MoPRU: Learning Motion Prediction and Recurrent Motion-State Update for 3D Single Object Tracking

Anonymous authors
Paper under double-blind review

Abstract

Motion-centric methods for LiDAR-based 3D single-object tracking (3D SOT) estimate relative target motion (RTM) from consecutive point-cloud frames for current target localization. Historical tracking information is further exploited for temporal motion modeling. However, learning the motion update only for current localization overlooks its effect on the recurrent motion state used for subsequent motion prediction. To address this issue, we propose MoPRU, a 3D SOT framework that unifies the learning of motion prediction and recurrent motion-state update. MoPRU learns how current motion information should be integrated into the recurrent motion state, extending motion-update learning beyond current-frame localization. Specifically, MoPRU aggregates historical tracking features into a recurrent motion state and predicts a motion prior for the current frame. The motion prior and current RTM determine a shared motion update, which refines the current RTM for target localization and updates the recurrent motion state for subsequent motion prediction. This motion update is learned by comparing short-horizon tracking results under alternative current updates. MoPRU achieves Success/Precision scores of 73.8/91.0 on KITTI and 62.93/75.66 on NuScenes, and improves zero-shot KITTI-to-Waymo transfer from 47.5/63.6 to 50.3/68.3.

1 Introduction

Given an initial 3D bounding box, LiDAR-based 3D single-object tracking (3D SOT) aims to localize a target throughout a point-cloud sequence. Sparse and incomplete observations limit the reliability of appearance matching. Motion-centric methods address this challenge by estimating relative target motion (RTM) between consecutive point clouds for current target localization. M^2-Track (Zheng et al., 2022) established this paradigm, followed by methods that further improve inter-frame motion estimation (Nie et al., 2025; Yang et al., 2026; Zhou et al., 2025; 2026).

Temporal trackers further exploit historical templates, bounding box sequences, and memory to improve current target localization (Lan et al., 2022; Xu et al., 2023; Lin et al., 2024; Luo et al., 2024; Fan et al., 2026; Yoo et al., 2026; Liu et al., 2026). However, current localization alone does not fully characterize the effect of a motion update. As shown in Figure 1(a), 33.4% of update pairs with comparable current localization lead to different subsequent tracking outcomes. Similar current target boxes therefore do not necessarily imply similar effects on subsequent tracking.

This difference arises from the dual effect of the current RTM update. As shown in Figure 1(b), the update refines the RTM for current target localization while also updating the recurrent motion state used for subsequent motion prediction. Learning a motion update only from current localization therefore overlooks its effect on subsequent tracking through the recurrent motion state. This motivates a unified learning framework for motion prediction and recurrent motion-state update, so that the current motion update accounts for both current target localization and subsequent motion prediction.

To this end, we propose MoPRU, a 3D SOT framework that jointly learns motion prediction and recurrent motion-state update. MoPRU aggregates historical tracking features into a recurrent motion state and predicts a motion prior for the current frame. Given the motion prior and current RTM, MoPRU predicts shared update weights for RTM refinement and recurrent motion-state update. The

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.

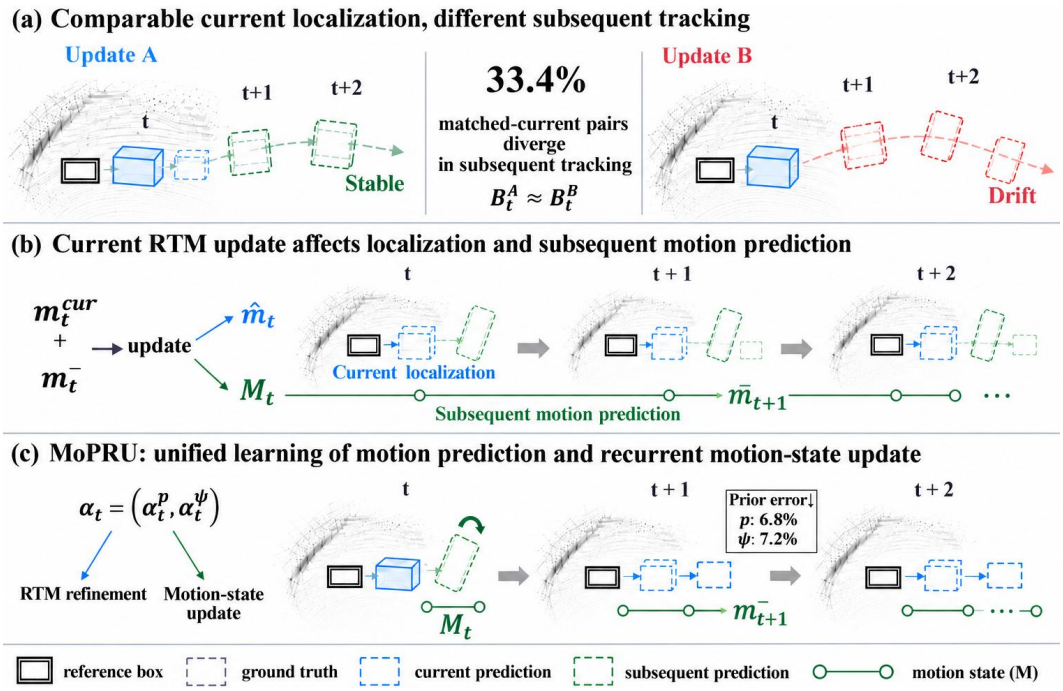


Figure 1: **RTM updates affect current localization and subsequent motion prediction.** (a) Updates with comparable current localization can lead to different subsequent tracking outcomes. (b) The same update changes the recurrent motion state used to predict the motion prior for the next frame. (c) MoPRU unifies RTM refinement and recurrent motion-state update, reducing translation/yaw errors of the predicted motion prior by 6.8%/7.2%.

same motion update refines the current RTM for target localization and updates the recurrent motion state for subsequent motion prediction. We learn this shared motion update by comparing the tracking outcomes of alternative current updates over subsequent frames, as shown in Figure 1(c).

Our contributions are threefold:

- **Learning Framework for Motion Prediction and Recurrent Motion-State Update:** We propose MoPRU, a 3D SOT framework that unifies the learning of motion prediction and recurrent motion-state update. A shared motion update refines the current RTM and controls the motion information integrated into the recurrent state for subsequent motion prediction.
- **Shared Motion Update for Recurrent Prediction:** We learn a shared motion update by comparing short-horizon tracking results under alternative current updates. This supervision guides current RTM refinement and motion-state update to improve subsequent motion prediction.
- **Generality across 3D SOT Frameworks:** MoPRU consistently improves different 3D SOT trackers on KITTI and NuScenes, and further improves zero-shot KITTI-to-Waymo transfer, demonstrating its generality across frameworks and datasets.

2 Related Work

Motion-centric 3D SOT. M^2-Track (Zheng et al., 2022) introduced inter-frame RTM estimation as an alternative to template–search appearance matching. P2P (Nie et al., 2025) captures fine-grained motion through part-to-part comparison, while SiamMo (Yang et al., 2026) decouples Siamese feature extraction from temporal fusion. FocusTrack (Zhou et al., 2025) combines motion modeling with foreground semantics, while CompTrack (Zhou et al., 2026) reduces spatial and foreground redundancy. These methods mainly improve current inter-frame motion estimation or representation. MoPRU extends motion-centric learning beyond current RTM estimation by learning how current motion information updates the recurrent motion state for subsequent motion prediction.

Temporal modeling in 3D SOT. Temporal trackers retain historical information through templates and memory (Lan et al., 2022; Xu et al., 2023; Luo et al., 2024), or jointly model point-cloud and bounding-box sequences (Lin et al., 2024). TrajTrack (Fan et al., 2026) exploits historical box trajectories to model motion continuity and refine current proposals. ChronoTrack (Yoo et al., 2026) improves long-term memory through temporal feature consistency, while PointRePar (Liu et al., 2026) captures point-level and box-level temporal relations. MoPRU instead couples current RTM refinement with recurrent motion-state update, allowing historical motion information to support both current localization and subsequent motion prediction.

Learning motion prediction and recurrent motion-state update. Beyond LiDAR-based 3D SOT, related studies learn temporal prediction or state updates from data. TGTrack (Geng et al., 2026) uses autoregressive future generation to learn temporal representations. Learned filtering methods also adapt motion updates: KalmanNet (Revach et al., 2022) learns Kalman gains from data, while DIMM (Zha et al., 2026) uses reinforcement learning to combine motion-model estimates along different spatial directions. MoPRU instead targets the coupling between current RTM refinement and recurrent motion-state update in 3D SOT. It learns component weights shared by these two processes using subsequent tracking outcomes under alternative current updates.

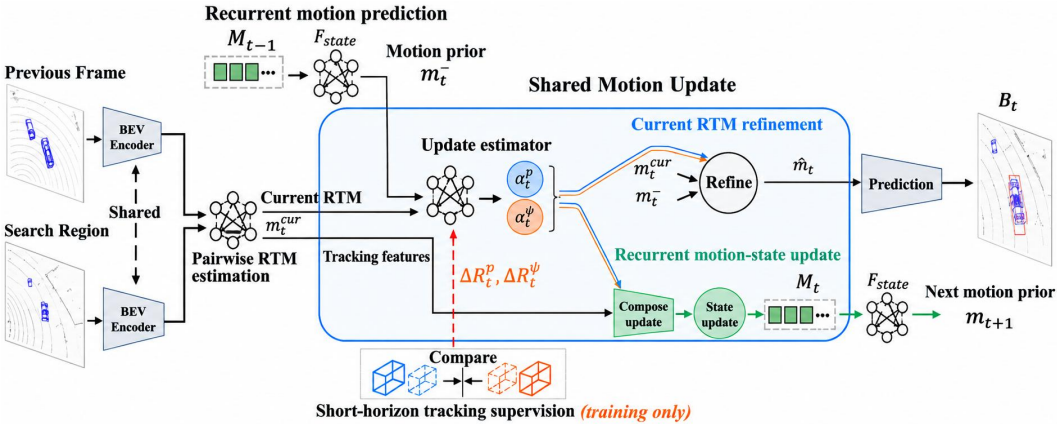


Figure 2: **Overview of MoPRU.** MoPRU unifies the learning of motion prediction and recurrent motion-state update. A predicted motion prior and the current pairwise RTM determine shared update weights for current RTM refinement and recurrent motion-state update. These weights are learned by comparing short-horizon tracking outcomes under different current updates.

3 METHOD

3.1 OVERVIEW AND RECURRENT MOTION PREDICTION

MoPRU unifies the learning of motion prediction and recurrent motion-state update (Figure 2). Shared update weights refine the current RTM and construct the recurrent write for subsequent motion prediction.

At frame t, the tracker receives the previous target box B_{t-1}, previous-frame point cloud P_{t-1}, and current search region P_t. The pairwise predictor F_{pair} (Yang et al., 2026) estimates the current 4-DoF RTM m_t^{cur}, comprising 3D translation p and yaw change ψ. Following MambaLCT (Li et al., 2025), a fixed-size recurrent motion state M_{t-1} aggregates historical tracking features. The state predictor produces $(m_t^-, h_{t-1}) = F_{\text{state}}(M_{t-1})$, where m_t^- is the motion prior and h_{t-1} is the recurrent context. During training, ground-truth RTM supervises the motion prior m_t^-.

3.2 LEARNED SHARED MOTION UPDATE

Update-weight prediction. We compute translation and yaw residuals between the current RTM and motion prior:

$$z_t^p = m_{t,p}^{\text{cur}} - m_{t,p}^-, \qquad z_t^\psi = \text{wrap}(m_{t,\psi}^{\text{cur}} - m_{t,\psi}^-). \tag{1}$$

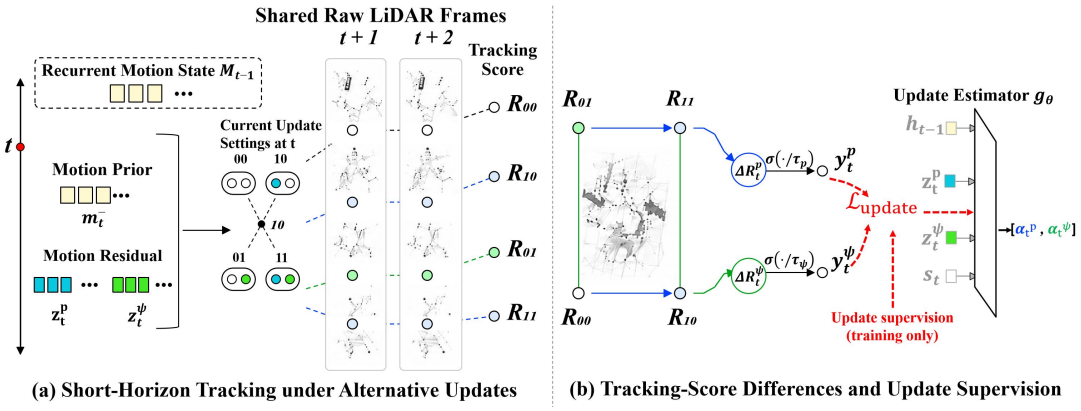


Figure 3: **Short-horizon tracking supervision.** Four current-update settings start from the same box and recurrent motion state. Each tracking branch uses the same LiDAR frames but produces its own boxes, crops, and recurrent states; subsequent weights are predicted by the same update predictor. Tracking-score differences supervise the shared update weights.

