GLEAN: RECOVERING UNDERUSED SIGNAL VIA TARGETED AUGMENTATION IN CLASSICAL 3D MULTI-OBJECT TRACKING

Anonymous authors
Paper under double-blind review

Classical tracking-by-detection pipelines for 3D multi-object tracking (MOT) provide a modular approach to trajectory estimation, yet potentially useful information available around individual tracking stages is often reduced to coarse representations or left unused. We present Glean, which systematically exploits such underused information at selected stages of an established LiDAR-camera tracking pipeline while preserving its core association mechanism. Built on Fusion-Poly, Glean enriches detection scoring with contextual quality cues, replaces rectangular geometric alignment targets with pixel-level instance masks, adapts Kalman uncertainty based on ego speed, and refines completed trajectories through Gaussian smoothing. These components target complementary sources of information while leaving the underlying FACM association cascade unchanged. Only the detection-quality component requires additional training. On the nuScenes validation set, Glean improves AMOTA from 77.294 to 77.775 over the reproduced Fusion-Poly baseline, corresponding to a gain of 0.481%p. Across five alternating timing runs, Fusion-Poly and the Glean tracking loop with cached SAM2 masks both average 409.8 ms per frame, with run-level standard deviations of 3.5 ms and 4.1 ms, respectively; SAM2 inference is excluded from this comparison. Experimental results demonstrate the effectiveness of the proposed Glean and its individual components, as evidenced by improvements in AMOTA.

1 INTRODUCTION

Three-dimensional multi-object tracking (3D MOT) aims to maintain object identities and estimate trajectories over time from sequential sensor observations (Weng et al., 2020; Li et al., 2023). LiDAR-camera tracking methods exploit complementary geometric and appearance information from the two modalities at different stages of the tracking pipeline (Kim et al., 2021; Wang et al., 2023; Xu et al., 2024). Recent learning-based trackers increasingly integrate such information within learned association or motion models, while classical tracking-by-detection pipelines retain modular detection, association, and state-estimation stages that can be modified independently (Weng et al., 2020; Li et al., 2023; 2024; Wu et al., 2026).

Fusion-Poly (Wu et al., 2026) is a recent LiDAR-camera tracking-by-detection framework that combines 3D and 2D detections through preprocessing, Frequency-Aware Cascade Matching (FACM), and Frequency-Aware Trajectory Estimation (FATE). Its preprocessing stage aligns cross-modal detections and filters low-confidence candidates, FACM associates synchronous and asynchronous detections with existing tracks, and FATE estimates track states using Kalman filtering. This modular structure provides a strong basis for examining whether information available around these stages can be used more effectively without replacing the underlying association mechanism.

Within this pipeline, several stages use only a limited representation of the information available at that point. The Score Filter applies a fixed class-specific confidence threshold to pure-3D detections, after which surviving detections are passed to association without further differentiation based on contextual detection cues. The Geometry-Aware Alignment Module (GAAM) aligns projected 3D boxes to rectangular image-space targets, while pixel-level segmentation can provide a more precise description of the visible object region. In FATE, fixed process and measurement noise matrices are used without explicitly adapting to the current ego speed. Finally, Fusion-Poly does not apply trajectory-level refinement after online tracking. We argue that these critical aspects and readily available sources of information are not adequately exploited or carefully considered in existing frameworks, leaving room for further improvement through their more effective use.

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.

Therefore, we introduce Glean, which systematically incorporates underused information at selected stages of the tracking-by-detection framework while preserving its overall tracking structure. Its components target complementary sources of information across detection scoring, geometric alignment, state estimation, and post-hoc trajectory refinement. The detection-quality filter uses contextual detection cues to re-weight pure-3D detections that survive the Score Filter and non-maximum suppression. Our Mask-GAAM replaces the rectangular alignment target with a pixel-level instance mask while retaining the original geometric optimization. We also propose Speed-Guided Linear Kalman Filtering (SG-LKF) that scales the existing Kalman process and measurement noise according to ego speed without changing the underlying filtering formulation. Finally, Gaussian-Smoothed Interpolation (GSI), originally introduced in StrongSORT (Du et al., 2023), is applied after online tracking to refine completed trajectories. The first three components operate within the online Fusion-Poly loop, whereas GSI is applied post-hoc. The detection-quality filter is the only component that requires additional training; Mask-GAAM uses precomputed segmentation masks, SG-LKF modifies the existing Kalman noise scaling, and GSI operates directly on completed trajectories. In this way, Glean incorporates each source of additional information at the stage where it is available without replacing the underlying association mechanism.

We evaluate Glean on the nuScenes validation set (Caesar et al., 2020) using Fusion-Poly as the underlying tracking pipeline. Glean improves AMOTA from 77.294 to 77.775 over our reproduced Fusion-Poly baseline, corresponding to a gain of 0.481%p. Across five alternating timing runs, both configurations run at an average of 409.8 ms per frame. Experimental results show that the proposed Glean and its individual components effectively improve tracking performance, as evidenced by consistent gains in AMOTA.

Our main contributions are:

- We propose Glean, which systematically incorporates underutilized information at various stages of a classical LiDAR-camera tracking-by-detection pipeline while preserving its core FACM association mechanism.
- We develop three complementary online components: contextual detection-quality re-weighting, mask-based geometric alignment, and ego-speed-aware Kalman noise scaling, and integrate post-hoc GSI for refinement of completed Fusion-Poly trajectories.
- On the nuScenes validation set, Glean improves AMOTA from 77.294 to 77.775 over the reproduced Fusion-Poly baseline, while the measured tracking loop with cached SAM2 masks shows no reproducible latency increase across five alternating timing runs.
- Through cumulative ablations and component-specific sensitivity analyses, we examine how the proposed components and their adopted settings affect downstream tracking performance.

2 Related Work

2.1 Classical Tracking-by-Detection and Motion Modeling

Classical 3D multi-object tracking commonly follows a tracking-by-detection framework, where frame-level detections are associated with existing tracks and used to update trajectory states. SORT (Bewley et al., 2016) introduced a simple tracking framework based on Kalman filtering and Hungarian matching, while DeepSORT (Wojke et al., 2017) incorporated appearance-based association. AB3DMOT (Weng et al., 2020) extended the tracking-by-detection paradigm to 3D multi-object tracking. Subsequent work has improved different components of this modular framework. The Poly-MOT lineage (Li et al., 2023; 2024) develops class-specific motion models, while Fusion-Poly (Wu et al., 2026) addresses the temporal heterogeneity of camera-LiDAR observations through frequency-aware processing of synchronous and asynchronous frames.

A separate line of work focuses on motion estimation and filtering. RobMOT (Nagy et al., 2025b;a) models observational noise and state-estimation drift using LiDAR point statistics, while IMM-MOT (Liu et al., 2025) considers multiple motion hypotheses through an interacting-multiple-model filter. OptiPMB (Ding et al., 2025) explores a Poisson multi-Bernoulli filtering formulation, and CANMOT (Osterburg et al., 2026) models process and measurement noise in a class-specific and

object-aligned manner. These methods primarily modify the underlying motion or filtering formulation. In contrast, Glean retains the Kalman filtering structure and introduces an ego-speed-dependent scaling of its process and measurement noise.

