GENERALIZED PSEUDO-LABEL IN POINT TRACKING

Anonymous authors
Paper under double-blind review

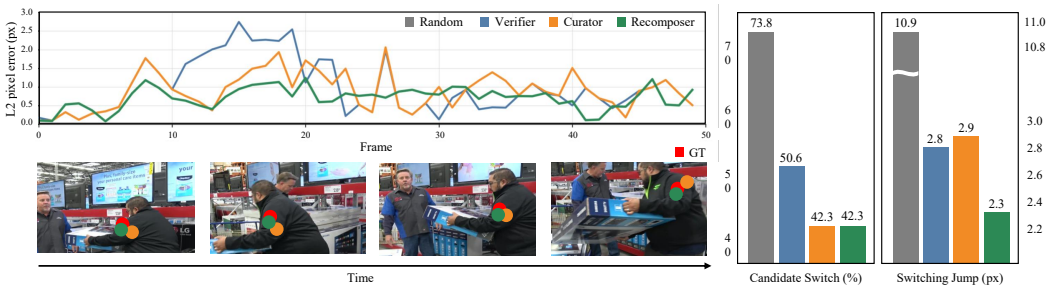


Figure 1: **Temporal stability from Candidate Curation and Coordinate Recomposition.** Frame-wise discrete candidate selection from multiple trackers can introduce temporal discontinuities. **(Left)** The Coordinate Recomposer improves localization after candidate selection, as shown by the frame-wise L2 localization error. **(Right)** The Candidate Curator reduces the tracker-switch rate, while the Recomposer further reduces switch-induced jumps, measured by the P95 displacement at switch transitions.

ABSTRACT

Reliable trajectory supervision is fundamental to robust point tracking, yet dense point-level annotations for real-world videos remain severely limited. To alleviate this scarcity, recent studies have increasingly explored two pseudo-labeling paradigms: dedicated methods that derive pseudo-labels from unlabeled videos and self-training approaches that reuse a tracker's own predictions as pseudo-labels. These approaches rely on different assumptions and mechanisms, resulting in distinct prediction characteristics and varying reliability across video sequences, frames and points. To account for this variability, we propose a **pseudo-label generation framework** consisting of a Candidate Curator and a Coordinate Recomposer. Specifically, the Candidate Curator constructs context-aware representations from local evidence around the query and candidate points. It then reasons over temporal context and frame-level candidate relations to select reliable candidates for each point and frame. However, independent frame-wise selection can introduce temporal discontinuities in the pseudo-label trajectory. We therefore introduce the Coordinate Recomposer, which models query-conditioned spatiotemporal correspondence to recompose the selected candidate into a temporally coherent point prediction. Extensive experiments demonstrate that the proposed framework produces more accurate pseudo-labels and provides effective supervision for real-world point-tracking adaptation. Fine-tuning existing point trackers with our generated pseudo-labels further improves their performance on real-world videos.

1 INTRODUCTION

Tracking any point (TAP), referred to as point tracking, aims to recover the trajectory and visibility of arbitrary points throughout a video (Doersch et al., 2022). Unlike optical flow, which primarily estimates dense correspondences over short temporal ranges (Teed & Deng, 2020), point tracking requires maintaining point identity across many frames under occlusion and reappearance. This long-term correspondence capability has made point tracking an important primitive for a wide range of

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.

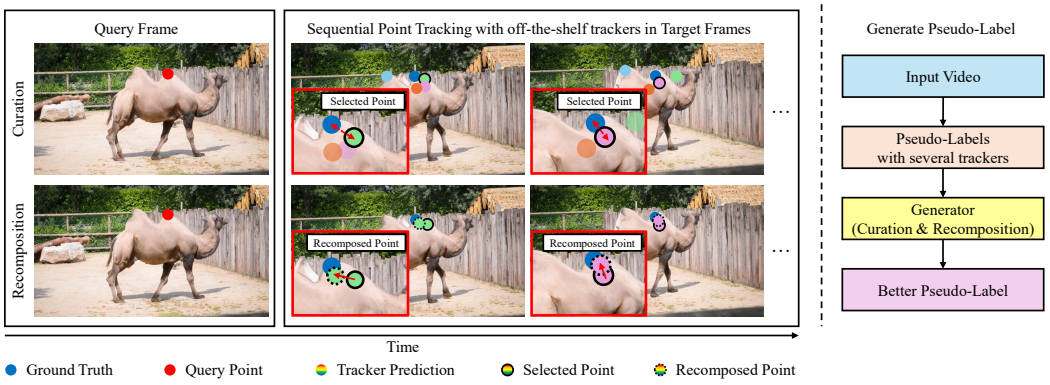


Figure 2: **Overall pipeline for pseudo-label generation.** Given an input video and a query point, multiple pretrained trackers generate candidate trajectories. Our Curator identifies a reliable candidate and the Recomposer generates the final pseudo-label.

downstream applications, including video editing, robot manipulation and visual imitation (Bharadhwaj et al., 2024; Vecerik et al., 2023), visual odometry and structure-from-motion (Chen et al., 2024; Wang et al., 2024) and 4D reconstruction (Feng et al., 2025).

Generalizing to real-world videos is therefore essential for such downstream tasks, yet training robust point trackers on them remains challenging. Dense point trajectories require precise frame-by-frame annotations, making manual labeling at scale prohibitively expensive. Consequently, most point trackers are first trained on synthetic data (Greff et al., 2022; Zheng et al., 2023) and recent methods further adapt them to the real-world domain through pseudo-labeling. These methods reuse predictions from pretrained teacher trackers, either through self-training (Doersch et al., 2024) or by randomly selecting one of multiple teacher trackers (Karaev et al., 2025). However, predictions from pretrained trackers exhibit varying reliability across videos, frames and points, so naively adopting them as pseudo-labels can propagate tracking errors during real-world adaptation. Recent work addresses this variability through verifier-guided pseudo-labeling, which scores predictions from multiple trackers and performs independent discrete selection at each point and frame (Aydemir et al., 2026a).

Despite these advances, selection-based multi-tracker pseudo-labeling still faces two fundamental limitations. First, candidate reliability is determined without explicitly modeling interactions among candidate representations, limiting the ability to reason about their relative relationships. Second, the final pseudo-label remains restricted to the discrete set of tracker predictions. Because selection is performed frame by frame, the selected tracker may change across adjacent frames, introducing temporal discontinuities. As shown in Fig. 1, candidate switching is closely associated with larger frame-to-frame fluctuations. Moreover, even when the most reliable candidate is correctly selected, its residual localization error is directly inherited because the final coordinate remains constrained to the candidate set.

To address these limitations, we introduce a **pseudo-label generation framework** consisting of a Candidate **Curator** and a Coordinate **Recomposer**, as illustrated in Fig. 2. The Candidate Curator first identifies a reliable prediction by jointly modeling interactions between the query and all candidate predictions, as well as relations among the candidates themselves. This allows candidate reliability to be assessed in the context of the entire candidate set rather than from isolated evidence. The selected candidate is then treated as an anchor rather than the final pseudo-label. Building on this anchor, the Coordinate Recomposer performs temporal interaction among selected candidate representations and combines the resulting spatiotemporal context with query-conditioned local correspondence. It estimates a continuous point coordinate beyond the original candidate set, thereby reducing the dependence on discrete frame-wise selection and promoting more temporally coherent pseudo-label trajectories.