Here $\mathrm{wrap}(\cdot)$ maps yaw differences to the principal interval. The update estimator predicts the component weights as

$$(\ell_t^p, \ell_t^\psi) = g_\theta(h_{t-1}, z_t^p, z_t^\psi, s_t), \qquad \alpha_t^j = \sigma(\ell_t^j), \quad j \in \{p, \psi\}. \tag{2}$$

Here s_t comprises the pairwise motion feature, point counts, object size, prior scale, and uncertainty estimates. The estimator input is 657-D.

Current RTM refinement. The weights control a bounded correction toward the prior:

$$\begin{aligned} \hat{m}_{t,p} &= m_{t,p}^{\mathrm{cur}} - (1 - \alpha_t^p)\,\mathrm{clip}(z_t^p; \mathbf{c}_p), \\ \hat{m}_{t,\psi} &= \mathrm{wrap}\Big(m_{t,\psi}^{\mathrm{cur}} - (1 - \alpha_t^\psi)\,\mathrm{clip}(z_t^\psi; c_\psi)\Big). \end{aligned} \tag{3}$$

The category-specific bounds are $\mathbf{c}_p = (c_x, c_y, c_z)$ and c_ψ, with element-wise clipping for translation. A weight of 1 retains the current component; 0 applies the maximum bounded correction toward the motion prior. The refined RTM $\hat{m}_t$ transforms B_{t-1} into the current target box B_t. Appendix C evaluates finer component decompositions.

Recurrent motion-state update. The recurrent write q_t combines the pairwise motion feature f_t with component-specific translation and yaw motion descriptors r_t^p and r_t^ψ using the shared weights:

$$\bar{\alpha}_t = \tfrac{1}{2}(\alpha_t^p + \alpha_t^\psi), \qquad q_t = \mathrm{RMSNorm}\Big(\bar{\alpha}_t E_f(f_t) + \alpha_t^p E_p(r_t^p) + \alpha_t^\psi E_\psi(r_t^\psi)\Big). \tag{4}$$

The embeddings E_f, E_p, E_ψ map these inputs to a common feature space. RMSNorm removes the common scale of the weighted sum, so the shared weights primarily control the relative feature composition. The state update and subsequent motion prediction are

$$M_t = \mathrm{Write}(M_{t-1}, q_t), \qquad (m_{t+1}^-, h_t) = F_{\mathrm{state}}(M_t). \tag{5}$$

The shared weights control both RTM refinement and recurrent motion-state update. Write updates the state once per frame; F_{state} predicts the next motion prior. Section 4.3 compares shared and separate update weights.

3.3 Short-Horizon Tracking Supervision

We learn the shared weights by comparing tracking outcomes under four current-update settings $a = (a^p, a^\psi) \in \{0, 1\}^2$, which replace α_t in Equations 3 and 4. For component j, $a^j = 1$ retains the current RTM component and includes its descriptor, whereas $a^j = 0$ applies the maximum bounded correction and excludes it. At $a = (0, 0)$, $q_t = 0$ and $M_t = \mathrm{Write}(M_{t-1}, 0)$.

All four branches start from the same previous box and recurrent motion state, applying different shared motion updates only at t. Thereafter, each predicts weights from its own observations and recurrent state using the same update predictor with fixed parameters. (Figure 3).

For predicted box B^a_{t+k} and ground-truth box B^*_{t+k}, the tracking error combines 3D overlap, center distance, and yaw error:

$$\begin{aligned}\rho^a_{t+k} = 1 &- \mathrm{clip}\big(\mathrm{IoU}_{3D}(B^a_{t+k}, B^*_{t+k}); [0,1]\big) \\ &+ \lambda_p \min\left(\frac{d_c(B^a_{t+k}, B^*_{t+k})}{\kappa^{\mathrm{err}}_p}, 1\right) \\ &+ \lambda_\psi \min\left(\frac{|\Delta\psi(B^a_{t+k}, B^*_{t+k})|}{\kappa^{\mathrm{err}}_\psi}, 1\right).\end{aligned} \tag{6}$$

Here d_c is the box-center distance and $\Delta\psi$ is the wrapped yaw difference. We use $\lambda_p = 0.50$, $\lambda_\psi = 0.20$, $\kappa^{\mathrm{err}}_p = 2\,\mathrm{m}$, and $\kappa^{\mathrm{err}}_\psi = 30^\circ$. The tracking score combines the current error with discounted errors over the next H frames:

$$R_a = -\left(\rho^a_t + \sum_{k=1}^{H} \gamma^k \rho^a_{t+k}\right), \qquad H = 2, \quad \gamma = 0.90. \tag{7}$$

Higher scores indicate better tracking. Appendix B evaluates the score definition and yaw weighting.

For each component, we average its tracking-score differences over the values of the other component:

$$\Delta R^p_t = \tfrac{1}{2}[(R_{10} - R_{00}) + (R_{11} - R_{01})], \tag{8}$$

$$\Delta R^\psi_t = \tfrac{1}{2}[(R_{01} - R_{00}) + (R_{11} - R_{10})]. \tag{9}$$

Positive ΔR^j_t favors retaining the current component; negative values favor its prior-corrected setting. These differences satisfy

$$\Delta R^p_t + \Delta R^\psi_t = R_{11} - R_{00}, \tag{10}$$

decomposing the joint tracking-score difference into translation and yaw contributions.

3.4 Training and Online Tracking

The tracking-score differences define continuous targets $y^j_t = \sigma(\Delta R^j_t / \tau_j)$, where τ_j controls target sharpness. The update estimator minimizes

$$\mathcal{L}_{\mathrm{update}} = \frac{1}{2} \sum_{j \in \{p,\psi\}} \mathrm{BCEWithLogits}(\ell^j_t, y^j_t). \tag{11}$$

Appendix C compares binary settings, the best common-weight setting, and a dense component-wise weight-grid setting. The joint training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{pair}} + \lambda_h \mathcal{L}_{\mathrm{prior}} + \lambda_u \mathcal{L}_{\mathrm{update}} + \lambda_r \mathcal{L}_{\mathrm{refine}}. \tag{12}$$

The terms supervise pairwise RTM, motion-prior prediction, shared update weights, and refined RTM, respectively. The prior loss combines a Laplace negative log-likelihood with 0.25 times a Smooth-L1 loss on normalized motion residuals. Appendix A gives the loss definitions and training schedule.

We detach the tracking-score targets when training g_θ. For $\mathcal{L}_{\mathrm{refine}}$, α_t is held constant while gradients update F_{pair} and F_{state}; $\mathcal{L}_{\mathrm{update}}$ supervises g_θ. During online tracking, g_θ predicts both weights from past and current tracking information. Both weights are set to 1 until recurrent history is available. Subsequent frames provide supervision during training only.

4 Experiments

4.1 Experimental Setup

Datasets and protocols. Following SiamMo (Yang et al., 2026), we evaluate on KITTI (Geiger et al., 2012) (0–16/17–18/19–20), NuScenes (Caesar et al., 2020) (700/150 train/val), and Waymo (Sun et al., 2020). Success and Precision are AUCs of 3D-IoU and center-distance curves. Zero-shot KITTI-to-Waymo evaluates the KITTI-trained model without Waymo-specific training or tuning.

Implementation. SiamMo is the default motion-centric backbone. Training uses five-frame clips ($H = 2$, $\gamma = 0.90$), with the recurrent motion state reset per clip and propagated over the full test tracklet. Across five seeds, MoPRU achieves $73.80 \pm 0.11/91.04 \pm 0.08$ on KITTI. Appendices A, B, and D report implementation details, short-horizon analysis, and multi-seed stability, respectively.

4.2 Comparison with State of the Art

Table 1: **Comparison with state of the art on KITTI.** Success/Precision (%); baseline Means follow sources, MoPRU Mean is five-seed frame-weighted; efficiency follows sources.

Method Frame Count	Car [6,424]	Pedestrian [6,088]	Van [1,248]	Cyclist [308]	Mean [14,068]	FLOPs	FPS	Device
P2B (Qi et al., 2020)	56.2/72.8	28.7/49.6	40.8/48.4	32.1/44.7	42.4/60.0	4.30G	40	GTX1080Ti
BEVTrack (Yang et al., 2025)	74.9/86.5	69.5/94.3	66.0/77.2	77.0/94.7	71.8/89.2	0.76G	200	RTX4090
V2B (Hui et al., 2021)	70.5/81.3	48.3/73.5	50.1/58.0	40.8/49.7	58.4/75.2	–	–	–
STNet (Hui et al., 2022)	72.1/84.0	49.9/77.2	58.0/70.6	73.5/93.7	61.3/80.1	3.14G	35	TITANRTX
M^2-Track (Zheng et al., 2022)	65.5/80.8	61.5/88.2	53.8/70.7	73.2/93.5	62.9/83.4	2.54G	57	V100
P2P-v (Nie et al., 2025)	73.6/85.7	**69.6**/94.0	70.3/83.9	75.5/94.6	71.7/89.4	1.23G	71	RTX3090
FocusTrack (Zhou et al., 2025)	74.1/85.9	69.3/94.1	68.4/83.5	75.9/94.7	71.3/89.4	–	105	RTX3090
CompTrack (Zhou et al., 2026)	73.4/85.2	69.5/**94.7**	68.5/82.5	76.0/94.8	71.4/89.3	0.94G	90	RTX3090
SiamMo (Yang et al., 2026)	76.3/88.1	68.6/93.9	67.9/80.5	78.5/94.8	72.3/90.1	0.82G	108	RTX4090
MBPTrack (Xu et al., 2023)	73.4/84.8	68.6/93.9	61.3/72.7	76.7/94.3	70.3/87.9	2.88G	50	RTX3090
PointRePar (Liu et al., 2026)	76.7/87.5	67.4/91.9	69.4/82.2	74.9/94.1	72.0/89.1	6.66G	37	RTX3090
ChronoTrack (Yoo et al., 2026)	76.0/88.6	68.6/94.0	64.2/77.7	77.7/94.8	71.8/90.1	–	–	–
SSMTrack (Xiao et al., 2026)	74.8/86.2	67.8/92.2	65.8/78.0	**79.8**/95.0	71.1/88.3	–	–	–
MambaTrack3D (Tian et al., 2025)	71.4/84.3	69.4/94.1	64.0/74.8	78.3/**95.2**	70.0/87.9	–	–	–
MoPRU (ours)	**78.4/89.1**	67.7/94.2	**78.8/84.6**	79.1/95.1	**73.8/91.0**	1.39G	101	RTX4090

Table 2: **Comparison with state of the art on NuScenes.** Per-category Success/Precision (%); baseline Means follow sources, MoPRU Mean is frame-weighted. $\text{PointRePar}_{\text{CS}}$ is category-specific.

Method Frame Count	Car [64,159]	Pedestrian [33,227]	Truck [13,587]	Trailer [3,352]	Bus [2,953]	Mean [117,278]
P2B (Qi et al., 2020)	38.81/43.18	28.39/52.24	42.95/41.59	48.96/40.05	32.95/27.41	36.48/45.08
BEVTrack (Yang et al., 2025)	64.31/71.14	46.28/76.77	66.83/67.04	74.54/**71.62**	61.09/56.68	59.71/71.19
M^2-Track (Zheng et al., 2022)	55.85/65.09	32.10/60.92	57.36/59.54	57.61/58.26	51.39/51.44	49.23/62.73
P2P-v (Nie et al., 2025)	65.15/72.90	46.43/75.08	64.96/65.96	70.46/66.86	59.02/56.56	59.84/72.13
FocusTrack (Zhou et al., 2025)	66.50/74.37	47.85/77.53	**69.34**/69.98	**74.76**/71.20	64.02/61.67	61.72/74.35
CompTrack (Zhou et al., 2026)	65.70/73.50	47.86/77.52	68.19/69.78	72.89/68.11	61.74/58.88	61.04/73.68
SiamMo (Yang et al., 2026)	64.95/72.24	46.23/76.25	68.22/68.81	74.21/70.63	65.63/62.07	60.31/72.68
MBPTrack (Xu et al., 2023)	62.47/70.41	45.32/74.03	62.18/63.31	65.14/61.33	55.41/51.76	57.48/69.88
SeqTrack3D (Lin et al., 2024)	62.55/71.46	39.94/68.57	60.97/63.04	68.37/61.76	54.33/53.52	55.92/68.94
TrajTrack (Fan et al., 2026)	68.02/75.87	48.32/78.78	67.19/68.44	70.70/68.02	61.16/57.67	62.25/75.15
ChronoTrack (Yoo et al., 2026)	64.5/73.0	47.0/76.1	66.2/68.2	67.3/64.0	60.0/57.6	59.7/72.7
SSMTrack (Xiao et al., 2026)	64.00/72.13	46.22/75.20	66.25/68.60	68.82/66.65	61.16/58.84	59.29/72.10
$\text{PointRePar}_{\text{CS}}$ (Liu et al., 2026)	**71.29/78.20**	**48.36/78.98**	68.89/**71.27**	71.57/70.56	56.40/55.26	**64.15/76.81**
MoPRU (ours)	68.75/76.03	47.88/**78.98**	69.23/70.21	72.52/69.65	**65.91/62.13**	62.93/75.66

Table 3: **Zero-shot KITTI-to-Waymo transfer.** Success/Precision (%) from direct KITTI-to-Waymo evaluation. Difficulty-level results follow the cited sources.

