STREAMDYNAMIC: STREAMING DYNAMIC SCENE RECONSTRUCTION WITHOUT HISTORICAL GHOSTING

Anonymous authors
Paper under double-blind review

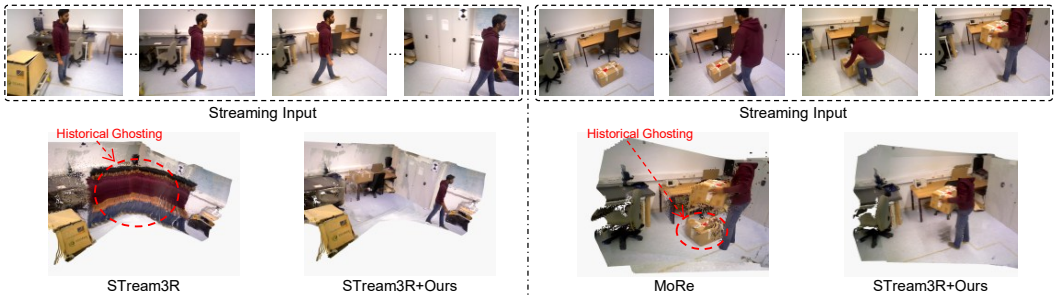

Figure 1: **Streaming dynamic scene reconstruction.** In existing works, a moving object leaves historical ghosting at its past locations (dynamic ghosting, left). Moreover, a previously static object that moves away also leaves historical ghosting at its original location (static ghosting, right). The proposed StreamDynamic achieves online removal of both types of historical ghosting.

ABSTRACT

Streaming 3D reconstruction incrementally predicts the point map of each newly arrived frame from an image stream, and is thus well suited to online applications. However, existing methods suffer from historical ghosting when reconstructing dynamic scenes, as they directly merge the predicted point maps of all frames into an accumulated point cloud. Specifically, the points reconstructed for a moving object at past time instants become dynamic ghosting at the current time instant. Moreover, when a previously static object moves away, its points reconstructed during the static period become static ghosting at its original location. In this work, we take the first step toward feed-forward streaming dynamic scene reconstruction without historical ghosting, and propose StreamDynamic, a novel training-free, plug-and-play framework that identifies moving objects online and removes both their dynamic and static ghosting from the reconstructions of existing streaming backbones. To remove dynamic ghosting, we propose a prior-anchored epipolar-geometry-based motion segmentation method, which estimates the epipolar geometry from the static regions indicated by a coarse motion prior extracted from the backbone predictions. To remove static ghosting accumulated over long histories, we design an online backtracking removal method, which constructs ghosting templates from only a few sparse views and efficiently erases the ghosting within the 3D neighborhoods of the templates. We further introduce a novel ghosting-aware evaluation protocol with ghosting-free ground truth for every time instant and metrics that penalize historical ghosting. Extensive experiments on three dynamic scene datasets show that StreamDynamic substantially reduces the historical ghosting of three representative streaming backbones, consistently improves their reconstruction quality, and outperforms existing methods.

1 INTRODUCTION

Streaming 3D reconstruction recovers the geometry of a scene from an image stream in an online manner by predicting the point map of each newly arrived frame in a unified coordinate system.

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.

Unlike offline feed-forward models, which have to reprocess the entire sequence whenever a new frame arrives, streaming methods process frames incrementally with low latency, and are thus well suited to online applications such as robotics and autonomous driving.

Although existing streaming methods can accurately predict the point map of each frame, they suffer from historical ghosting when reconstructing dynamic scenes, as shown in Fig. 1. Specifically, to obtain a complete reconstruction of the scene, existing works directly merge the predicted point maps of all frames received so far into an accumulated point cloud. However, as a moving object changes its location over time, its points reconstructed at past time instants become ghosting at the current time instant (dynamic ghosting). Moreover, when a previously static object, e.g., a box that has been stationary for a long time, moves away, its points reconstructed during the static period also become ghosting at its original location (static ghosting). These two types of historical ghosting hinder downstream applications, such as robotic navigation (Lim et al., 2023) and world models with point-cloud-based 3D memory (Wu et al., 2026). In this work, we take the first step toward feed-forward streaming dynamic scene reconstruction without historical ghosting.

Historical ghosting in streaming reconstruction poses three main challenges that existing methods fail to address. First, prior streaming methods struggle to accurately segment moving objects while reconstructing the scene. For example, MoRe (Fang et al., 2026) trains an additional motion head to predict motion masks, but its accuracy is limited by the scarcity of reliable motion annotations. Although dedicated motion segmentation models can accurately identify moving objects, incorporating them substantially slows down streaming reconstruction due to their complex and time-consuming pipelines. Second, a greater challenge, which has been overlooked by prior work, lies in removing static ghosting. Specifically, an object may remain static for a long time, e.g., from the beginning of the stream, before it suddenly starts to move. For such an object, naively tracking it backward through all historical frames to segment its ghosting incurs a computational cost that grows with the length of the stream. Third, prior work evaluates streaming reconstruction following the protocol of static scene reconstruction, which cannot effectively assess historical ghosting. Since the ground truth accumulated from all frames itself contains historical ghosting, the ghosting in the prediction is matched rather than penalized.

To address these challenges, we propose StreamDynamic, a novel training-free, plug-and-play framework that identifies moving objects online and removes both their dynamic and static ghosting from the reconstructions of existing streaming backbones. We first propose a prior-anchored epipolar-geometry-based motion segmentation method to remove dynamic ghosting. It anchors the estimation of epipolar geometry to static regions with a coarse motion prior extracted online from the backbone predictions, and then obtains motion masks based on the epipolar distance. Moreover, we design a novel online backtracking removal method based on sparse-view ghosting templates to remove static ghosting. It first identifies newly moving objects, then constructs templates of their static ghosting from a limited number of views selected by the proposed Geometry-Verified Sparse View Selection algorithm regardless of the length of the stream, and finally erases the ghosting in the 3D neighborhood of the templates. Furthermore, we introduce a novel ghosting-aware evaluation protocol with ghosting-free ground truth for every time instant and metrics that penalize historical ghosting. Extensive experiments show that StreamDynamic reduces the ghosting-aware Chamfer Distance of three representative backbones by up to 50% and outperforms existing methods, while segmenting moving objects 6–9$\times$ faster than a dedicated model at comparable quality. Our contributions are summarized as follows:

- We take the first step toward feed-forward streaming dynamic scene reconstruction without historical ghosting, and propose StreamDynamic, a novel training-free, plug-and-play framework that effectively removes both dynamic and static ghosting and generalizes to several backbones.
- We propose a novel online backtracking removal method that efficiently erases static ghosting accumulated over long histories via a ghosting template constructed from only a few sparse views. We further propose a prior-anchored epipolar-geometry-based motion segmentation method for online removal of dynamic ghosting.
- We propose a novel ghosting-aware evaluation protocol for streaming reconstruction of dynamic scenes, including ghosting-free ground truth for every time instant and metrics that penalize historical ghosting. Extensive experiments show the effectiveness and superiority of our work.

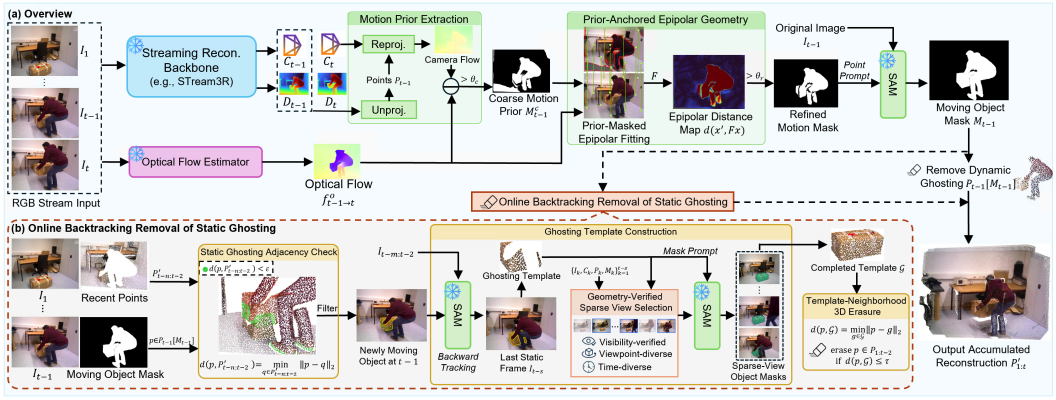

Figure 2: (a) Overview of the proposed StreamDynamic. The epipolar geometry is estimated from the static regions indicated by a coarse motion prior extracted from the backbone predictions, and the pixels with large epipolar distances form a refined motion mask, which SAM completes into the moving object mask M_{t-1}. The points $P_{t-1}[M_{t-1}]$ are removed as dynamic ghosting, while newly moving objects further undergo online backtracking removal of static ghosting. (b) Static Ghosting Adjacency Check identifies newly moving objects, and the templates of their static ghosting are constructed via Geometry-Verified Sparse View Selection for Template-Neighborhood 3D Erasure.

