DYNSTREAM: ONLINE STREAMING 4D GAUSSIAN RECONSTRUCTION OF DYNAMIC WORLDS FROM UNPOSED VIDEO

Anonymous authors
Paper under double-blind review

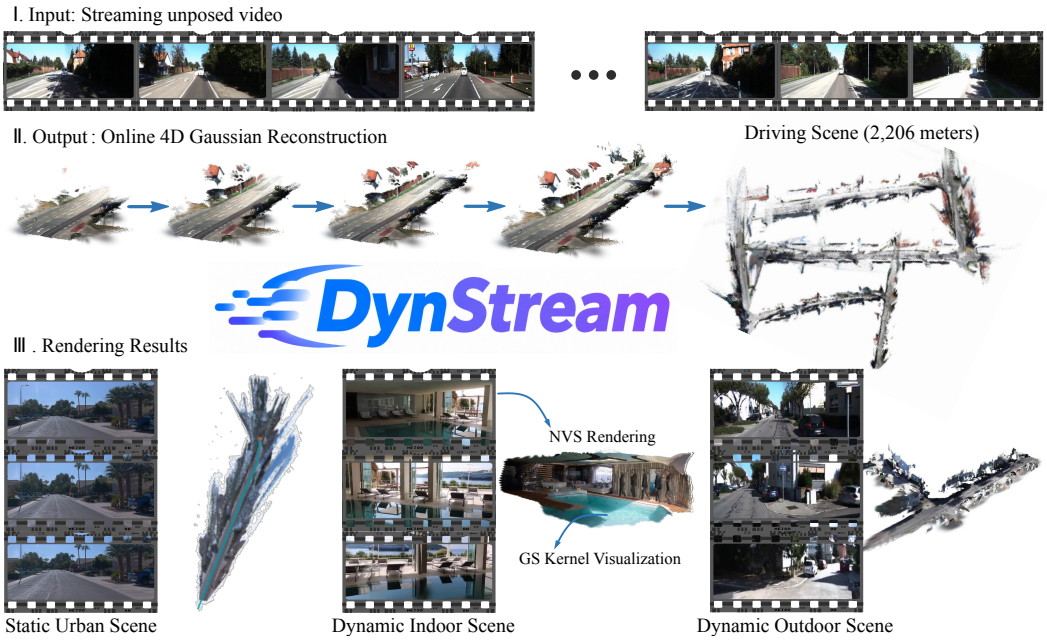


Figure 1: **DynStream** enables online, photorealistic 4D reconstruction from long, unposed video streams without per-scene optimization. It incrementally integrates incoming observations into a globally consistent representation, supports arbitrarily long streaming inputs, and generalizes across diverse dynamic and static indoor/outdoor scenes.

ABSTRACT

Online reconstruction of dynamic 4D scenes from long, unposed streaming videos requires both continuous processing and photorealistic rendering, which existing methods struggle to achieve simultaneously. Existing feed-forward Gaussian methods are restricted to offline processing, whereas online point-cloud approaches struggle to maintain dense geometry and high-fidelity rendering. We present **DynStream**, a framework for streaming 4D Gaussian reconstruction from long, unposed videos. Given a continuous video stream, DynStream reconstructs the scene within local temporal windows and incrementally aligns and fuses these local reconstructions into a globally consistent scene, enabling online 4D reconstruction without per-scene optimization. By jointly enforcing cross-window geometric consistency and modeling time-varying scene content, DynStream supports efficient reconstruction and photorealistic rendering over extended video streams. Experiments demonstrate that DynStream enables high-fidelity online dynamic reconstruction and rendering from long video streams, achieving state-of-the-art performance across diverse dynamic indoor and outdoor scenes.

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.

1 INTRODUCTION

Reconstructing dynamic 4D worlds from continuous video streams is a fundamental step toward building persistent, photorealistic representations of real-world environments. Unlike curated image sequences or short video clips, real-world streams are typically long and unposed, with unconstrained camera motion and continuously changing scene content. A practical reconstruction system must therefore incrementally integrate observations over time while preserving both geometric consistency and high-fidelity rendering. Such capability is essential for applications including autonomous driving, embodied intelligence, and interactive content creation. However, achieving continuous 4D reconstruction remains challenging, as the system must simultaneously maintain long-term temporal consistency, capture dynamic scene evolution, and operate efficiently under online processing constraints.