Post-processing provides another direction for improving tracking quality after online tracking. Gaussian-smoothed interpolation (GSI), introduced in StrongSORT (Du et al., 2023), performs training-free trajectory smoothing, while subsequent work has explored offline trajectory refinement (Huang et al., 2025; Li et al., 2026). Glean follows this post-processing direction by applying GSI to the trajectories produced by the existing tracking pipeline without modifying online association or state estimation.

2.2 Multi-Modal Fusion and Detection Refinement

LiDAR-camera tracking methods exploit complementary information from the two modalities at different stages of the tracking pipeline. EagerMOT (Kim et al., 2021), CAMO-MOT (Wang et al., 2023), and EMMS-MOT (Xu et al., 2024) incorporate cross-modal information into tracklet fusion, association, or state estimation, while DINO-MOT (Lee et al., 2024; Oquab et al., 2024) uses visual features for pedestrian re-identification. FusionTrack (Zeng et al., 2024) further exploits appearance information and 2D detection results within a camera-LiDAR tracking framework. Fusion-Poly (Wu et al., 2026) instead focuses on the temporal heterogeneity between camera and LiDAR observations. Glean addresses a different aspect of multimodal tracking by retaining the existing fusion and association structure while incorporating additional information at selected decision points.

Detection refinement provides another way to exploit complementary sensor information. CLOCs (Pang et al., 2020), for example, combines 2D and 3D detection candidates before non-maximum suppression. In contrast, Glean augments distinct stages of the existing preprocessing pipeline without introducing a new association mechanism. Our Mask-GAAM refines the geometric alignment target using pixel-level segmentation, while the detection-quality filter re-weights pure-3D detections that survive the original Score Filter using contextual detection cues. These modifications preserve the underlying association procedure rather than replacing the existing multimodal tracking pipeline.

3 Method

3.1 Overall Architecture

At each frame t, Fusion-Poly (Wu et al., 2026) receives a set of 3D detections D_t^{3D} from the CenterPoint 3D detector (Yin et al., 2021) at synchronized frames and, for each camera, a set of 2D detections D_t^{2D} from the Cascade R-CNN 2D detector (Cai & Vasconcelos, 2018) at every frame. Fusion-Poly consists of three main stages: preprocessing, Frequency-Aware Cascade Matching (FACM), and Frequency-Aware Trajectory Estimation (FATE). The preprocessing stage performs cross-modal matching, geometric alignment, and score filtering. FACM associates detections with existing tracks by associating synchronous and asynchronous detections, while FATE predicts and updates track states with a Kalman filter and manages track birth and death.

Glean retains this overall tracking structure and augments selected stages where additional information can be incorporated without replacing the underlying association mechanism. As illustrated in Figure 1, Mask-GAAM uses pixel-level instance geometry in place of rectangular image-space targets during geometric alignment, while retaining the original optimization procedure. The detection-quality filter uses contextual detection cues to re-weight pure-3D detections that survive the original Score Filter/NMS. SG-LKF incorporates ego speed into state estimation by scaling the existing Kalman process and measurement noise matrices. These three components operate within the online tracking loop, whereas GSI uses completed trajectories for post-hoc smoothing after online tracking.

The FACM association cascade remains unchanged throughout these interventions. Each component is designed to preserve the corresponding Fusion-Poly behavior when disabled: Mask-GAAM restores the original rectangular alignment target, disabling detection-quality re-weighting leaves the original detection scores unchanged, disabling ego-speed-aware noise scaling restores the original Kalman noise matrices, and omitting GSI leaves the online trajectories unchanged. The following sections describe the motivation and formulation of each component in detail.

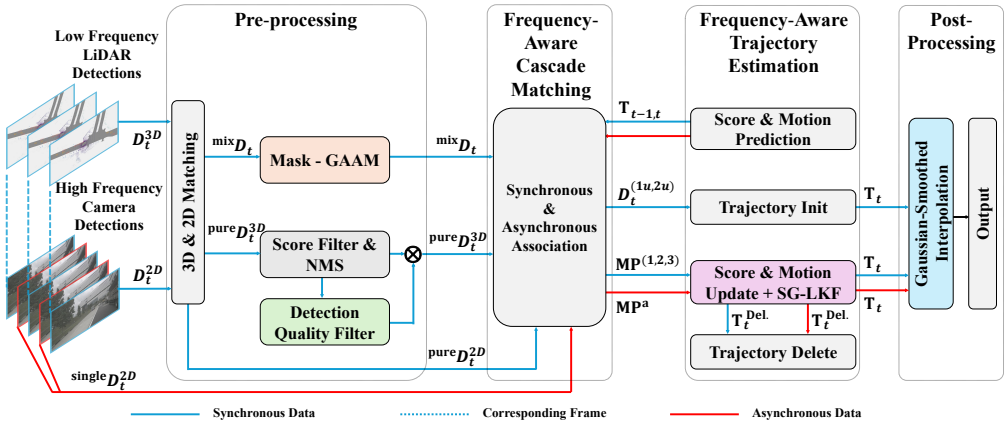

Figure 1: Overall architecture of Glean built on Fusion-Poly. Mask-GAAM and the detection-quality filter modify the preprocessing stage, SG-LKF modifies state estimation in FATE, and GSI is applied after online tracking. The FACM association cascade is unchanged.

3.2 Mask-GAAM: Pixel-Level Geometric Alignment

Fusion-Poly refines matched *mix* detections (*i.e.*, matched 3D and 2D detections) using its Geometry-Aware Alignment Module (GAAM), which adjusts a 3D box by comparing its image-plane projection with a 2D alignment target. In the original GAAM, this target is represented by a rectangular region. While the rectangle provides an image-space constraint for optimization, it inevitably includes background pixels outside the visible object region and does not capture the object's pixel-level shape. The alignment objective therefore operates on a coarse approximation of the image-space geometry available for the matched object.

Mask-GAAM replaces this rectangular target with a pixel-level instance mask while retaining the original GAAM optimization procedure. For each *mix* detection, we use its projected 2D box as a prompt to SAM2 (Ravi et al., 2025) and obtain an instance mask for the corresponding object. The masks are precomputed offline and cached for use during tracking. This substitution allows the alignment objective to evaluate the projected 3D box against the visible object region rather than its enclosing rectangle, without introducing a separate alignment model or modifying the subsequent FACM association procedure. An illustrative comparison of the GAAM and Mask-GAAM alignment targets is provided in Appendix B.

Formally, let B^{3D} denote the 3D box parameters, $\mathcal{P}(\cdot)$ denote the 3D-to-2D projection operator determined by the camera intrinsics and extrinsics, and $T(d)$ denote the original rectangular alignment target for detection d.

The original GAAM objective is

$$B^{3D*} = \arg\min_{B^{3D}} \left[1 - \mathrm{IoU}\left(\mathcal{P}(B^{3D}), T(d)\right)\right]. \tag{1}$$

For Mask-GAAM, let $M(d)$ denote the SAM2 (Ravi et al., 2025) instance mask obtained for detection d. We retain the same objective and replace only the alignment target:

$$B^{3D*} = \arg\min_{B^{3D}} \left[1 - \mathrm{IoU}\left(\mathcal{P}(B^{3D}), M(d)\right)\right]. \tag{2}$$

The optimized box parameters, objective form, and Trust-Region-Reflective (TRF) solver remain unchanged from the original GAAM. Thus, Mask-GAAM changes the geometric evidence used by the alignment objective rather than the optimization procedure itself. Note that replacing $M(d)$ with $T(d)$ recovers the original GAAM formulation.