2 Related Works

Feed-forward Reconstruction. Unlike traditional methods such as Structure-from-Motion (Schonberger & Frahm, 2016) which rely on iterative optimization, feed-forward reconstruction methods directly regress 3D geometry from images with neural networks trained on large-scale 3D data. Representative feed-forward methods are further reviewed in Appendix A.

Streaming 3D Reconstruction. Streaming reconstruction methods (Li et al., 2026b; Ding et al., 2026; Cheng et al., 2026b; Liu et al., 2026) take an image stream as input and incrementally predict the point map of each newly arrived frame, rather than reprocessing the entire sequence for every new frame as offline feed-forward methods do. CUT3R (Wang et al., 2025b) adopts a recurrent architecture that updates a persistent state with each new frame and predicts the point map of this frame by reading from the state. Based on CUT3R, TTT3R (Chen et al., 2026b) formulates the state update as test-time training with a confidence-guided learning rate to alleviate forgetting on long sequences. Another line of work (Cheng et al., 2026a; Wang et al., 2026b) employs causal transformers with KV caches. For example, the concurrent works STream3R (LAN et al., 2026) and StreamVGGT (Zhuo et al., 2026) process each new frame with causal attention over the cached keys and values of historical frames. DROID-W (Li et al., 2026a), a representative dynamic SLAM (Zhong et al., 2026) method, estimates per-pixel uncertainty from multi-view feature inconsistency to down-weight moving regions in bundle adjustment. MoRe (Fang et al., 2026) leverages ground-truth motion masks during training to guide the attention of its camera token toward static regions and to supervise an additional motion head. Recently, LingBot-Map (Chen et al., 2026a) maintains an anchor context, a window of recent frames, and a compact trajectory memory via geometric context attention for efficient reconstruction of long sequences. Although these methods achieve accurate per-frame reconstruction, none of them explicitly addresses the historical ghosting in their accumulated point clouds when reconstructing dynamic scenes.

Moving Object Segmentation. Motion segmentation aims to separate moving objects in a video from the static background. Representative methods are further reviewed in Appendix A.

3 Method

Given an image stream $I_{1:t}$, streaming reconstruction predicts the point map of each newly arrived frame I_t and merges it into an accumulated point cloud. In dynamic scenes, our goal is to output, at each time instant t, an accumulated point cloud $P'_{1:t}$ without historical ghosting. The overview of the

proposed training-free, plug-and-play framework is illustrated in Fig. 2 (a). Applied to a streaming reconstruction backbone, the framework removes dynamic ghosting via prior-anchored epipolar-geometry-based motion segmentation (Sec. 3.1) and static ghosting via online backtracking based on sparse-view ghosting templates (Sec. 3.2). We further introduce a ghosting-aware evaluation protocol for streaming reconstruction of dynamic scenes in Sec. 3.3.

3.1 Online Removal of Dynamic Ghosting

Preliminaries: Epipolar Geometry. Epipolar geometry is the intrinsic projective geometry between two views, algebraically represented by the fundamental matrix $F \in \mathbb{R}^{3\times3}$ (Hartley et al., 2003). The projections x and x' of a static 3D point in the first and second views, both in homogeneous coordinates $(u, v, 1)^\top$, must satisfy the epipolar constraint $x'^\top Fx = 0$, i.e., x' lies on the epipolar line Fx of x. Points on moving objects, in contrast, generally violate this constraint and can thus be identified by the epipolar error, e.g., the distance from x' to the epipolar line, $d(x', Fx) = \frac{|x'^\top Fx|}{\sqrt{(Fx)_1^2+(Fx)_2^2}}$, where $(\cdot)_i$ denotes the i-th entry. In practice, F can be estimated solely from static correspondences between the two views. In dynamic scenes, however, the key challenge is to accurately estimate F, since only static correspondences are guaranteed to satisfy the epipolar constraint. Previous epipolar-geometry-based motion segmentation methods (Goli et al., 2025; Wang et al., 2025c) estimate F from correspondences that often include numerous dynamic ones violating the epipolar constraint, which leads to inaccurate estimates. To remove dynamic ghosting, we propose a prior-anchored epipolar-geometry-based motion segmentation method, estimating F from the static regions indicated by a coarse motion prior extracted from the backbone predictions.

Motion Prior Extraction. We extract the coarse motion prior by comparing two flows between adjacent frames. With the depth and camera predicted by the backbone, each pixel can be unprojected into 3D and reprojected into the adjacent frame, yielding the camera flow, i.e., the displacement the pixel would undergo if its 3D point were static. The optical flow, in contrast, captures the observed motion of the pixel. Ideally, the two flows agree in static regions but deviate on moving objects, and their discrepancy can thus identify dynamic pixels. However, errors in the predicted depth and camera also cause the camera flow to deviate from the optical flow, so that some static pixels are misclassified as dynamic and some dynamic pixels as static. We show in Appendix D that, if the errors of the camera flow and the optical flow are identically distributed for static and dynamic pixels, a dynamic pixel is less likely to be misclassified as static than a static pixel as dynamic. Motivated by this, we use the flow discrepancy as a coarse motion prior to exclude dynamic pixels with high recall, so that F can be accurately estimated.

Formally, for each incoming frame I_t, the backbone predicts its camera C_t and depth map D_t, from which the point map P_t is unprojected and accumulated into the reconstruction $P_{1:t}$. To obtain the coarse motion prior at time $t-1$, we compute the camera flow $f^c_{t-1\to t}$ by reprojecting P_{t-1} into I_t with C_t, and the optical flow $f^o_{t-1\to t}$ with an off-the-shelf estimator (Teed & Deng, 2020). The coarse motion prior is then obtained as $M^c_{t-1} = \left\{x \,\middle|\, \left\|f^o_{t-1\to t}(x) - f^c_{t-1\to t}(x)\right\|_2 > \theta_c\right\}$, where the threshold θ_c is set small so that M^c_{t-1} covers moving objects as fully as possible, even though some static regions are also covered.

Prior-Anchored Epipolar Geometry. We anchor the estimation of F to the static regions indicated by the coarse motion prior, i.e., the regions outside M^c_{t-1}. Specifically, we densely sample pixels x in these regions at a fixed interval and pair each with its correspondence $x' = x + f^o_{t-1\to t}(x)$ in I_t, yielding abundant correspondences for an accurate and robust estimate. If these pixels are insufficient, we supplement them with the pixels inside M^c_{t-1} that have the smallest flow discrepancies and are spread over the image. Since the coarse motion prior may still miss some dynamic pixels, we estimate F with RANdom SAmple Consensus (RANSAC) (Fischler & Bolles, 1981), which repeatedly fits a candidate F from a minimal random subset of the correspondences and keeps the best-supported candidate, and is thus robust to the remaining outliers. Given F, the refined motion mask is obtained with the epipolar distance mentioned before as $M^r_{t-1} = \left\{x \,\middle|\, d(x', Fx) > \theta_r\right\}$, where the threshold θ_r balances false positives and false negatives. Since an object is often only partially in motion, we prompt SAM (Carion et al., 2026) with points evenly sampled within the refined motion mask to complete it into the moving object mask M_{t-1}, so that the dynamic ghosting of the entire object is removed.

We remove points from the reconstruction by assigning them expiration times: for each frame k, we maintain a per-pixel expiration map E'_k, where $E'_k[x]$, initialized to ∞, denotes the time instant from which the point $P_k[x]$ is removed, so that the output accumulated reconstruction at time t is $P'_{1:t} = \{\, P_k[x] : k \le t,\ E'_k[x] > t \,\}$. Once M_{t-1} is obtained, we set $E'_{t-1}[x] = t$ for $x \in M_{t-1}$, i.e., the points $P_{t-1}[M_{t-1}]$ are removed as dynamic ghosting from frame t onward, since the moving objects will have moved to new positions by then. Moreover, the historical static ghosting of a newly moving object, i.e., a previously static object that has just started to move, is further removed by the online backtracking removal in Sec. 3.2.

3.2 Online Backtracking Removal of Static Ghosting

Static Ghosting Adjacency Check. Historical ghosting is also left by objects that suddenly start to move after being static. For example, after a box that has remained static for a long time is moved, all its points accumulated during this period persist as static ghosting. To remove such static ghosting by backtracking, we first design the static ghosting adjacency check to detect newly moving objects.