Existing approaches to online dynamic scene reconstruction largely follow two paradigms, both of which face fundamental limitations for long, continuously arriving video streams. Optimization-based methods (Newcombe et al., 2015; Slavcheva et al., 2017; Yu et al., 2018) incrementally update scene representations as new observations arrive, including recent Gaussian-based approaches (Li et al., 2026) for photorealistic online mapping. However, their reliance on scene-specific optimization and iterative tracking causes computational costs to grow with sequence length and can lead to accumulated tracking and reconstruction errors over time. Many methods also rely on depth estimates or motion priors, further limiting their applicability to unconstrained in-the-wild videos. Fundamentally, these approaches lack a scalable mechanism for incrementally incorporating long-term observations without repeatedly optimizing the scene.

Recent feed-forward methods (Ye et al., 2025; Hong et al., 2025; Jiang et al., 2025; Ye et al., 2026; Yang et al., 2025; 2026) avoid costly per-scene optimization by directly predicting scene representations from visual observations. However, they typically assume a fixed or bounded set of frames processed jointly in a single forward pass and operate on pre-recorded video clips, preventing newly arriving observations from being incorporated without reprocessing previous frames. As a result, computation and memory grow with sequence length, making them poorly suited to long video streams. Recent online feed-forward methods based on point clouds (Wang et al., 2024; Zhang et al., 2025; Wang et al., 2025) extend this paradigm to streaming reconstruction, but typically require ground-truth 3D geometry and camera poses for training, limiting scalable self-supervised learning and making joint estimation of camera motion and scene geometry difficult. Moreover, these methods often fail to simultaneously provide dense geometric predictions and photorealistic rendering. Thus, existing feed-forward approaches remain ill-suited to high-quality online reconstruction from long, unposed video streams due to their bounded temporal context, strong supervision requirements, or limited reconstruction fidelity.

To overcome these limitations, we present **DynStream**, a framework for online reconstruction of long, unposed video streams into globally consistent 4D Gaussian scenes, jointly estimating camera motion and time-varying scene representations while enabling photorealistic rendering without per-scene optimization or 3D/pose supervision. Our method is built on two key components. First, a *causal-attention online reconstruction module* incrementally processes incoming observations and predicts camera poses and newly observed Gaussian primitives using only past and current frames. Second, a *dynamic-aware relative-pose strategy* establishes spatial consistency across successive observations by accounting for time-varying scene content during pose estimation and scene integration. Specifically, we divide the incoming stream into consecutive temporal windows, or *chunks*. Within each chunk, the first frame serves as a keyframe, and subsequent camera poses are predicted relative to it along with the corresponding Gaussian representation. Because independently reconstructed chunks may reside in different coordinate gauges, particularly in scale, we additionally predict a metric scale factor to align each local reconstruction with a shared metric coordinate system. By incrementally processing and integrating successive chunks, DynStream enables online reconstruction of long dynamic video streams while preserving globally consistent geometry and high-fidelity scene representations.

The main contributions of this work are:

- We introduce DynStream, a framework for online dynamic 4D Gaussian reconstruction from long, unposed video streams, enabling continuous scene construction and photorealistic rendering through incremental integration of incoming observations.

- We develop a dynamic-aware relative-pose strategy that propagates camera and scene coordinates across successive chunks, enabling long-sequence dynamic reconstruction without independent chunk registration or per-scene optimization.
- Experiments demonstrate that DynStream achieves competitive rendering quality while enabling online dynamic scene reconstruction from long video streams, supporting efficient and scalable online 4D reconstruction across diverse scenes.