Method	Vehicle				Pedestrian			
	Easy	Medium	Hard	Mean	Easy	Medium	Hard	Mean
Frame Count	[67,832]	[61,252]	[56,647]	[185,731]	[85,280]	[82,253]	[74,219]	[241,752]
P2B (Qi et al., 2020)	57.1/65.4	52.0/60.7	47.9/58.5	52.6/61.7	18.1/30.8	17.8/30.0	17.7/29.3	17.9/30.1
BAT (Zheng et al., 2021)	61.0/68.3	53.3/60.9	48.9/57.8	54.7/62.7	19.3/32.6	17.8/29.8	17.2/28.3	18.2/30.3
V2B (Hui et al., 2021)	64.5/71.5	55.1/63.2	52.0/62.0	57.6/65.9	27.9/43.9	22.5/36.2	20.1/33.1	23.7/37.9
STNet (Hui et al., 2022)	65.9/72.7	57.5/66.0	54.6/64.7	59.7/68.0	29.2/45.3	24.7/38.2	22.2/35.8	25.5/39.9
M^2-Track (Zheng et al., 2022)	68.1/75.3	58.6/66.6	55.4/64.9	61.1/69.3	35.5/54.2	30.7/48.4	29.3/45.9	32.0/49.7
P2P-v (Nie et al., 2025)	66.2/73.8	57.8/67.0	56.8/68.1	60.0/69.1	43.7/65.2	36.4/57.1	31.3/51.0	37.4/58.1
MBPTrack (Xu et al., 2023)	68.5/77.1	58.4/68.1	57.6/69.7	61.9/71.9	37.5/57.0	33.0/51.9	30.0/48.8	33.7/52.7
FocusTrack (Zhou et al., 2025)	69.2/77.9	**59.5**/69.3	58.7/**70.8**	**62.8**/72.9	44.5/66.1	38.5/59.4	33.9/53.5	39.3/60.0
CompTrack (Zhou et al., 2026)	68.2/72.8	57.4/66.0	57.2/69.4	61.2/69.6	43.8/68.6	**38.7/62.6**	33.8/56.0	39.0/62.7
SiamMo (Yang et al., 2026)	66.1/73.4	58.1/67.9	56.6/68.7	60.6/70.1	44.8/66.2	35.6/56.8	31.2/51.8	37.5/58.6
PointRePar (Liu et al., 2026)	67.3/78.0	58.3/70.2	56.0/70.3	60.9/73.1	45.8/70.8	36.5/58.9	35.1/58.0	40.1/64.4
MoPRU (ours)	**69.5/78.1**	58.9/**70.3**	**58.9**/70.5	**62.8/73.2**	**45.9/70.9**	38.4/**62.6**	**37.2/59.1**	**40.7/64.5**

In-domain tracking. On KITTI, MoPRU improves SiamMo from 72.3/90.1 to 73.8/91.0 mean Success/Precision (Table 1). Van obtains the largest Success gain of 10.9 points, while Pedestrian shows a $-0.9/+0.3$ change in Success/Precision. Category-wise results are further analyzed in Appendix F. On NuScenes, MoPRU improves the mean from 60.31/72.68 to 62.93/75.66 (Table 2). The gains on both datasets show consistent improvements across different tracking scenarios. MoPRU runs at 101 FPS; matched-hardware efficiency and training cost are reported in Appendix H.

Zero-shot KITTI-to-Waymo transfer. Without Waymo-specific training or tuning, the KITTI-trained MoPRU achieves 62.8/73.2 on Vehicle and 40.7/64.5 on Pedestrian (Table 3). The frame-weighted mean over 427,483 frames is 50.3/68.3, improving SiamMo by 2.8/4.7 points. On hard Pedestrian, Success/Precision increases from 31.2/51.8 to 37.2/59.1.

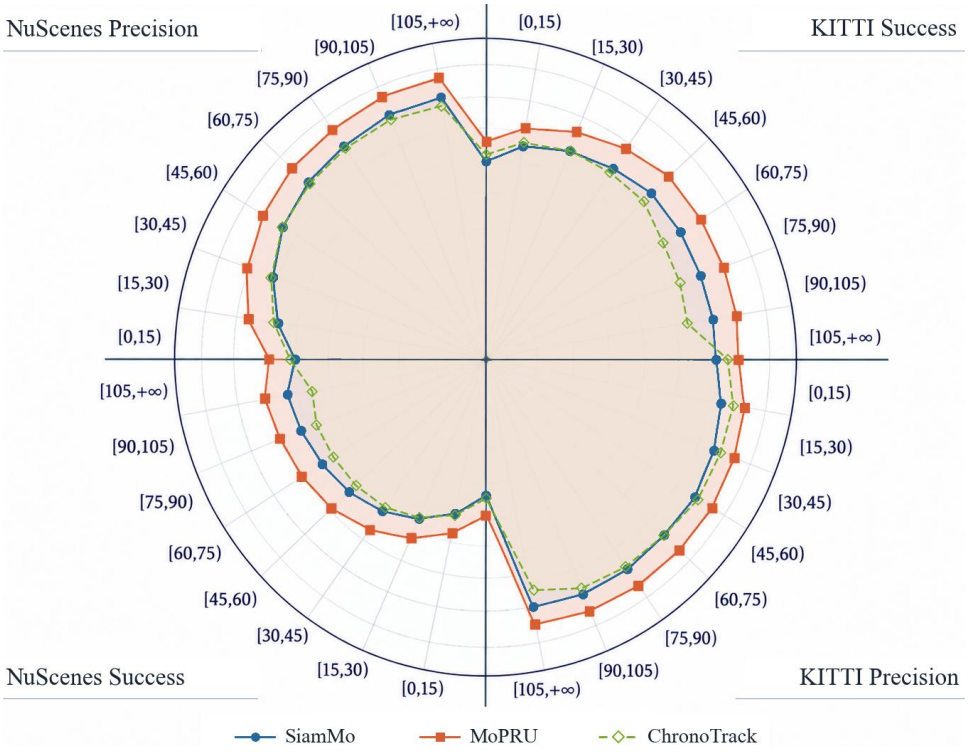


Figure 4: **Tracking performance under varying target sparsity.** Success and Precision across eight point-count bins on KITTI and NuScenes are reported for SiamMo, ChronoTrack, and MoPRU. MoPRU performs best across all sparsity levels.

Target-point sparsity. Across eight target-point-count bins, MoPRU outperforms SiamMo and ChronoTrack on KITTI and NuScenes (Figure 4). In the sparsest KITTI bin, Success/Precision reaches 70.6/88.5, versus 68.8/87.0 for SiamMo. The advantage persists as target points decrease, indicating robustness to sparse target observations.

4.3 Ablation Studies

Table 4 isolates the effects of supervision source, the two motion-update paths, and weight sharing under the same SiamMo backbone, with the remaining tracking components unchanged.

Short-horizon tracking supervision. With the same shared update structure, current-frame supervision, scalar short-horizon updates and direct future-error supervision reach 73.0/90.5, 72.9/90.7 and 72.8/90.4, respectively. Short-horizon outcome differences yield 73.8/91.0 (Appendix E, Table A13).

Update paths and weight sharing. RTM-only and State-only reach 73.1/90.6 and 73.6/90.8, respectively. State-only applies learned weights only to the recurrent motion-state update and directly uses pairwise RTM for localization. Using both paths with shared weights reaches 73.8/91.0, versus 73.5/90.7 with separate weights.

Table 4: **Ablation of motion-update learning on KITTI.** Success/Precision (%). "Refine" and "State" denote the update paths; "Shared" uses the same update-weight vector for both.

Configuration	Supervision	Refine	State	Weight sharing	Car	Ped.	Van	Cyc.	Mean
SiamMo	–	–	–	–	76.3/88.1	68.6/93.9	67.9/80.5	78.5/94.8	72.3/90.1
History only	–	–	–	–	76.9/88.3	**68.9**/94.0	69.3/81.0	78.8/94.9	72.8/90.3
Scalar update	Scalar	✓	✓	Shared	77.0/88.7	68.8/**94.4**	70.0/81.5	78.9/95.0	72.9/90.7
Current-frame update	Current	✓	✓	Shared	77.2/88.6	68.6/94.1	71.4/81.7	78.9/94.9	73.0/90.5
RTM-only update	Subseq.	✓	–	–	77.3/88.5	68.7/94.3	71.8/81.8	79.0/95.0	73.1/90.6
State-only update	Subseq.	–	✓	–	77.9/88.9	68.4/94.1	75.5/83.4	79.0/95.0	73.6/90.8
Separate weights	Subseq.	✓	✓	Separate	77.8/88.8	68.3/94.0	75.1/83.2	78.9/95.0	73.5/90.7
MoPRU	Subseq.	✓	✓	Shared	**78.4/89.1**	67.7/94.2	**78.8/84.6**	**79.1/95.1**	**73.8/91.0**

The signs of ΔR_t^p and ΔR_t^ψ recover the preferred binary setting on 76.8% of analysis frames, with a 0.060 mean normalized score gap (Appendix B, Table A6). Continuous update weights further improve Success/Precision by 0.25/0.19 points and reduce the normalized gap to the dense weight-grid upper bound from 0.046 to 0.028 (Appendix C, Table A10).

Table 5: **Transfer across tracking frameworks on KITTI and NuScenes.** Trackers use different temporal designs; Δ denotes gain over each base tracker.

		KITTI			NuScenes		
Base tracker	Temporal design	Base S/P	+MoPRU S/P	Δ S/P	Base S/P	+MoPRU S/P	Δ S/P
SiamMo	Pairwise motion	72.3/90.1	**73.8/91.0**	+1.5/+0.9	60.31/72.68	**62.93/75.66**	+2.62/+2.98
MBPTrack	Memory + box prior	70.3/87.9	**71.0/88.4**	+0.7/+0.5	57.48/69.88	**58.70/71.54**	+1.22/+1.66

Transfer across tracking frameworks. MoPRU consistently improves SiamMo and MBPTrack on KITTI and NuScenes (Table 5), showing that the learned motion update transfers across different tracking designs and datasets. Per-category KITTI results are reported in Appendix E, Table A18.

4.4 Mechanism Analysis

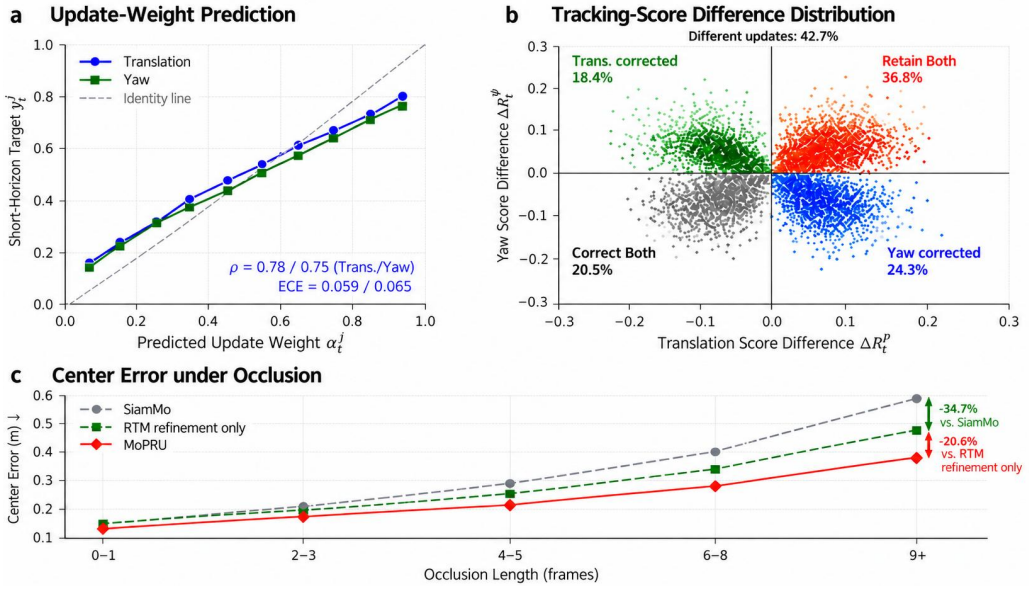


Figure 5: **Mechanism analysis of MoPRU.** (a) Predicted update weights versus short-horizon targets. (b) Distribution of short-horizon tracking-score differences on the KITTI analysis set. 42.7% favor different binary updates. (c) Center error versus occlusion length.