3.3 Detection-Quality Filter

Fusion-Poly applies a fixed class-specific Score Filter to pure-3D detections before association. Once a detection passes this threshold, however, its confidence is carried forward without considering contextual evidence that may distinguish reliable detections from ambiguous ones. In particular,

detections with similar detector scores can exhibit different box geometry, local duplicate structure, or score distributions relative to other candidates in the same frame. Treating such detections with the same confidence discards valuable information that is already available before association.

To incorporate this information, we estimate detection quality from contextual properties of each surviving pure-3D detection. For an eligible detection d, we construct a 26-dimensional feature vector $\phi(d)$ from six groups: detector score and calibration features, box geometry relative to class statistics, ego-relative geometry, local duplicate statistics, frame-level score-rank context, and class identity. The adopted feature set contains no camera-derived or cross-modal features. A two-layer MLP predicts a calibrated quality probability

$$p(d) = \mathrm{MLP}_\theta(\phi(d)), \qquad p(d) \in [0, 1]. \tag{3}$$

We use this estimate to define a quality-dependent score multiplier

$$q(d) = \max(1 - w(1 - p(d)),\ f), \tag{4}$$

where w controls the strength of quality-based re-weighting and f specifies a lower bound on the multiplier. The original detector confidence is then updated as

$$\mathrm{score}'(d) = \mathrm{score}(d)\, q(d). \tag{5}$$

For $p(d) \in [0, 1]$, the multiplier satisfies $q(d) \in [\max(1 - w, f), 1]$. The quality estimator therefore preserves or reduces the original confidence rather than increasing it. With the adopted $w = 0.4$ and $f = 0$, $q(d)$ is restricted to $[0.6, 1.0]$: detections assigned high predicted quality remain close to their original confidence, whereas those assigned lower quality receive a larger downward adjustment.

This re-weighting is applied only to pure-3D detections that have already survived the original Score Filter/NMS. It does not restore rejected detections or generate additional candidates, and the resulting scores are passed to the unchanged FACM association procedure. This formulation is also conservative with respect to the Fusion-Poly baseline: setting $w = 0$ yields $q(d) = 1$ for every eligible detection and exactly recovers the original detection scores.

The quality estimator is the only learned component introduced by Glean. The adopted classifier has hidden dimensions $[64, 32]$, `tanh` activations, a sigmoid output, and Platt calibration (Platt, 1999). It is trained on 483,038 pure-3D Score Filter survivors collected from the 700 nuScenes training scenes, with 105 training scenes held out at the scene level for architecture selection, feature-group analysis, and calibration. The nuScenes validation scenes used for tracking evaluation are not used to train the classifier or compute its input features. Complete optimization settings and additional feature analyses are provided in Appendices E, F, and H.

3.4 SG-LKF: Ego-Speed-Aware Kalman Noise Scaling

In FATE, Fusion-Poly estimates each track state using a Kalman filter with fixed process and measurement noise matrices Q and R. These matrices control the uncertainty assigned to state prediction and measurement update, respectively, but the same noise setting is used regardless of the ego vehicle's motion. As a result, the uncertainty model does not explicitly distinguish between frames acquired at low and high ego speeds, even though ego-motion information is available at synchronized frames.

We introduce Speed-Guided Linear Kalman Filter (SG-LKF) to condition the existing Kalman uncertainty on ego speed while preserving the state representation and filtering procedure of FATE. At each synchronized frame, we obtain the current ego speed v_{ego} from the nuScenes ego pose information and compute a speed-dependent scaling factor

$$s_t = 1 + \alpha \frac{v_{\mathrm{ego},t}}{v_{\mathrm{ref}}}, \tag{6}$$

where $v_{\mathrm{ref}} = 10.0$ m/s is a fixed normalization constant and $\alpha \geq 0$ controls the sensitivity of the uncertainty scaling to ego speed. The process and measurement noise matrices used at frame t are then

$$Q_t = s_t Q, \qquad R_t = s_t R. \tag{7}$$

The scaled matrices Q_t and R_t are supplied to the same Kalman prediction and measurement-update procedure used by the original FATE. SG-LKF therefore modifies the uncertainty assigned to the existing motion and observation models rather than introducing a new state-transition or measurement

model. As $v_{\text{ego},t}$ increases, s_t increases accordingly, allowing the filter to operate with larger process and measurement uncertainty at higher ego speeds.

Because $s_t \geq 1$, SG-LKF never reduces either noise matrix below its original Fusion-Poly value. At asynchronous camera-only frames, ego-speed scaling is not applied and we set $s_t = 1$, so the original noise matrices are used. Likewise, setting $\alpha = 0$ yields $s_t = 1$ at all frames and recovers the original Fusion-Poly filtering behavior. We use $\alpha = 0.6$, selected by validation AMOTA; the corresponding sensitivity analysis is reported in Appendix D.

3.5 Gaussian-Smoothed Interpolation

Fusion-Poly produces trajectories through online frame-by-frame association and state estimation, but does not apply trajectory-level refinement after a track is completed. Consequently, local irregularities that remain in the online state estimates are preserved in the final trajectory, even though the completed trajectory provides temporal context that was unavailable during online estimation. To exploit this post-hoc information without changing track identities or preceding association decisions, we apply Gaussian-Smoothed Interpolation (GSI), originally introduced in StrongSORT (Du et al., 2023), to completed Fusion-Poly trajectories.

GSI operates independently on each completed trajectory. It first linearly interpolates temporal gaps of at most `interval` frames. The resulting trajectory positions are then smoothed using Gaussian-process regression with a radial basis function (RBF) kernel

$$k(t, t') = \exp\left(-\frac{(t - t')^2}{2\ell^2}\right), \tag{8}$$

where ℓ denotes the effective kernel length scale. Following the original GSI implementation, the effective length scale is determined from the smoothing parameter τ together with the trajectory length rather than setting $\ell = \tau$ directly. We use `interval`$= 2$ and $\tau = 10$, selected by validation AMOTA; the corresponding sensitivity analysis is reported in Appendix D.

GSI is applied only after online tracking is complete and therefore does not alter FACM association or FATE state estimation. In the adopted configuration, no trajectory gaps satisfy the interpolation condition at `interval`$= 2$; the contribution observed in our experiments therefore comes from the Gaussian smoothing step rather than gap interpolation. Because this refinement is performed post hoc, GSI is excluded from the online per-frame latency analysis in Appendix G.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate Glean on the nuScenes dataset (Caesar et al., 2020), which contains 700 training, 150 validation, and 150 test scenes. Following the high-frequency setting of Fusion-Poly (Wu et al., 2026), we use the 2 Hz synchronized (sync) LiDAR-camera keyframes together with the intermediate camera-only asynchronous (async) frames at 4 Hz. All reported results are evaluated on the validation split over the seven official tracking classes.

We use AMOTA (Weng et al., 2020) as the primary metric and additionally compute AMOTP, Recall, MOTAR, MOTA, false positives (FP), false negatives (FN), and identity switches (IDS). All metrics are computed using the official nuScenes tracking evaluation protocol. Detailed overall tracking metrics for the controlled Fusion-Poly comparison are reported in Appendix A.

4.2 Implementation Details

We use the unmodified Fusion-Poly (Wu et al., 2026) tracking pipeline as our baseline, which achieves an AMOTA of 77.294 in our evaluation. Following Fusion-Poly, we use CenterPoint (Yin et al., 2021) for 3D detection and Cascade R-CNN (Cai & Vasconcelos, 2018) for 2D detection. The same precomputed detector outputs are shared across the reproduced baseline and all Glean configurations.