2 Related Work

Online Dynamic Scene Reconstruction Online reconstruction of dynamic scenes has traditionally been formulated as an iterative tracking-and-mapping problem, where an evolving scene representation is continuously updated as new observations arrive (Newcombe et al., 2015; Slavcheva et al., 2017; Yu et al., 2018). These approaches typically couple scene integration with correspondence estimation, motion tracking, and iterative optimization to maintain geometric consistency over time (Newcombe et al., 2015; Slavcheva et al., 2017; Yu et al., 2018), and often rely on explicit depth observations to support robust online reconstruction. More recent Gaussian-based methods extend this paradigm with directly renderable scene representations, jointly modeling scene geometry, camera motion, and dynamic content to enable photorealistic online reconstruction (Li et al., 2026). Nevertheless, these approaches remain fundamentally optimization- or tracking-driven, requiring repeated scene-specific computation as new observations arrive (Li et al., 2026). In contrast, our goal is to retain the incremental nature of online reconstruction while replacing repeated scene-specific optimization with direct feed-forward inference from RGB observations.

Feed-Forward Reconstruction Recent feed-forward reconstruction methods shift substantial geometric reasoning from test-time optimization into learned inference, directly estimating camera poses, depth, point-based geometry, and motion from visual observations (Wang et al., 2024; Zhang et al., 2025; Wang et al., 2025). Building on this paradigm, recent works further introduce causal attention, temporal feature propagation, keyframe-relative pose estimation, and metric scale prediction to extend feed-forward geometry reconstruction toward long and streaming sequences (Zhuo et al., 2026; Cheng et al., 2026). These approaches demonstrate that geometric reconstruction can be performed efficiently without repeatedly optimizing over the entire observation history. However, their outputs primarily describe geometric quantities rather than a continuously maintained, photorealistic scene representation that can be directly rendered and incrementally updated.

Feed-forward Gaussian reconstruction addresses this limitation by directly predicting renderable Gaussian primitives, together with camera parameters when necessary, from posed or unposed visual observations (Ye et al., 2025; Hong et al., 2025; Jiang et al., 2025; Ye et al., 2026). More recent approaches extend this formulation to dynamic scenes by jointly modeling time-varying geometry, appearance, motion, and camera poses (Yang et al., 2025; Chen et al., 2026; Yang et al., 2026), while streaming formulations have begun to process dynamic observations causally with bounded temporal context (Qiu et al., 2026). Despite this progress, existing methods either operate on pre-collected image sets or video clips, or require known camera parameters for streaming reconstruction, and thus do not continuously build a globally aligned, renderable scene from long, unposed video streams. In contrast, DynStream combines causal feed-forward reconstruction with direct Gaussian prediction, incrementally propagating camera and scene coordinates across temporal chunks and integrating them into a shared representation without reprocessing past frames or per-scene optimization.

3 Method

3.1 Problem Formulation

Given a long unposed video stream $\mathcal{V}_{1:T} = \{I_i\}_{i=1}^{T}$, our goal is to incrementally reconstruct a dynamic scene as a collection of Gaussians while simultaneously estimating the camera trajectory. Formally, our model implements the mapping:

$$F_\theta : \quad \mathcal{V}_{1:t} \longmapsto \mathcal{S}_t = (\mathcal{G}_t, \mathcal{T}_{1:t}), \tag{1}$$

where the output $\mathcal{S}_t$ contains a time-aware scene representation and the corresponding camera trajectory. Specifically, $\mathcal{G}_t$ denotes the Gaussian scene representation maintained at time t, while