Online update-weight prediction. On the KITTI analysis set, the predicted shared update weights correlate with their short-horizon targets (Spearman 0.78/0.75). The two RTM update weights favor different settings on 42.7%/45.1% of KITTI/NuScenes analysis frames. In these cases, distinct RTM update weights outperform a uniform RTM update on 80.0%/82.3% of the frames, with normalized score gains of 0.23/0.26 (Figure 5a–b; Appendix B, Table A8).

Tracking under long occlusion. For occlusion streaks of at least nine frames, MoPRU reduces mean center error from $0.585\,\mathrm{m}$ with SiamMo and $0.481\,\mathrm{m}$ with RTM-only update to $0.382\,\mathrm{m}$ (Figure 5c). The additional gain over RTM-only update shows that recurrent motion-state update improves tracking when current point-cloud observations are sparse or incomplete.

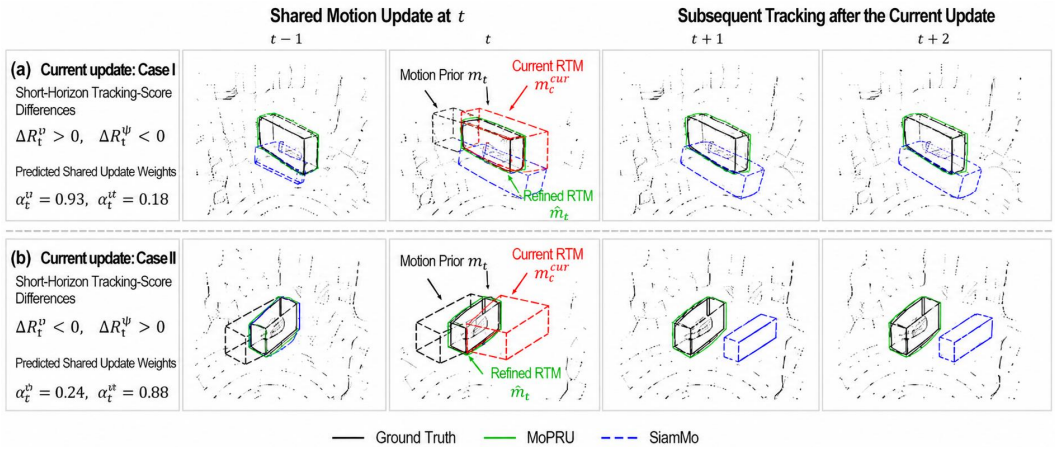


Figure 6: **Closed-loop effect of the shared motion update.** (a–b) Two representative cases show how the shared update at frame t jointly refines the current RTM for target localization and updates the recurrent motion state for subsequent motion prediction. The resulting sequences from t to $t{+}2$ show how this update influences later predictions and target localization during closed-loop tracking.

Closed-loop update consistency. Fixed-observation evaluation and closed-loop tracking-score differences show 0.86/0.82 Spearman correlation for translation/yaw, with 82.7% best-update agreement and a 0.04 mean normalized score gap (Appendix B). Subsequent-tracking and current-refinement score differences further show 85.6%/82.9% sign agreement, showing that the learned update remains consistent during closed-loop tracking. Representative cases are shown in Figure 6, with additional stress and failure cases in Appendix G.

Update granularity. Opposite-sign translation–yaw score differences occur on 39.8–43.5% of analysis frames under alternative short-horizon tracking-score definitions and 36.7–44.5% across tracking horizons $H = 0$–3. Finer $\{xy, z, \psi\}$ and $\{x, y, z, \psi\}$ decompositions reach 73.5/90.8 and 73.4/90.7, respectively, compared with 73.8/91.0 for $\{xyz, \psi\}$, while increasing the short-horizon tracking branches from 4 to 8 and 16 (Appendices B and C). We therefore use $\{xyz, \psi\}$ as the default update granularity; the learned update remains stable across score definitions and tracking horizons.

5 Conclusion

In this work, we propose MoPRU, a novel recurrent motion learning framework for 3D single object tracking that addresses the gap between current RTM refinement and recurrent motion-state update for subsequent motion prediction. At its core, MoPRU introduces a shared motion update that applies the same predicted component weights to current RTM refinement and recurrent motion-state update. These weights are learned from short-horizon tracking outcomes under alternative current updates, allowing the recurrent state to retain motion information that remains useful for subsequent motion prediction. Experiments on KITTI and NuScenes show consistent improvements. Gains across different 3D SOT frameworks and in zero-shot KITTI-to-Waymo transfer further support the generality of the learned motion update. Improvements under sparse observations and long occlusions highlight the value of recurrent motion-state update beyond current-frame localization.

AI USE STATEMENT

The authors used Generative AI (GenAI) tools to assist with manuscript preparation and to correct minor grammatical errors in the manuscript. No experimental data or reported results were generated using GenAI tools. The authors reviewed and verified all AI-assisted content and take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

Detailed implementation, training, evaluation, and short-horizon tracking protocols are provided in the appendix to facilitate reproduction of the reported results.

REFERENCES

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. NuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

BaiChen Fan, Sifan Zhou, Jian Li, Shibo Zhao, Muqing Cao, and Qin Wang. Beyond frame-wise tracking: A trajectory-based paradigm for efficient point cloud tracking. *IEEE Robotics and Automation Letters*, 11(3):3430–3437, 2026. doi: 10.1109/LRA.2026.3655309.

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

Wanting Geng, Xin Chen, Chuanyu Sun, Jie Zhao, Ben Kang, Dong Wang, and Huchuan Lu. TGTrack: Temporal generative learning for unified single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28134–28144, 2026.

Le Hui, Lingpeng Wang, Mingmei Cheng, Jin Xie, and Jian Yang. 3D Siamese Voxel-to-BEV Tracker for Sparse Point Clouds. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Le Hui, Lingpeng Wang, Linghua Tang, Kaihao Lan, Jin Xie, and Jian Yang. 3D Siamese Transformer Network for Single Object Tracking on Point Clouds. In *European Conference on Computer Vision*, pp. 293–310, 2022.

Kaihao Lan, Haobo Jiang, and Jin Xie. Temporal-aware siamese tracker: Integrate temporal context for 3D object tracking. In *Proceedings of the Asian Conference on Computer Vision*, pp. 399–414, 2022.

Xiaohai Li, Bineng Zhong, Qihua Liang, Guorong Li, Zhiyi Mo, and Shuxiang Song. MambaLCT: Boosting tracking via long-term context state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4986–4994, 2025. doi: 10.1609/aaai.v39i5.32528.

Yu Lin, Zhiheng Li, Yubo Cui, and Zheng Fang. SeqTrack3D: Exploring sequence information for robust 3D point cloud tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 6959–6965, 2024. doi: 10.1109/ICRA57147.2024.10611238.

Juntao Liu, Zikun Zhou, Zhuotao Tian, Guangming Lu, Jun Yu, and Wenjie Pei. PointRePar: Spatiotemporal point relation parsing for robust category-unified 3D tracking. In *International Conference on Learning Representations*, 2026.

Zhipeng Luo, Gongjie Zhang, Changqing Zhou, Zhonghua Wu, Qingyi Tao, Lewei Lu, and Shijian Lu. Modeling continuous motion for 3D point cloud object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4026–4034, 2024. doi: 10.1609/aaai.v38i5.28196.

Jiahao Nie, Fei Xie, Sifan Zhou, Xueyi Zhou, Dong-Kyu Chae, and Zhiwei He. P2P: Part-to-part motion cues guide a strong tracking framework for LiDAR point clouds. *International Journal of Computer Vision*, 133(8):5326–5342, 2025. doi: 10.1007/s11263-025-02430-6.

Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. P2B: Point-to-box network for 3D object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adrià López Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. KalmanNet: Neural network aided kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022. doi: 10.1109/TSP.2022.3158588.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Shengjing Tian, Yinan Han, Xiantong Zhao, Xuehu Liu, and Qi Lang. MambaTrack3D: A state space model framework for LiDAR-based object tracking under high temporal variation. *arXiv preprint arXiv:2511.15077*, 2025.

Jie Xiao, Yinchao Ma, Yuyang Tang, Chuxin Wang, and Tianzhu Zhang. State space models for long-term temporal context in 3D single object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(4):4941–4955, 2026. doi: 10.1109/TCSVT.2025.3627667.

Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. MBPTrack: Improving 3D point cloud tracking with memory networks and box priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9911–9920, 2023.

Yuxiang Yang, Yingqi Deng, Hongjie Gu, Zhekang Dong, and Jing Zhang. SiamMo: Siamese motion-centric 3D object tracking. *Journal of Image and Graphics*, 31(2):512–524, 2026. doi:10.11834/jig.250112.

Yuxiang Yang, Yingqi Deng, Mian Pan, Zheng-Jun Zha, and Jing Zhang. BEVTrack: A simple and strong baseline for 3D single object tracking in bird's-eye view. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 2242–2250, 2025. doi: 10.24963/ijcai.2025/250.

Jaejoon Yoo, SuBeen Lee, Yerim Jeon, Miso Lee, and Jae-Pil Heo. Temporally consistent long-term memory for 3D single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 8388–8397, 2026.

Jirong Zha, Yuxuan Fan, Kai Li, Han Li, Chen Gao, and Xinlei Chen. DIMM: Decoupled multi-hierarchy Kalman filter via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18746–18754, 2026. doi: 10.1609/aaai.v40i22.38943.

Chaoda Zheng, Xu Yan, Jiantao Gao, Weibing Zhao, Wei Zhang, Zhen Li, and Shuguang Cui. Box-aware feature enhancement for single object tracking on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13199–13208, 2021.

Chaoda Zheng, Xu Yan, Haiming Zhang, Baoyuan Wang, Shenghui Cheng, Shuguang Cui, and Zhen Li. Beyond 3D siamese tracking: A motion-centric paradigm for 3D single object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8111–8120, 2022.

Sifan Zhou, Jiahao Nie, Ziyu Zhao, Yichao Cao, and Xiaobo Lu. FocusTrack: One-stage focus-and-suppress framework for 3D point cloud object tracking. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 7366–7375, 2025. doi: 10.1145/3746027.3754781.

Sifan Zhou, Yichao Cao, Jiahao Nie, Yuqian Fu, Ziyu Zhao, Xiaobo Lu, and Shuo Wang. CompTrack: Information bottleneck-guided low-rank dynamic token compression for point cloud tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13773–13781, 2026. doi: 10.1609/aaai.v40i16.38385.

A Implementation and Reproducibility

Reproducibility scope. MoPRU extends a motion-centric 3D SOT tracker with recurrent motion prediction and a shared update for RTM refinement and recurrent motion-state update. Four alternative current updates provide short-horizon translation- and yaw-specific supervision from subsequent tracking outcomes during training, whereas online tracking uses one predicted shared update per frame. Backbone settings otherwise follow SiamMo.

Table A1: **Short-horizon tracking analysis set.** Each analysis frame satisfies the same eligibility criteria for all four current-update settings.

Item	Setting
Analysis set	KITTI Train split (sequences 0–16) used for short-horizon analysis
Eligibility	Valid recurrent history and complete $H = 2$ tracking outcomes for all four current-update settings
Number of analysis frames	4,800

The same 4,800 KITTI analysis frames are used throughout Appendices B and C.

Loss definitions. The pairwise-motion loss $\mathcal{L}_{\text{pair}}$ follows SiamMo. For frames with valid recurrent history, the normalized residual of the predicted motion prior is

$$e_t = \frac{m_t^- - m_t^\star}{\boldsymbol{\kappa}^{\text{norm}}}, \qquad \boldsymbol{\kappa}^{\text{norm}} = (1\,\text{m}, 1\,\text{m}, 0.5\,\text{m}, 10^\circ),$$

where the yaw residual is wrapped to the principal interval. The motion-prior loss $\mathcal{L}_{\text{prior}}$ combines a Laplace negative log-likelihood with $0.25\,\text{SmoothL1}(e_t, 0)$.

The translation- and yaw-specific tracking-score differences define the continuous update targets

$$y_t^j = \sigma(\Delta R_t^j / \tau_j), \qquad j \in \{p, \psi\},$$

where τ_j controls the target scale. The update estimator is optimized by

$$\mathcal{L}_{\text{update}} = \frac{1}{2} \sum_{j \in \{p,\psi\}} \text{BCEWithLogits}(\ell_t^j, y_t^j).$$

The refined-RTM loss is

$$\mathcal{L}_{\text{refine}} = \text{SmoothL1}\left(\frac{\hat{m}_t - m_t^\star}{\boldsymbol{\kappa}^{\text{norm}}}, 0\right),$$

again using the wrapped yaw residual. The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{pair}} + \lambda_h \mathcal{L}_{\text{prior}} + \lambda_u \mathcal{L}_{\text{update}} + \lambda_r \mathcal{L}_{\text{refine}}.$$

The MoPRU-specific loss weights and target temperatures are summarized in Table A4.