Extensive experiments demonstrate that our framework generates more accurate pseudo-labels across real-world benchmarks. Moreover, fine-tuning existing point trackers with our generated pseudo-labels improves their performance on real-world videos. In summary, our contributions are as follows: (1) We identify the **discrete selection problem** in multi-tracker pseudo-labeling, where

frame-wise selection can introduce temporal discontinuities and constrain pseudo-label accuracy. (2) We propose a Candidate **Curator** and Coordinate **Recomposer** that jointly model candidate relations, temporal context and local correspondence to generate continuous point coordinates. (3) We demonstrate state-of-the-art pseudo-label generation and more effective supervision for real-world point-tracking adaptation.

2 Related Work

Point Tracking. Point tracking aims to estimate the location and visibility of the same physical surface point throughout a video from a given query point. Unlike optical flow, which mainly estimates short-term dense correspondences (Teed & Deng, 2020), point tracking requires maintaining point identity over long temporal horizons and handling occlusion and reappearance. Early approaches such as PIPs (Harley et al., 2022) adopted iterative refinement inspired by RAFT, while TAP-Vid (Doersch et al., 2022) established standardized benchmarks and TAPIR (Doersch et al., 2023) combined per-frame matching with iterative trajectory refinement.

Subsequent works have improved point tracking through richer spatial and temporal reasoning. CoTracker (Karaev et al., 2024) introduced joint multi-point tracking to exploit dependencies across trajectories, while TAPTR (Li et al., 2024b) introduced a query-based Transformer architecture for point tracking, followed by further improvements in TAPTRv2 and TAPTRv3 (Li et al., 2024a; Qu et al., 2026). For online point tracking, Track-On (Aydemir et al., 2025) introduced a memory-augmented causal formulation, followed by Track-On2 (Aydemir et al., 2026b) and TAPNext (Zholus et al., 2025), which further explored efficient sequential tracking.

Recent methods have also explored richer correspondence representations and broader tracking formulations. LocoTrack (Cho et al., 2024b) introduced local 4D all-pair correlation to capture richer spatial correspondence than conventional point-to-region matching, while CoTracker3 (Karaev et al., 2025) simplified the CoTracker design while retaining effective spatial and temporal tracking components. FlowTrack (Cho et al., 2024a) revisits optical flow for long-range dense tracking, Dense Optical Tracking (Le Moing et al., 2024) extends point tracking toward dense correspondence estimation, Chrono (Kim et al., 2025) explores temporal representations for point tracking and AllTracker (Harley et al., 2025) targets efficient dense tracking at high resolution.

Point Tracking Datasets and Real-World Supervision. High-performance point trackers require large-scale training data covering diverse motion, appearance changes, illumination variation and occlusion. However, obtaining pixel-accurate point trajectories and visibility annotations in real videos is highly time-consuming. Most existing trackers therefore rely on synthetic datasets, where dense trajectory annotations can be generated automatically. Kubric (Greff et al., 2022) is widely used for point-tracking training, while PointOdyssey (Zheng et al., 2023) provides longer and more diverse synthetic trajectories. Nevertheless, synthetic data cannot fully capture the visual and motion complexity of real videos.

Pseudo-labeling and self-training have been widely studied for exploiting unlabeled data (Lee et al., 2013; Sohn et al., 2020; Xie et al., 2020) and recent point-tracking methods have adopted these strategies to obtain supervision from unlabeled real-world videos. BootsTAPIR (Doersch et al., 2024) adapts a synthetically pretrained tracker to large-scale unlabeled real videos through self-training. CoTracker3 (Karaev et al., 2025) instead generates pseudo-labels by randomly selecting a single pretrained tracker from a pool of teachers for each training batch. These methods rely directly on teacher predictions for supervision. More recently, Aydemir et al. (2026a) introduced verifier-guided pseudo-labeling, which estimates the reliability of multiple tracker predictions and selects the highest-scoring candidate for each point and frame.

However, its query-to-candidate reasoning does not explicitly model interactions among candidate representations and the final pseudo-label remains restricted to discrete frame-wise selection. Consequently, tracker switching across adjacent frames can introduce temporal discontinuities, while residual localization errors are directly inherited from the selected candidate. Our Candidate Curator explicitly models interactions among the query and candidate representations, while the Coordinate Recomposer estimates a continuous point location beyond the discrete candidate set.

3 METHOD

Given a query point $\mathbf{q}$ and candidate trajectories $\{\mathbf{c}_t^{(m)}\}_{m=1}^{M}$ predicted by M pretrained trackers at frame t, our goal is to generate a pseudo-label beyond discrete candidate selection. Our framework first identifies a reliable candidate and then uses it as an anchor to generate a continuous point coordinate. We construct context-aware query and candidate representations in a shared token stream (Sec. 3.1). We then model interactions among candidates and across time through the Token-Stream Transformer (Sec. 3.2). The Candidate Curator then selects a reliable anchor, while the Coordinate Recomposer generates a continuous point coordinate beyond the discrete candidate set (Sec. 3.3). We use the generated pseudo-labels for real-world point-tracker adaptation (Sec. 3.4). An overview is shown in Fig. 2.

3.1 CONTEXT-AWARE TOKENIZATION

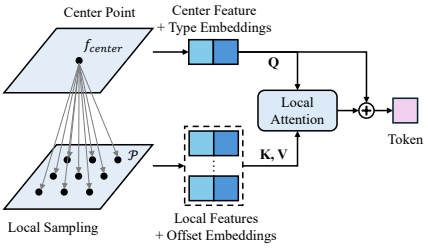


Figure 3: **Local Attention.** Local features sampled around the center point are aggregated by local attention.

Candidate curation requires representations that capture query correspondence and relations among tracker predictions, while coordinate recomposition requires fine-grained local evidence around the selected anchor for precise coordinate estimation. To provide such informative representations, we construct context-aware tokens for the query and candidate predictions. Dense features are extracted using a pretrained CoTracker3 (Karaev et al., 2025) encoder and projected into a common D-dimensional space, from which multi-scale local features are sampled around each point for localized visual representation.

Query Token. For each query point $\mathbf{q}$, we aggregate multi-scale local features around the query using relative offset-aware local attention, as illustrated in Fig. 3:

$$\mathbf{Q} = \mathrm{LocalAttn}\left(\mathbf{f}_q, \mathcal{P}(\mathbf{q}) + \mathbf{E}_\Delta\right) + \mathbf{f}_q, \tag{1}$$

where $\mathbf{f}_q$ is the query-center feature, $\mathcal{P}(\mathbf{q})$ denotes the sampled local features and $\mathbf{E}_\Delta$ denotes the relative offset embeddings.

Candidate Token. Using the same local aggregation as in Eq. 1, we obtain the candidate-local representation $\tilde{\mathbf{C}}_t^{(m)}$ around each predicted location. However, local appearance alone may be insufficient to determine whether the candidate corresponds to the queried point. We therefore condition the candidate representation on the $\mathbf{Q}$ and attend to the candidate neighborhood:

$$\mathbf{C}_t^{(m)} = \mathrm{QueryCondAttn}\left(\mathbf{Q}, \tilde{\mathbf{C}}_t^{(m)}, \mathcal{P}(\mathbf{c}_t^{(m)}) + \mathbf{E}_\Delta\right). \tag{2}$$

This query-conditioned representation emphasizes local evidence that is consistent with the queried point, allowing each candidate token to encode both candidate-local appearance and query correspondence. We further encode the query-to-candidate displacement using $\mathrm{SinCosEmbed}\left(\gamma(\mathbf{c}_t^{(m)} - \mathbf{q})\right)$, providing geometric information complementary to visual correspondence. The query-conditioned visual representation is then combined and projected into the common token space for subsequent candidate curation and coordinate recomposition.

