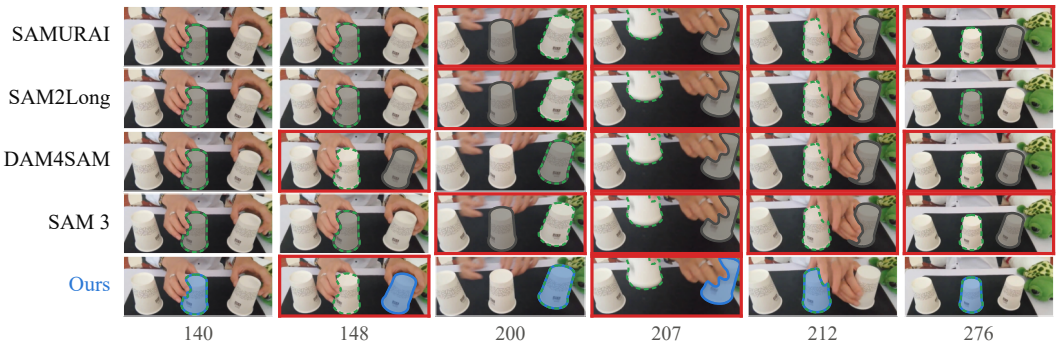

Figure 4: **Three identical cups being shuffled** Masks of SAMURAI, SAM2Long, DAM4SAM, SAM 3 and PCIA-Track on six frames (numbered from 0), cropped to remove burned-in subtitles; dashed green: ground truth; red border: the mask is not on the annotated cup. SAM 2.1-L behaves like SAMURAI ($\mathcal{J}\&\mathcal{F}$ 0.546).

Table 3: **VOTS2026 results.** Official entries are copied from the challenge report (Kristan et al., 2026, Table 1): the five leading entries and those of DAM4SAM and SAM 3 (SAM3a), which Table 1 also compares. The report rounds to three decimals; PCIA-Track is a post-challenge submission to the official server. VLM: uses a vision–language model. Best in **bold**.

Tracker	VLM	$Q\uparrow$	A↑	R↑
DAM4SAM (Videnović et al., 2025)	–	0.626	0.742	0.806
SAM3a	–	0.645	0.766	0.792
SENTRY-SAM3 (Alansari et al., 2026)	–	0.666	0.769	0.812
SAM3Dual	–	0.682	0.776	0.840
ACTrack	✓	0.702	0.776	0.866
SAFEDAM	–	0.722	0.797	0.865
VLM-SAFEDAM (winner)	✓	0.733	0.810	0.878
PCIA-Track (ours)	–	**0.7352**	**0.829**	**0.883**

4.6 Results on VOTS2026

Table 3 compares PCIA-Track with official VOTS2026 entries. PCIA-Track reaches $Q = 0.7352$ with the highest accuracy (0.829) and robustness (0.883), using SAM 3 as its only model; the challenge winner and ACTrack add a vision–language model. Table 2 gives the contribution of each component on VOTS2026, and Table 4 in Appendix B lists Q, A and R of each configuration.

5 Conclusion

Identity conflicts in memory-based trackers are detected after the identity error, and the evidence that resolves them arrives during and after the conflict. PCIA-Track therefore holds the memory of a conflicting track while it keeps reporting its prediction, relocates the target when that prediction duplicates another one, and decides the identity after the conflict by looking back from fresh probes. PCIA-Track improves on SAM 3 on every benchmark we report. Its decisive corrections are rare and concentrated, so identity mechanisms should be evaluated on complete benchmarks with per-video attribution.

Limitations. PCIA hands over rarely, four times in 112 arbitrations on LVOSv2, once in 109 on VOTS2026 and twice in 98 on DiDi, so its hand-over precision rests on few events: two of the four LVOSv2 hand-overs moved the identity onto the target, the other two, in one video, followed a reference window that already lay off the target, its VOTS2026 gain comes from one sequence, and on DiDi one of its two hand-overs moves the track onto another object, so that PCIA lowers Q there by 0.37, 0.34 of it from that hand-over. It runs two additional SAM 3 instances while it arbitrates; on the LVOSv2 video with the most arbitrations, our implementation raises peak GPU memory by $5.6\times$ and run time by $5.1\times$, mostly because the replay trackers arbitrate themselves (Appendix A); removing these nested arbitrations leaves every output bit-identical and gives factors of $2.5\times$ and $4.2\times$. Relocation runs a mask generator on the frames it treats, 0.9% of the object-frames of VOTS2026, and adds 11.6% to the total run time (Appendix A). HaR trades false outputs in absent frames for recovered visible frames, which lowers the absence-detection quality (ADQ) of VOTS2026 from 81.51 without HaR to 79.97. Both components act only on conflicts that the conflict check detects (Appendix F). The conflict check and the duplicate test of HaR rely on the one-sided containment ratio of Eq. (1), so a small target nested inside another, such as a part inside the whole, can be treated as a duplicate of it: its memory is held and its output withheld or relocated. Separating nesting from duplication requires part–whole reasoning, which we leave to future work.