Table A2: **Training schedule.** The pairwise motion predictor and recurrent motion predictor are initialized before joint optimization of the complete MoPRU tracker.

Stage	Trainable module(s)	Training length
Pairwise motion	F_{pair}	50 epochs
Motion prior	F_{state}	20 epochs
Joint MoPRU	F_{pair}, F_{state}, g_θ	10 epochs

During joint optimization, the short-horizon tracking-score targets are detached when optimizing g_θ. For $\mathcal{L}_{\text{refine}}$, α_t is treated as constant while gradients update F_{pair} and F_{state}. This separates update-weight supervision from the gradients used to improve current RTM estimation and recurrent motion prediction.

Short-horizon tracking supervision. For each eligible frame t, four settings $a = (a^p, a^\psi) \in \{0, 1\}^2$ start from the same previous box and recurrent motion state. For component j, $a^j = 1$ retains the current RTM component, whereas $a^j = 0$ applies the maximum prior correction and removes its recurrent descriptor; hence $a = (0, 0)$ gives $M_t = \text{Write}(M_{t-1}, 0)$.

The four branches share the raw LiDAR frames but maintain independent tracking states. For all $k > 0$, each branch predicts weights from its own observations and recurrent state using the same fixed-parameter predictor; only the update at frame t is varied.

Following Section 3.3, translation and yaw targets are obtained by averaging each tracking-score difference over the two settings of the other component:

$$\Delta R_t^p = \frac{1}{2}\left[(R_{10} - R_{00}) + (R_{11} - R_{01})\right],$$
$$\Delta R_t^\psi = \frac{1}{2}\left[(R_{01} - R_{00}) + (R_{11} - R_{10})\right].$$

Positive ΔR_t^j favors the current RTM component, while negative values favor its motion-prior-corrected setting. These differences supervise the shared translation and yaw update weights.

Table A3: **Short-horizon tracking settings.** The same settings are used to construct the translation- and yaw-specific update targets.

Item	Setting
Tracking horizon	$H = 2$
Discount factor	$\gamma = 0.90$
Subsequent update	Branch-wise predicted weights ($k > 0$)
Tracking-error weights	$\lambda_p = 0.50$, $\lambda_\psi = 0.20$
Error scales	$\kappa_p^{\mathrm{err}} = 2\,\mathrm{m}$, $\kappa_\psi^{\mathrm{err}} = 30^\circ$

Training and online tracking. Training uses five-frame clips, with the recurrent motion state reset at the start of each clip. Short-horizon branches are used only for update supervision. Online tracking predicts one shared update per frame; $\alpha_t = (1, 1)$ before recurrent history is available, and the motion state is propagated through the complete tracklet.

Table A4: **Principal MoPRU model and optimization settings.** Backbone settings inherited unchanged from SiamMo are omitted.

Item	Setting
State predictor	$d_{\mathrm{model}} = 128$, $d_{\mathrm{state}} = 16$, convolution width 4
Update estimator	LayerNorm–MLP $657 \to 192 \to 96 \to 2$ with SiLU
Optimization	AdamW, learning rate 1×10^{-4}, batch size 8 clips
Loss weights	$(\lambda_h, \lambda_u, \lambda_r)$: Car $(0.25, 0.12, 0.12)$, Pedestrian $(0.30, 0.16, 0.12)$, Van $(0.38, 0.24, 0.28)$, Cyclist $(0.30, 0.14, 0.10)$
Update-target temperatures	(τ_p, τ_ψ): Car $(0.10, 0.08)$, Pedestrian $(0.10, 0.07)$, Van $(0.12, 0.09)$, Cyclist $(0.10, 0.07)$
Motion clipping bounds	(c_x, c_y, c_z, c_ψ): Car $(0.10, 0.10, 0.04, 1.0^\circ)$, Pedestrian $(0.15, 0.15, 0.05, 2.0^\circ)$, Van $(0.40, 0.40, 0.14, 3.5^\circ)$, Cyclist $(0.12, 0.12, 0.04, 1.5^\circ)$

B Analysis of Shared Motion-Update Learning

Motivating observations. We quantify the two observations underlying Figure 1 using the KITTI analysis set and the four current-update settings $a \in \{00, 10, 01, 11\}$. For each eligible frame, all settings start from the same previous target box and recurrent motion state.

For update setting a, define the current localization score and subsequent tracking score as

$$C_t(a) = -\rho_t^a, \qquad S_t(a) = -\sum_{k=1}^{H} \gamma^k \rho_{t+k}^a.$$

For two settings a and b, we measure

$$D_{\mathrm{cur}}(a, b) = |C_t(a) - C_t(b)|, \qquad D_{\mathrm{sub}}(a, b) = |S_t(a) - S_t(b)|.$$

We regard two updates as having comparable current localization when $D_{\mathrm{cur}}(a, b) \le \epsilon_{\mathrm{cur}}$ with $\epsilon_{\mathrm{cur}} = 0.05$, and as producing different subsequent tracking outcomes when $D_{\mathrm{sub}}(a, b) > \epsilon_{\mathrm{sub}}$ with

$\epsilon_{\text{sub}} = 0.10$. Among 8,617 comparable-current pairs, 2,878 (33.4%) produce different subsequent outcomes. Thus, similar current localization does not necessarily imply similar subsequent tracking.

We further evaluate how recurrent motion-state update affects the next-frame motion prior. For m^{-}_{t+1}, translation and yaw errors are

$$e^{p}_{t+1} = \left\| m^{-}_{t+1,p} - m^{\star}_{t+1,p} \right\|_2, \qquad e^{\psi}_{t+1} = \left| \text{wrap}\left(m^{-}_{t+1,\psi} - m^{\star}_{t+1,\psi} \right) \right|.$$

We compare the RTM-only variant with the full MoPRU model. Both use the same learned component weights for current RTM refinement, while only MoPRU applies them to recurrent motion-state update. This matched comparison measures the contribution of recurrent state update to subsequent motion prediction.

Table A5: **Motivating observations and closed-loop consistency.** (a) Comparable current localization can lead to different subsequent tracking outcomes. (b) The shared update reduces next-frame motion-prior translation and yaw errors over RTM-only update. (c) Fixed-observation and closed-loop tracking yield consistent tracking-score differences on the KITTI analysis set.

(a) Comparable current localization and subsequent tracking

Statistic	Definition	Value
Current-localization threshold	$D_{\text{cur}}(a,b) \le \epsilon_{\text{cur}}$	0.05
Subsequent-tracking threshold	$D_{\text{sub}}(a,b) > \epsilon_{\text{sub}}$	0.10
Comparable-current pairs	N_{match}	8,617
Different-subsequent pairs	N_{diff}	2,878
Different subsequent outcomes	$N_{\text{diff}}/N_{\text{match}}$	**33.4%**

(b) Next-frame motion-prior error

Configuration	Translation error (m) ↓	Yaw error (°) ↓
RTM-only update	0.279	3.47
MoPRU	**0.260**	**3.22**
Relative reduction	**6.8%**	**7.2%**

(c) Fixed-observation and closed-loop consistency

Criterion	Result
Rank correlation (p/ψ)	0.86 / 0.82
Update-preference agreement (p/ψ, %)	90.4 / 87.8
Best-update agreement (%)	82.7
Translation–yaw pattern agreement (%)	85.3
Mean normalized score gap	0.04

Evaluation protocols. We next evaluate whether the short-horizon supervision used to learn the shared motion update remains consistent when subsequent target localization is performed in closed loop. For each eligible frame t, the four current-update settings $a \in \{00, 10, 01, 11\}$ start from the same previous target box and recurrent motion state.

Under *fixed-observation tracking*, the four branches share subsequent observation inputs but maintain independent recurrent states. Under *closed-loop tracking*, each branch evolves its own search region, pairwise input, target box, and recurrent state under the same raw LiDAR frames. For $k > 0$, each branch predicts its own weights with the same update predictor; only the update at frame t is varied. Tracking-score differences follow Equations 8 and 9; results are shown in Figure A1 and Table A5(c).

Closed-loop consistency. Let ΔR^{j}_{FO} and ΔR^{j}_{CL} denote the tracking-score differences under fixed-observation and closed-loop tracking, respectively, for $j \in \{p, \psi\}$. The two protocols yield rank correlations of 0.86/0.82, while the preferred binary current-update settings agree on 90.4%/87.8% of the KITTI analysis frames. For the joint current update, we define

$$a^{\star}_{\text{FO}} = \arg \max_{a \in \{00,10,01,11\}} R_{\text{FO}}(a), \qquad a^{\star}_{\text{CL}} = \arg \max_{a \in \{00,10,01,11\}} R_{\text{CL}}(a).$$

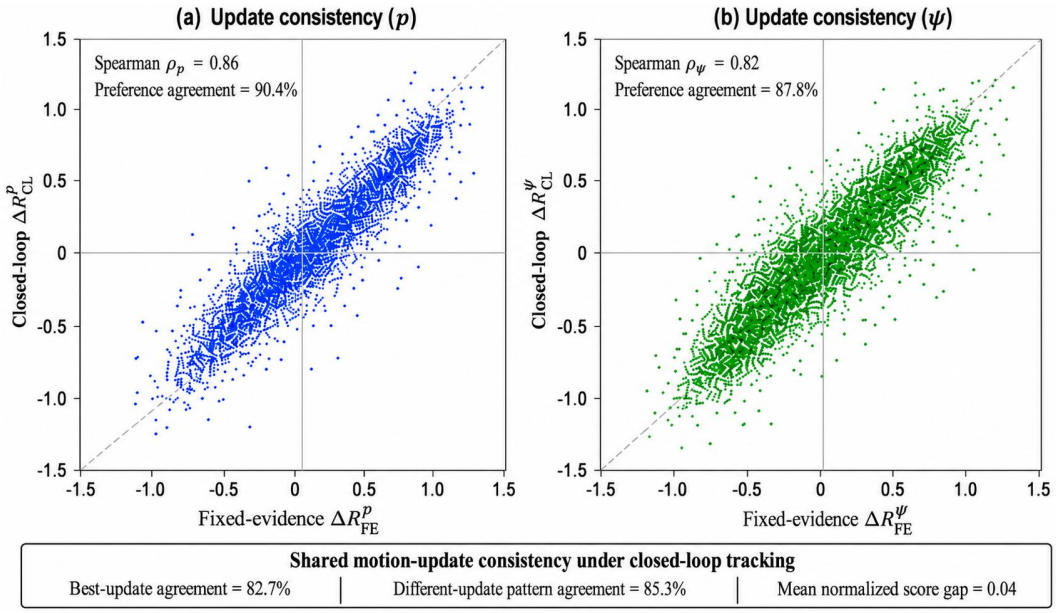


Figure A1: **Closed-loop consistency of short-horizon update supervision.** Translation and yaw tracking-score differences are compared between fixed-observation and closed-loop tracking on the KITTI analysis set. The dashed diagonal denotes equal values, while the zero axes separate binary update preferences.

The two tracking protocols select the same best current update on 82.7% of the analysis frames. Among frames for which translation and yaw favor different binary settings, 85.3% retain the same component-wise update pattern under closed-loop tracking.

For frames with different selections, the normalized closed-loop score gap is

$$G_{\mathrm{CL}} = \frac{R_{\mathrm{CL}}(a^\star_{\mathrm{CL}}) - R_{\mathrm{CL}}(a^\star_{\mathrm{FO}})}{\max_a R_{\mathrm{CL}}(a) - \min_a R_{\mathrm{CL}}(a) + \epsilon}.$$

The mean normalized score gap is 0.04.

Current-frame consistency. Short-horizon supervision accounts for subsequent tracking through the recurrent motion state, while the same current update also determines target localization at frame t. We therefore compare the update preference derived from short-horizon tracking with the preference obtained from current-frame localization alone.

For each $a \in \{00, 10, 01, 11\}$, Equation 3 produces $\hat{m}^a_t$ and the corresponding target box B^a_t. Using the current localization score $C_t(a)$ defined above and applying the same score-difference decomposition as Equations 8 and 9, with R replaced by C, gives ΔC^p_t and ΔC^ψ_t.

The short-horizon update preference is

$$\hat{a}_t = \left(\mathbb{1}[\Delta R^p_t > 0], \mathbb{1}[\Delta R^\psi_t > 0]\right),$$

while the best current-frame update is

$$a^{\star,C}_t = \arg\max_{a \in \{00,10,01,11\}} C_t(a).$$

Their normalized current-frame score gap is

$$G^C_t = \frac{C_t(a^{\star,C}_t) - C_t(\hat{a}_t)}{\max_a C_t(a) - \min_a C_t(a) + \epsilon}.$$

Table A6: **Consistency between short-horizon supervision and current-frame localization on KITTI.** The first three criteria compare update preferences obtained from short-horizon tracking and current-frame localization. The last two quantify the advantage of component-wise binary updates over uniform updates.