Our design builds on the temporal continuity of object motion in the image stream: between adjacent frames, an object can only move a short distance and thus stays close to its previous positions. Hence, the points of a newly moving object in the current frame lie close to its static ghosting in the recent frames, and we detect newly moving objects by checking this adjacency. Formally, let $P'_{t-n:t-2}$ denote the points of the recent frames $t-n$ to $t-2$ after ghosting removal, where n is a small hyperparameter. For each point $p \in P_{t-1}[M_{t-1}]$ of the moving objects at frame $t-1$, we compute its distance to $P'_{t-n:t-2}$ as $d(p, P'_{t-n:t-2}) = \min_{q \in P'_{t-n:t-2}} \|p - q\|_2$. The pixels whose points satisfy $d(p, P'_{t-n:t-2}) < \varepsilon$ form the adjacent region, where the tolerance ε scales with the depth of p to correspond to a fixed number of pixels in the image. However, the adjacency condition also holds at the contact between a moving object and the static scene, e.g., the sole of a foot on the floor. Since such contact regions are typically thin, we filter out the thin connected components of the adjacent region, i.e., those with a small inscribed radius, which yields the newly moving region. As before, we then prompt SAM with points evenly sampled within the newly moving region to complete it into the mask of the newly moving object for the subsequent backtracking removal. When multiple newly moving objects are detected, each of them undergoes the backtracking removal separately.

Ghosting Template Construction. Given the newly moving object, a straightforward way to remove its static ghosting is to track the object backward through all the preceding frames with SAM and remove its points in each frame. However, this approach faces a severe efficiency challenge: the object may have remained static for a long period, e.g., since the beginning of the stream, in which case SAM has to track it autoregressively over up to t frames, a cost that grows with the history length. To address this challenge, we propose an online backtracking removal method based on a sparse-view ghosting template.

We first track the newly moving object backward with SAM over only a short window of preceding frames $I_{t-m:t-2}$, where m is a small hyperparameter, to locate the last static frame I_{t-s} ($s \le m$) before the object starts to move. The object is regarded as static once the displacement of its 3D points between adjacent frames stays below a small fraction of its size for several consecutive frames. We then construct the initial ghosting template $\mathcal{G}_0$ from the 3D points of the object located by SAM in I_{t-s}. Owing to occlusion and the limited viewpoint of a single frame, $\mathcal{G}_0$ covers only part of the static ghosting of the object. To make the template sufficiently cover the ghosting of the object in all the historical frames $I_{1:t-s}$, we design the Geometry-Verified Sparse View Selection algorithm, which selects a bounded number of sparse views from the history through lightweight geometric verification, so that the template can be completed at a cost independent of the history length.

As shown in Alg. 1 in the appendix, the algorithm selects at most N views in three stages. It first verifies the visibility of the ghosting template in each historical frame by projecting $\mathcal{G}_0$ into the frame with its predicted camera. A template point is counted as observed if it projects inside the image but outside the moving object mask M_k, and the 3D point observed at the projected pixel coincides with it within the tolerance τ; the frame becomes a candidate only if the fraction of observed template points reaches ρ. Although this does not strictly guarantee the presence of the object, it excludes the frames in which the template location is heavily occluded or out of view. Starting from the last

static frame, the algorithm then fills at most half of the budget by the viewpoint-diverse criterion. It iteratively selects the candidate whose viewing direction ω_k, i.e., the unit vector from the camera center to the template centroid, deviates most from those of the selected views, so that the selected views reveal different surfaces of the object. This selection stops once no candidate deviates from the selected views by more than θ_v, e.g., when the camera barely moves over the history. The remaining budget is finally filled by the time-diverse criterion, which iteratively selects the candidate farthest in time from the selected views, since temporally distant frames are more likely to observe the object in different states, e.g., with different parts occluded. Consequently, the time-diverse selection receives at least half of the budget, and most of it when the viewpoint varies little.

We then segment the object in each selected view with SAM: the object mask in I_{t-s} is propagated to the view as a mask prompt to localize the object, and the localized mask is then completed by prompting SAM with points evenly sampled within it. As the selected views are independent of each other, their segmentation can run in parallel with only a few inference passes of SAM, instead of up to t autoregressive steps of the naive backward tracking. The 3D points of the segmented object in each view are merged into $\mathcal{G}_0$, yielding the completed ghosting template $\mathcal{G}$, which covers the static ghosting of the object observed from different viewpoints and at different times.

Template-Neighborhood 3D Erasure. Given the completed ghosting template $\mathcal{G}$, we erase the static ghosting directly in 3D space: since the object stayed around the template location throughout its static period, its points reconstructed during this period lie close to the template and can be erased by a geometric neighborhood check against the template. Specifically, for each historical frame $k \leq t-2$ in which $\mathcal{G}$ is verified as visible, similarly to the visibility verification in Alg. 1, we erase the points $p \in P_k$ in the neighborhood of the template, i.e., those with $d(p, \mathcal{G}) = \min_{g \in \mathcal{G}} \|p - g\|_2 \leq \tau$, where τ is the same tolerance as in the verification. The verification skips the frames in which the object was absent, e.g., before it was placed there, so that the non-ghosting points near the template location are not erased. For the frames after the last static frame, in which the object is already moving, we directly erase its points within the masks from the backward tracking. In both cases, erasing a point $P_k[x]$ at time t sets $E'_k[x] \leftarrow \min(E'_k[x],\, t)$, so that the static ghosting is excluded from the reconstruction from time t onward.

3.3 Ghosting-Aware Evaluation

Prior work evaluates streaming point cloud reconstruction following the protocol of static scene reconstruction (Wang et al., 2025b; Wang & Agapito, 2025; Chen et al., 2026a): the point maps predicted for all frames are merged into one accumulated point cloud, which is aligned to its counterpart accumulated from the ground-truth (GT) depth and camera poses, and the accuracy, completeness, and normal consistency are then computed between the two. This protocol, however, suffers from two problems. First, the accumulated GT itself contains historical ghosting, so that the historical ghosting in the predicted point cloud is matched by that in the GT rather than penalized, and a reconstruction with severe historical ghosting still performs well under these metrics. Second, merging the predictions of all frames amounts to evaluating only the accumulated point cloud at the final time instant. In a dynamic scene, however, the expected accumulated point cloud differs from one time instant to another, as the points reconstructed for an object are valid before it moves away but become historical ghosting afterward. To address these problems, we first propose a new approach to constructing the ghosting-free GT for every time instant, and then a set of ghosting-aware metrics.

Streaming Ghosting-Free Ground Truth. As the expected accumulated point cloud of a dynamic scene varies over time, we construct it at every time instant as the GT for that instant. The GT follows the principle that, if an object is moving at the current instant, all its 3D points at the current and previous instants should expire at the next instant, since the object will have moved to a new position by then. Formally, let P^*_k denote the point map of frame k unprojected from its GT depth and camera pose. Similar to E'_k in Sec. 3.1, we record the expiration time of each point in a per-pixel GT expiration map E_k, where $E_k[x]$, initialized to ∞, denotes the time instant from which the point $P^*_k[x]$ is excluded from the GT. Following this principle, the points inside the annotated dynamic mask of each frame j expire at the next instant, i.e., $E_j[x] = j + 1$ for every pixel x inside the mask. We further track the moving objects of each frame j backward with SAM to obtain their masks in the previous frames $k < j$, and the points inside these masks expire at $j + 1$ as well, i.e., $E_k[x] \leftarrow \min(E_k[x],\, j+1)$ for every pixel x inside the tracked mask of frame k. The ghosting-free

Table 1: **Quantitative comparison.** The accuracy $\mathrm{Acc}_{\mathrm{sd}}$, completeness $\mathrm{Comp}_{\mathrm{sd}}$, Chamfer Distance $\mathrm{CD}_{\mathrm{sd}}$, and normal consistency $\mathrm{NC}_{\mathrm{sd}}$ are our newly proposed metrics for streaming dynamic scene reconstruction. Since a method may favor $\mathrm{Acc}_{\mathrm{sd}}$ over $\mathrm{Comp}_{\mathrm{sd}}$ or vice versa, their average $\mathrm{CD}_{\mathrm{sd}}$ measures the overall reconstruction quality.