The detection-quality filter is the only Glean component that requires training. We train its two-layer MLP on 483,038 pure-3D detections that survive the original Score Filter, collected from the

700 nuScenes training scenes. We hold out 105 training scenes at the scene level for architecture selection, feature-group analysis, and calibration. The nuScenes validation scenes used for tracking evaluation are not used to train the classifier or compute its input features. Mask-GAAM, SG-LKF, and GSI require no additional model training.

For Mask-GAAM, we precompute and cache SAM2 masks using the projected 2D boxes of *mix* detections as prompts. In an online deployment, the same detection-derived prompts would instead be used to generate the masks on the current frame. Thus, caching changes when the masks are computed, not the information used for tracking. The hyperparameters w, α, `interval`, and τ are selected using validation AMOTA. The sensitivity analyses for α, `interval`, and τ are reported in Appendix D, while the analysis of w is provided in Appendix E. Complete training settings and implementation details are provided in Appendices F and H.

4.3 Main Results

4.3.1 Comparison with Prior Methods

Table 1 compares Glean with prior 3D MOT methods on the nuScenes validation set. Following Fusion-Poly (Wu et al., 2026), we report the published results of the compared methods, while the Glean result is obtained from our own evaluation.

Table 1: Comparison with prior 3D MOT methods on the nuScenes **val** set (Zhang et al., 2023; Kim et al., 2021; Li et al., 2023; Lee et al., 2024; Li et al., 2024; Wang et al., 2025; Xu et al., 2024; Wang et al., 2023; Wu et al., 2026). CP: CenterPoint; CR: Cascade R-CNN; BEVF: BEVFusion; FC: FocalsConv; C+L: camera+LiDAR; L: LiDAR only.

Method	Detector	Input	AMOTA (%)↑	MOTA (%)↑	IDS↓
MotionTrack	MotionTrack	C+L	64.1	59.7	—
EagerMOT	CP&CR	C+L	71.2	—	899
Poly-MOT	CP	L	73.1	61.9	232
DINO-MOT	CP&DINOv2	C+L	73.1	—	216
Fast-Poly	CP	L	73.7	63.2	414
MC-Track	CP	L	74.0	64.0	275
EMMS-MOT	CP&CR	C+L	74.3	—	247
CAMO-MOT	BEVF&FC	C+L	76.3	—	239
Fusion-Poly	CP&CR	C+L	77.1	67.3	416
Glean	CP&CR	C+L	**77.8**	**67.7**	417

Glean achieves an AMOTA of 77.8 on the nuScenes validation set. Since GSI is applied as post-hoc trajectory smoothing, the complete Glean configuration is not strictly online; we therefore use the reproduced Fusion-Poly baseline as the primary controlled comparison for evaluating Glean.

4.3.2 Comparison with Fusion-Poly

Table 2: Overall and per-class AMOTA (%) of Fusion-Poly and Glean on the nuScenes validation set. † denotes our reproduced Fusion-Poly baseline using the official implementation.

	AMOTA	bicycle	bus	car	motorcycle	pedestrian	trailer	truck
Fusion-Poly†	77.294	72.95	87.31	86.50	81.35	84.63	54.56	73.76
Glean	**77.775**	**73.03**	**87.98**	**86.54**	**81.42**	**84.78**	**56.75**	**73.93**

Glean improves AMOTA from 77.294 to 77.775, corresponding to a gain of 0.481 percentage points over the reproduced Fusion-Poly baseline. Per-class results in Table 2 show improvements across all seven tracking classes, with the largest gain on trailer, where AMOTA increases from 54.56 to 56.75 (+2.19 pp). Additional analysis shows that trailer trajectories in the evaluated output have a mean length of 3.14 frames, the shortest among the evaluated classes. Together with the observation that the adopted GSI configuration performs smoothing without gap interpolation, this is consistent with

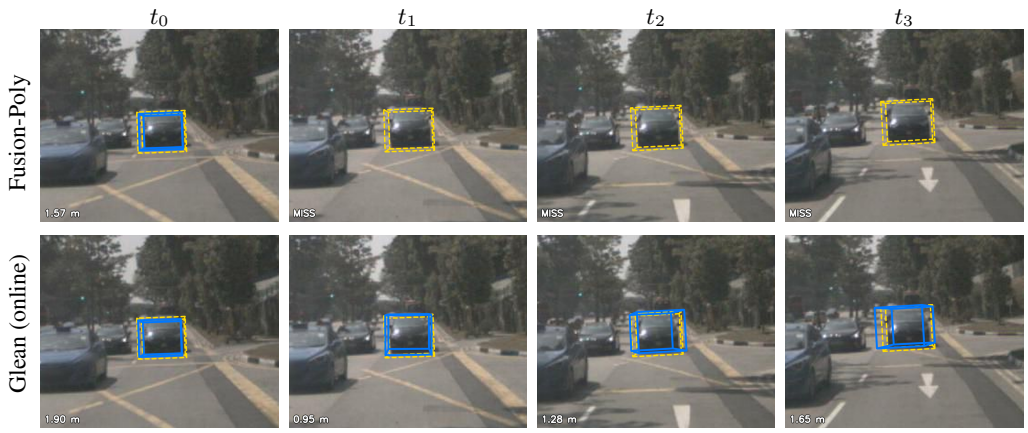

Figure 2: Qualitative sequence comparison between Fusion-Poly and online Glean on the nuScenes validation set. Columns show four consecutive frames of the same car trajectory. Both methods are matched to the ground-truth object in the first frame, after which Fusion-Poly becomes unmatched for three consecutive frames while online Glean remains matched. Dashed yellow boxes denote ground truth, solid blue boxes denote tracker predictions, and d denotes the BEV center distance for matched predictions. The online Glean results are shown before post-hoc GSI refinement.

trajectory smoothing contributing to the larger trailer improvement. Further component-level analysis for the trailer class is provided in Appendix E. The additional tracking metrics in Appendix A show higher Recall and lower FP, FN, and IDS for Glean, while MOTAR decreases slightly.

Figure 2 illustrates a temporal difference between the two online configurations. Both methods are matched to the same ground-truth car at t_0, but Fusion-Poly becomes unmatched in the following three frames, whereas online Glean remains matched throughout the interval. Because the displayed Glean output is taken before GSI, the difference shown here is present in the online configuration rather than being introduced by post-hoc trajectory refinement. The example therefore complements the aggregate results in Table 2 by illustrating the effect of the online modifications over consecutive frames; the quantitative results remain the primary evidence.

4.4 Ablation Study

We evaluate both the cumulative effect of the four Glean components and the design choices used in the adopted configuration. Scalar module parameters are selected using validation AMOTA, with the corresponding sensitivity analyses reported in the appendix. Because these values are selected on the validation split, we do not interpret the selected points as evidence of optimality on unseen data.

Cumulative ablation. Table 3 progressively adds the four Glean components to the Fusion-Poly baseline in their pipeline order. AMOTA increases from 77.294 for Fusion-Poly to 77.391 with Mask-GAAM, 77.444 with the detection-quality filter, 77.494 with SG-LKF, and 77.775 after GSI. Because each row builds on the preceding configuration, these differences characterize the cumulative behavior of the composed system rather than an independent additive contribution from each component. Mask-GAAM yields the lowest AMOTP and IDS among the intermediate configurations, while the complete system remains better than the baseline on both metrics.