Criterion	Result
Update-preference agreement (p/ψ, %)	85.6 / 82.9
Best-update agreement (%)	76.8
Mean normalized score gap	0.060
Component-wise > uniform (%)	80.0
Mean normalized score gain	0.23

The short-horizon and current-frame component preferences agree on 85.6%/82.9% of the KITTI analysis frames. The binary update selected from the short-horizon tracking scores matches the best current-frame update on 76.8% of the frames, with a mean normalized current-frame score gap of 0.060.

Translation–yaw update disagreement. We further examine frames for which translation and yaw favor different binary current-update settings. These frames test whether the two motion components benefit from independently controlled update weights. We define the best component-wise setting as

$$R_{\text{comp}} = \max(R_{10}, R_{01}),$$

and the best uniform setting as

$$R_{\text{uni}} = \max(R_{00}, R_{11}).$$

The normalized score gain of the component-wise setting is

$$G_{\text{comp}} = \frac{R_{\text{comp}} - R_{\text{uni}}}{\max_a R(a) - \min_a R(a) + \epsilon}.$$

On KITTI, the best component-wise setting exceeds the best uniform setting on 80.0% of translation–yaw disagreement frames, with a mean normalized score gain of 0.23. This difference supports the separate translation and yaw update weights used by MoPRU.

Supervision sensitivity. We evaluate short-horizon supervision under alternative tracking-score definitions and tracking horizons while keeping the current-update protocol unchanged. The score variants modify the tracking-error terms or yaw-error weight, while the horizon analysis varies the number of subsequent tracking steps.

Table A7: **Sensitivity of short-horizon tracking supervision.** We vary the tracking-score definition in (a) and the tracking horizon in (b) while keeping the current-update protocol unchanged. Translation–yaw disagreement denotes analysis frames for which the two motion components favor different binary current-update settings.

(a) Tracking-score definition

Definition	Trans.–yaw disagreement (%)	Preference agreement (p/ψ, %)
IoU only	39.8	90.0 / 88.0
IoU + center	41.5	97.0 / 89.0
IoU + yaw	42.2	93.0 / 97.0
Default	42.7	100.0 / 100.0
$\lambda_\psi \times 0.5$	42.3	98.0 / 96.0
$\lambda_\psi \times 2$	43.5	98.0 / 95.0

(b) Tracking horizon

Horizon	Trans.–yaw disagreement (%)	Success	Precision
$H = 0$	36.7	73.1	90.5
$H = 1$	39.1	73.4	90.7
$H = 2$	42.7	73.8	91.0
$H = 3$	44.5	73.6	90.8

Across the tested tracking-score definitions, the translation–yaw disagreement frequency ranges from 39.8% to 43.5%, while the resulting component preferences remain close to those obtained with the default score. Increasing the tracking horizon from $H = 0$ to $H = 3$ changes the disagreement frequency from 36.7% to 44.5%. Tracking performance reaches 73.8/91.0 at $H = 2$, which is used as the default short-horizon setting.

Cross-dataset consistency. We repeat the same four-update closed-loop analysis on NuScenes using the same eligibility criteria, tracking-score definition, and $H = 2$ tracking horizon. Table A8 compares the frequency of translation–yaw update disagreement, the gain of component-wise binary updates over uniform updates, and the correlation between predicted update weights and short-horizon targets.

Table A8: **Cross-dataset consistency of shared motion-update learning.** KITTI and NuScenes use the same four-update protocol and short-horizon tracking score.

Dataset	Trans.–yaw disagreement (%)	Component-wise > uniform (%)	Norm. score gain	Target corr. (p/ψ)
KITTI	42.7	80.0	0.23	0.78 / 0.75
NuScenes	45.1	82.3	0.26	0.74 / 0.71

Translation–yaw disagreement occurs on 42.7%/45.1% of the KITTI/NuScenes analysis frames. Within these frames, the best component-wise binary setting exceeds the best uniform setting on 80.0%/82.3%, with mean normalized score gains of 0.23/0.26. The predicted translation and yaw update weights correlate with their corresponding short-horizon targets at 0.78/0.75 on KITTI and 0.74/0.71 on NuScenes. The same component-wise update behavior is therefore observed under both KITTI and NuScenes tracking protocols.

C Shared Motion-Update Analysis

Short-horizon supervision is constructed from four binary current updates, whereas online MoPRU predicts continuous update weights. We analyze the granularity of the shared motion update, the benefit of continuous weight prediction, and how the learned weights affect current motion estimation and recurrent motion-state update.

Motion-update granularity. The default MoPRU groups the 4-DoF RTM as $\{xyz, \psi\}$ and predicts one update weight for each group. Table A9 compares this design with a single update weight and finer motion groupings. A grouping with K independently controlled weights requires 2^K binary current-update settings during short-horizon supervision.

Table A9: **Motion-update granularity on KITTI.** Each grouping determines the number of independently controlled update weights and the corresponding binary current-update settings. Success/Precision (%) are reported.

Update grouping	Update weights	Binary settings	Success/Precision
$\{xyz\psi\}$	1	2	72.9/90.7
$\{xyz, \psi\}$	**2**	**4**	**73.8/91.0**
$\{xy, z, \psi\}$	3	8	73.5/90.8
$\{x, y, z, \psi\}$	4	16	73.4/90.7

Using one update weight gives 72.9/90.7 Success/Precision. The default two-weight grouping improves tracking to 73.8/91.0. Further separating the motion update into three or four independently controlled groups gives 73.5/90.8 and 73.4/90.7, while increasing the number of binary current-update settings from four to eight and sixteen. We therefore use $\{xyz, \psi\}$ as the default motion-update granularity.

Continuous shared motion update. Although the short-horizon targets are constructed from binary current updates, online MoPRU predicts $\alpha_t = (\alpha_t^p, \alpha_t^\psi) \in [0, 1]^2$. We evaluate continuous update settings using the same closed-loop tracking protocol as in Appendix B.

For each KITTI analysis frame, we evaluate a 25×25 grid over (α^p, α^ψ). The normalized short-horizon score gap is

$$G_{\text{grid}}(\alpha) = \frac{R(\alpha^\star_{\text{grid}}) - R(\alpha)}{\max_{\alpha'} R(\alpha') - \min_{\alpha'} R(\alpha') + \epsilon},$$

where $\alpha^\star_{\text{grid}}$ is the highest-scoring setting on the 25×25 grid.

Figure A2 shows three representative tracking-score landscapes from the KITTI analysis set.

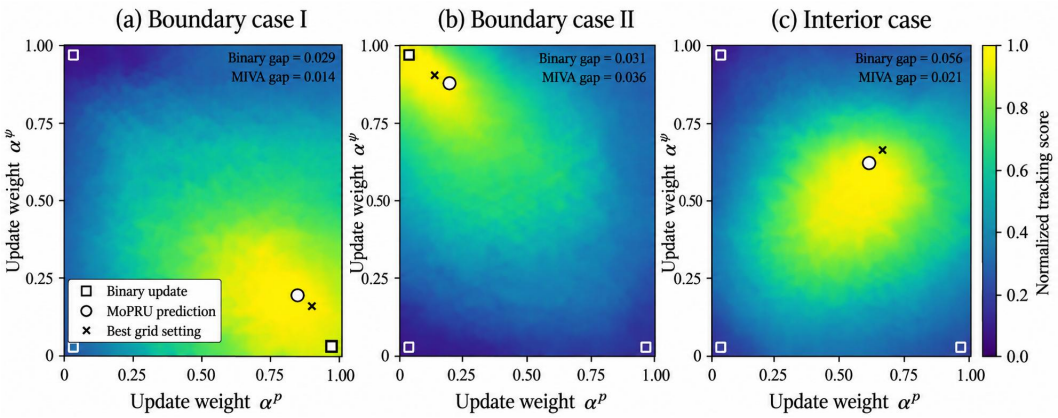


Figure A2: **Continuous shared motion-update landscapes.** Three KITTI analysis frames show two boundary cases and one interior case over 25×25 grids of $(\alpha^p, \alpha^\psi) \in [0, 1]^2$. Heatmaps report normalized short-horizon tracking scores. Squares denote binary updates, circles denote MoPRU predictions, and crosses denote the highest-scoring grid settings. The emphasized square marks the best binary update.

Continuous-update comparison. Table A10(a) compares binary and continuous update settings under the same closed-loop tracking protocol, including the short-horizon score selection, binary and common-weight settings, continuous targets, online MoPRU predictions, and the 25×25 grid-selected setting.

Table A10: **Analysis of continuous shared motion updates on KITTI.** (a) Binary and continuous updates under the same closed-loop protocol. (b) Responses of the two update paths relative to $\boldsymbol{\alpha}_0 = (0.5, 0.5)$.

(a) Continuous-update comparison

Update setting	Success	Precision	Normalized score gap ↓
Binary update from score differences	73.42	90.72	0.058
Best binary update	73.55	90.81	0.046
Best common-weight setting (α, α)	73.34	90.70	0.062
Short-horizon target weights $\boldsymbol{y}_t$	73.63	90.85	0.019
MoPRU predicted weights	**73.80**	**91.00**	**0.028**
Grid-selected setting (25×25)	73.68	90.87	0.000

(b) Response of the two update paths

(α^p, α^ψ)	Δq_t	$\Delta \hat{m}_t$	Δm^-_{t+1}	ΔR
$(0.1, 0.1)$	0.002	0.145	0.003	-0.018
$(0.5, 0.5)$	0.000	0.000	0.000	0.000
$(0.9, 0.9)$	0.001	0.145	0.002	$+0.006$
$(0.9, 0.1)$	0.286	0.145	0.064	$+0.011$
$(0.1, 0.9)$	0.301	0.145	0.071	$+0.007$

All changes are measured relative to $\boldsymbol{\alpha}_0 = (0.5, 0.5)$. We use $\Delta q_t = \text{RMS}(q_t^\alpha - q_t^{\alpha_0})$, $\Delta \hat{m}_t = \|(\hat{m}_t^\alpha - \hat{m}_t^{\alpha_0})/\boldsymbol{\kappa}^{\text{norm}}\|_2$, $\Delta m^-_{t+1} = \|(m^{-,\alpha}_{t+1} - m^{-,\alpha_0}_{t+1})/\boldsymbol{\kappa}^{\text{norm}}\|_2$, and $\Delta R = R^\alpha - R^{\alpha_0}$.

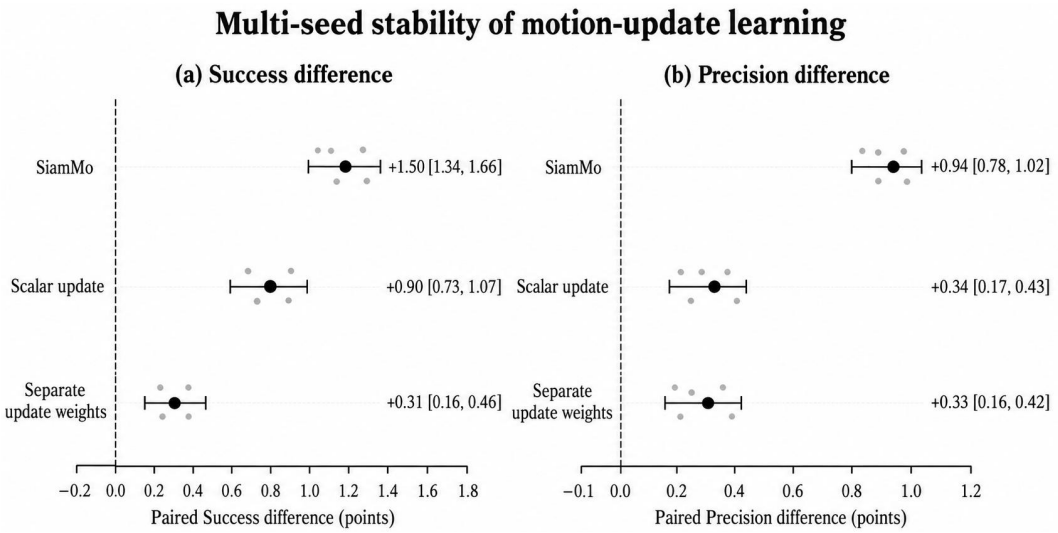


Figure A3: **Multi-seed stability of motion-update learning on KITTI.** Panels (a,b) report paired Success and Precision differences between MOPRU and the corresponding comparison setting over five independent training seeds. Light-gray points show seed-level effects, black circles denote their means, and horizontal bars report paired tracklet-bootstrap 95% confidence intervals. The dashed line denotes zero difference.

Continuous-update behavior. The best binary update reaches 73.55/90.81, while the continuous short-horizon target weights reach 73.63/90.85. Online MOPRU further reaches 73.80/91.00, improving the best binary update by 0.25/0.19 Success/Precision points. Its mean normalized short-horizon score gap is 0.028, compared with 0.046 for the best binary update.