Token Stream. We organize the query and candidate representations into a shared token stream:

$$\mathbf{T}_t = [\mathbf{Q}, \mathbf{C}_t^{(1)}, \ldots, \mathbf{C}_t^{(M)}, \mathbf{S}_t, \mathbf{R}_t], \tag{3}$$

where $\mathbf{S}$ and $\mathbf{R}$ are learnable Select Token and Refine Token, respectively. We add token-type embeddings to distinguish the roles of $\mathbf{Q}$, $\mathbf{C}$, $\mathbf{S}$ and $\mathbf{R}$ tokens. This token stream allows self-attention to explicitly model both query–candidate and inter-candidate relations, supporting more reliable candidate selection and temporally consistent pseudo-label generation.

3.2 TOKEN-STREAM TRANSFORMER

As illustrated in Fig. 4, the representations are processed through L Token-Stream Blocks. In the first $L-1$ shared blocks, token self-attention models interactions among the query and all candi-

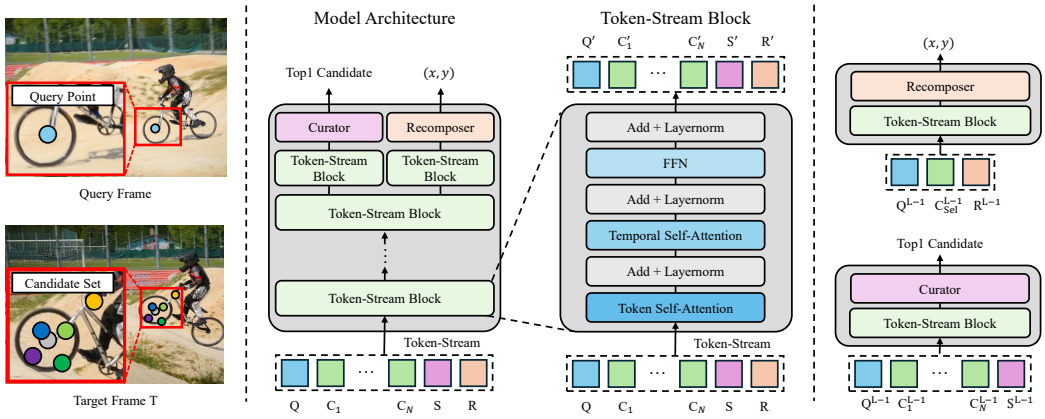


Figure 4: **Overall architecture and Token-Stream Block. (Left)** Query point and candidate set in the query and target frame. **(Mid)** The shared Token-Stream Blocks jointly model $\mathbf{Q}$, $\mathbf{C}$, $\mathbf{S}$ and $\mathbf{R}$ using within-frame and temporal self-attention. **(Right)** After the first $L-1$ shared blocks, the Curator processes the $\mathbf{Q}$, $\mathbf{C}$ and $\mathbf{S}$ to select the top-1 prediction, whereas the Recomposer uses the $\mathbf{Q}$, selected $\mathbf{C}$ and $\mathbf{R}$ to generate the final point coordinate $\hat{\mathbf{x}}$.

dates at each frame, while temporal self-attention propagates information across frames for each token sequence. Through these shared blocks, each token accumulates both candidate-set relational information and temporal context. At the final L-th block, the network separates into the Curator and Recomposer branches. The preceding shared blocks first establish a common representation that captures candidate relations and temporal context, while the final block specializes this representation for the two distinct objectives.

Token Self-Attention. At each frame, multiple trackers provide different candidates for the same query point. Considering each candidate independently discards relation among the predictions, thereby making it more difficult to distinguish reliable candidates when multiple trackers produce similar or conflicting predictions. We therefore jointly process the query and all candidate tokens through token self-attention, enabling each candidate to incorporate both query-specific correspondence and inter-candidate context. Token self-attention is applied over $\mathbf{Q}$, all candidate $\mathbf{C}$ tokens, $\mathbf{S}$ and $\mathbf{R}$ at each frame. Each shared block applies standard multi-head self-attention(MHSA) (Vaswani et al., 2017):

$$\widetilde{\mathbf{T}}_t^{(\ell)} = \mathrm{LN}\left(\mathbf{T}_t^{(\ell-1)} + \mathrm{MHSA}\left(\mathbf{T}_t^{(\ell-1)}\right)\right), \qquad \ell = 1, \ldots, L-1.$$

At the final block, the Curator applies self-attention over $\mathbf{Q}$, $\mathbf{C}$ and $\mathbf{S}$ to estimate their relative reliability, whereas the Recomposer operates on $\mathbf{Q}$, the selected $\mathbf{C}$ and $\mathbf{R}$ to estimate a continuous coordinate from the selected candidate anchor.

Temporal Self-Attention. Candidate reliability can be ambiguous from a single frame, particularly under challenging conditions such as occlusion, motion blur, or locally similar appearances. We therefore apply temporal self-attention independently along the temporal sequence of each token, allowing each representation to incorporate evidence from neighboring frames. In the shared blocks, temporal interaction is applied to each token sequence.

At the final block, the two branches use temporal context according to their respective objectives. The Curator propagates temporal information across the tokens in the curation branch to assess candidate reliability with respect to trajectory-level consistency. The Recomposer instead operates on the tokens in the refinement branch. Since the selected candidate may originate from different trackers across adjacent frames, temporal interaction among the selected candidate representations provides a continuous context across such switches. This allows the Recomposer to estimate coordinates using temporally accumulated evidence, thereby helping maintain continuity in the resulting trajectory.

3.3 PSEUDO-LABEL GENERATION

Our pseudo-label generation process consists of two stages: the **Curator**, which identifies the most reliable candidate from multiple tracker predictions and the **Recomposer**, which reconstructs the final pseudo-label by combining the selected hypothesis with local correspondence evidence.

Candidate Scoring. After spatiotemporal token interaction, we score each candidate using its relation to both the $\mathbf{Q}$ and $\mathbf{S}$. For candidate m, we explicitly model its relation to the $\mathbf{Q}$ and $\mathbf{S}$ using both token differences and element-wise products and construct the ranking representation:

$$\mathbf{r}_m = [\mathbf{C}_m, \mathbf{Q}, \mathbf{S}, \mathbf{C}_m - \mathbf{Q}, \mathbf{C}_m \odot \mathbf{Q}, \mathbf{C}_m - \mathbf{S}, \mathbf{C}_m \odot \mathbf{S}], \tag{4}$$

where $\odot$ denotes element-wise multiplication. The resulting representation is passed through an MLP ranking head to obtain the candidate score,

$$s_m = \alpha \ \mathrm{MLP}_{\mathrm{rank}}(\mathbf{r}_m), \tag{5}$$

where α is a learnable score scale. We denote the score vector over all M candidates as $\mathbf{s} = [s_1, \ldots, s_M]$. This scoring formulation combines candidate-specific evidence, query correspondence and candidate-set context for candidate ranking.