Method	Bonn				TUM-Dynamic				DyCheck				Speed (FPS)↑	GPU Mem.↓
	$\mathrm{Acc}_{\mathrm{sd}}\downarrow$	$\mathrm{Comp}_{\mathrm{sd}}\downarrow$	$\mathrm{CD}_{\mathrm{sd}}\downarrow$	$\mathrm{NC}_{\mathrm{sd}}\uparrow$	$\mathrm{Acc}_{\mathrm{sd}}\downarrow$	$\mathrm{Comp}_{\mathrm{sd}}\downarrow$	$\mathrm{CD}_{\mathrm{sd}}\downarrow$	$\mathrm{NC}_{\mathrm{sd}}\uparrow$	$\mathrm{Acc}_{\mathrm{sd}}\downarrow$	$\mathrm{Comp}_{\mathrm{sd}}\downarrow$	$\mathrm{CD}_{\mathrm{sd}}\downarrow$	$\mathrm{NC}_{\mathrm{sd}}\uparrow$		
DROID-W	0.11	0.09	0.097	0.544	0.31	**0.08**	0.193	0.504	0.04	0.06	0.048	0.655	3.96	13.9
MoRe	0.07	0.06	0.065	0.588	0.20	0.18	0.190	0.527	0.03	0.07	0.053	0.681	8.82	17.8
D4RT	0.11	0.17	0.139	0.595	0.22	0.25	0.236	0.546	0.04	0.08	0.063	0.687	0.02	9.2
StreamVGGT	0.14	0.05	0.099	0.594	0.29	0.20	0.243	0.546	0.04	0.06	0.049	0.717	4.99	19.7
TTT3R	0.13	0.06	0.091	0.582	0.28	0.21	0.242	0.501	0.08	0.05	0.063	0.678	24.48	6.5
Lingbot-map	0.11	0.06	0.083	0.586	0.22	0.14	0.178	0.546	0.05	0.04	0.043	0.730	12.39	14.4
STream3R	0.12	**0.04**	0.078	0.595	0.22	0.18	0.198	0.529	0.03	**0.03**	0.029	0.736	8.31	24.1
TTT3R + Ours	0.03	0.06	0.046	0.587	0.20	0.22	0.209	0.502	0.03	0.06	0.044	0.679	6.05	16.8
Lingbot-map + Ours	**0.03**	0.06	0.044	0.591	**0.15**	0.14	**0.148**	**0.548**	0.02	0.05	0.035	0.739	4.98	25.0
STream3R + Ours	0.03	0.05	**0.039**	**0.597**	0.18	0.19	0.182	0.530	**0.02**	0.04	**0.026**	**0.747**	4.30	32.1

accumulated GT at time instant t is then the set of points of all frames $k \leq t$ that have not yet expired, i.e., $G_t = \{\, P_k^*[x] : k \leq t,\ E_k[x] > t \,\}$.

Ghosting-Aware Metrics. As before, the predicted point cloud of each evaluated method is accumulated from its point maps P_k according to its expiration maps E'_k, i.e., $P'_{1:t} = \{\, P_k[x] : k \leq t,\ E'_k[x] > t \,\}$, with $E'_k \equiv \infty$ for methods without ghosting removal. Besides defining G_t, the GT expiration maps E_k also indicate which predicted points should have expired by t. The predicted points that should have expired but are retained form the residual ghosting $\mathcal{R}_t = \{\, P_k[x] : k \leq t,\ E'_k[x] > t,\ E_k[x] \leq t \,\}$, and the others are the non-ghosting points $\overline{\mathcal{R}}_t = P'_{1:t} \setminus \mathcal{R}_t$.

In the presence of residual ghosting, the accuracy, completeness, and normal consistency used by prior work cannot be directly applied, since they treat residual ghosting points in the same way as non-ghosting points. Consider first the accuracy, which measures the mean or median distance from each predicted point to its nearest GT point. Residual ghosting points lying closer to the GT than the non-ghosting points reduce the overall mean or median distance, so that retaining such ghosting can yield a better accuracy than removing it. To penalize residual ghosting, the accuracy should measure not only the distance from the non-ghosting points to the GT, but also the amount of residual ghosting and its distance to the GT. We therefore propose a ghosting-aware accuracy metric, averaged over all time instants of the stream, as $\mathrm{Acc}_{\mathrm{sd}} = \frac{1}{2T}\sum_{t=1}^{T}\Big(\frac{1}{|\overline{\mathcal{R}}_t|}\sum_{p\in\overline{\mathcal{R}}_t} d(p, G_t) + \frac{1}{|\overline{G}_t|}\sum_{p\in\mathcal{R}_t} d(p, G_t)\Big)$, where T is the number of frames, $d(p, G_t)$ is the distance from p to its nearest point in G_t as before, and $\overline{G}_t = \{\, P_k^*[x] : k \leq t,\ E_k[x] \leq t \,\}$ is the set of points in P_k^*, $k \leq t$, that have expired by t according to the GT expiration maps E_k, with the second term set to 0 when $\overline{G}_t = \emptyset$. Instead of averaging over the residual ghosting, the second term normalizes their total distance to the GT by $|\overline{G}_t|$, which is independent of the prediction. This term thus grows with both the amount of residual ghosting and its distance to the GT, and vanishes only when all ghosting is either removed or lies at zero distance from the GT. The two terms are averaged as they are equally important for accuracy.

Similarly, we propose a ghosting-aware completeness metric $\mathrm{Comp}_{\mathrm{sd}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|G_t|}\sum_{q\in G_t} d(q, \overline{\mathcal{R}}_t)$, which measures the distance from each GT point to the nearest non-ghosting point in the prediction. In contrast to prior work, we exclude the residual ghosting points from the nearest-neighbor candidates, since they would otherwise make the prediction appear more complete. Moreover, since residual ghosting points have no true counterparts in the GT, we report the normal consistency only in the direction from the GT to the prediction, i.e., $\mathrm{NC}_{\mathrm{sd}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|G_t|}\sum_{q\in G_t}\big|\mathbf{n}(q)^\top \mathbf{n}(\hat{q})\big|$, where $\mathbf{n}(\cdot)$ denotes the unit normal of a point and $\hat{q}$ the nearest point to q in $P'_{1:t}$. Unlike for $\mathrm{Comp}_{\mathrm{sd}}$, residual ghosting points remain candidates for $\hat{q}$, so that $\mathrm{NC}_{\mathrm{sd}}$ also reflects the impact of residual ghosting on the reconstructed surfaces. As with the metrics of prior work, a method may achieve a high accuracy at the cost of a low completeness by removing numerous points far from the GT. Following common practice, we therefore also report the Chamfer Distance $\mathrm{CD}_{\mathrm{sd}} = (\mathrm{Acc}_{\mathrm{sd}} + \mathrm{Comp}_{\mathrm{sd}})/2$ as the overall quality of the reconstruction.

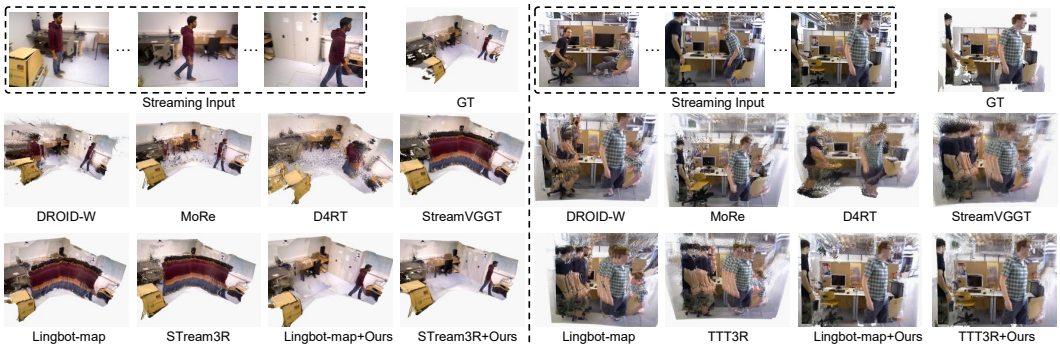

Figure 3: **Qualitative comparison of streaming reconstruction.** By effectively removing historical ghosting, our work substantially improves all three backbones and outperforms the other methods. More qualitative comparisons are shown in Fig. 4 in the appendix.

4 Experiments

4.1 Experimental Setup

Implementation Details. We apply StreamDynamic to three representative streaming reconstruction backbones: the recurrent TTT3R (Chen et al., 2026b), the causal transformer STream3R (LAN et al., 2026) with KV caches, and the recent LingBot-Map (Chen et al., 2026a) with geometric context attention. The hyperparameters of our framework are determined on a validation set constructed from Sintel (Butler et al., 2012) and shared by all backbones and benchmarks without backbone-specific adjustment. Further implementation details, including the hyperparameter settings, are provided in Appendix E.

Comparison Methods. For streaming point cloud reconstruction, we first compare StreamDynamic built upon each backbone with the original backbone. We further compare with other dynamic scene reconstruction methods, including the StreamVGGT (Zhuo et al., 2026), the query-based D4RT (Zhang et al., 2026), MoRe (Fang et al., 2026) with a motion head, and the dynamic SLAM method DROID-W (Li et al., 2026a). To remove historical ghosting, the intermediate motion masks of MoRe and DROID-W are applied to their reconstructions, while D4RT tracks all historical pixels to the current time instant. For moving object segmentation, we compare with the dedicated motion segmentation models GeoMotion (He et al., 2026) and RoMo (Goli et al., 2025), as well as MoRe, DROID-W, and VGGT4D (Hu et al., 2026), which output motion masks along with their reconstructions.