The response analysis further separates the two roles of the shared motion update. Common weight changes leave the recurrent write and next motion prior nearly unchanged after normalization ($\Delta q_t \leq 0.002$ and $\Delta m^-_{t+1} \leq 0.003$), while changing the refined RTM by 0.145. In contrast, asymmetric settings produce substantially larger recurrent-state changes, with $\Delta q_t = 0.286/0.301$ and $\Delta m^-_{t+1} = 0.064/0.071$. The same predicted weights therefore control current RTM refinement and the recurrent motion state used for subsequent motion prediction through the two update paths.

D MULTI-SEED STABILITY

We evaluate the stability of the principal motion-update comparisons over five independent training seeds on KITTI. Success and Precision are reported as mean±standard deviation. We additionally compute paired effects relative to each comparison method, with 95% confidence intervals obtained by bootstrap resampling complete target tracklets.

Stability of motion-update learning. Figure A3 compares MOPRU with the SiamMo baseline, scalar-update supervision, and separate update weights. These settings correspond to the principal comparisons in Table 4.

Across the five training seeds, MOPRU improves over SiamMo by 1.50/0.94 Success/Precision points on average. Relative to scalar-update supervision and separate update weights, the corresponding gains are 0.90/0.34 and 0.31/0.33 points, respectively. Figure A3 reports the paired effect distributions and their confidence intervals.

Category-level stability. Table A12 reports the five-seed results for the KITTI categories used in the main evaluation. The weighted mean uses the same category-frame weighting as Table 1.

Multi-seed comparison. The five-seed results preserve the trends observed in the main ablation. MOPRU reaches 73.80 ± 0.11 Success and 91.04 ± 0.08 Precision, compared with

Table A11: **Multi-seed results for motion-update learning on KITTI.** Success and Precision (%) are reported as mean±standard deviation over five independent training seeds. The compared settings follow the principal motion-update ablations in Table 4.

Configuration	Seeds	Success	Precision
SiamMo	5	72.30 ± 0.12	90.10 ± 0.09
Scalar update	5	72.90 ± 0.13	90.70 ± 0.10
Separate update weights	5	73.49 ± 0.12	90.71 ± 0.09
MoPRU	**5**	$\mathbf{73.80 \pm 0.11}$	$\mathbf{91.04 \pm 0.08}$

Table A12: **Category-level multi-seed results on KITTI.** Success/Precision (%) are reported as mean±standard deviation over five independent training seeds. The weighted mean follows the category-frame weighting in Table 1, and Δ denotes MoPRU minus SiamMo.

Category	SiamMo S/P	MoPRU S/P	Δ S/P
Car	76.36 ± 0.15 / 88.14 ± 0.10	78.36 ± 0.14 / 89.10 ± 0.09	$+2.00$ / $+0.96$
Pedestrian	68.60 ± 0.18 / 93.90 ± 0.12	67.70 ± 0.17 / 94.20 ± 0.11	-0.90 / $+0.30$
Van	67.90 ± 0.31 / 80.50 ± 0.20	78.80 ± 0.28 / 84.60 ± 0.18	$+10.90$ / $+4.10$
Cyclist	78.50 ± 0.22 / 94.80 ± 0.13	79.10 ± 0.19 / 95.10 ± 0.12	$+0.60$ / $+0.30$
Weighted mean	72.30 ± 0.12 / 90.10 ± 0.09	$\mathbf{73.80 \pm 0.11}$ / $\mathbf{91.04 \pm 0.08}$	$\mathbf{+1.50}$ / $\mathbf{+0.94}$

72.90 ± 0.13/90.70 ± 0.10 for scalar-update supervision and 73.49 ± 0.12/90.71 ± 0.09 for separate update weights. The category-level results show the same weighted mean as the main KITTI evaluation, while reporting the variation across independent training runs.

E Additional Motion-Update Ablations

We further examine the supervision used to learn the shared motion update, the capacity of the update estimator, the construction of the short-horizon tracking-score differences, category-shared parameterization, cross-dataset behavior, and transfer to a different 3D SOT tracker.

Supervision-target controls. The main MoPRU target is constructed by comparing short-horizon tracking outcomes under alternative current updates. We compare this supervision with two direct error-based targets while keeping the same update estimator and the same current-motion and recurrent-state update paths.

For the direct current-error target, let $e^j_{t,\mathrm{cur}}$ and $e^j_{t,\mathrm{prior}}$ denote the normalized current-motion errors of the pairwise estimate and motion prior for update dimension $j \in \{p, \psi\}$. We use

$$s^j_{t,\mathrm{cur}} = e^j_{t,\mathrm{prior}} - e^j_{t,\mathrm{cur}}, \qquad y^j_{t,\mathrm{cur}} = \sigma(s^j_{t,\mathrm{cur}}/\tau_j).$$

The direct future-error target instead aggregates tracking error over the same $H = 2$ horizon:

$$\bar{e}^p_{t,\mathrm{fut}} = \frac{\sum_{k=1}^{H} \gamma^k \min\big(d_c(B_{t+k}, B^\star_{t+k})/\kappa^{\mathrm{err}}_p, 1\big)}{\sum_{k=1}^{H} \gamma^k},$$

$$\bar{e}^\psi_{t,\mathrm{fut}} = \frac{\sum_{k=1}^{H} \gamma^k \min\Big(|\Delta\psi(B_{t+k}, B^\star_{t+k})|/\kappa^{\mathrm{err}}_\psi, 1\Big)}{\sum_{k=1}^{H} \gamma^k},$$

with

$$y^j_{t,\mathrm{fut}} = 1 - \bar{e}^j_{t,\mathrm{fut}}.$$

Learning from short-horizon outcome differences reaches 73.8/91.0 Success/Precision. Direct current-error and future-error supervision reach 72.9/90.3 and 72.8/90.4, respectively, while the scalar short-horizon update reaches 72.9/90.7. The comparison isolates the supervision signal while keeping the motion-update execution path unchanged.

Table A13: **Supervision-target controls on KITTI.** All two-weight variants use the same update estimator and both update paths; only the supervision target changes. Scalar update uses one shared update weight under the same short-horizon tracking protocol.

Configuration	Supervision target	Weight form	Success	Precision
Direct current-error target	Current motion error	Two-weight	72.9	90.3
Direct future-error target	Future tracking error	Two-weight	72.8	90.4
Scalar update	Short-horizon tracking outcome	Scalar	72.9	90.7
MOPRU	Short-horizon outcome difference	Two-weight	**73.8**	**91.0**

Matched update capacity. We next test whether the gain can be explained by adding a learned two-weight update alone. The current-frame update uses the same two-weight execution path as MOPRU but learns the update from the current tracking objective rather than subsequent tracking outcomes.

Table A14: **Matched-capacity control on KITTI.** The learned-update configurations use the same two-weight current-motion and recurrent-state update paths and differ in supervision.

Configuration	Supervision	Success/Precision
History only	None	72.8/90.3
Current-frame update	Current tracking objective	73.0/90.5
MOPRU	Short-horizon tracking outcomes	**73.8/91.0**

The current-frame update improves the history-only model by 0.2/0.2 points. Using subsequent tracking outcomes with the same two-weight execution path raises the result to 73.8/91.0.

Tracking-score-difference construction. The default short-horizon target averages each update dimension over the two settings of the other dimension, as defined in Equations 8 and 9. We compare this two-reference construction with score differences formed from a single reference setting.

Table A15: **Construction of short-horizon tracking-score differences on KITTI.** Single-reference variants compute each update preference from one current-update context, while the default averages both contexts.

Construction	p score difference	ψ score difference	Success	Precision
00 reference	$R_{10} - R_{00}$	$R_{01} - R_{00}$	73.4	90.7
11 reference	$R_{11} - R_{01}$	$R_{11} - R_{10}$	73.6	90.8
Two-reference average	Eq. 8	Eq. 9	**73.8**	**91.0**

Using both reference settings gives 73.8/91.0, compared with 73.4/90.7 and 73.6/90.8 for the two single-reference constructions. We use the two-reference average throughout the main experiments.

Category-shared update parameters. The default implementation uses the category-dependent update settings described in Appendix A. We additionally evaluate one shared parameter set for all KITTI categories without changing the trained motion-update formulation.

Table A16: **Category-shared update parameters on KITTI.** The shared variant uses one update-parameter set across all KITTI categories; the default uses category-dependent settings. Success/Precision (%) are reported.

Parameter scope	Car	Ped.	Van	Cyc.	Mean
Shared across categories	78.0/88.9	67.8/94.1	75.6/83.4	78.9/95.0	73.4/90.8
Category-dependent	**78.4/89.1**	67.7/94.2	**78.8/84.6**	**79.1/95.1**	**73.8/91.0**

A single shared parameter set reaches 73.4/90.8, compared with 73.8/91.0 for the default category-dependent setting. No category-specific parameter values are required for this comparison.

Cross-dataset supervision controls. We repeat the principal motion-update controls on NuScenes using the same evaluation protocol as the main experiment.

Table A17: **Motion-update supervision controls on NuScenes.** Mean Success/Precision (%) are reported using the same evaluation protocol as Table 2.

Configuration	Supervision	Success/Precision
SiamMo	–	60.31/72.68
History only	None	60.93/73.33
Current-frame update	Current tracking objective	61.34/73.97
Scalar update	Short-horizon tracking outcome	61.31/74.57
MoPRU	Short-horizon outcome differences	**62.93/75.66**

The same ordering is observed on NuScenes. Current-frame and scalar updates reach 61.34/73.97 and 61.31/74.57, respectively, while MoPRU reaches 62.93/75.66. The gain from short-horizon outcome supervision therefore extends beyond the KITTI training and evaluation setting.

Transfer across tracking frameworks. Table A18 gives the per-category KITTI results for the tracker-transfer experiment summarized in the main paper.

Table A18: **Per-category transfer across 3D SOT frameworks on KITTI.** MoPRU is applied to two 3D SOT trackers with different temporal designs. Each result uses the same evaluation protocol as its corresponding tracker baseline. Success/Precision (%) are reported.

Base tracker	Car	Ped.	Van	Cyc.	Mean
SiamMo	76.3/88.1	68.6/93.9	67.9/80.5	78.5/94.8	72.3/90.1
SiamMo + MoPRU	**78.4/89.1**	67.7/94.2	**78.8/84.6**	**79.1/95.1**	**73.8/91.0**
MBPTrack	73.4/84.8	68.6/93.9	61.3/72.7	76.7/94.3	70.3/87.9
MBPTrack + MoPRU	**74.2/85.4**	**68.7/94.1**	**64.0/74.5**	**77.0/94.5**	**71.0/88.4**

Integration with a memory-based tracker. For MBPTrack, the current relative target motion is recovered from its localization output and the previous target box. Earlier memory features provide the recurrent input, and a lightweight motion predictor produces the motion prior. Linear projections align the tracker features with the MoPRU update estimator. The predicted shared update weights then control current motion estimation and recurrent motion-state update through the same formulation used with SiamMo. MBPTrack's native feature encoder, voting, localization, box-prior sampling, and tracking objectives remain unchanged.

F Robustness and Category-Level Update Analysis

We further analyze MoPRU under challenging tracking conditions and examine how the two motion-update paths contribute across KITTI categories. The robustness analysis covers sparse observations, long occlusions, different-update cases, and abrupt target rotation. The category analysis uses the RTM-only and state-only variants from the main ablation to separate current RTM refinement from recurrent motion-state update.

Tracking under challenging observations. Table A19 evaluates MoPRU under four targeted conditions. Sparse-observation frames follow the lowest target-point interval in Figure 4, long occlusions contain at least nine consecutive frames, and translation–yaw disagreement denotes opposite binary update preferences under the short-horizon tracking score. Abrupt-rotation frames satisfy

$$\left|\mathrm{wrap}(\psi_t^\star - \psi_{t-1}^\star)\right| \geq \tau_{\mathrm{rot}}, \qquad \tau_{\mathrm{rot}} = 10^\circ.$$

In the sparsest KITTI interval, MoPRU improves SiamMo by 1.8/1.5 Success/Precision points. The gain increases to 3.3/2.9 points on different-update cases, where the short-horizon tracking score favors different binary settings across the two update dimensions. On abrupt-rotation cases, MoPRU

Table A19: **Targeted robustness analysis on KITTI.** (a) Success/Precision (%) are reported for sparse observations, translation–yaw update disagreement, and abrupt target rotation. Δ denotes the absolute improvement of MoPRU over SiamMo. (b) Long-occlusion center error compares SiamMo, RTM-only update, and the full shared motion update for occlusion streaks of at least nine frames.