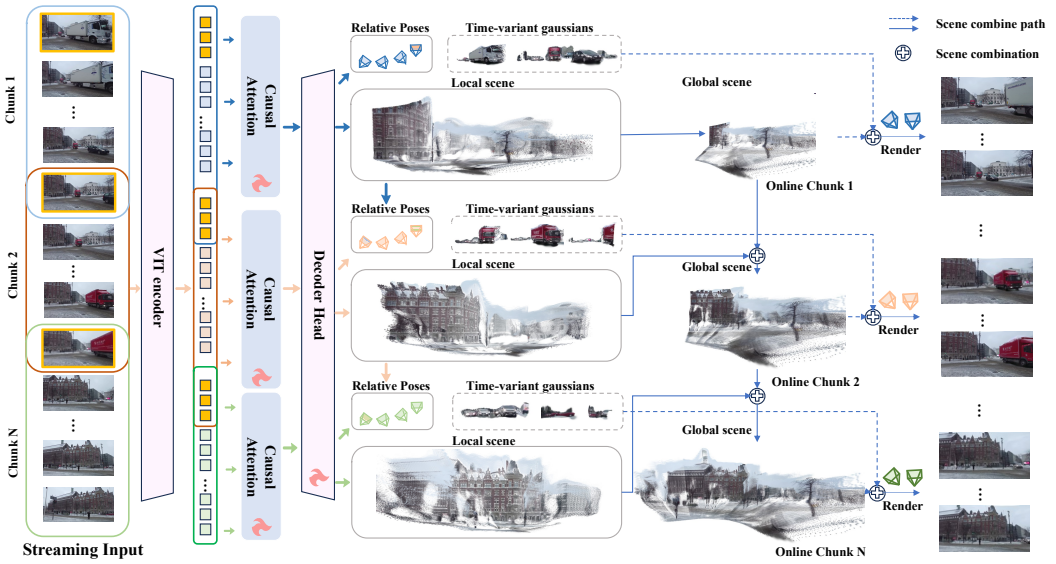


Figure 2: **Overview of DynStream.** The incoming video stream is processed sequentially in fixed-length chunks. For each chunk, the causal stream backbone predicts keyframe-relative camera poses, a local Gaussian scene, and time-varying Gaussian primitives. The local scene is aligned to the global coordinate system and incrementally fused with the previously reconstructed scene, forming a persistent global representation. Time-varying Gaussians preserve their temporal associations and are incorporated into the global scene only at the corresponding time. This design enables continuous reconstruction and rendering of long dynamic video streams while maintaining globally consistent static geometry and temporally coherent dynamic content.

$\mathcal{T}_{1:t} = (\mathbf{T}_1^w, \ldots, \mathbf{T}_t^w)$ denotes the estimated camera trajectory, with $\mathbf{T}_i^w \in \mathrm{SE}(3)$ representing the camera-to-world pose of frame I_i.

However, directly processing the entire growing prefix $\mathcal{V}_{1:t}$ with any ViT-based architecture is computationally expensive. With P tokens per frame, attention over t frames involves $O(t^2P^2)$ pairwise interactions and quadratic memory consumption (Dao et al., 2022; Dosovitskiy, 2020). We therefore divide the incoming stream into fixed-length consecutive temporal windows, referred to as **chunks**. Each chunk $\mathcal{C}_k = (I_{a_k}, \ldots, I_{b_k})$ contains a bounded number of consecutive frames, and its first frame I_{a_k} serves as the keyframe. During online processing, only the frames that have already arrived within the current chunk are visible to the model.

After organizing the stream into chunks, we can realize the mapping F_θ in Eq. (1) through two sequential stages: causal local reconstruction and global scene integration:

$$\mathcal{S}_t = F_\theta(\mathcal{V}_{1:t}) = \Psi\left(\mathcal{S}_{t-1}, \Phi_\theta(\mathcal{C}_{k,\le t}, \mathcal{H}_{k-1})\right). \tag{2}$$

In this formulation, Φ_θ reconstructs the newly observed content from the visible portion of the current chunk $\mathcal{C}_{k,\le t}$ and the retained reference information $\mathcal{H}_{k-1}$, while Ψ aligns the local prediction with the existing global scene and updates the scene state $\mathcal{S}_{t-1}$. We initialize $\mathcal{H}_0$ as empty and $\mathcal{S}_0$ with an empty scene and trajectory.

This formulation allows the model to process only a bounded temporal context at each step while preserving a coherent scene representation over long video streams. We implement Φ_θ with a causal-attention online reconstruction module and Ψ with a dynamic-aware relative-pose strategy, which are detailed in Sections 3.2 and 3.3 respectively.

3.2 Causal-Attention Online Reconstruction

Following Eq. (2), the local reconstruction operator Φ_θ converts the currently available observations into a keyframe-relative reconstruction. We implement Φ_θ with a DINOv2 image encoder (Oquab et al., 2023), a causal alternating-attention backbone, and task-specific prediction heads.