Component behavior. The cumulative results in Table 3 reflect interventions at different stages of the pipeline rather than a single shared mechanism. To further examine their effect on the final tracking output, we remove each component individually from the complete configuration and compare the resulting evaluation counts. This analysis shows that removing the detection-quality filter changes the reported true-positive count even though the filter neither restores rejected detections nor generates new candidates. Instead, its bounded score correction changes how retained detections propagate through downstream association and evaluation. Because each removal is measured in the presence of the remaining components, these differences are not interpreted as an additive decom-

Table 3: Cumulative ablation of Glean. Each row adds one component to the preceding configuration.

Configuration	AMOTA (%)↑	AMOTP (m)↓	IDS↓
Fusion-Poly†	77.294	0.5487	423
+Mask-GAAM	77.391	**0.5402**	**398**
+Det-Quality	77.444	0.5415	418
+SG-LKF (pre-GSI)	77.494	0.5417	421
+GSI (Glean)	**77.775**	0.5409	417

Table 4: Detection-quality feature-set comparison. AUC and AMOTA are reported on a 0–100 scale.

Feature set	Dim.	AUC (%)↑	AMOTA (%)↑
Score-only	2	89.65	77.362
Adopted	26	91.50	**77.494**
+Cross-modal	29	**92.80**	77.467

position of the overall improvement. Detailed counts and component-level analyses are provided in Appendix E.

Detection-quality feature set. Table 4 examines whether the detection-quality filter mainly acts as a recalibration of the original detection score. Using only the two score/calibration features reaches an AMOTA of 77.362, whereas the adopted 26-dimensional feature set reaches 77.494. The corresponding held-out AUC also increases from 89.65 to 91.50. The improvement in held-out AUC indicates that the contextual feature groups provide predictive information beyond the score/calibration features alone.

Adding three cross-modal features increases held-out AUC further to 92.80 but reduces downstream AMOTA to 77.467. We therefore adopt the 26-dimensional feature set, which contains no cross-modal features. These results also indicate that a higher standalone classifier AUC does not necessarily translate into higher downstream tracking AMOTA. This distinction is relevant because the quality estimator is not used as an independent detection classifier at inference time; its prediction only modulates the confidence of detections that have already survived the original Score Filter/NMS before they enter the downstream association pipeline. The adopted feature set is therefore selected according to downstream tracking performance rather than classifier AUC alone. A leave-one-group-out analysis of the adopted feature set is provided in Appendix E.

GSI sensitivity. We select the GSI parameters by validation AMOTA. The sensitivity analysis in Appendix D shows that the configuration with $\texttt{interval} = 2$ and $\tau = 10$ achieves the highest AMOTA among the evaluated configurations, while the top four reported settings lie within 0.058%p of one another. With the adopted setting, no gaps in the online trajectories satisfy the interpolation condition, indicating that the observed improvement comes from trajectory smoothing rather than gap interpolation. We therefore treat the selected parameters as validation-chosen settings rather than evidence of a sharply defined optimum.

SG-LKF sensitivity. We evaluate $\alpha \in \{0.3, 0.6, 1.0\}$ with SG-LKF added directly to the Fusion-Poly baseline. The sensitivity analysis in Appendix D shows that $\alpha = 0.6$ achieves the highest AMOTA among the evaluated values, with only a 0.002%p difference from $\alpha = 1.0$.

5 Conclusion

We presented Glean, an augmentation framework for classical LiDAR-camera tracking-by-detection that incorporates additional information at selected stages while preserving the underlying tracking structure. Built on Fusion-Poly, Glean introduces contextual detection-quality estimation, pixel-level geometric alignment, ego-speed-aware Kalman noise scaling, and post-hoc trajectory smoothing without modifying the core FACM association mechanism. On the nuScenes validation set, Glean improves AMOTA from 77.294 to 77.775 (+0.481 pp). Across five alternating timing runs, the measured tracking loop with cached SAM2 masks shows no reproducible latency increase over Fusion-Poly; SAM2 inference is excluded from this comparison. Online deployment requires additional segmentation inference for Mask-GAAM, although masks are prompted only for eligible detection boxes. Future work could reduce this overhead using a lightweight segmentation model. Cumulative ablations show incremental AMOTA improvements as the four components are progressively introduced. Overall, the results show that targeted use of underused information can improve tracking performance while preserving the core association mechanism.

AI USE STATEMENT

Generative AI tools were used to assist with code development and debugging, experiment execution, and drafting and copy-editing of the manuscript. The research questions, methodological decisions, experimental design, analysis, and interpretation of the results were determined by the authors. All AI-assisted code, experimental outputs, and manuscript content were reviewed by the authors, who take responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We use the unmodified official Fusion-Poly implementation as the reproduced baseline and use the same precomputed detector outputs across all controlled configurations. The training and held-out split for the detection-quality classifier, hyperparameter selection procedures, and final module settings are documented in Section 4.2 and Appendices E, F, and H. Appendix I provides the online-loop pseudocode, while Appendix G reports the timing protocol and its measurement limitations.

A ADDITIONAL TRACKING METRICS

Table 5 reports additional tracking metrics for the controlled comparison between the reproduced Fusion-Poly baseline and Glean. In addition to the AMOTA improvement reported in the main text, Glean increases Recall from 80.34 to 81.16 and MOTA from 67.21 to 67.70, while reducing AMOTP from 0.5487 m to 0.5409 m. The final configuration also reduces FP from 13,057 to 12,997, FN from 15,739 to 15,564, and IDS from 423 to 417. MOTAR decreases slightly from 83.56 to 83.28.

Table 5: Overall tracking performance of Fusion-Poly and Glean. Percentage-based metrics are reported on a 0–100 scale.

	AMOTA↑	AMOTP (m)↓	Recall↑	MOTAR↑	MOTA↑	TP↑	FP↓	FN↓	IDS↓
Fusion-Poly†	77.294	0.5487	80.34	**83.56**	67.21	85735	13057	15739	423
Glean	**77.775**	**0.5409**	**81.16**	83.28	**67.70**	**85916**	**12997**	**15564**	**417**

B ADDITIONAL QUALITATIVE RESULTS

Figure 3 provides additional qualitative examples for the trailer class, which shows the largest per-class AMOTA improvement in Table 2. These examples complement the temporal comparison in Figure 2; the aggregate per-class results remain the primary evidence for the improvement.

C GSI STRUCTURAL VARIANTS

In addition to the `interval`/τ sensitivity analysis reported in Appendix D (Table 6), we examined two possible modifications to the standard GSI configuration. These analyses were conducted during development and are reported here to clarify why neither modification was included in the final configuration.

Ego-speed-dependent smoothing. We first examined whether ego speed could provide a useful signal for adapting the GSI smoothing strength. For each trajectory, we computed the mean ego speed over its frames and a trajectory residual defined by fitting a constant-velocity linear model independently to the x- and y-coordinates over time and taking the root-mean-square residual. This residual serves as a proxy for deviation from constant-velocity motion. Across 13,201 trajectories with at least four frames and available ego-speed measurements, the Pearson correlation between mean ego speed and the trajectory residual was $r = -0.063$. The magnitude of the correlation was small and its direction did not support using higher ego speed as an indicator for stronger smoothing. We therefore did not implement ego-speed-dependent GSI in the final system.