Soft Ranking Supervision. We retain the soft ranking objective of Aydemir et al. (2026a). Let $\hat{\mathbf{x}}_m \in \mathbb{R}^2$ denote the coordinate predicted by candidate m and $\mathbf{x}^{\mathrm{gt}} \in \mathbb{R}^2$ the ground-truth coordinate. We compute the localization error of each candidate as $e_m = \|\hat{\mathbf{x}}_m - \mathbf{x}^{\mathrm{gt}}\|_2$, and denote the error vector as $\mathbf{e} = [e_1, \ldots, e_M]$. We then construct an error-based soft target with temperature $T_s = 0.3$:

$$\mathbf{y} = \mathrm{Softmax}\left(-\frac{\mathbf{e}}{T_s}\right), \tag{6}$$

and optimize the cross-entropy between this target distribution and the predicted candidate scores,

$$\mathcal{L}_{\mathrm{rank}} = \mathrm{CE}\left(\mathbf{y}, \mathrm{Softmax}(\mathbf{s})\right). \tag{7}$$

This soft supervision reflects the relative quality of candidate predictions without enforcing a hard one-hot target, following the original verifier formulation.

Hard-Pair Ranking Supervision. We further complement the soft ranking objective with a focused pairwise loss. We define the oracle-best candidate m^+ as the candidate with the smallest localization error and select the highest-scoring non-oracle candidate as the hard negative m^-:

$$m^+ = \arg\min_m e_m, \qquad m^- = \arg\max_{m \neq m^+} s_m. \tag{8}$$

To avoid enforcing an arbitrary ordering between candidates with nearly identical localization quality, we weight each pair according to its error gap. We then encourage the score of the oracle-best candidate to exceed that of the hard negative:

$$w_{\mathrm{pair}} = \mathrm{clip}\left(\frac{(e_{m^-} - e_{m^+}) - \delta_{\mathrm{tie}}}{\delta_{\mathrm{gap}}}, 0, 1\right), \quad \mathcal{L}_{\mathrm{pair}} = w_{\mathrm{pair}} \, \mathrm{softplus}\left(\frac{\mu - (s_{m^+} - s_{m^-})}{T_p}\right). \tag{9}$$

Thus, near-tied candidates contribute little pairwise supervision, whereas candidates with a clear difference in localization error receive stronger supervision. The complete curation objective is

$$\mathcal{L}_{\mathrm{curator}} = \lambda_{\mathrm{rank}} \mathcal{L}_{\mathrm{rank}} + \lambda_{\mathrm{pair}} \mathcal{L}_{\mathrm{pair}}. \tag{10}$$

Coordinate Recomposition.

Candidate curation provides a reliable anchor, but discrete selection constrains the pseudo-label to the provided tracker predictions. We therefore generate a continuous point coordinate around the selected candidate using temporal context and query-conditioned local correspondence. After candidate scoring, we select the highest-ranked candidate:

$$m^* = \arg\max_m s_m, \tag{11}$$

and use its coordinate $\tilde{\mathbf{x}} = \mathbf{x}_{m^*}$ as the anchor for recomposition.

As illustrated in Fig. 5, we sample local patches around the query and selected candidate and apply patch-to-patch cross-attention. The resulting candidate-side features are conditioned on $\mathbf{Q}$, selected $\mathbf{C}$ and $\mathbf{R}$ to obtain context-enhanced local features.

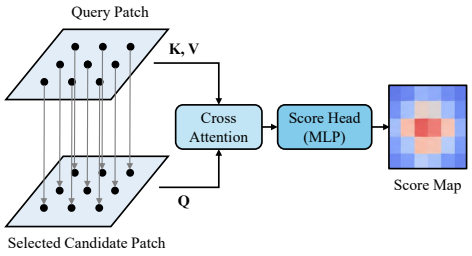


Figure 5: **Local correspondence module.** Query and selected-candidate patches are matched via cross-attention to produce a score map for coordinate recomposition.

Temporal interaction among the selected candidate representations captures context across frames, while query-conditioned local correspondence provides fine-grained spatial evidence around the selected anchor. We sample local patches around the query and selected candidate and apply patch-to-patch cross-attention to extract query-consistent local evidence. The resulting candidate-side features are conditioned on $\mathbf{Q}$, selected $\mathbf{C}$ and $\mathbf{R}$ to obtain context-enhanced local features. These features provide the Recomposer with fine-grained local evidence for coordinate estimation.

Let $\{\boldsymbol{\delta}_k\}_{k=1}^{K}$ denote predefined offsets around the selected candidate. For each offset, an MLP predicts a matching score a_k from corresponding context-conditioned local feature,

$$a_k = \mathrm{MLP}_{\mathrm{score}}\left(\bar{\mathbf{p}}_{c,k}\right), \tag{12}$$

where $\bar{\mathbf{p}}_{c,k}$ denotes the context-enhanced local feature around the selected candidate after patch-to-patch and context cross-attention. We normalize the scores into a probability distribution and compute the residual displacement:

$$p_k = \mathrm{Softmax}(\mathbf{a})_k, \qquad \Delta\mathbf{x} = \sum_{k=1}^{K} p_k \boldsymbol{\delta}_k. \tag{13}$$

The final coordinate is obtained as $\hat{\mathbf{x}} = \tilde{\mathbf{x}} + \Delta\mathbf{x}$. The recomposed objective is

$$\mathcal{L}_{\mathrm{recomposer}} = \mathrm{SmoothL1}\left(\Delta\mathbf{x}, \Delta\mathbf{x}^{\mathrm{gt}}\right). \tag{14}$$

3.4 Real-World Fine-Tuning with Generated Pseudo-Labels

We use the generated pseudo-labels to adapt a synthetically pretrained point tracker to unlabeled real-world videos. For real-world adaptation, we use RGBD1K (Zhu et al., 2023) and generate pseudo-label trajectories using our trained pseudo-label generator. For each real-world clip, we construct query points from four complementary sources. SIFT-based queries (Lowe, 2004) provide appearance-distinctive points, while motion-salient queries are sampled from regions with large temporal changes between adjacent frames. Uniform queries maintain spatial coverage and hard queries emphasize challenging tracking regions such as object boundaries and locally ambiguous areas. Multiple trackers generate candidate trajectories, from which the Candidate Curator selects a reliable candidate and the Coordinate Recomposer generates the final point coordinate.

Before fine-tuning, we filter only trajectories that satisfy candidate agreement, visibility and spatial-validity criteria. We further apply a query curriculum that emphasizes appearance and motion-based queries early in training, while gradually increasing the proportion of uniform and hard queries.

The student is jointly trained on pseudo-labeled RGBD1K videos and synthetic Kubric data using a balanced 1:1 sampling strategy. Real and synthetic samples are processed in separate batches, while only the student inputs are augmented and the cached pseudo-labels remain fixed.

4 Experiments

4.1 Experimental Setup

Implementation Details. We implement our method in PyTorch and conduct all experiments using a single NVIDIA TITAN GPU with 24 GB of memory. The pseudo-label generator is trained on 2K sequences from TAP-Vid Kubric (Greff et al., 2022). Each training clip contains 24 frames and is resized to 384×512 pixels. During training, candidate trajectories are synthetically constructed by perturbing the ground-truth trajectories, allowing the generator to learn candidate curation and coordinate recomposition. At evaluation time, candidate trajectories are instead obtained from six frozen pretrained point trackers. Further optimization settings and training details are provided in the supplementary material.