AI use statement

We used large language models to aid and polish the writing of this paper and to assist with research ideation and execution, including implementing and running experiments and analysing their results. The authors reviewed all AI-assisted content, checked every reported number against the experiment records and the official evaluation servers, and take full responsibility for the content of this paper.

Reproducibility statement

Section 3 and Appendix A specify all components and parameters. PCIA-Track is implemented on the released code of a VOTS2026 challenge tracker and uses public SAM 3 weights. VOTS2026 results come from the official evaluation server, the other multi-object results from their official evaluators, DiDi results from the VOT toolkit, and $\text{LaSOT}_{\text{ext}}$ results from the evaluation code of pytracking; the LVOSv2 analyses use the VOTS measures computed from its ground truth; the submitted packages and the per-video comparisons will be released with the code. Repeated runs give bit-identical outputs in nearly all cases; a few sequences differ slightly across machines or between runs.

References

Mohamad Alansari, Yonathan Michael, Hasan AlMarzouqi, Muzammal Naseer, Naoufel Werghi, and Sajid Javed. SENTRY: SAM2-enhanced neighbor-aware and temporally reasoned memory for visual tracking. In *European Conference on Computer Vision*, 2026.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6182–6191, 2019.

Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. MeMOT: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8090–8100, 2022.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, pp. 138846–138923, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e0982cbc81401df3430ee1ff780dc7a2-Paper-Conference.pdf.

Ruixiang Chen, Guolei Sun, Yawei Li, Jie Qin, and Luca Benini. HiM2SAM: Enhancing SAM2 with hierarchical motion estimation and memory optimization towards long-term tracking. In *Pattern Recognition and Computer Vision: PRCV 2025 Proceedings*, pp. 276–291, 2026. doi: 10.1007/978-981-95-5755-4_19.

Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an Atkinson-Shiffrin memory model. In *Computer Vision – ECCV 2022*, pp. 640–658, 2022. URL https://link.springer.com/chapter/10.1007/978-3-031-19815-1_37.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3151–3161, 2024. doi: 10.1109/CVPR52733.2024.00304.

Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip H. S. Torr, and Song Bai. MOSE: A new dataset for video object segmentation in complex scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20224–20234, 2023.

Henghui Ding, Kaining Ying, Chang Liu, Shuting He, Xudong Jiang, Yu-Gang Jiang, Philip H. S. Torr, and Song Bai. MOSEv2: A more challenging dataset for video object segmentation in complex scenes. *arXiv preprint arXiv:2508.05630*, 2025a. URL https://arxiv.org/abs/2508.05630.

Shuangrui Ding, Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Yuwei Guo, Dahua Lin, and Jiaqi Wang. SAM2Long: Enhancing SAM 2 for long video segmentation with

a training-free memory tree. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13614–13624, 2025b. URL https://openaccess.thecvf.com/content/ICCV2025/html/Ding_SAM2Long_Enhancing_SAM_2_for_Long_Video_Segmentation_with_a_ICCV_2025_paper.html.

Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5374–5383, 2019.

Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, Yong Xu, Chunyuan Liao, Lin Yuan, and Haibin Ling. LaSOT: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129(2):439–461, 2021. doi: 10.1007/s11263-020-01387-y.

Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. STMTrack: Template-free visual tracking with space-time memory networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13774–13783, 2021.

Lingyi Hong, Wenchao Chen, Zhongying Liu, Wei Zhang, Pinxue Guo, Zhaoyu Chen, and Wenqiang Zhang. LVOS: A benchmark for long-term video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13480–13492, 2023.

Lingyi Hong, Zhongying Liu, Wenchao Chen, Chenzhi Tan, Yuang Feng, Xinyu Zhou, Pinxue Guo, Jinglun Li, Zhaoyu Chen, Shuyong Gao, Wei Zhang, and Wenqiang Zhang. LVOS: A benchmark for large-scale long-term video object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(1):946–961, 2026. doi: 10.1109/TPAMI.2025.3611020.

Yuqing Huang, Xin Li, Zikun Zhou, Yaowei Wang, Zhenyu He, and Ming-Hsuan Yang. RTracker: Recoverable tracking via PN tree structured memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19038–19047, 2024.

Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In *2010 20th International Conference on Pattern Recognition*, pp. 2756–2759, 2010. doi: 10.1109/ICPR.2010.675.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Matej Kristan, Jiri Matas, Pavel Tokmakov, Adam W. Harley, Alan Lukežič, Michael Felsberg, et al. The fourth visual object tracking segmentation VOTS2026 challenge results. VOTS2026 workshop at ECCV 2026, 2026. URL https://openreview.net/forum?id=zGOan36xpF. Under review.

Xin Li, Deshui Miao, Zhenyu He, Yaowei Wang, Huchuan Lu, and Ming-Hsuan Yang. Learning spatial-semantic features for robust video object segmentation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/154d63285d3ed7826e7f026c0b350d69-Abstract-Conference.html.

Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13444–13454, 2021.

Meta AI. SAM 3.1 release notes: Object Multiplex. https://github.com/facebookresearch/sam3/blob/main/RELEASE_SAM3p1.md, 2026. Released March 27, 2026; checkpoint facebook/sam3.1.

Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4293–4302, 2016. doi: 10.1109/CVPR.2016.465.

Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9226–9235, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Oh_Video_Object_Segmentation_Using_Space-Time_Memory_Networks_ICCV_2019_paper.html.

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

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, pp. 28085–28128, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/45c1f6a8cbf2da59ebf2c802b4f742cd-Paper-Conference.pdf.

Yihong Sun, Xinyu Yang, Jennifer J. Sun, and Bharath Hariharan. Tracking and understanding object transformations. In *Advances in Neural Information Processing Systems*, volume 38, pp. 89907–89933, 2025. doi: 10.52202/085713-2708.

Pavel Tokmakov, Jie Li, and Adrien Gaidon. Breaking the "object" in video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22836–22845, 2023. doi: 10.1109/CVPR52729.2023.02187.

Jovana Videnović, Alan Lukežič, and Matej Kristan. A distractor-aware memory for visual object tracking with SAM2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24255–24264, 2025. doi: 10.1109/CVPR52734.2025.02259.

Jovana Videnović, Matej Kristan, and Alan Lukežič. Distractor-aware memory-based visual object tracking. *International Journal of Computer Vision*, 134(5):211, 2026. doi: 10.1007/s11263-026-02790-7.

Paul Voigtlaender, Jonathon Luiten, Philip H. S. Torr, and Bastian Leibe. Siam R-CNN: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6578–6588, 2020.

Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2566–2576, 2019.

Qianxiong Xu, Lanyun Zhu, Chenxi Liu, Guosheng Lin, Cheng Long, Ziyue Li, and Rui Zhao. SAMITE: Position prompted SAM2 with calibrated memory for visual object tracking. *arXiv preprint arXiv:2507.21732*, 2025.

Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. SAMURAI: Motion-aware memory for training-free visual object tracking with SAM 2. *IEEE Transactions on Image Processing*, 35:970–982, 2026. doi: 10.1109/TIP.2026.3651835.

Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36324–36336, 2022.

Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2491–2502, 2021.

Yuan Yuan, Sabu Emmanuel, Yuming Fang, and Weisi Lin. Visual object tracking based on backward model validation. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(11):1898–1910, 2014. doi: 10.1109/TCSVT.2014.2319632.

Zhixiong Zhang, Shuangrui Ding, Xiaoyi Dong, Songxin He, Jianfan Lin, Junsong Tang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Advancing complex video object segmentation via progressive concept construction. In *International Conference on Learning Representations*, pp. 130411–130434, 2026.

Junbao Zhou, Ziqi Pang, and Yu-Xiong Wang. RMem: Restricted memory banks improve video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18602–18611, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Zhou_RMem_Restricted_Memory_Banks_Improve_Video_Object_Segmentation_CVPR_2024_paper.html.

Deyi Zhu, Yuji Wang, Yong Liu, Yansong Tang, Bingyao Yu, Jiwen Lu, and Jie Zhou. Segment anything with motion, geometry, and semantic adaptation for complex nonlinear visual object tracking. *arXiv preprint arXiv:2605.22538*, 2026.

Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware Siamese networks for visual object tracking. In *European Conference on Computer Vision*, volume 11213 of *Lecture Notes in Computer Science*, pp. 103–119, 2018. doi: 10.1007/978-3-030-01240-3_7.

A Implementation details

Conflict check. For every overlapping pair of two targets, a target and one of its candidate tracks, or two candidate tracks of the same target, the conflict check compares their object-score logits and considers the lower-scoring track f. The track is held when $\max \rho_{f|\cdot} \geq 0.9$, and the hold ends when the ratio falls below 0.2. A held track skips the memory update, and HaR decides its output. Overlaps used by HaR are computed on a copy of all predictions taken before any output is emptied.

HaR relocation. Relocation acts on a held track f whose prediction the keep step has emptied because it duplicates another target, and whose object-score logit, rounded to four decimals, is at least τ. Proposals come from the tracker's own SAM 3 image model, prompted with a 32×32 point grid and three masks per point; all non-empty masks are kept (at most 3,072), once per frame for all tracks treated in that frame. For f we take the $3K$ proposals whose box centers are nearest to the centroid of f's last non-empty output within the previous 399 frames, and among them the K with the highest predicted IoU (without such an output, the K with the highest predicted IoU overall). Each of the K proposals is scored by prompting the tracker with its box, conditioned on f's held memory and without a memory update, as the product of the resulting object-score logit and predicted IoU; ties go to the first proposal. The best proposal replaces f's output in this frame only and does not count as f's last output in later frames. On VOTS2026, relocation treats 6,424 object-frames (0.9%) in 52 of the 170 sequences; with three processes per GPU, the proposals take about 2.2 s per treated frame and each scored proposal 33 ms.