Causal alternating-attention backbone. For each input frame $I_i \in \mathbb{R}^{H\times W\times 3}$, we divide the image into non-overlapping patches and embed them into visual tokens using DINOv2 (Oquab et al., 2023). Following VGGT (Wang et al., 2025), the tokens are processed by alternating intra-frame and cross-frame attention blocks. Intra-frame attention aggregates spatial information within each image, while cross-frame attention exchanges information across observations. To satisfy the online setting in Eq. (1), we apply a causal mask so that frame I_t can only attend to observations available up to time t.

For the visible prefix of the current chunk, the backbone produces:

$$\mathbf{F}_{k,t} = A_\theta\left(E_\theta(\mathcal{C}_{k,\le t})\right), \tag{3}$$

where E_θ denotes the DINOv2 patch encoder and A_θ denotes the causal alternating-attention. This allows Φ_θ to aggregate spatial and temporal context without accessing future observations.

Multi-head local reconstruction. Task-specific heads decode $\mathbf{F}_{k,t}$ into the predictions required for local reconstruction:

$$(\mathbf{T}_{k,t}, D_{k,t}, G_{k,t}, s_{k,t}) = \Phi_\theta(\mathcal{C}_{k,\le t}, \mathcal{H}_{k-1}). \tag{4}$$

Here, $\mathbf{T}_{k,t} \in \mathrm{SE}(3)$ denotes the camera pose of frame I_t relative to the current keyframe, distinguishing it from the global camera-to-world pose $\mathbf{T}_t^w$ in Eq. (1). The depth head predicts $D_{k,t} \in \mathbb{R}^{H\times W\times 1}$, while the Gaussian head predicts a pixel-aligned Gaussian map $G_{k,t} \in \mathbb{R}^{H\times W\times C_G}$, with one Gaussian primitive associated with each spatial location.

Using RGB appearance, we set $C_G = 15$ and parameterize each Gaussian as:

$$\mathbf{g}_{k,t}(u,v) = [\boldsymbol{\mu}, \mathbf{q}, \mathbf{s}, \alpha, \mathbf{c}, p_{\mathrm{dyn}}] \in \mathbb{R}^{15}, \tag{5}$$

where $\boldsymbol{\mu} \in \mathbb{R}^3$ is the Gaussian center, $\mathbf{q} \in \mathbb{R}^4$ its rotation quaternion, $\mathbf{s} \in \mathbb{R}^3$ its anisotropic scale, $\alpha \in \mathbb{R}$ its opacity, $\mathbf{c} \in \mathbb{R}^3$ its RGB appearance, and $p_{\mathrm{dyn}} \in [0,1]$ its dynamic probability.

Although the predicted pose, depth, and Gaussian geometry share the same local coordinate system, independently reconstructed chunks can still differ by an unknown scale. This scale ambiguity affects camera translation, depth, Gaussian centers, and Gaussian sizes, preventing direct cross-chunk integration. We therefore introduce a scale head that predicts a positive metric factor $s_{k,t} > 0$, which is applied consistently to all length-related predictions before global alignment.

The resulting local reconstruction is then passed to Ψ, which aligns it with the global coordinate system and updates the maintained scene representation.

3.3 Relative-Pose Global Alignment and Dynamic-Aware Integration

Following Eq. (2), the alignment operator Ψ transforms the local reconstruction produced by Φ_θ into the shared world coordinate system and updates the persistent scene state. This process consists of relative-pose propagation, metric Gaussian alignment, and dynamic-aware scene integration.

Relative-pose global alignment. From Eq. (4), $\mathbf{T}_{k,t}$ represents the pose of frame I_t relative to the keyframe of chunk k. We write $\mathbf{T}_{k,t} = (\mathbf{R}_{k,t}, \mathbf{u}_{k,t})$, where $\mathbf{R}_{k,t}$ and $\mathbf{u}_{k,t}$ denote the relative rotation and translation, respectively.