Evaluation Benchmarks. Following prior work, we evaluate streaming point cloud reconstruction on three commonly used dynamic scene datasets: the Bonn RGB-D Dynamic (Palazzolo et al., 2019), TUM-Dynamic (Sturm et al., 2012), and DyCheck (Gao et al., 2022) datasets. To assess historical ghosting, we adopt the proposed ghosting-aware evaluation protocol (Sec. 3.3) instead of that of prior work. We also report the average online speed over all benchmarks in frames per second (FPS), i.e., the number of frames reconstructed and merged into the accumulated point cloud per second, and the per-sequence peak GPU memory (GiB) averaged over all sequences. For moving object segmentation, we use the validation set of DAVIS16 (Perazzi et al., 2016) and report $\mathcal{J}_M$, the mean per-frame intersection over union (IoU) between the predicted and ground-truth motion masks, and $\mathcal{J}_R$, the recall measured as the fraction of frames with an IoU above 0.5.

4.2 Comparisons

Streaming Reconstruction. Tab. 1 and Fig. 3 present the quantitative and qualitative comparisons for streaming dynamic scene reconstruction, respectively. On all benchmarks, StreamDynamic substantially improves $\mathrm{Acc}_{\mathrm{sd}}$ over all three original backbones with $\mathrm{Comp}_{\mathrm{sd}}$ unchanged or slightly worse, and thus achieves a considerably better overall $\mathrm{CD}_{\mathrm{sd}}$, which is roughly halved on Bonn. This indicates that our method effectively removes historical ghosting, as also confirmed by the qualita-

Table 2: **Quantitative ablation results.** "-" denotes that the variant differs from our full framework only in static ghosting removal and thus yields the same motion masks as ours.

Method	$\mathrm{Acc}_{\mathrm{sd}} \downarrow$	$\mathrm{Comp}_{\mathrm{sd}} \downarrow$	$\mathrm{CD}_{\mathrm{sd}} \downarrow$	$\mathrm{NC}_{\mathrm{sd}} \uparrow$	$\mathcal{J}_M \uparrow$	$\mathcal{J}_R \uparrow$	Speed (FPS)↑
Ours. w/o Motion Prior Extraction	0.030	0.066	0.048	0.590	0.62	0.68	5.34
Ours. w/o Prior-Anchored Epipolar.	0.027	0.173	0.100	0.588	0.51	0.51	2.61
Ours. w/o Online Backtracking.	0.052	**0.062**	0.057	0.589	-	-	5.76
Ours. w/o Sparse-View Template	0.026	0.071	0.049	0.589	-	-	1.97
Ours	**0.025**	0.063	**0.044**	**0.591**	**0.79**	**0.87**	5.15

tive results. By retaining all the points predicted for historical frames, the original backbones achieve a slightly better $\mathrm{Comp}_{\mathrm{sd}}$ at the cost of accuracy. Moreover, our method clearly outperforms the other methods, achieving the best $\mathrm{Acc}_{\mathrm{sd}}$ and $\mathrm{CD}_{\mathrm{sd}}$ on all benchmarks. Although DROID-W achieves the best $\mathrm{Comp}_{\mathrm{sd}}$ on TUM-Dynamic, it retains more historical ghosting than our method and thus has the worst $\mathrm{Acc}_{\mathrm{sd}}$ on this benchmark. In terms of speed, D4RT runs at only 0.02 FPS, since it has to track all historical pixels to the current time instant. In contrast, our method achieves clearly better reconstruction quality than DROID-W, MoRe, and StreamVGGT at a comparable speed.

Moving Object Segmentation. The quantitative and qualitative comparisons with prior methods are presented in Tab. 3 and Fig. 5 in the appendix, respectively. Our motion segmentation method achieves quality comparable to that of the dedicated motion segmentation model GeoMotion while running 6–9× faster, making it better suited to online streaming reconstruction. It also outperforms RoMo, VGGT4D, and DROID-W in both quality and speed. Although MoRe runs faster by predicting motion masks with an additional motion head, it falls behind ours in segmentation quality.

4.3 Ablation Study

We conduct ablation studies with LingBot-Map as the backbone on Bonn for streaming reconstruction and speed, and on DAVIS16 for moving object segmentation. The quantitative results are reported in Tab. 2, and the qualitative results of the reconstructed geometry and the motion masks are shown in Fig. 6 and Fig. 7 in the appendix, respectively. The ablation studies of the motion prior extraction, online backtracking removal of static ghosting, and the sparse-view template-based removal are provided in Appendix B.

Prior-Anchored Epipolar Geometry. To validate the prior-anchored epipolar geometry, we design a variant that removes it and directly completes the coarse motion prior into the moving object mask with SAM. As shown by "Ours w/o Prior-Anchored Epipolar Geometry" in Tab. 2 and "w/o Prior-Epipolar" in Fig. 6 and Fig. 7, this variant degrades both the reconstruction quality and the motion segmentation accuracy. In particular, $\mathrm{Comp}_{\mathrm{sd}}$ severely degrades from 0.063 to 0.173, and the speed drops from 5.15 to 2.61 FPS. This is because the coarse motion prior often misclassifies static regions as dynamic, thereby removing a large portion of the static scene and slowing down the reconstruction by unnecessarily triggering the online backtracking removal.

5 Conclusion

We take the first step toward feed-forward streaming dynamic scene reconstruction without historical ghosting, and propose StreamDynamic, a training-free, plug-and-play framework that removes both dynamic and static ghosting. Within this framework, we propose a novel online backtracking removal method for static ghosting, and a prior-anchored epipolar-geometry-based motion segmentation method for the online removal of dynamic ghosting. We further propose a ghosting-aware evaluation protocol for streaming dynamic scene reconstruction. Extensive experiments demonstrate the effectiveness of StreamDynamic and its generalization across different streaming backbones. Limitations are discussed in Appendix F.

References

Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*

Pattern Recognition (CVPR), June 2020.

D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.) (ed.), *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pp. 611–625. Springer-Verlag, October 2012.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Lin-Zhuo Chen, Jian Gao, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, et al. Geometric context transformer for streaming 3d reconstruction. *arXiv preprint arXiv:2604.14141*, 2026a.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. In *International Conference on Learning Representations*, volume 2026, pp. 50694–50718, 2026b.

Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

Chong Cheng, Xianda Chen, Tao Xie, Wei Yin, Weiqiang Ren, Qian Zhang, Xiaoyang Guo, and Hao Wang. Longstream: Long-sequence streaming autoregressive visual geometry. *arXiv preprint arXiv:2602.13172*, 2026a.

Chong Cheng, Peilin Tao, Nanjie Yao, Guanzhi Ding, Xianda Chen, Yuansen Du, Xiaoyang Guo, Wei Yin, Weiqiang Ren, Qian Zhang, et al. Horizonstream: Long-horizon attention for streaming 3d reconstruction. *arXiv preprint arXiv:2605.23889*, 2026b.

Tianye Ding, Yiming Xie, Yiqing Liang, Moitreya Chatterjee, Pedro Miraldo, and Huaizu Jiang. Laser: Layer-wise scale alignment for training-free streaming 4d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36433–36443, 2026.

Juntong Fang, Zequn Chen, Weiqi Zhang, Donglin Di, Xuancheng Zhang, Chengmin Yang, and Yu-Shen Liu. More: Motion-aware feed-forward 4d reconstruction transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28914–28924, June 2026.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35: 33768–33780, 2022.

Lily Goli, Sara Sabour, Mark Matthews, Marcus A Brubaker, Dmitry Lagun, Alec Jacobson, David J Fleet, Saurabh Saxena, and Andrea Tagliasacchi. Romo: Robust motion segmentation improves structure from motion. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6155–6164. IEEE, 2025.

Jisang Han, Honggyu An, Jaewoo Jung, Takuya Narihira, Junyoung Seo, Kazumi Fukuda, Chaehyun Kim, Sunghwan Hong, Yuki Mitsufuji, and Seungryong Kim. Enhancing 3d reconstruction for dynamic scenes. *Advances in Neural Information Processing Systems*, 38:1210–1234, 2026.

Richard Hartley, Andrew Zisserman, et al. *Multiple view geometry in computer vision*. Cambridge university press Cambridge, 2003.

Xiankang He, Peile Lin, Ying Cui, Dongyan Guo, Chunhua Shen, and Xiaoqin Zhang. Geomotion: Rethinking motion segmentation via latent 4d geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28145–28155, June 2026.

Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 414–424, 2026.

Nan Huang, Wenzhao Zheng, Chenfeng Xu, Kurt Keutzer, Shanghang Zhang, Angjoo Kanazawa, and Qianqian Wang. Segment any motion in videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3406–3416. IEEE, 2025.

Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14578–14589, 2026.

Yushi LAN, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Bo Dai, Shuai Yang, Chen Change Loy, and Xingang Pan. STream3r: Scalable sequential 3d reconstruction with causal transformer. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RTTYGeC2Io.