Selection of K and τ. Both were selected on LVOSv2 by a rule fixed before the evaluation. On the object-frames whose output is still empty after the keep step, each candidate threshold is the lowest τ at which at most N frames with an absent target would be relocated, for $N \in \{20, 50, 100, 200, 500\}$ and $K \in \{30, 100, 300, 1000, 3072\}$. The rule takes the smallest setting whose gain on LVOSv2 passes pre-registered criteria, among them a positive net gain that exceeds the contribution of any single video, comes from at least five videos, and keeps its sign when any one video is left out. It selects $K = 30$ and $N = 20$, that is, $\tau = 2.1094$; at $N = 20$, every K from 30 to 1000 passes, and no setting with $N \geq 50$ does. All other benchmarks use these values unchanged, so only the LVOSv2 results share data with this selection.

Order of operations. SCF and the keep and withhold steps of HaR run at the end of each frame, before PCIA caches the incumbent's predictions; relocation runs after the tracker step and changes neither the memory nor the cache of PCIA.

PCIA state machine. PCIA updates its buffer, discontinuity record and pair states on sampled frames only ($S = 5$; $S = 1$ on LVOSv2); the tracker outputs every frame as usual. A pair (incumbent, candidate track) is armed at the first frame on which the conflict check holds either track.

It stays armed until the arbitration at s_2 or until B sampled steps pass without separation. A new conflict for the same pair requires at least two consecutive hold-free sampled steps. A new distractor-resolving memory frame is recorded as a discontinuity at the frame index of the memory entry, which enters the memory bank one frame later. The two probes are complete instances of the tracker, including its gates and candidate tracks, and are discarded after the decision. A hand-over copies the candidate's full object state into the incumbent's slot, removes the candidate from the pool of candidate tracks and takes effect from the frame after s_2; earlier outputs are never modified. In our implementation, the two replay trackers themselves run the conflict check and PCIA; these nested arbitrations never hand over in any run, so every decision equals that of a non-recursive implementation, but they add GPU memory: in one run on LVOSv2 (without HaR), a replay exceeded the 48 GB of our GPU and PCIA abstained. On the LVOSv2 video with the most arbitrations, removing them leaves every output bit-identical and lowers the peak memory from 35.4 to 16.0 GB. The VOTS2026 configuration without HaR comes from an earlier build in which the replay trackers do not arbitrate.

SCF. The token count of a mask with N_+ foreground pixels in a $W \times H$ frame is $N_+ \cdot \frac{1008}{W} \cdot \frac{1008}{H} \cdot \frac{1}{196}$, the number of 14×14 patches it covers at SAM 3's input resolution. SCF empties the output when this count is below 16 and the raw object-score logit is below 0.5.

Evaluation protocols. PCIA-Track tracks the objects of the first annotated frame jointly, so that the conflict check can act between them; the 6 of 239 LVOSv2 objects that appear later are tracked separately from their first appearance. LVOSv1 results are computed from the LVOSv2 outputs of its 50 videos, with the v1 release of the official evaluator (commit 9ff733c) and its seen/unseen split (36 seen and 14 unseen videos). The evaluator was changed for LVOSv2 at commit c05e889: it no longer scores the first and the last sampled frame of each object, and its split counts one of these 14 videos as seen. Both versions compute $\mathcal{J}$ and $\mathcal{F}$ with the same functions: rescoring our outputs with the v1 release without the first and last sampled frames reproduces the current version with the v1 split exactly. Published LVOSv1 results may come from either version; Table 7 gives the results of PCIA-Track and SAM 3 under both. The v1 release also scores the first frame of each object, where SAM 3 outputs its own reconstruction of the given mask ($\mathcal{J} = 0.977$ on average); replacing it by the ground truth changes its $\mathcal{J}\&\mathcal{F}$ by only $+0.002$ points. VOST is evaluated with the official code as released (TRI-ML/VOST, commit fe274574): $\mathcal{J}$ and $\mathcal{J}_{\mathrm{tr}}$ are computed on the annotated frames (5 fps) without the first and the last one, $\mathcal{J}_{\mathrm{tr}}$ on the last 25% of the remaining frames, and both are averaged over objects; the code counts pixels labelled Ignore (255) as background, unlike the description in the VOST paper. SAM 3 is run with the SAM 2 VOS evaluation protocol: first-frame masks, 480p frames on DAVIS 2017, and one inference state for all objects, except on LVOSv2, where every object is tracked separately from its first appearance. On DAVIS 2017 val it obtains 91.5–91.6 against 92.2 reported; the remaining gap is not explained by the code version, the memory-selection setting, or the overlap resolution.