(a) Tracking robustness

Evaluation subset	SiamMo	MoPRU	Δ S/P
Target points $[0, 15)$	68.8/87.0	**70.6/88.5**	$+1.8/+1.5$
Different-update cases	69.0/86.8	**72.3/89.7**	$+3.3/+2.9$
Abrupt-rotation cases	67.3/85.8	**70.7/88.8**	$+3.4/+3.0$

(b) Long-occlusion center error

Configuration	Center error (m) ↓
SiamMo	0.585
RTM-only update	0.481
MoPRU	**0.382**

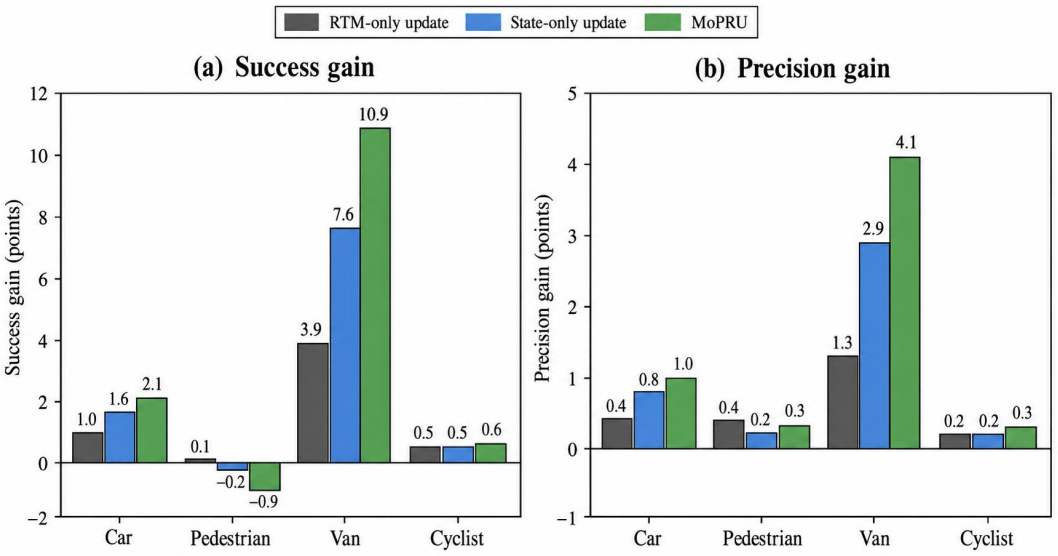


Figure A4: **Category-wise update-path analysis on KITTI.** Panels (a,b) report Success and Precision gains over SiamMo, respectively. RTM-only applies the learned update to current RTM refinement, State-only applies it to recurrent motion-state update, and MoPRU applies the shared motion update to both paths. Positive values indicate improvements over SiamMo.

further improves SiamMo by 3.4/3.0 points. These results show that the learned motion update remains effective when the current observation is sparse or when the preferred update changes across motion dimensions.

Under occlusion streaks of at least nine frames, the mean center error decreases from 0.585 m for SiamMo to 0.481 m with RTM-only update and to 0.382 m with the full MoPRU update. The additional reduction from the RTM-only variant to the full model shows the contribution of recurrent motion-state update when current observations provide limited tracking evidence.

Category-wise update-path analysis. The main ablation evaluates the two roles of the shared motion update separately through RTM-only and state-only variants. Figure A4 reports their category-wise gains over SiamMo together with the full MoPRU result.

The category-wise results show that the two update paths contribute differently across target categories. On Car, RTM-only and State-only updates improve Success by 1.0 and 1.6 points, while applying the shared motion update to both paths reaches a 2.1-point gain. Cyclist shows smaller but consistent

improvements across the three settings. Van exhibits the strongest dependence on recurrent motion-state update: RTM-only improves Success by 3.9 points, State-only by 7.6 points, and the full MoPRU update by 10.9 points.

Van update-path profile. For Van, SiamMo reaches 67.9/80.5 Success/Precision. RTM-only update improves the result to 71.8/81.8, while State-only update reaches 75.5/83.4. Applying the shared motion update to both paths further raises performance to 78.8/84.6. The larger gain from State-only update shows that recurrent motion-state evolution contributes substantially to Van tracking, while combining it with current RTM refinement gives the largest overall improvement.

Pedestrian update-path diagnostic. The Pedestrian diagnostic shows that the Success reduction is associated with the heading-related update on both execution paths. Fixing the current-motion weight to one increases Success from 67.7 to 68.1, while fixing the recurrent-state weight increases it to 68.3. Fixing both paths reaches 68.5 Success while retaining 94.2 Precision. The two paths therefore contribute jointly to the category-specific Success/Precision trade-off, with the recurrent-state path showing the larger individual effect.

Table A20: **Pedestrian update-path diagnostic on KITTI.** The heading-related update weight is fixed separately on the current-motion and recurrent-state paths while all other predictions remain unchanged.

Setting	Current-motion α^{ψ}	State α^{ψ}	Success	Precision
SiamMo	–	–	68.6	93.9
MoPRU	learned	learned	67.7	94.2
Current-motion path fixed	1	learned	68.1	94.0
Recurrent-state path fixed	learned	1	68.3	94.1
Both paths fixed	1	1	68.5	94.2

The diagnostic distinguishes the contribution of the heading-related update on current target localization from its effect on recurrent motion-state evolution. Fixing the two paths separately measures how each path contributes to the Pedestrian Success/Precision trade-off without retraining the tracker.

G Qualitative Tracking Analysis

Figure 6 presents representative closed-loop examples from the main paper. Figure A5 further examines sparse observations, long occlusion, abrupt rotation, and target loss, complementing the quantitative analysis in Appendix F.

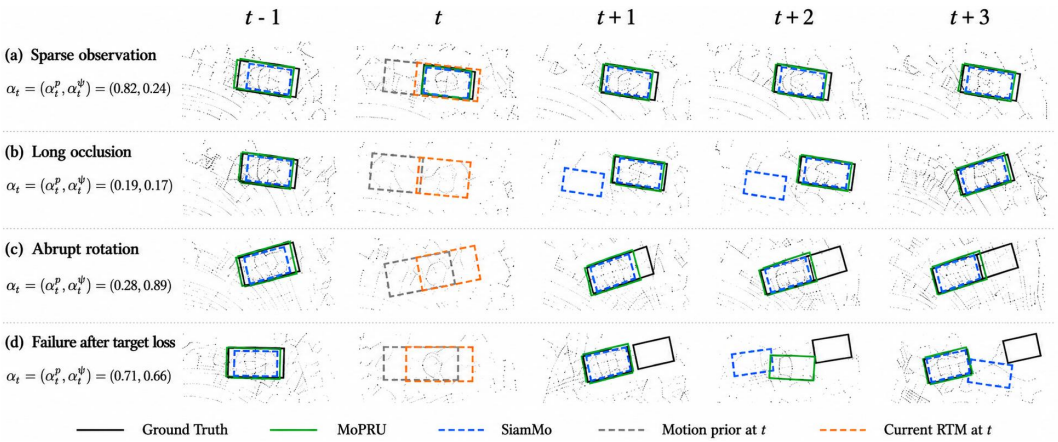


Figure A5: **Qualitative tracking under challenging observations.** Rows show sparse observation, long occlusion, abrupt rotation, and tracking failure from $t-1$ to $t+3$. Motion priors, current RTMs, and predicted shared update weights are additionally shown at frame t.

Sparse observation and long occlusion. Sparse target points reduce the reliability of the current RTM, while the motion prior provides motion information from the recurrent state. The shared update refines the current RTM and updates the recurrent motion state for subsequent prediction. During long occlusion, the recurrent state helps limit tracking drift until reliable observations resume.

Abrupt rotation and target loss. Abrupt rotation increases the discrepancy between the current RTM and motion prior. The shared update adjusts current RTM refinement and recurrent motion-state update accordingly. When the target leaves the search region, both motion estimates become unreliable and localization errors propagate to subsequent frames.

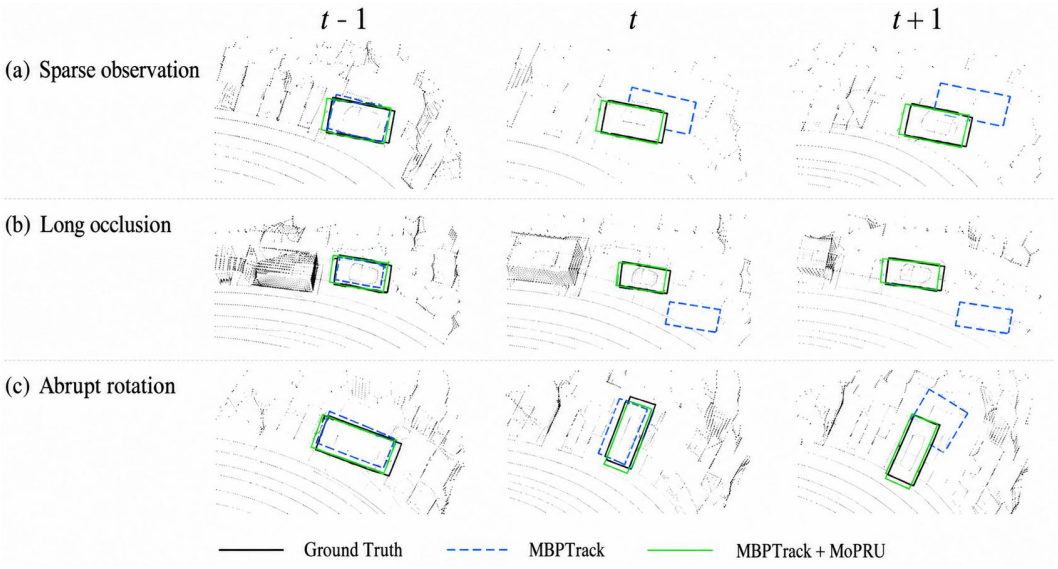


Figure A6: **Qualitative transfer to MBPTrack.** Rows show sparse observation, long occlusion, and abrupt rotation from $t-1$ to $t+1$.

Transfer to MBPTrack. Figure A6 complements Table 5 and Table A18. Under sparse observation, long occlusion, and abrupt rotation, MBPTrack + MoPRU remains closer to the ground-truth trajectory than the base tracker.

H Efficiency and Training Cost

We report online tracking efficiency and training cost separately. For inference, we include the efficiency reported with each tracker and matched-hardware measurements on an RTX 4090. For training, we report the additional cost of short-horizon tracking supervision and the wall-clock cost of the full MoPRU training pipeline.

Online tracking efficiency. Table A21 compares FLOPs and tracking throughput. Reported results retain the hardware and efficiency measurements from the corresponding tracker reports. Matched measurements use batch size 1 and the same precision mode, warm-up procedure, and timing window on an RTX 4090.

On matched hardware, MoPRU runs at 100 FPS compared with 106 FPS for SiamMo, while increasing computation from 0.82 to 1.39 GFLOPs per tracking step. PointRePar runs at 52 FPS under the same hardware setting. The additional recurrent motion prediction and shared motion update therefore retain real-time online tracking throughput.

Short-horizon supervision cost. The four current-update settings and their subsequent tracking outcomes are used only to construct training targets. They are not executed during online tracking. We therefore report their computational cost separately from online FPS.

Table A21: **Online tracking efficiency under reported and matched hardware.** GFLOPs are measured per tracking step. Reported rows follow the corresponding tracker reports, while matched rows are measured on the same RTX 4090 with batch size 1.

Tracker	GFLOPs	Throughput	GPU	Measurement
SiamMo	0.82	108 FPS	RTX 4090	Reported
PointRePar	6.66	37 FPS	RTX 3090	Reported
MoPRU	1.39	101 FPS	RTX 4090	Reported
SiamMo	0.82	106 FPS	RTX 4090	Matched
PointRePar	6.66	52 FPS	RTX 4090	Matched
MoPRU	1.39	100 FPS	RTX 4090	Matched

Table A22: **Training-time cost of short-horizon update supervision.** The measurement includes the four current-update settings and their short-horizon tracking propagation used to construct update targets. Throughput is reported in processed training anchors per second.

Training operation	GFLOPs	Throughput	GPU
Short-horizon update supervision	8.74	16.8 anchors/s	RTX 4090

This additional computation is confined to training. Online tracking uses a single current update and does not require future frames or multiple tracking branches.

End-to-end training cost. Table A23 reports wall-clock time and peak GPU memory for the executed training configurations on the same RTX 4090. The complete MoPRU pipeline follows the three stages in Table A2: pairwise motion, motion-prior prediction, and joint MoPRU optimization.

Table A23: **Training cost on matched hardware.** Wall-clock time and peak allocated GPU memory are measured on one RTX 4090. The stage-only row measures the incremental MoPRU training run after the pairwise model is available. Rows are timed independently, and relative cost is normalized by SiamMo end-to-end training time.

Training configuration	Wall-clock (h)	Relative	Peak memory (GB)	GPU-hours
SiamMo	11.8	1.00×	12.7	11.8
MoPRU stage only	7.1	–	18.4	7.1
Full MoPRU pipeline	19.9	1.69×	18.4	19.9

The complete MoPRU pipeline requires 19.9 GPU-hours on one RTX 4090, compared with 11.8 GPU-hours for SiamMo, corresponding to 1.69× end-to-end training cost. Peak allocated memory increases from 12.7 to 18.4 GB. The stage-only MoPRU run requires 7.1 GPU-hours after the pairwise model is available. Since the configurations are timed independently, the stage-only and SiamMo times are not intended to sum to the full-pipeline measurement.