Table 1: **Quantitative results of Generator.** Comparison with pseudo-label methods across multiple benchmarks.

Model	ITTO-MOSE			Kinetics			DAVIS		
	AJ	δ^x_{avg}	OA	AJ	δ^x_{avg}	OA	AJ	δ^x_{avg}	OA
Random	41.1	56.7	79.3	54.8	67.7	88.7	64.1	76.3	91.0
Verifier (Aydemir et al., 2026a)	45.3	60.8	81.3	58.1	**71.6**	90.2	68.8	81.2	91.8
Generator (Ours)	**45.5**	**61.0**	**81.4**	**58.4**	**71.6**	**90.3**	**69.4**	**81.8**	**92.4**

Table 2: **Quantitative results on real-world datasets.** Comparison of original real-world fine-tuned trackers and models fine-tuned with our generated pseudo-labels.

Model	ITTO-MOSE			Kinetics			DAVIS		
	AJ	$\boldsymbol{\delta^x_{\text{avg}}}$	**OA**	**AJ**	$\boldsymbol{\delta^x_{\text{avg}}}$	**OA**	**AJ**	$\boldsymbol{\delta^x_{\text{avg}}}$	**OA**
Real-World Fine-Tuning									
BootsTAPIR (Doersch et al., 2024)	36.9	51.1	76.4	54.6	68.4	86.5	61.4	73.6	88.7
Anthro-LocoTrack (Kim et al., 2026)	42.8	58.1	79.4	53.9	68.4	86.4	64.8	77.3	89.1
BootsTAPNext-B (Zholus et al., 2025)	40.5	54.0	79.4	57.3	70.6	87.4	65.2	78.5	91.2
CoTracker3 (Window) (Karaev et al., 2025)	42.4	55.8	80.4	55.8	68.5	88.3	63.8	76.3	90.2
Track-On-R (Aydemir et al., 2026a)	45.5	61.4	82.0	57.8	71.0	90.5	68.1	80.3	92.5
Real-World Fine-Tuning with Ours									
BootsTAPIR (Doersch et al., 2024)	38.8	53.3	77.3	55.5	69.4	87.1	62.6	75.0	89.2
Anthro-LocoTrack (Kim et al., 2026)	44.0	59.0	80.2	54.8	69.5	87.0	65.8	78.5	89.7
BootsTAPNext-B (Zholus et al., 2025)	41.7	55.2	80.1	58.3	71.4	87.9	66.2	79.3	91.6
CoTracker3 (Window) (Karaev et al., 2025)	43.3	56.6	80.9	56.7	69.3	88.8	64.8	77.5	90.3
Track-On-R (Aydemir et al., 2026a)	45.3	60.8	82.1	58.1	71.2	90.6	68.4	80.5	92.6

Datasets and Evaluation Protocol. We evaluate pseudo-label generation on three real-world point-tracking benchmarks. All three benchmarks consist of real-world videos. **ITTO-MOSE** (Demler et al., 2025; Ding et al., 2023) uses 44 videos from the MOSE subset of ITTO with verified annotations, featuring dynamic motion and frequent occlusion and reappearance. **TAP-Vid Kinetics** (Doersch et al., 2022; Carreira & Zisserman, 2017) contains 1,144 YouTube videos sampled from the Kinetics-700-2020 validation set, while **TAP-Vid DAVIS** (Doersch et al., 2022; Pont-Tuset et al., 2017) comprises 30 object-centric videos with challenging deformation and occlusion.

For all benchmarks, we follow the queried-first evaluation protocol, where each trajectory is queried at its first visible frame and evaluated on subsequent frames, with predictions mapped to a 256×256 evaluation space before metric computation.

Evaluation Metrics. We report average position accuracy δ^x_{avg}, Average Jaccard (AJ) and Occlusion Accuracy (OA). δ^x_{avg} measures visible-point localization accuracy averaged over thresholds of 1, 2, 4, 8 and 16 pixels. OA measures visibility prediction accuracy, while AJ jointly evaluates localization and visibility, following the TAP-Vid evaluation protocol (Doersch et al., 2022)

4.2 Results

Generated pseudo-labels. Pseudo-label candidates were obtained from an ensemble of six trackers: BootsTAPIR (Doersch et al., 2024), Anthro-LocoTrack (Kim et al., 2026), BootsTAPNext-B (Zholus et al., 2025), CoTracker3 (Window) (Karaev et al., 2025), AllTracker (Harley et al., 2025), and Track-On-R (Aydemir et al., 2026a). The curator then selects the most reliable candidate for each point and the recomposer generates the final point location.

As shown in Table 1, our Generator consistently matches or improves upon the Verifier (Aydemir et al., 2026a) baseline across all three benchmarks. On ITTO-MOSE, δ^x_{avg}, AJ and OA improve from 60.8, 45.3 and 81.3 to 61.0, 45.5 and 81.4, respectively. On Kinetics, our method maintains δ^x_{avg} at 71.6 while improving AJ from 58.1 to 58.4 and OA from 90.2 to 90.3. The largest improvement is observed on DAVIS, where δ^x_{avg} increases from 81.2 to 81.8, AJ from 68.8 to 69.4 and OA from 91.8

Table 3: **Training objective ablation.** Effects of the hard-pair ranking and difference-product (D/P) feature objectives on localization accuracy.

Variant	Hard Pair	D/P	δ^x_{avg}
(i)	✗	✓	81.4
(ii)	✓	✗	81.2
(iii)	✓	✓	**81.8**

Table 4: **Real-world fine-tuning ablation.** Effects of filtering, synthetic–real training and query curriculum on localization accuracy.

Variant	Filter	Syn–Real	Curr.	δ^x_{avg}
(i)	✗	✓	✓	80.2
(ii)	✓	✗	✓	79.6
(iii)	✓	✓	✗	80.4
(iv)	✓	✓	✓	**80.5**

to 92.4. These results indicate that our generator improves pseudo-label quality over selection-based pseudo-labeling.

Supervised Training with Generated Pseudo-Labels. We further evaluate our generated pseudo-labels for real-world adaptation by fine-tuning each point tracker with our pseudo-labels and adaptation strategy and comparing it with its original real-world fine-tuned counterpart.

As shown in Table 2, our approach generally improves performance across different trackers and benchmarks. For example, BootsTAPIR improves δ^x_{avg} from 51.1 to 53.3 on ITTO-MOSE, BootsTAPNext-B from 70.6 to 71.4 on Kinetics and CoTracker3 from 76.3 to 77.5 on DAVIS. Similar improvements are observed in AJ and OA across most tracker–benchmark pairs. Overall, these results demonstrate that our generated pseudo-labels, together with the proposed adaptation strategy, provide effective supervision across different point-tracking architectures.

4.3 Ablation Study

Generated Pseudo-Labels. We ablate the hard-pair ranking loss and difference-product (D/P) features on TAP-Vid DAVIS. The D/P features are the relational terms used in the candidate-scoring formulation in Sec. 3.3, where feature differences and element-wise products capture discrepancy and agreement between each $\mathbf{C}$ and the $\mathbf{Q}$ and $\mathbf{S}$. As shown in Table 3, removing either component degrades localization accuracy. These results indicate that focused supervision on difficult candidate pairs and explicit relational features provide complementary cues for candidate ranking.