Single-object evaluation. PCIA-Track receives the first-frame mask of DiDi or, on LaSOT$_{\mathrm{ext}}$, the mask that SAM 3 predicts from the first-frame box; SAM 3 uses the same initialization on the benchmarks scored with the VOT toolkit. On single-target sequences the duplicate test of HaR never fires, because it only compares initialized targets, so relocation never runs; the conflict check can still hold the target's memory against one of its own candidate tracks, HaR then keeps reporting the target's prediction, and PCIA arbitrates between the target and that candidate track. Both trackers convert masks to boxes in the same way. On DiDi we use the conversion of the official DiDi code (Videnović et al., 2025), the axis-aligned box of the mask with $w = x_{\max} - x_{\min}$, as for the published DiDi results. On LaSOT$_{\mathrm{ext}}$, the first frame keeps the ground-truth box, and every other frame reports the axis-aligned box of its mask, with $x = x_{\min}$ and $w = x_{\max} - x_{\min} + 1$ in 0-based pixel coordinates; an empty mask repeats the previous box, or the first-frame box if all earlier masks are empty, and the evaluation replaces empty boxes by the previous box in any case. LaSOT$_{\mathrm{ext}}$ is scored with the evaluation code of pytracking[1]: success AUC over 21 overlap thresholds, precision at 20 pixels and normalized precision at 0.2; on LaSOT$_{\mathrm{ext}}$, frames without a visible target count as successes in the normalized precision, as in the published results. Recomputed from their released raw results, the published LaSOT (Fan et al., 2019) scores of KeepTrack (Mayer et al., 2021), DiMP-50 (Bhat et al., 2019) and MDNet (Nam & Han, 2016) and the published LaSOT$_{\mathrm{ext}}$ scores of KeepTrack are reproduced within 0.1 points. DiDi is scored with the

[1] https://github.com/visionml/pytracking

VOT toolkit, which reads `groundtruth_old.txt` where a sequence provides one (90 of the 180 sequences); this is the evaluation workflow released with DiDi, and our DiDi results use it. The newer `groundtruth.txt` of these sequences marks 2.2% of all frames as target-absent where the old file still gives a box; scoring with it gives Q 72.8 for PCIA-Track and 72.0 for SAM 3. Published SAM-based trackers differ in this conversion: the released code of SAMURAI, SAMITE and HiM2SAM computes $w = x_{\max} - x_{\min}$ and h alike, whereas that of DAM4SAM and SENTRY adds 1, as we do; all use 0-based coordinates.[2] Table 11 gives our scores under two alternative conventions.

Parameters. PCIA: $K_{\mathrm{w}} = 3$ and $B = 100$ sampled steps; the sampling stride is $S = 5$ frames, and $S = 1$ on LVOSv2, whose frames are stored every fifth frame, so that B spans 500 original frames everywhere; link test: IoU ≥ 0.5 on ≥ 2 of 3 frames; discontinuities: IoU < 0.3 between consecutive sampled masks or centroid displacement $> \sqrt{\text{area}}$. HaR: duplicate test $\max_p \rho_{f|p} \geq 0.9$; relocation $K = 30$, $\tau = 2.1094$. SCF: 16 tokens, logit 0.5. All other parameters are those of the released base code. Using $S = 5$ also on LVOSv2 changes its $\mathcal{J}\&\mathcal{F}$ from 89.90 to 89.79: with $S = 5$, PCIA makes none of the four hand-overs, so its outputs equal those without PCIA, and the outputs of the other 137 videos are bit-identical to those with $S = 1$.

B Full VOTS2026 results of each configuration

Table 4 gives Q, A and R on the VOTS2026 server for each configuration of Table 2.

Table 4: **VOTS2026 server results (Q, A, R) of the configurations in Table 2.** A component that never acts in a sequence leaves its output unchanged; the configurations without SCF and without PCIA therefore re-run only the sequences in which the component changes the output (139 sequences for SCF and, for PCIA, the one sequence with a hand-over) and reuse the outputs of the full system on the others, which we checked to be byte-identical on the same machine. Removing SCF lowers the absence-detection quality (ADQ) from 79.97 to 77.66.

Configuration	$Q\uparrow$	A↑	R↑
PCIA-Track (full)	0.7352	0.8287	0.8831
without SCF	0.7350	0.8260	0.8868
without HaR	0.7338	0.8316	0.8774
HaR without relocation	0.7345	0.8310	0.8794
without PCIA	0.7334	0.8284	0.8811

C Detailed results

Table 5 lists the source of every value of other methods in Table 1. Table 6 breaks the LVOSv2 results down by category, and Table 7 gives the LVOSv1 results under both versions of its evaluator. Table 8 gives $\mathcal{J}$ and $\mathcal{J}_{\mathrm{tr}}$ on VOST val. Tables 9 and 10 list all measures on the single-object benchmarks, and Table 11 gives our scores under other mask-to-box conventions. Figure 5 shows a DiDi sequence in which SAM 3 moves to a look-alike.

D Output rules for held tracks

The LVOSv2 analyses of Sections 3.2 and 4.4 come from earlier runs that tracked only the objects of the first annotated frame (233 of 239 objects). All paired comparisons of these runs, including Table 12 and Figure 3b, use the 137 of 140 videos in which every object is present in the first annotated frame; the evaluation script excluded the three videos with later-appearing objects, because these objects were not tracked.