Let $\mathbf{K}_k \in \mathrm{SE}(3)$ denote the camera-to-world pose of the keyframe of chunk k. At each chunk transition, the relative-pose head predicts a transformation $\mathbf{L}_k$ between the new keyframe and the previous keyframe using the retained reference information $\mathcal{H}_{k-1}$. Starting from $\mathbf{K}_1 = \mathbf{I}_4$, the global keyframe pose and the global pose of frame I_t are obtained as:

$$\mathbf{K}_k = \mathbf{K}_{k-1}\mathbf{L}_k, \qquad \mathbf{T}_t^w = \mathbf{K}_k \begin{bmatrix} \mathbf{R}_{k,t} & s_{k,t}\mathbf{u}_{k,t} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{6}$$

The predicted scale $s_{k,t}$ converts the local translation into metric units. The same calibration is applied to the predicted depth as $D_{k,t}^{\mathrm{m}} = s_{k,t} D_{k,t}$. In this way, local camera predictions from different chunks are progressively connected to the same global coordinate system without global trajectory optimization.

Metric Gaussian alignment. We apply the same scale and coordinate transformation to the local Gaussian map $G_{k,t}$. We denote this operation by:

$$G^w_{k,t} = \mathcal{W}\left(G_{k,t}; \mathbf{K}_k, s_{k,t}\right), \tag{7}$$

where $\mathcal{W}$ transforms the local Gaussian parameters into the world coordinate system. Specifically, Gaussian centers and spatial scales are multiplied by $s_{k,t}$, while their positions and orientations are transformed according to $\mathbf{K}_k$. Appearance, opacity, and dynamic probability remain unchanged. As a result, $G^w_{k,t}$ can be directly integrated with Gaussian predicted from previous chunks.

Dynamic-aware scene integration. Although the Gaussian primitives are spatially aligned, dynamic content observed at different times should not be indiscriminately accumulated into the same persistent representation. We therefore use the dynamic probability p_{dyn} contained in each Gaussian to distinguish persistent and time-varying content. Given a threshold τ, Gaussians with $p_{\text{dyn}} \leq \tau$ are treated as static and integrated into the persistent scene, while Gaussians with $p_{\text{dyn}} > \tau$ are treated as dynamic and retain their temporal association with the corresponding observation.

The alignment operator Ψ therefore updates the scene state by jointly incorporating the globally aligned Gaussian map $G^w_{k,t}$ and the camera pose $\mathbf{T}^w_t$. Static structure is progressively accumulated across chunks, whereas dynamic content remains time-specific. This allows successive local reconstructions produced by Φ_θ to form a spatially consistent and temporally aware global scene.



Figure 3: Qualitative comparison of NVS rendering results on the Waymo Open Dataset. Each column shows one scene, and each row corresponds to GT, **Ours**, DGGT, and NeoVerse, respectively.

3.4 Loss Function

We employ a frozen VGGT-Ω (Wang et al., 2026b) teacher to provide pseudo geometric and camera supervision during training. Based on these teacher predictions and the available metric and dynamic annotations, we train the model with five objectives that supervise appearance, geometry, camera motion, metric scale, and dynamic scene decomposition:

$$\mathcal{L} = \mathcal{L}_{\text{render}} + \lambda_1 \mathcal{L}_{\text{depth}} + \lambda_2 \mathcal{L}_{\text{pose}} + \lambda_3 \mathcal{L}_{\text{scale}} + \lambda_4 \mathcal{L}_{\text{dyn}}. \tag{8}$$

Online rendering loss. To preserve the online setting during training, rendering supervision is applied to the causal scene state $\mathcal{G}_t$, which contains only Gaussian primitives reconstructed from observations available up to time t. We render the current observation using its estimated global camera pose:

$$(\widetilde{I}_t, \widetilde{D}_t) = \mathcal{R}(\mathcal{G}_t, \mathbf{T}^w_t), \tag{9}$$

where $\mathcal{R}$ denotes the differentiable Gaussian rasterizer, and $\widetilde{I}_t$ and $\widetilde{D}_t$ are the rendered RGB image and depth map, respectively. Following (Jiang et al., 2025), we supervise rendering using a combination of pixel-wise reconstruction and perceptual losses:

$$\mathcal{L}_{\text{render}} = \|\widetilde{I}_t - I_t\|_2^2 + \lambda_5 \,\text{LPIPS}(\widetilde{I}_t, I_t). \tag{10}$$

Table 1: Rendering quality comparison on dynamic-scene datasets. For each scene, metrics are first averaged over 14 uniformly sampled frames, followed by an equal-weight average across scenes. Best results are shown in **bold**, and second-best results are underlined.