Supervised Training with Generated Pseudo-Labels. We further ablate pseudo-label filtering, synthetic–real joint training and query curriculum on TAP-Vid DAVIS. As shown in Table 4, removing any of these components degrades performance. Filtering improves the reliability of pseudo supervision, synthetic–real joint training preserves stable tracking supervision during adaptation and the query curriculum progressively introduces more diverse and challenging queries. Together, these components contribute to effective real-world adaptation.

5 Conclusion

We presented a pseudo-label generation framework that extends discrete candidate selection toward continuous coordinate estimation. The Candidate Curator identifies reliable predictions by modeling candidate relations and temporal context, while the Coordinate Recomposer uses the selected candidate as an anchor to estimate a continuous point coordinate through spatiotemporal context and local correspondence. Our experiments show that the proposed framework produces more accurate pseudo-labels and provides effective supervision for real-world point-tracking adaptation.

A remaining limitation is that pseudo-label quality still depends on the diversity and reliability of the teacher ensemble, particularly when multiple trackers fail consistently. Future work may explore more adaptive pseudo-label generation that remains robust to such collective failures.

REFERENCES

Görkay Aydemir, Xiongyi Cai, Weidi Xie, and Fatma Güney. Track-On: Transformer-based online point tracking with memory. In *International Conference on Learning Representations (ICLR)*, 2025.

Görkay Aydemir, Fatma Güney, and Weidi Xie. Real-world point tracking with verifier-guided pseudo-labeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a.

Görkay Aydemir, Weidi Xie, and Fatma Güney. Track-On2: Enhancing online point tracking with memory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(8):9263–9278, 2026b. doi: 10.1109/TPAMI.2026.3675257.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision (ECCV)*, pp. 306–324, 2024.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. doi: 10.1109/CVPR.2017.502.

Weirong Chen, Le Chen, Rui Wang, and Marc Pollefeys. LEAP-VO: Long-term effective any point tracking for visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19844–19853, 2024.

Seokju Cho, Jiahui Huang, Seungryong Kim, and Joon-Young Lee. FlowTrack: Revisiting optical flow for long-range dense tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19268–19277, 2024a.

Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. In *European Conference on Computer Vision (ECCV)*, 2024b.

Ilona Demler, Saumya Chauhan, and Georgia Gkioxari. Is this tracker on? a benchmark protocol for dynamic tracking. In *NeurIPS*, 2025.

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 (ICCV)*, pp. 20224–20234, October 2023.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, João Carreira, Andrew Zisserman, and Yi Yang. TAP-Vid: A benchmark for tracking any point in a video. In *Advances in Neural Information Processing Systems*, 2022.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, and Andrew Zisserman. BootsTAP: Bootstrapped training for tracking-any-point. In *Asian Conference on Computer Vision (ACCV)*, 2024.

Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J. Black, Trevor Darrell, and Angjoo Kanazawa. ST4RTrack: Simultaneous 4d reconstruction and tracking in the world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 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 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.

Adam W. Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision (ECCV)*, 2022.

Adam W. Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Pavel Tokmakov, Suya You, Rares Ambrus, Katerina Fragkiadaki, and Leonidas J. Guibas. AllTracker: Efficient dense point tracking at high resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *European Conference on Computer Vision (ECCV)*, 2024.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Inès Hyeonsu Kim, Seokju Cho, Jiahui Huang, Jung Yi, Joon-Young Lee, and Seungryong Kim. Exploring temporally-aware features for point tracking. *arXiv preprint arXiv:2501.12218*, 2025.

Inès Hyeonsu Kim, Seokju Cho, Jahyeok Koo, Junghyun Park, Jiahui Huang, Honglak Lee, Joon-Young Lee, and Seungryong Kim. AnthroTAP: Learning point tracking with real-world motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense optical tracking: Connecting the dots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *International Conference on Machine Learning (ICML) Workshops*, 2013.

Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Feng Li, Tianhe Ren, Bohan Li, and Lei Zhang. TAPTRv2: Attention-based position update improves tracking any point. In *Advances in Neural Information Processing Systems*, 2024a.

Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, and Lei Zhang. TAPTR: Tracking any point with transformers as detection. In *European Conference on Computer Vision (ECCV)*, 2024b.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

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.

Jinyuan Qu, Hongyang Li, Shilong Liu, Tianhe Ren, Zhaoyang Zeng, and Lei Zhang. TAPTRv3: Spatial and temporal context foster robust tracking of any point in long video. In *International Conference on Learning Representations (ICLR)*, 2026.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A. Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 596–608, 2020.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. RoboTAP: Tracking arbitrary points for few-shot visual imitation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. VGGSfM: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21686–21697, 2024.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves ImageNet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Artem Zholus, Carl Doersch, Yi Yang, Skanda Koppula, Viorica Patraucean, Xu Owen He, Ignacio Rocco, Mehdi S. M. Sajjadi, Sarath Chandar, and Ross Goroshin. TAPNext: Tracking any point (TAP) as next token prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Xue-Feng Zhu, Tianyang Xu, Zhangyong Tang, Zucheng Wu, Haodong Liu, Xiao Yang, Xiao-Jun Wu, and Josef Kittler. Rgbd1k: A large-scale dataset and benchmark for rgb-d object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 3870–3878, 2023.

A ADDITIONAL ANALYSIS

A.1 SELECTION AND RECOMPOSITION ANALYSIS

We further compare our Curator with the Verifier (Aydemir et al., 2026a) using the same tracker candidates and without applying the Recomposer. As shown in Fig. 6, the fraction of selected candidates within 1 pixel of the ground truth increases from 46.3 to 47.1, while δ^x_{avg} improves from 81.2 to 81.4. These results indicate that the Curator provides more reliable candidate selection than the Verifier under the same candidate set.

After candidate selection, the Coordinate Recomposer further improves δ^x_{avg} from 81.4 to 81.8 by estimating a continuous point coordinate around the selected anchor. Together, these results separate the contribution of improved candidate curation from the additional localization gain provided by coordinate recomposition.

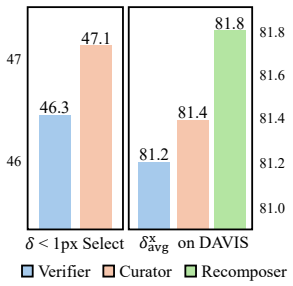


Figure 6: **Selection and recomposition on DAVIS.**

A.2 TEMPORAL DISCONTINUITY ANALYSIS

We further analyze the temporal discontinuities introduced by frame-wise candidate selection on DAVIS. As illustrated in Figure 1, discrete selection can cause the selected tracker to change across adjacent frames, leading to abrupt coordinate transitions. To quantify this behavior, we measure the *candidate switch rate*, defined as the fraction of adjacent visible-frame transitions for which the selected tracker identity changes. We additionally measure the *switching jump* as the 95th-percentile (P95) frame-to-frame displacement at these switching transitions.