[2] `https://github.com/yangchris11/samurai`, `https://github.com/Sam1224/SAMITE`, `https://github.com/LouisFinner/HiM2SAM`, `https://github.com/jovanavidenovic/DAM4SAM`, `https://github.com/HamadYA/SENTRY`.

Table 5: **Sources of the values of other methods in Table 1.** Venues are given in the references; "arXiv version" marks values taken from the arXiv version of a paper that is published elsewhere.

Benchmark	Sources
LVOSv2	SAM 2.1-L: Ravi et al. (2025, Table 19(c)), released checkpoint; self-reported: S3 (Li et al., 2025, Table 4, model ViTb+), SAM2Long (Ding et al., 2025b, Table 3), SeC (Zhang et al., 2026, Table 5) and DAM4SAM (Videnović et al., 2026, Table 12); SAMURAI: as reported by Zhang et al. (2026, Table 5); rows marked *: Alansari et al. (2026, arXiv version, Table 25).
LVOSv1	SAM 2.1-L: Ravi et al. (2025, Table 19(b)), released checkpoint; self-reported: S3 (Li et al., 2025, Table 4, model ViTb+), SAM2Long (Ding et al., 2025b, Table 3) and DAM4SAM (Videnović et al., 2026, Table 12); SAMURAI and rows marked *: Alansari et al. (2026, arXiv version, Table 25). Published LVOSv1 results may come from either version of the official evaluator (Appendix A).
VOST	SAM 2.1, SAMURAI, SAM2Long and DAM4SAM: reported by Sun et al. (2025, arXiv version, Table 2), which does not state the model size or weights of SAM 2.1 and SAM2Long.
DiDi	DAM4SAM: self-reported, and SAM 2.1-L, SAMURAI and SAM2Long: run by Videnović et al. (2025, Table 2); SAMITE, HiM2SAM and SENTRY-HiM: Alansari et al. (2026, arXiv version, Table 27); SENTRY-D4S: Alansari et al. (2026, Table 5) (Table 23 of its arXiv version gives 70.2).
$\text{LaSOT}_{\text{ext}}$	Self-reported: SAMURAI (Yang et al., 2026, Table I), SAM2Long (Ding et al., 2025b, Table 5), DAM4SAM (Videnović et al., 2025, Table 6), SAMITE (Xu et al., 2025, Table 6) and HiM2SAM (Chen et al., 2026, arXiv version, Table 1), where the released code of SAMURAI, SAMITE and HiM2SAM computes $w = x_{\max} - x_{\min}$; SAM 2.1-L: run by Videnović et al. (2025, Table 6); SeC: run by Zhu et al. (2026, Table I); SENTRY-HiM: Alansari et al. (2026, arXiv version, Table 24); SENTRY-D4S: Alansari et al. (2026, Table 1).

Table 6: **LVOSv2 val by category**: $\mathcal{J}\&\mathcal{F}$, region similarity $\mathcal{J}$ and boundary accuracy $\mathcal{F}$ on seen (s) and unseen (u) categories, from the official evaluator. The rows of SAM 2.1-L and SAM2Long are copied from Ravi et al. (2025, Table 19(c), released checkpoint) and Ding et al. (2025b, Table 3, SAM 2.1 version). Ravi et al. (2025) label the model SAM 2; its second arXiv version and the released code call it SAM 2.1.

Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$
SAM 2.1-L	80.6	81.7	88.2	71.4	81.0
SAM2Long	85.9	81.7	88.6	83.0	90.5
SAM 3	88.3	85.5	92.9	83.3	91.5
PCIA-Track (ours)	89.9	85.9	93.2	86.4	94.1

Table 7: **LVOSv1 val under both versions of the official evaluator** ($\mathcal{J}\&\mathcal{F}$). The v1 release (commit 9ff733c) scores every sampled frame; the current version (from commit c05e889) skips the first and the last sampled frame of each object, and its split counts one more video as seen. Table 1 uses the first column.

Evaluator	v1	current	current
Seen/unseen split	v1	v1	current
SAM 3	83.26	83.26	83.81
PCIA-Track (ours)	87.53	87.54	87.44

Table 8: **VOST val**: region similarity $\mathcal{J}$ over the annotated frames and $\mathcal{J}_{\text{tr}}$ over their last quarter, from the official code (Appendix A). Other rows are reported by Sun et al. (2025, arXiv version, Table 2), which does not state the model size or weights of SAM 2.1 and SAM2Long.

Method	$\mathcal{J}$	$\mathcal{J}_{\text{tr}}$
SAM 2.1	48.4	32.4
SAMURAI	49.8	34.0
SAM2Long	46.4	29.1
DAM4SAM	48.8	33.6
SAM 3	53.1	36.3
PCIA-Track (ours)	53.3	39.0

Table 9: **Single-object benchmarks scored with the VOT toolkit** (all scores $\times 100$). DiDi: quality Q, accuracy A and robustness R. Sources in Table 5; a mark on the first measure of a benchmark applies to all three. †: reported by a third party. –: not reported.