Figure 3: Additional qualitative comparisons between Fusion-Poly and Glean for the trailer class on the nuScenes validation set. Top: Fusion-Poly. Bottom: Glean. Dashed boxes denote ground truth, solid boxes denote tracker predictions, and d denotes the BEV center distance between the displayed prediction and the corresponding ground-truth trailer. Across these examples, Glean produces predictions with smaller center distances to the ground-truth trailers than Fusion-Poly.

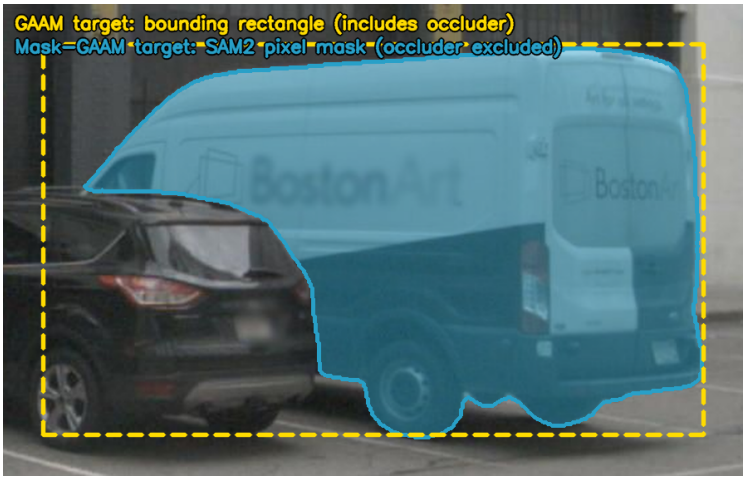

Figure 4: Illustration of the alignment targets used by GAAM and Mask-GAAM. The original GAAM uses a rectangular image-space target (yellow dashed box), which includes pixels belonging to the occluding vehicle. Mask-GAAM instead uses a SAM2 instance mask (blue region) that follows the visible region of the target object and excludes the occluder.

Gap-conditioned smoothing. We also evaluated a variant that applies GSI only to trajectories containing detection gaps, thereby skipping smoothing for trajectories without gaps. This variant reduced AMOTA by 0.15 pp relative to applying GSI to all eligible trajectories, with the largest degradation observed for the trailer class. In the adopted configuration, `interval`=2 bridges no trajectory gaps, as also reflected in the sensitivity analysis in Appendix D. Together, these observations indicate that the improvement from the adopted GSI configuration is associated with trajectory smoothing rather than gap interpolation.

Neither modification was included in the final Glean configuration.

D Hyperparameter Sensitivity

GSI parameters. Table 6 reports representative configurations from the GSI parameter grid. Among the evaluated settings, the configuration with `interval`$=$ 2 and $\tau = 10$ achieves the

highest validation AMOTA and is used in the complete Glean configuration. The top four reported configurations differ by only 0.058 pp in AMOTA. Under the adopted setting, no trajectory gaps satisfy the interpolation condition, so GSI operates as trajectory smoothing rather than gap interpolation.

Table 6: Representative results from the GSI hyperparameter grid, ranked by AMOTA.

`interval`	τ	AMOTA (%)↑
2	10	**77.775**
4	10	77.744
5	10	77.743
3	10	77.717
10	10	77.557
20	20	77.411

SG-LKF scaling. Table 7 reports the sensitivity to α when SG-LKF is added directly to the Fusion-Poly baseline. The three evaluated settings produce similar AMOTA values, with $\alpha = 0.6$ achieving the highest value. The difference between $\alpha = 0.6$ and $\alpha = 1.0$ is 0.002 pp.

Table 7: SG-LKF sensitivity to α, evaluated on the Fusion-Poly baseline.

α	AMOTA (%)↑
0.3	77.288
0.6	**77.313**
1.0	77.311

E Detection-Quality Training and Additional Analysis

Training and validation separation. The detection-quality classifier is trained exclusively on detections from the 700 nuScenes training scenes. The 150-scene validation split used for the tracking results in this paper is not used for feature computation or classifier training. Within the training split, 105 scenes are held out at the scene level for architecture selection, feature-group ablation, and calibration, as described in Appendix H. The blend weight w is subsequently selected using validation-set AMOTA, whereas the floor $f = 0$ is fixed.

Platt scaling is applied to the classifier logit using the held-out training scenes. The fitted calibration parameters are $a = 0.985$ and $b = -0.164$.

Sensitivity of validation-selected hyperparameters. Several hyperparameters are selected using validation AMOTA. The evaluated results are relatively flat near the adopted settings: the top four reported GSI configurations span 0.058 pp (Table 6), $w = 0.2$ and $w = 0.4$ differ by 0.008 pp (Table 8), and $\alpha = 0.6$ and $\alpha = 1.0$ differ by 0.002 pp (Table 7). These results indicate that the adopted values lie in relatively flat regions of the evaluated parameter ranges. Nevertheless, because these parameters are selected on the validation set, we do not claim that the same values are optimal on unseen data.

Detection-quality blend weight. The feature-set comparison in Table 4 evaluates the classifier input design, whereas Table 8 examines the strength of its downstream score correction. The sweep is performed on the same pre-GSI configuration with Mask-GAAM and SG-LKF active.

Among the evaluated nonzero blend weights, $w = 0.4$ gives the highest AMOTA. Performance decreases as the correction becomes stronger beyond this setting, with AMOTA dropping to 77.404 at $w = 0.6$ and 77.060 at $w = 0.8$. This behavior is consistent with using the classifier output as a limited correction to the original detection score rather than allowing it to dominate the score.

Table 8: Detection-quality blend-weight sweep, evaluated with Mask-GAAM and SG-LKF active before GSI.

Blend weight	AMOTA (%)↑
$w = 0.2$	77.486
$w = 0.4$ (adopted)	**77.494**
$w = 0.6$	77.404
$w = 0.8$	77.060

Feature-group ablation. Table 9 reports a leave-one-group-out analysis of the six feature groups in the adopted 26-dimensional classifier. The analysis uses the same 105-scene held-out subset of the training split described in Appendix H. The complete 26-dimensional feature set obtains a held-out AUC of 91.50.

Table 9: Held-out AUC decrease when each feature group is removed individually from the adopted 26-dimensional detection-quality classifier. The complete feature set obtains an AUC of 91.50.

Removed feature group	AUC drop (pp)
(1) Score/calibration	**2.076**
(4) Local duplicate statistics	0.387
(6) Class one-hot	0.253
(5) Frame-level rank context	0.169
(3) Ego-relative geometry	0.116
(2) Box geometry vs. class prior	0.039

Removing the score/calibration group produces the largest decrease in held-out AUC, indicating that the original detector confidence remains the dominant input to the quality classifier. Removing each of the other contextual feature groups also decreases AUC, although by smaller amounts. Together with the score-only comparison in Table 4, these results show that the classifier retains the original score as its primary signal while obtaining additional predictive information from the contextual features.

Trailer-class analysis. The trailer class shows the largest per-class AMOTA increase in the complete system, from 54.56 to 56.75, corresponding to $+2.19$ pp. In the additional module-removal analysis, the detection-quality filter is associated with a $+0.32$ pp difference for trailer. Because the trailer threshold in the original Score Filter is zero and the correction in Equation (5) can only maintain or reduce a surviving detection's score, this effect cannot result from the filter restoring detections rejected by the original Score Filter. Instead, the result is consistent with score re-weighting changing how retained trailer detections contribute across the evaluation thresholds.