The Verifier exhibits a candidate switch rate of 50.6%, whereas the Curator reduces it to 42.3%, indicating that modeling candidate relations and temporal context leads to more stable candidate selection. Since the Recomposer operates on the candidate sequence selected by the Curator without changing candidate identity, its switch rate remains 42.3%. However, the switching jump is reduced from 2.9 pixels for the Curator to 2.3 pixels after recomposition. This result shows that the Curator primarily reduces the frequency of candidate switching, while the Recomposer alleviates the coordinate discontinuity associated with the remaining switches.

Figure 7 provides a complementary frame-level visualization. We plot the signed localization errors along the x and y axes, computed as the difference between the predicted and ground-truth coordinates at each frame. Although the Curator provides more reliable candidate selection, its error trajectories still exhibit abrupt fluctuations across several adjacent frames. In contrast, the Recomposer generally produces smoother variations in both x- and y-direction errors, particularly around frames with large fluctuations in the selected predictions. Together with the reduced switching jump reported in Fig. 1 this behavior supports the role of coordinate recomposition in mitigating temporal discontinuities induced by discrete frame-wise candidate selection.

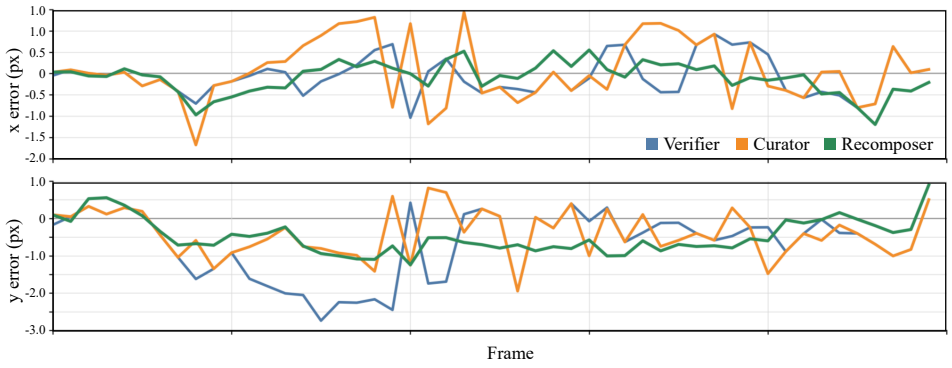


Figure 7: **Frame-wise localization error on DAVIS.** Localization errors along the x and y axes for the Verifier, Curator and Recomposer. The Recomposer suppresses several abrupt fluctuations observed after discrete candidate selection.

B CANDIDATE-SET AUGMENTATION

B.1 SYNTHETIC CANDIDATE GENERATION

Following . (Aydemir et al., 2026a), we generate $M \in \{6, 8, 10, 12\}$ synthetic candidate trajectories by perturbing ground-truth tracks. We retain the six perturbation types of the original verifier and additionally introduce near-confusing trajectories for fine-grained candidate ranking:

1. **Stable noise** ($p = 0.5$): 1-px accumulated noise with temporal smoothing.
2. **Transient spikes** ($p = 0.3$): 2-px accumulated noise with two 8-px spikes.
3. **Gradual drift** (mild $p = 0.25$ / strong $p = 0.2$): 16/32-px switch ranges with 4/8-px noise.
4. **Long-term drift** ($p = 0.05$): gradually increasing displacement with a 32-px scale.
5. **Spiky** (local $p = 0.2$ / large-range $p = 0.05$): identity exchange within 24 px or large-range switches within 128 px.
6. **Complete trajectory switch** ($p = 0.05$): replacement with another point trajectory.
7. **Near-confusing trajectories** ($p = 0.4$): temporally coherent trajectories near the ground-truth track.

All augmentation types, occurrence probabilities and perturbation magnitudes are determined from the training data and fixed before evaluation. For near-confusing trajectories, we sample three difficulty levels with probabilities 0.25/0.4/0.35, random-walk noise scales of 0.4/0.5/0.6 px, and maximum displacement radii of 2.5/3.5/4.5 px, respectively.

B.2 CANDIDATE DISTRIBUTION ANALYSIS

The resulting synthetic candidates exhibit a median localization error of approximately 2.7 px with a long tail of larger failures. Oracle coverage within 2, 4 and 8 px is 84%, 96% and 99.8%, respectively. The median second-best error is 0.96 px and the median oracle-to-second-best gap is only 0.41 px, indicating that the augmentation produces both clear failures and closely competing candidate predictions.

After fixing the augmentation settings, we perform a post-hoc analysis of the candidate distributions produced by the six-tracker ensemble on DAVIS and ITTO-MOSE. On DAVIS, oracle coverage within 1, 2, 4 and 8 px is 64.9%, 85.8%, 95.8% and 99.0%, respectively. These statistics are reported only to characterize the candidate distribution at evaluation and are not used to determine the augmentation settings.

ITTO-MOSE presents substantially more challenging candidate distributions. Across 72,890 valid frame-point pairs, individual candidates have a median error of 2.84 px and a mean error of 16.10 px, while frame-wise oracle selection reduces the mean error to 7.01 px. Oracle coverage within 1, 2, 4 and 8 px is only 41.3%, 57.5%, 70.3% and 79.9%, respectively.

C VISIBILITY AGGREGATION

The Coordinate Recomposer estimates point coordinates, while visibility is determined from the tracker ensemble. We do not introduce an additional visibility-classification head. Instead, we aggregate the visibility predictions of the candidate trackers according to the candidate-ranking scores predicted by the Candidate Curator:

$$w_m = \mathrm{softmax}\left(\frac{s_m}{\tau}\right), \qquad \hat{v} = \mathbb{I}\left[\sum_{m=1}^{M} w_m v_m > 0.5\right], \tag{15}$$

where s_m denotes the ranking logit of candidate m, v_m denotes its visibility prediction and τ controls the sharpness of the ranking-based weights. This formulation assigns larger contributions to visibility predictions from candidates considered more reliable by the Curator.

Visibility Aggregation Strategy. To isolate the contribution of visibility aggregation from coordinate recomposition, we keep the predicted coordinates fixed and vary only the visibility decision

Table 5: **Visibility aggregation strategy.** OA across different visibility decision rules.

Strategy	ITTO-MOSE	DAVIS
Top-ranked	79.9	91.4
Majority vote	81.3	91.8
Rank-weighted	**81.4**	**92.4**

Table 6: **Visibility temperature sensitivity.** OA across different aggregation temperatures τ.

τ	ITTO-MOSE	DAVIS
1	80.62	92.12
2	80.97	**92.50**
5	**81.37**	92.42
10	81.36	92.32
20	81.36	92.28

rule. We compare three strategies: using the visibility prediction of the selected candidate, majority voting across the tracker ensemble and the proposed rank-weighted aggregation. As shown in Table 5, majority voting improves OA over the top-ranked candidate from 79.9 to 81.3 on ITTO-MOSE and from 91.4 to 91.8 on DAVIS. Rank-weighted aggregation further improves OA to 81.4 and 92.4, respectively. These results show that the Curator's ranking scores provide useful reliability cues for aggregating visibility predictions beyond treating all trackers equally.