	DiDi		
Method	Q	A	R
SAM 2.1-L	64.9†	72.0	88.7
SAMURAI	68.0†	72.2	93.0
SAM2Long	64.6†	71.9	88.3
DAM4SAM	69.4	72.7	94.4
SAMITE	68.4†	72.3	93.6
HiM2SAM	67.0†	71.2	91.9
SENTRY-HiM	67.3	71.6	92.2
SENTRY-D4S	70.5	73.7	95.6
SAM 3	71.7	74.8	95.2
PCIA-Track (ours)	72.3	75.4	95.3

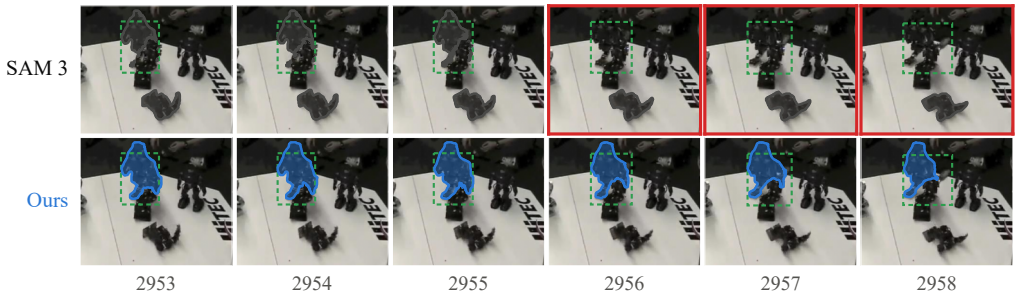

Figure 5: **A look-alike on DiDi** (`LaSOT_robot-19`, frames 2953–2958 numbered from 0, cropped). Masks of SAM 3 and PCIA-Track; dashed green: ground-truth box; red border: the mask does not overlap the ground-truth box. From frame 2956, SAM 3 follows the knocked-over robot of the same type, while PCIA-Track stays on the target; no arbitration of PCIA occurs in this window.

E Arbitration outcomes of PCIA

LVOSv2: gain is measured against $S = 5$ (no hand-over, outputs bit-identical on the other 137 videos). DiDi: gain against the full system without PCIA; the `dancingshoe` hand-over is correct, the `hat-1` one moves the track onto another object.

Arbitrations that failed in our batch runs because other jobs took the GPU memory were re-run on a free GPU; all reported results use these re-runs.

What limits PCIA. Most arbitrations keep the incumbent, whose probe usually traces back to its own stable window. The link test compares both probes with the incumbent's own history, so its accuracy is bounded by that history. On LVOSv2 we can label 81 of the 82 keeps with the ground truth: 60 kept an incumbent that was on the target, and in 9 of the other 21 the candidate was on the target. In all 9, the reference window itself already lay off the target (median IoU with the target 0.00, against 0.93 in the correct keeps), so the incumbent's probe linked to a wrong reference. The two hand-overs that missed the target (both in one video) have the same cause: their reference window had IoU 0.00 with the target, and the candidates' probes linked to the object in that window.

One hand-over, 2,079 frames. Figure 6 shows the hand-over in the VOTS2026 sequence `goldfish`, a 4,520-frame video with several similar fish. A conflict between the target track and a candidate track starts at frame 2,385. At frame 2,440 the candidate's probe reconnects to the target's stable window and the incumbent's probe does not, and PCIA hands the identity to the candidate. From the next frame on, the outputs of the sequence change (the target's on all remaining 2,079 frames), and its server score rises by 0.180 Q-points.

Table 10: **Single-object benchmarks scored with the one-pass evaluation.** $\text{LaSOT}_{\text{ext}}$: success AUC, normalized precision P_{norm} and precision P. Self-reported: SAMURAI (Yang et al., 2026, Table I), SAM2Long (Ding et al., 2025b, Table 5), DAM4SAM (Videnović et al., 2025, Table 6), SAMITE (Xu et al., 2025, Table 6) and HiM2SAM (Chen et al., 2026, arXiv version, Table 1); SAM 2.1-L was run by Videnović et al. (2025, Table 6), SENTRY-HiM is from Alansari et al. (2026, arXiv version, Table 24), and SENTRY-D4S and the P_{norm} and P of DAM4SAM from its Table 1; SeC was run by Zhu et al. (2026, Table I). A mark on the first measure applies to all three. †: reported by a third party. The released code of SAMURAI, SAMITE and HiM2SAM computes $w = x_{\max} - x_{\min}$ (Appendix A); –: not reported.