Moyang Li, Zihan Zhu, Marc Pollefeys, and Daniel Barath. DROID-SLAM in the wild. In *Conference on Computer Vision and Pattern Recognition 2026*, 2026a. URL https://openreview.net/forum?id=KBGRXZ4kMx.

Zizun Li, Jianjun Zhou, Yifan Wang, Haoyu Guo, Wenzheng Chang, Yang Zhou, Haoyi Zhu, Junyi Chen, Chunhua Shen, and Tong He. Wint3r: Window-based streaming reconstruction with camera token pool. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=PjviszIZf1.

Hyungtae Lim, Lucas Nunes, Benedikt Mersch, Xieyuanli Chen, Jens Behley, Hyun Myung, and Cyrill Stachniss. Erasor2: Instance-aware robust 3d mapping of the static world in dynamic scenes. In *Robotics: Science and Systems*, 2023.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yirunib8l8.

Xinhang Liu, Pedro Miraldo, Suhas Lohit, Huaizu Jiang, Naoko Sawada, Yu-Wing Tai, Chi-Keung Tang, and Moitreya Chatterjee. Point4cast: Streaming dynamic scene reconstruction and forecasting. In *Conference on Computer Vision and Pattern Recognition 2026*, 2026. URL https://openreview.net/forum?id=lOhjcRX7f7.

Yihang Luo, Shangchen Zhou, Yushi Lan, Xingang Pan, and Chen Change Loy. 4rc: 4d reconstruction via conditional querying anytime and anywhere. *arXiv preprint arXiv:2602.10094*, 2026.

Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7855–7862. IEEE, 2019.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 573–580. IEEE, 2012.

Edgar Sucar, Eldar Insafutdinov, Zihang Lai, and Andrea Vedaldi. V-dpm: 4d video reconstruction with dynamic point maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14502–14511, June 2026.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):376–380, 1991.

Feiran Wang, Junyi Wu, Dawen Cai, Yuan Hong, and Yan Yan. Cognimap3d: Cognitive 3d mapping and rapid retrieval. *arXiv preprint arXiv:2601.08175*, 2026a.

Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pp. 78–89. IEEE, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522. IEEE, 2025b.

Runze Wang, Yuxuan Song, Youcheng Cai, and Ligang Liu. Stac: Plug-and-play spatio-temporal aware cache compression for streaming 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7567–7576, June 2026b.

Shizun Wang, Zhenxiang Jiang, Xingyi Yang, and Xinchao Wang. C4d: 4d made from 3d through dual correspondences. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7570–7580. IEEE, 2025c.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. pi3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026c.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *Advances in Neural Information Processing Systems*, 38:49371–49393, 2026.

Junyu Xie, Tengda Han, Weidi Xie, and Andrew Zisserman. Gmos: Grounding moving object segmentation in 3d space and time. *arXiv preprint arXiv:2605.30352*, 2026.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K Barral, Raia Hadsell, et al. Efficiently reconstructing dynamic scenes one d4rt at a time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7382–7392, 2026.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Forrester Cole, Deqing Sun, Ming-Hsuan Yang, et al. Monst3r: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations*, volume 2025, pp. 82863–82886, 2025.

Jianhao Zheng, Liyuan Zhu, Zihan Zhu, and Iro Armeni. Wildpose: A unified framework for robust pose estimation in the wild. *arXiv preprint arXiv:2605.12774*, 2026.

Xingguang Zhong, Liren Jin, Marija Popovic, Jens Behley, and Cyrill Stachniss. Dynamic visual slam using a general 3d prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21805–21815, 2026.

Kaichen Zhou, Yuhan Wang, Grace Chen, Gaspard Beaudouin, Fangneng Zhan, Paul Liang, and Mengyu Wang. Page-4d: Disentangled pose and geometry estimation for vggt-4d perception. In *International Conference on Learning Representations*, volume 2026, pp. 36401–36414, 2026.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *International Conference on Learning Representations*, volume 2026, pp. 88055–88072, 2026.

APPENDIX

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools to implement methods, assist in the writing of proofs, and in translation. We have not used generative AI tools for generating synthetic datasets or proposing hypotheses, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for editing the paper to improve readability and creating the icons in the method overview figure. We have reviewed all AI-assisted work: AI-generated code was tested by the authors, and AI-edited text and proofs were checked by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Table 3: **Comparison of moving object segmentation.** Our method achieves quality comparable to that of the dedicated motion segmentation model GeoMotion at a substantially faster speed, and outperforms the other methods in quality.

Method	DAVIS16		Speed	GPU
	$\mathcal{J}_M\uparrow$	$\mathcal{J}_R\uparrow$	(FPS)↑	Mem.↓
VGGT4D	0.64	0.81	0.34	10.5
DROID-W	0.43	0.41	4.57	12.5
MoRe	0.54	0.66	**12.05**	16.9
RoMo	0.72	0.78	0.03	4.1
Geo-Motion	**0.83**	**0.92**	0.92	8.0
Lingbot-map + Ours	0.79	0.87	6.72	15.7
STream3R + Ours	0.75	0.83	5.61	20.4
TTT3R + Ours	0.75	0.84	8.61	17.3

A MORE RELATED WORKS

Feed-forward Reconstruction. The early work DUSt3R (Wang et al., 2024) regresses the point maps of an uncalibrated image pair in a common coordinate system, but requires a time-consuming global alignment to handle multiple views. Subsequent multi-view methods (Lin et al., 2026; Wang et al., 2026c; Karhade et al., 2026), e.g., VGGT (Wang et al., 2025a), jointly reconstruct all views in a single forward pass. However, whenever a new frame arrives, they have to reprocess the entire sequence at a cost that grows quadratically with the number of frames, making incremental reconstruction inefficient. Among recent query-based 4D reconstruction methods (Luo et al., 2026; Sucar et al., 2026), D4RT (Zhang et al., 2026) encodes a video into a global scene representation, from which the 3D position of any pixel at any time instant can be queried. It attempts to alleviate ghosting by querying the positions of the pixels from all frames at a specified time instant. Similar to VGGT, however, it has to recompute this representation over the entire sequence with quadratic complexity whenever a new frame arrives, which precludes real-time incremental reconstruction. In this work, we focus on streaming 3D reconstruction, which incrementally processes each newly arrived frame and thus supports real-time online reconstruction.

Moving Object Segmentation. Among dedicated motion segmentation models (Xie et al., 2026; Huang et al., 2025), GeoMotion (He et al., 2026) uses a pretrained feed-forward reconstruction model to extract latent 4D geometric features from the frames within a temporal window, and fuses them with optical flow to decode motion masks. Although these methods segment moving objects accurately, they usually involve complex and time-consuming pipelines, and directly incorporating them into streaming reconstruction would substantially slow it down. Some feed-forward reconstruction methods (Han et al., 2026; Zhou et al., 2026; Wang et al., 2026a; Zheng et al., 2026) also segment moving objects while reconstructing the scene. MonST3R (Zhang et al., 2025) directly obtains motion masks by comparing the camera flow computed from the predicted depth and camera poses with the optical flow, but its accuracy is limited by errors in the camera flow. Some prior works leverage epipolar geometry for motion segmentation, but they struggle to estimate the epipolar geometry accurately. RoMo (Goli et al., 2025) uses epipolar geometry as weak supervision to

train a classifier over the entire video, but it estimates the epipolar geometry from the correspondences over the whole image, which are contaminated by moving objects. C4D (Wang et al., 2025c) trains a dynamic-aware point tracker on synthetic data to exclude dynamic points when estimating the epipolar geometry, which may generalize poorly to real scenes. In contrast, we propose a novel prior-anchored epipolar-geometry-based motion segmentation method, which extracts a coarse motion prior from the predictions of the streaming reconstruction backbone in a training-free manner to anchor the estimation of epipolar geometry to static regions, achieving fast online segmentation of moving objects.

B More Ablation Study

Sparse-View Template-Based Removal. To validate the sparse-view template-based removal, we design a variant that, once a newly moving object is detected, tracks it backward through all the preceding frames with SAM and removes its points in each frame, instead of constructing its ghosting template. As shown by "Ours w/o Sparse-View Template" in Tab. 2, this variant most notably reduces the speed from 5.15 to 1.97 FPS, since the object may have remained static for a long period, in which case SAM has to track it autoregressively over numerous historical frames. It performs slightly worse than ours in the reconstruction metrics, while its qualitative results of "w/o Template" in Fig. 6 are visually comparable to ours. The slight degradation arises because SAM may segment incorrect objects when tracking autoregressively over a long history.