Temperature Sensitivity. We further analyze the effect of the temperature τ used to convert the Curator's ranking scores into visibility aggregation weights. As shown in Table 6, performance initially improves as τ increases and then remains stable over a broad range of temperatures. On ITTO-MOSE, OA increases from 80.62 at $\tau = 1$ to 81.37 at $\tau = 5$, after which the performance remains nearly unchanged. On DAVIS, the highest OA of 92.50 is obtained at $\tau = 2$, while $\tau = 5$ achieves a comparable OA of 92.42. We therefore use $\tau = 5$, providing consistently high and stable performance.

D Implementation Details

During training, we sample clips of 24 frames with 384 point trajectories from Kubric(Greff et al., 2022). Instead of running pretrained trackers during synthetic training, we construct candidate trajectories by applying synthetic perturbations to the ground-truth trajectories. The number of candidates is randomly sampled from $\{6, 8, 10, 12\}$. At evaluation time, candidate trajectories are generated by six pretrained trackers: BootsTAPIR, Anthro-LocoTrack, BootsTAPNext-B, CoTracker3 (Window), AllTracker and Track-On-R.

Our Curator and Recomposer are built on a six-layer Token-Stream architecture with a 256-dimensional embedding space. The Curator is trained with a soft ranking loss and a hard-pair loss contrasting the candidate closest to the ground truth with the highest-scoring competing candidate. We use a soft-target temperature of 0.3. The hard-pair loss uses a temperature of 1.0, a zero logit margin, a 0.1-pixel tie threshold and a 0.4-pixel error-gap scale. The Recomposer performs patch-to-patch local correspondence around the curated candidate and predicts a residual coordinate correction within a 2-pixel radius.

We optimize the Curator and Recomposer parameter groups using separate AdamW optimizers with cosine learning rate schedules. The ranking and refinement learning rates are 5×10^{-4} and 3×10^{-4}, respectively and the weight decay is 1×10^{-3}. Each training epoch contains 2,000 Kubric clips. We train the Curator for 1,600 optimizer updates. Using a batch size of one on a single GPU, we accumulate gradients over 32 iterations, resulting in an effective batch size of 32. The Recomposer is activated after the first 800 Curator updates and is then trained for 12,500 optimizer updates with a batch size of two. During inference, the Curator selects the highest-scoring candidate trajectory and the Recomposer uses the selected trajectory as an anchor to generate the final coordinate prediction.

E Details of Supervised Training with Generated Pseudo-Labels

We fine-tune each student tracker using pseudo-labeled RGBD1K videos (Zhu et al., 2023) together with synthetic Kubric data (Greff et al., 2022).

Table 7: **Ablation studies of the Curator and Recomposer on DAVIS. (a)** Effect of candidate interaction and local visual evidence in the Curator. **(b)** Effect of temporal attention and patch-to-patch local correspondence in the Recomposer. Higher is better.

(a) Curator

Candidate Interaction	Local Evidence	AJ	δ^x_{avg}
×	×	68.5	80.9
×	✓	68.9	81.2
✓	×	68.7	81.0
✓	✓	**69.3**	**81.8**

(b) Recomposer

Temporal Attention	Patch-to-Patch Correspondence	AJ	δ^x_{avg}
×	×	68.9	81.2
×	✓	69.1	81.4
✓	×	69.0	81.4
✓	✓	**69.3**	**81.8**

1. **Pseudo-label generation.** Six pretrained trackers produce candidate trajectories for each query. The Curator selects a candidate for each point and frame and the Recomposer refines its coordinate to form the final pseudo-label.
2. **Query sampling.** For each RGBD1K clip, we construct a query pool of 1,536 points: 768 SIFT (Lowe, 2004), 384 motion-salient, 192 uniform and 192 hard queries. Each training crop then samples up to $N = 256$ quality-valid queries from this pool. Motion-salient queries are sampled from regions with large inter-frame changes, while hard queries emphasize moving image boundaries.
3. **Quality filtering.** We retain queries with at least two valid post-query frames. A valid prediction must be visible, spatially valid and supported by the teacher ensemble: at least three teachers must lie within 8 px of the selected coordinate.
4. **Spatially diverse SIFT queries.** For SIFT queries, we partition each frame into a 12×16 grid and sample across cells to avoid concentration in highly textured regions.
5. **Query curriculum.** We progressively shift the SIFT:motion:uniform:hard sampling ratio from an appearance- and motion-dominated 2:1:0:0 mix to a more diverse 4:2:1:1 mix. This curriculum first provides stable supervision from reliable correspondences, then increases exposure to less distinctive and more challenging tracking regions.
6. **Synthetic–Real Mixing.** The student is jointly trained on synthetic videos from Kubric data and pseudo-labeled real-world videos from RGBD1K with a fixed 1:1 sampling ratio. Synthetic samples use ground-truth trajectories and visibility, while real-world samples use generated pseudo-labels.
7. **Fine-Tuning Protocol.** For each student tracker, we retain its original architecture and initialize it from the corresponding pretrained model. We then fine-tune the tracker on pseudo-labeled real-world videos using our generated pseudo-labels and the real-world adaptation strategy described above.

F ADDITIONAL ABLATION STUDY

We further analyze the design of the Curator and Recomposer. Table 7 summarizes the results.

F.1 CURATOR ABLATIONS.

Candidate interaction and local evidence both contribute to candidate curation. As shown in Table 7, removing candidate interaction decreases AJ from 69.3 to 68.9 and δ^x_{avg} from 81.8 to 81.2, demonstrating the benefit of relational reasoning across tracker predictions. Removing local evidence also reduces performance to 68.7 AJ and 81.0 δ^x_{avg}, highlighting the importance of localized visual context for candidate assessment.

F.2 RECOMPOSER ABLATIONS

We evaluate the contributions of temporal attention and patch-to-patch local correspondence in the Recomposer. As shown in Table 7, removing temporal attention decreases performance from 69.3

to 69.1 AJ and from 81.8 to 81.4 δ^x_{avg}, indicating that temporal context contributes to coordinate recomposition. Removing patch-to-patch correspondence similarly reduces performance to 69.0 AJ and 81.4 δ^x_{avg}, demonstrating the importance of fine-grained local correspondence around the selected anchor.

G VISUALIZATION

Pseudo-Label Generation. Figure 8 shows qualitative examples of pseudo-label generation on DAVIS and ITTO-MOSE. The top and bottom rows correspond to DAVIS and ITTO-MOSE, respectively. Tracker predictions are color-coded as follows: BootsTAPIR in green, Anthro-LocoTrack in light blue, BootsTAPNext in orange, CoTracker3 in purple, AllTracker in brown and Track-On-R in blue. The square marker denotes the candidate selected by the Curator and retains the color of the corresponding tracker. The yellow square indicates the coordinate generated by the Recomposer, while the red cross denotes the ground-truth location.



Figure 8: **Qualitative visualization of pseudo-label generation.** Examples from DAVIS (top) and ITTO-MOSE (bottom), showing tracker candidates, Curator selection, Recomposer output and ground truth.

Qualitative Comparison. Figure 9 compares pseudo-label trajectories generated by the Verifier and our Generator on challenging real-world sequences. Each color denotes the same query point across frames. While the Verifier exhibits noticeable trajectory variations for several points under occlusion and complex motion, our Generator produces more temporally consistent point trajectories by recomposing the selected candidates with temporal context and local correspondence.



Figure 9: **Qualitative comparison of generated pseudo-labels.** Verifier and our Generator are compared across real-world sequences. Each color represents the same query point across frames, with our Generator producing more temporally consistent trajectories.