	$\text{LaSOT}_{\text{ext}}$		
Method	AUC	P_{norm}	P
SAM 2.1-L	56.9†	–	–
SAMURAI	61.0	73.9	72.2
SAM2Long	60.9	–	–
SeC	60.3†	72.7	71.2
DAM4SAM	60.9	75.3†	72.2†
SAMITE	62.1	75.2	73.5
HiM2SAM	62.8	75.5	74.3
SENTRY-HiM	62.1	76.3	73.5
SENTRY-D4S	61.8	76.6	73.8
SAM 3	62.2	76.8	73.5
PCIA-Track (ours)	65.6	81.0	78.6

Table 11: **Mask-to-box conventions.** Success AUC on $\text{LaSOT}_{\text{ext}}$ under our convention ($w = x_{\max} - x_{\min} + 1$, Appendix A) and two alternatives. B: the same boxes in 1-based coordinates ($x = x_{\min} + 1$, $y = y_{\min} + 1$); C: $w = x_{\max} - x_{\min}$ and $h = y_{\max} - y_{\min}$, as in SAMURAI. On $\text{LaSOT}_{\text{ext}}$ the ground-truth boxes are aligned with our 0-based mask boxes, often one pixel wider at the top-left corner, so shifting by one pixel (B) lowers AUC while the narrower boxes of C change it little.

	$\text{LaSOT}_{\text{ext}}$		
Method	ours	B	C
SAM 3	62.2	60.0	61.8
PCIA-Track (ours)	65.6	63.1	65.2

Table 12: **Output rules for held tracks on LVOSv2 val** (VOTS measures computed from the ground truth, 137 paired videos, Appendix D). ΔQ in points relative to reporting nothing, with 95% bootstrap interval over videos; ↑ / ↓: videos improved / degraded. False out.: frames of a hold, excluding its first frame, in which the target is absent and the held track reports a mask where reporting nothing gives an empty output. All rows share the same base configuration and differ only in the output rule for held tracks.

Output of a held track	ΔQ [95% CI]	↑ / ↓	False out.
nothing (empty mask)	reference	–	0
own prediction	$+0.128$ [-0.003, $+0.285$]	15 / 12	83
own prediction, duplicates withheld	$+0.136$ [$+0.025$, $+0.279$]	14 / 5	14
own prediction, duplicates relocated (HaR)	$+0.148$ [$+0.033$, $+0.294$]	19 / 7	27

Table 13: **Arbitration outcomes of PCIA on LVOSv2, DiDi and VOTS2026.** Arbitrations with L_i=1 keep the incumbent; "abstain" counts arbitrations in which neither probe links or no usable reference window exists. Gain: score change caused by the hand-overs. The VOTS2026 rows are the configurations of Table 2 with HaR without relocation and without HaR.

Benchmark	Videos	Arbitrations	Keep	Abstain	Hand-over	Gain
LVOSv2 val	140	112	82	26	4	+0.306 Q-points
DiDi	180	98	70	26	2	−0.366 Q-points
VOTS2026, HaR w/o relocation	170	109	66	42	1	+0.180 Q-points
VOTS2026, without HaR	170	99	61	36	2	+0.193 Q-points



Figure 6: **A hand-over in `goldfish` (VOTS2026).** Target track of PCIA-Track (blue) and without PCIA (orange outline); frames numbered from 0. At frame 2,440, PCIA hands the identity to the candidate track that follows the target; afterwards, the track without PCIA stays on other fish.

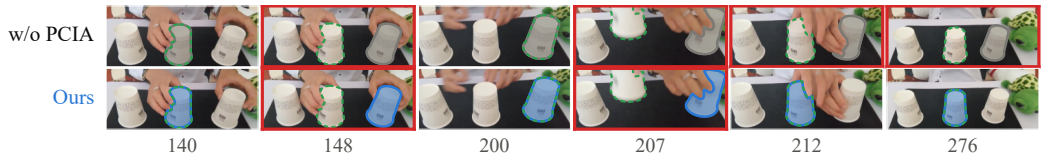

Figure 7: **PCIA-Track without and with PCIA on `aFytsETk`** (LVOSv2, Figure 4). Without PCIA, the track stays on another cup after the shuffles; with PCIA, the identity is handed back at frame 209 (frames numbered from 0). Dashed green: ground truth; red border: the mask is not on the annotated cup.

F How often tracks conflict

Table 14 counts, on LVOSv2 and VOTS2026, the videos, objects and object-frames in which the conflict check holds a target; holds of candidate tracks are not counted. HaR and PCIA act only on these conflicts. The counts come from the run logs. The state of the conflict check depends only on the memory, which SCF and HaR do not change; for LVOSv2 we count a run without PCIA, whose counts are identical to those of a run with PCIA on the 138 videos where both were logged. LVOSv2 counts cover the 233 objects of the first annotated frames.

Table 14: **Frequency of conflicts per benchmark**: videos, objects and object-frames with a held target. Object-frames exclude the first frame of each video.

Benchmark	Videos	Objects	Object-frames
LVOSv2 val	31 / 140 (22.1%)	42 / 233 (18.0%)	771 / 101,308 (0.76%)
VOTS2026	72 / 170 (42.4%)	130 / 375 (34.7%)	14,420 / 719,259 (2.00%)