GSI is also associated with a substantial portion of the trailer improvement in our module-removal analysis. Trailer trajectories in the analyzed output have a mean length of 3.14 frames, the shortest among the evaluated classes. In addition, the adopted GSI configuration bridges no gaps at `interval`=2, as reported in Appendix D. These observations are consistent with trajectory smoothing contributing to the trailer improvement, although the effects of individual modules are not additive because they operate on the output of preceding stages.

True-positive accounting. The complete Glean configuration increases the number of true positives from 85,735 to 85,916 relative to the Fusion-Poly baseline. To examine how this count changes with individual components, we additionally remove each module from the complete configuration one at a time. Relative to the corresponding leave-one-out configuration, restoring the detection-quality filter changes the reported TP count by $+70$, SG-LKF by $+22$, Mask-GAAM by $+16$, and GSI by $+34$. These values are not an additive decomposition of the overall TP increase because each measurement is made in the presence of the other modules.

The $+70$ change associated with the detection-quality filter does not imply that the filter generates additional detection candidates. By construction, the correction in Equation (5) is applied only to de-

tections that have already survived the original Score Filter/NMS and does not introduce new detections. Instead, score re-weighting can change which retained detections contribute at the confidence threshold used to report the corresponding tracking counts. We therefore interpret the leave-one-out TP differences as evaluation-level changes produced by the complete tracking and thresholding pipeline rather than as a direct count of detections recovered by each module.

F HYPERPARAMETER SUMMARY

Table 10 summarizes the final settings used for the four Glean components. We distinguish fixed implementation settings from parameters selected through held-out training data or validation-set tracking performance. The sensitivity analyses for α, `interval`, and τ are reported in Appendix D, while the blend-weight analysis for w is provided in Appendix E.

Table 10: Final settings for the four Glean components and the procedure used to determine each setting.

Module	Parameter / setting	Value and selection procedure
Mask-GAAM	SAM2 checkpoint	`facebook/sam2.1-hiera-large` (fixed)
	TRF tolerances	`ftol` $= 10^{-6}$, `gtol` $= 10^{-5}$ (fixed; original GAAM setting)
Det.-Quality Filter	Hidden dimensions	$[64, 32]$ (selected from 4 candidates on the held-out training subset)
	Optimizer	Adam, lr $= 3 \times 10^{-3}$, wd $= 10^{-5}$ (fixed)
	Epochs / batch size	60 / 4096 (fixed)
	Blend weight w	0.4 (validation AMOTA sweep; Table 8)
	Floor f	0.0 (fixed; see Section 3.3)
SG-LKF	α	0.6 (validation AMOTA sweep over $\{0.3, 0.6, 1.0\}$; Table 7)
	v_{ref}	10.0 m/s (fixed normalization constant)
GSI	`interval`	2 (validation AMOTA grid search; Table 6)
	τ	10 (validation AMOTA grid search; Table 6)

G ONLINE LATENCY

We report an indicative timing comparison to characterize the computational cost of the Glean tracking loop relative to the Fusion-Poly baseline under our experimental setup. Detector forward passes are excluded because the same precomputed detector outputs are used by both configurations. SAM2 inference is also excluded because instance masks are precomputed and cached for repeated tracking experiments. In an online deployment, Mask-GAAM would require a SAM2 forward pass on the current frame using the projected 2D detection box as a prompt. GSI is excluded because it is applied after tracking is complete. The reported measurements therefore characterize tracking-loop latency with cached masks rather than end-to-end online latency including SAM2 inference.

Timing is measured over the full high-frequency nuScenes validation sequence, including LiDAR keyframes and intermediate camera-only asynchronous frames. Each run contains 11,888 frames, of which the first 50 are discarded as warm-up, leaving 11,838 measured frames. We perform five runs for each configuration and alternate Fusion-Poly and Glean during measurement to reduce the effect of temporal variation in system load. All runs use the same evaluation setup and precomputed inputs.

The mean per-frame latency is 409.8 ms for both configurations to the reported precision. The difference between the two means is approximately 0.01%, which is substantially smaller than the run-to-run variation observed for either configuration. Under this measurement protocol, we therefore do not observe a reproducible increase in per-frame latency from the three online Glean components.

This timing comparison characterizes the incremental tracking cost after detector outputs and SAM2 masks have been precomputed. It should therefore not be interpreted as end-to-end sensor-to-output latency for a deployment that performs detection or segmentation online.

Table 11: Per-frame tracking-loop timing for the reported Fusion-Poly and Glean runs. Each row reports one full validation-set run after discarding 50 warm-up frames. Detector forward passes, SAM2 inference, and post-hoc GSI are excluded; therefore, the measurements do not represent end-to-end online inference latency.

Method	R1	R2	R3	R4	R5	Mean $\pm$ std
Fusion-Poly†	407.4	404.9	412.4	412.0	412.4	409.8 ± 3.5
Glean (online)	409.7	411.0	415.2	409.6	403.7	409.8 ± 4.1

H Implementation Details

Detectors. Following Fusion-Poly's own configuration, we use CenterPoint (Yin et al., 2021) for 3D detection and Cascade R-CNN (Cai & Vasconcelos, 2018) for 2D detection, using the released pretrained weights. Detector outputs are precomputed offline and shared across all evaluated configurations, thereby controlling for detector variation in the controlled comparisons.

Modules. Mask-GAAM, SG-LKF, and GSI involve no training; adopted values and their selection procedure are collected in Table 10 (Appendix F). SAM2 (Ravi et al., 2025) masks are precomputed offline, prompted with each mix detection's projected 2D box, and cached, so the online loop performs a lookup rather than a forward pass. The detection-quality filter is the only trained component: a two-layer MLP trained on 483,038 pure-3D Score Filter survivors from the 700 training scenes, with 105 scenes held out for architecture selection, feature-group analysis, and calibration.

Hardware and timing. All experiments run on a single NVIDIA RTX A6000 GPU and an AMD EPYC 7313 CPU (32 threads). Training the detection-quality filter takes under one minute on this hardware; at inference, it runs on CPU alongside the rest of the online loop. The online tracking loop (loader and tracker stages; Appendix G) runs entirely on CPU using NumPy; detector forward passes and SAM2 mask inference are precomputed offline and excluded from the timed loop.

I Online Loop Pseudocode

Algorithm 1 summarizes how the three online Glean components are inserted into the Fusion-Poly tracking loop. Mask-GAAM modifies the geometric alignment target for mix detections, the detection-quality filter re-weights surviving pure-3D detections, and SG-LKF modifies the Kalman noise scaling during state estimation. The FACM association cascade itself remains unchanged. GSI is omitted because it is applied only after online tracking is complete.