Online Backtracking Removal. To validate the online backtracking removal of static ghosting, we design a variant that removes it and only performs the online removal of dynamic ghosting. As shown by "Ours w/o Online Backtracking" in Tab. 2, although this variant slightly improves $\mathrm{Comp}_{\mathrm{sd}}$, it severely degrades $\mathrm{Acc}_{\mathrm{sd}}$ from 0.025 to 0.052. This is because this variant can neither remove the static ghosting left by previously static objects that suddenly start to move, nor backtrack to remove the ghosting of moving objects occasionally missed by the motion segmentation. Such residual ghosting is also evident in the qualitative results of "w/o Backtrack" in Fig. 6.

Motion Prior Extraction. To validate the coarse motion prior, we design a variant that removes the prior and directly estimates F from correspondences densely sampled over the whole image. As shown by "Ours w/o Motion Prior Extraction" in Tab. 2 and "w/o Motion Prior" in Fig. 6 and Fig. 7, this variant degrades both the reconstruction quality and the motion segmentation accuracy, e.g., $\mathcal{J}_M$ drops from 0.79 to 0.62. This is because numerous correspondences sampled over the whole image lie on moving objects and violate the epipolar constraint, thereby interfering with the estimation of F.

C Geometry-Verified Sparse View Selection

D Analysis of the Coarse Motion Prior

Consider a pixel x of I_{t-1} and let $\bar{f}^c_{t-1\to t}(x)$ denote its true camera flow, i.e., the displacement it would undergo if its 3D point were static. The estimated optical flow is $f^o_{t-1\to t}(x) = \bar{f}^c_{t-1\to t}(x) + w + \delta_o$, where w is the additional displacement caused by the object motion ($w = 0$ for a static pixel) and δ_o is the optical flow estimation error, and the camera flow computed from the predicted depth and camera is $f^c_{t-1\to t}(x) = \bar{f}^c_{t-1\to t}(x) + \delta_c$, where δ_c is the error caused by the depth and camera errors. Hence $f^o_{t-1\to t}(x) - f^c_{t-1\to t}(x) = w - \delta$ with the combined error $\delta = \delta_c - \delta_o$, and the pixel is classified as dynamic if and only if $\|w - \delta\|_2 > \theta_c$. A static pixel is thus misclassified as dynamic if and only if $\|\delta\|_2 > \theta_c$, and a dynamic pixel as static if and only if $\|w - \delta\|_2 \le \theta_c$. Treating w as fixed and δ as random, let $\Pr(\mathrm{FP})$ and $\Pr(\mathrm{FN})$ denote the probabilities of the two events, i.e., the misclassification probabilities of a static pixel and of a dynamic pixel with displacement w, respectively, and let φ denote the angle between δ and w. We assume that (A1) δ follows the same distribution for static and dynamic pixels; (A2) φ is uniformly distributed over $(-\pi, \pi]$ and independent of $\|\delta\|_2$, as the direction of the object motion is arbitrary and generally unrelated to the error; and (A3) $\|w\|_2 > 2\theta_c$, which excludes only the dynamic pixels whose motion-induced displacement between adjacent frames is at most $2\theta_c$, i.e., a few pixels, since θ_c is set small, and

Algorithm 1 Geometry-Verified Sparse View Selection

Input: initial template $\mathcal{G}_0$; historical frames I_k with predicted cameras C_k, point maps P_k, and moving object masks M_k, $k = 1, \ldots, t-s$
Output: indices $\mathcal{S}$ of the selected views
Hyperparameters: tolerance τ, visibility threshold ρ, angle threshold θ_v, budget N

Visibility-verified
for $k = 1, \ldots, t-s$ **do**
  projected pixel $x_g \leftarrow$ projection of $g \in \mathcal{G}_0$ into I_k with C_k
  visibility ratio $\alpha_k \leftarrow$ fraction of $g \in \mathcal{G}_0$ with $x_g \notin M_k$ and $\|g - P_k[x_g]\|_2 \le \tau$
  viewing direction $\omega_k \leftarrow$ unit vector from camera C_k to the centroid of $\mathcal{G}_0$
end for
candidate frame indices $\mathcal{K} \leftarrow \{k : \alpha_k \ge \rho\}$, selected view indices $\mathcal{S} \leftarrow \{t-s\}$
Viewpoint-diverse
while $|\mathcal{S}| < \lfloor N/2 \rfloor$ and $\mathcal{K} \setminus \mathcal{S} \neq \emptyset$ **do**
  $k^* \leftarrow \arg\max_{k \in \mathcal{K} \setminus \mathcal{S}} \min_{j \in \mathcal{S}} \angle(\omega_k, \omega_j)$ ▷ largest viewing-angle gap
  if $\min_{j \in \mathcal{S}} \angle(\omega_{k^*}, \omega_j) \le \theta_v$ **then break**
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{k^*\}$
end while
Time-diverse
while $|\mathcal{S}| < N$ and $\mathcal{K} \setminus \mathcal{S} \neq \emptyset$ **do**
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\arg\max_{k \in \mathcal{K} \setminus \mathcal{S}} \min_{j \in \mathcal{S}} |k - j|\}$ ▷ largest frame gap
end while
return $\mathcal{S}$

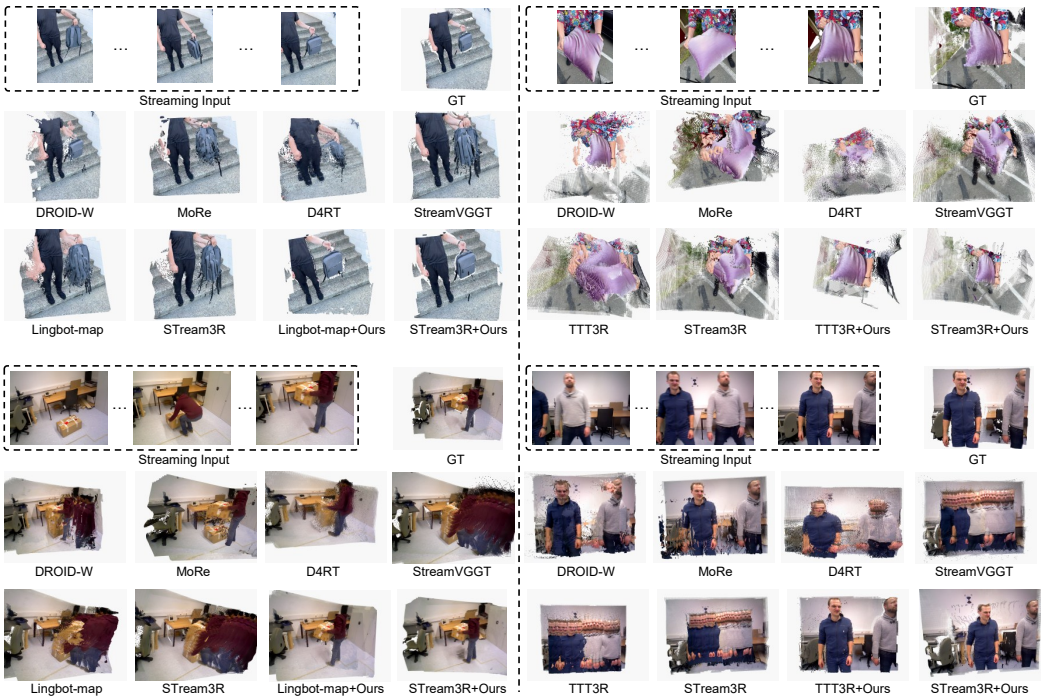

Figure 4: **More qualitative comparisons.**

can be further relaxed by decreasing θ_c. No further assumption is made on the distribution of δ; in particular, its magnitude $\|\delta\|_2$ may follow an arbitrary distribution.

Proposition 1. *Under (A1)–(A3),* $\Pr(\mathrm{FN}) \le \frac{\theta_c}{2\|w\|_2} \Pr(\mathrm{FP})$.

Proof. Suppose a dynamic pixel is misclassified as static, i.e., $\|w - \delta\|_2 \le \theta_c$. By the triangle inequality, $\|w\|_2 \le \|w-\delta\|_2 + \|\delta\|_2$, and hence $\|\delta\|_2 \ge \|w\|_2 - \theta_c$. Moreover, $\|w-\delta\|_2^2 = (\|\delta\|_2 - \|w\|_2 \cos\varphi)^2 + \|w\|_2^2 \sin^2\varphi \ge \|w\|_2^2 \sin^2\varphi$, and $\cos\varphi \le 0$ would give $\|w - \delta\|_2 \ge \|w\|_2 > \theta_c$;

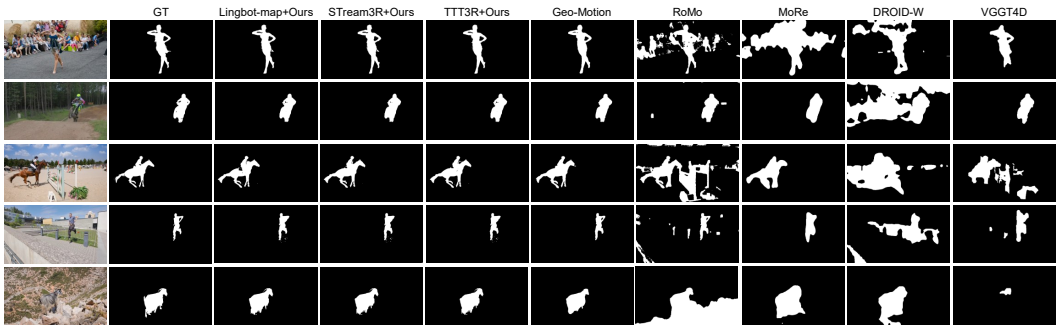

Figure 5: **Qualitative comparison of moving object segmentation.** Our method produces masks comparable to those of the dedicated motion segmentation model GeoMotion and more accurate than those of the other methods, while running substantially faster than GeoMotion (Tab. 3).