Algorithm 1 Fusion-Poly online loop at frame t with the three online Glean components

Require: 2D detections D_t^{2D}; 3D detections D_t^{3D} at synchronized frames; active tracks $\mathcal{T}$ with predicted states $\{S_t\}$
Ensure: Updated tracks $\mathcal{T}$

if frame t is synchronized **then**
  classify detections as MIX, PURE-3D, or PURE-2D using the original Fusion-Poly preprocessing
  for $d \in$ MIX detections **do**
    retrieve cached SAM2 mask M_d
    align d using M_d as the GAAM image-space target ▷ **Mask-GAAM**
  end for
  $\mathcal{D}_t \leftarrow$ detections surviving the original Score Filter/NMS
  for $d \in \mathcal{D}_t \cap$ PURE-3D **do**
    $p(d) \leftarrow \mathrm{MLP}_\theta(\phi(d))$
    $\mathrm{score}(d) \leftarrow \mathrm{score}(d)\ \max(1 - w(1 - p(d)),\ f)$ ▷ **Detection-Quality**
  end for
else
  $\mathcal{D}_t \leftarrow$ detections surviving the original asynchronous-frame preprocessing
end if
$\mathcal{M}_t \leftarrow \mathrm{FACM}(\mathcal{D}_t, \{S_t\})$ ▷ unchanged Mix → Pure-3D → Pure-2D cascade
for $(\mathrm{track}, d) \in \mathcal{M}_t$ **do**
  if frame t is synchronized **then**
    $s_t \leftarrow \max\Big(1 + \alpha \frac{v_{\mathrm{ego},t}}{v_{\mathrm{ref}}},\ 1\Big)$ ▷ **SG-LKF**
  else
    $s_t \leftarrow 1$
  end if
  apply the original Kalman filtering procedure using $Q_t = s_t Q$ and $R_t = s_t R$
end for
update the original track lifecycle for unmatched tracks and detections
return $\mathcal{T}$

J LIMITATIONS

Three scope limitations are relevant to the reported results. First, GSI is applied after online tracking is complete, so the complete Glean configuration is not a strictly online tracker. The online latency analysis therefore covers only Mask-GAAM, the detection-quality filter, and SG-LKF, while GSI is excluded from the timed loop.

Second, our evaluation is limited to the nuScenes validation set. Although the controlled experiments provide a consistent comparison with the reproduced Fusion-Poly baseline, evaluation on the nuScenes test set and additional 3D MOT benchmarks would be needed to assess how well the observed improvements generalize beyond the current setting.

Third, the latency analysis measures the incremental cost of the online tracking loop after detector outputs and SAM2 masks have been precomputed. Although the repeated measurements do not show a reproducible latency increase from the three online Glean components, this comparison does not include detector inference, online SAM2 inference, or post-hoc GSI. It should therefore not be interpreted as an end-to-end real-time deployment measurement.

REFERENCES

Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468. IEEE, 2016.

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 au-

tonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2018.

Guanhua Ding, Yuxuan Xia, Runwei Guan, Qinchen Wu, Tao Huang, Weiping Ding, Jinping Sun, and Guoqiang Mao. OptiPMB: Enhancing 3D multi-object tracking with optimized poisson multi-bernoulli filtering. *IEEE Transactions on Intelligent Transportation Systems*, 26(12): 22312–22328, 2025. doi: 10.1109/TITS.2025.3619674.

Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. StrongSORT: Make DeepSORT great again. *IEEE Transactions on Multimedia*, 25:8725–8737, 2023. doi: 10.1109/TMM.2023.3240881.

Kemiao Huang, Yinqi Chen, Meiying Zhang, and Qi Hao. Bitrack: Bidirectional offline 3d multi-object tracking using camera-lidar data. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Aleksandr Kim, Aljoša Ošep, and Laura Leal-Taixé. Eagermot: 3d multi-object tracking via sensor fusion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11315–11321. IEEE, 2021.

M. Y. Lee, C. D. W. Lee, J. Li, and M. H. Ang Jr. Dino-mot: 3d multi-object tracking with visual foundation model for pedestrian re-identification using visual memory mechanism. *IEEE Robotics and Automation Letters*, 2024.

Xiaoyu Li, Tao Xie, Dedong Liu, Jinghan Gao, Kun Dai, Zhiqiang Jiang, Lijun Zhao, and Ke Wang. Poly-MOT: A polyhedral framework for 3D multi-object tracking. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9391–9398. IEEE, 2023.

Xiaoyu Li, Dedong Liu, Yitao Wu, Xian Wu, Lijun Zhao, and Jinghan Gao. Fast-Poly: A fast polyhedral algorithm for 3D multi-object tracking. *IEEE Robotics and Automation Letters*, 9(11): 10519–10526, 2024. doi: 10.1109/LRA.2024.3475882.

Xiaoyu Li, Yitao Wu, Xian Wu, Hongyu Zhuo, Lijun Zhao, and Lining Sun. Offline-poly: A polyhedral framework for offline 3d multi-object tracking. *arXiv preprint arXiv:2602.13772*, 2026.

Xiaohong Liu, Xulong Zhao, Gang Liu, Zili Wu, Tao Wang, Lei Meng, and Yuhan Wang. Immmot: A novel 3d multi-object tracking framework with interacting multiple model filter. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 20540–20547. IEEE, 2025.

Mohamed Nagy, Naoufel Werghi, Bilal Hassan, Jorge Dias, and Majid Khonji. Towards accurate state estimation: Motion dynamics kalman filter for 3d multi-object tracking. *arXiv preprint arXiv:2505.07254*, 2025a.

Mohamed Nagy, Naoufel Werghi, Bilal Hassan, Jorge Dias, and Majid Khonji. RobMOT: 3D multi-object tracking enhancement through observational noise and state estimation drift mitigation in LiDAR point clouds. *IEEE Transactions on Intelligent Transportation Systems*, 26(10):16047–16059, 2025b. doi: 10.1109/TITS.2025.3581980.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Timo Osterburg, Stefan Schütte, and Torsten Bertram. Canmot: Class-aware noise modeling for multi-object tracking in autonomous driving. *arXiv preprint arXiv:2606.03590*, 2026.

Su Pang, Daniel Morris, and Hayder Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10386–10393. IEEE, 2020.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press, 1999.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, 2025.

Li Wang, Xinyu Zhang, Wenyuan Qin, Xiaoyu Li, Jinghan Gao, Lei Yang, Zhiwei Li, Jun Li, Lei Zhu, Hong Wang, and Huaping Liu. CAMO-MOT: Combined appearance-motion optimization for 3D multi-object tracking with camera-LiDAR fusion. *IEEE Transactions on Intelligent Transportation Systems*, 24(11):11981–11996, 2023. doi: 10.1109/TITS.2023.3285651.

X. Wang, S. Qi, J. Zhao, H. Zhou, S. Zhang, G. Wang, K. Tu, S. Guo, J. Zhao, J. Li, et al. MCTrack: A unified 3d multi-object tracking framework for autonomous driving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4551–4558. IEEE, 2025.

Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10359–10366. IEEE, 2020.

Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649. IEEE, 2017.

Xian Wu, Yitao Wu, Xiaoyu Li, Zijia Li, Lijun Zhao, and Lining Sun. Fusion-Poly: A polyhedral framework based on spatial-temporal fusion for 3D multi-object tracking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026.

Xinglong Xu, Weihong Ren, Xiai Chen, Huijie Fan, Zhi Han, and Honghai Liu. Exploiting multimodal synergies for enhancing 3D multi-object tracking. *IEEE Robotics and Automation Letters*, 9(10):8643–8650, 2024. doi: 10.1109/LRA.2024.3448132.

Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11784–11793, 2021.

Weizhen Zeng, Jiaqi Fan, Xuelin Tian, Hongqing Chu, and Bingzhao Gao. Fusiontrack: An online 3d multi-object tracking framework based on camera-lidar fusion. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4920–4925, 2024. doi: 10.1109/IROS58592.2024.10802805.

C. Zhang, C. Zhang, Y. Guo, L. Chen, and M. Happold. Motiontrack: End-to-end transformer-based multi-object tracking with lidar-camera fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 151–160, 2023.