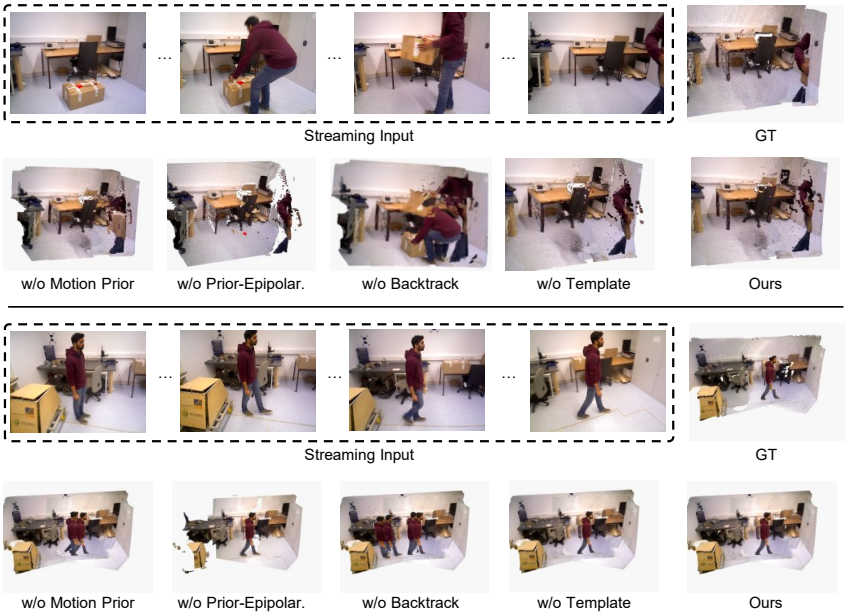

Figure 6: **Qualitative ablation results of streaming reconstruction.** The variant without the sparse-view template-based removal ("w/o Template") yields results comparable to ours but runs considerably slower (1.97 vs. 5.15 FPS).

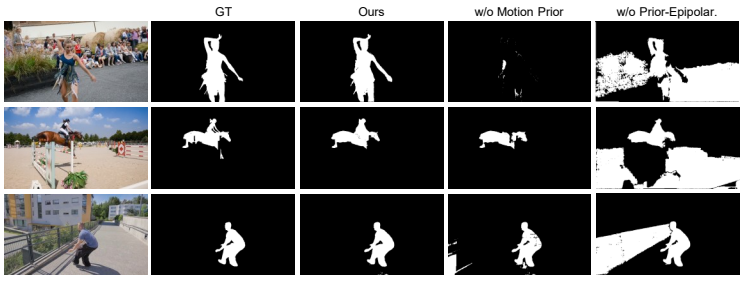

Figure 7: **Qualitative ablation results of moving object segmentation.** Without the motion prior extraction ("w/o Motion Prior"), the epipolar geometry cannot be accurately estimated, resulting in inaccurate motion segmentation. Without the prior-anchored epipolar geometry ("w/o Prior-Epipolar"), the coarse motion prior extracted from the backbone predictions tends to misclassify the static background as dynamic.

hence $|\varphi| \leq \arcsin(\theta_c/\|w\|_2)$. By (A2), the two necessary conditions are independent, and the second one has probability $\arcsin(\theta_c/\|w\|_2)/\pi \leq \theta_c/(2\|w\|_2)$, using $\arcsin z \leq \pi z/2$ for $z \in [0,1]$. By (A3), $\|w\|_2 - \theta_c > \theta_c$, and by (A1), $\Pr(\|\delta\|_2 \geq \|w\|_2 - \theta_c) \leq \Pr(\|\delta\|_2 > \theta_c) = \Pr(\mathrm{FP})$. Therefore $\Pr(\mathrm{FN}) \leq \frac{\theta_c}{2\|w\|_2} \Pr(\mathrm{FP})$. □

A false positive only requires the error to exceed θ_c in magnitude, whereas a false negative requires the error to reproduce the object displacement in both magnitude and direction. The false negative probability is bounded by $\theta_c/(2\|w\|_2)$ times the false positive probability, which also motivates a small θ_c. Without (A2), i.e., under (A1) and (A3) alone, the first necessary condition still gives $\Pr(\mathrm{FN}) \leq \Pr(\mathrm{FP})$. Conversely, without (A1), the second necessary condition alone gives $\Pr(\mathrm{FN}) \leq \theta_c/(2\|w\|_2)$ under (A2) whenever $\|w\|_2 > \theta_c$, i.e., the coarse motion prior covers a dynamic pixel with probability at least $1 - \theta_c/(2\|w\|_2)$ regardless of the error magnitude.

E More Implementation Details

All experiments are conducted on a single NVIDIA H800 GPU. We use RAFT (Teed & Deng, 2020) for the optical flow and SAM 3 (Carion et al., 2026) for all segmentation and tracking, where each point prompt of SAM consists of 8 points selected by farthest point sampling within the interior of the prompted region. The optical flow, camera flow, and epipolar geometry are computed on frames resized to a width of 518 pixels, and all the pixel thresholds below refer to this resolution.

For dynamic ghosting removal, the optical flow correspondences that fail a forward-backward consistency check with a threshold of 3 pixels are discarded. The coarse motion prior is obtained with $\theta_c = 1.4$ pixels, after the flow discrepancy is set to zero at pixels occluded in the adjacent frame. To estimate F, the static pixels outside M^c_{t-1} are sampled at an interval of 7 pixels. If fewer than 50 of them are available, supplementary pixels inside M^c_{t-1}, spaced at least 50 pixels apart whenever possible, are added until the total reaches 50. F is then estimated with MAGSAC++ (Barath et al., 2020), a variant of RANSAC, with an inlier threshold of 1 pixel and a confidence of 0.999, and the refined motion mask is obtained with $\theta_r = 1.68$ pixels. Before the expiration times are assigned, the motion object mask M_{t-1} is dilated by 5 pixels.

For static ghosting removal, the adjacency check uses $n = 4$ and an ε corresponding to 10 pixels, and filters out the connected components with an inscribed radius below 5 pixels or an area below 60 pixels. In the ghosting template construction, we set $m = 13$, and the object is regarded as static once its displacement stays below 3% of its bounding-box diagonal for three consecutive frames. If the backward tracking does not reach a static frame within the window, e.g., when the object keeps moving or leaves the view, no ghosting template is constructed, and only the points within the masks obtained by the backward tracking are erased. Only the points that have not yet expired are used to construct $\mathcal{G}_0$, and for the visibility verification, the template is voxelized with a voxel size of 1% of the bounding-box diagonal of $\mathcal{G}_0$. In Alg. 1, we set τ to 2.5% of this diagonal, $\rho = 0.05$, $\theta_v = 3°$, and $N = 8$, and dilate M_k by 5 pixels. Since $\mathcal{G}_0$ covers only part of the object, the views that mainly observe its other parts may fail the visibility verification against $\mathcal{G}_0$. We therefore perform the view selection and template completion for two rounds. In the second round, Alg. 1 is rerun with the template completed in the first round in place of $\mathcal{G}_0$, excluding the views already selected, to select such views and further complete the template.

For evaluation, the prediction of each sequence is aligned to the GT once, and the alignment is shared by all time instants. Specifically, a similarity transform is first estimated by the Umeyama algorithm (Umeyama, 1991) from the pixel-wise correspondences over all frames, regardless of the expiration times, and is then refined by point-to-plane ICP (Chen & Medioni, 1992) with a maximum correspondence distance of 0.1 m.

F Limitations and Future Work

Although StreamDynamic improves over prior work in removing historical ghosting from streaming dynamic scene reconstruction, its backtracking removal of static ghosting inevitably introduces additional latency compared with the backbone, since an object may remain static for a long time

before it starts to move. Developing a native streaming dynamic scene reconstruction model that is inherently free of historical ghosting is a promising direction for future work.