Method	Real-world Dynamic									Synthetic Dynamic					
	Waymo			SpatialVID			KITTI			PointOdyssey			TartanAir		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
AnySplat	24.109	0.7822	0.1998	22.096	0.7581	0.1528	16.003	0.4768	0.6493	20.554	0.6911	0.3157	22.302	0.6664	0.3025
DGGT	24.396	0.8125	0.1462	21.232	0.7350	0.1569	17.010	0.6497	0.2911	17.148	0.5704	0.4264	20.537	0.6488	0.2988
MVSplat	18.474	0.5291	0.3776	15.965	0.4217	0.3917	14.845	0.3697	0.3322	17.973	0.6093	0.4369	16.991	0.4731	0.4868
NeoVerse	19.732	0.6067	0.3596	17.400	0.5009	0.2982	13.469	0.4817	0.4432	21.868	0.6913	0.2834	18.932	0.5762	0.3259
STORM	15.435	0.5219	0.6682	14.606	0.3480	0.6310	11.201	0.2619	0.7729	17.233	0.5846	0.6094	14.562	0.3397	0.6944
Ours	**26.777**	**0.8675**	**0.0972**	**23.994**	**0.8213**	**0.1034**	**17.708**	**0.6930**	**0.2748**	**26.787**	**0.8456**	**0.1119**	**22.348**	**0.7466**	**0.1872**

Depth loss. Following AnySplat (Jiang et al., 2025), we supervise the depth branch using pseudo ground-truth depth D_{gt} predicted by the teacher model, together with a depth consistency term that enforces consistency between the predicted depth and the Gaussian representation:

$$\mathcal{L}_{\mathrm{depth}} = \|\hat{D}_t - D_{\mathrm{gt}}\|_2^2 + \lambda_6 \|\hat{D}_t - \widetilde{D}_t\|_2^2. \tag{11}$$

Relative pose loss. We supervise camera motion in the keyframe-relative gauge. For frame I_t, the teacher relative pose is obtained from the teacher camera poses as $(\mathbf{T}^{\mathrm{tea}}_{r_k})^{-1}\mathbf{T}^{\mathrm{tea}}_t$. We represent the relative camera prediction as $\mathbf{p}_{k,t} = [\mathbf{u}_{k,t}, \mathbf{q}_{k,t}, \mathbf{f}_{k,t}]$, where $\mathbf{u}$, $\mathbf{q}$, and $\mathbf{f}$ denote translation, unit quaternion, and focal offset, respectively. The pose loss is:

$$\mathcal{L}_{\mathrm{pose}} = \|\overline{\mathbf{u}}_{k,t} - \overline{\mathbf{u}}^{\mathrm{tea}}_{k,t}\|_1 + \|\mathbf{q}_{k,t} - \mathbf{q}^{\mathrm{tea}}_{k,t}\|_1 + \|\mathbf{f}_{k,t} - \mathbf{f}^{\mathrm{tea}}_{k,t}\|_1, \tag{12}$$

where $\overline{\mathbf{u}}$ denotes scale-normalized translation. This decouples relative camera estimation from the metric scale learned by the scale head.

Metric scale loss. Following the orthogonal scale learning strategy of LongStream (Cheng et al., 2026), the scale head is supervised independently in log space:

$$\mathcal{L}_{\mathrm{scale}} = \left|\log s_{k,t} - \log s^{\mathrm{gt}}_{k,t}\right|. \tag{13}$$

We apply $\mathcal{L}_{\mathrm{scale}}$ only to training samples with metric-scale supervision; for non-metric datasets, this term is disabled.

Dynamic loss. To supervise dynamic scene decomposition, we use SAM3 to generate a pseudo ground-truth mask $M_t \in \{0,1\}^{H\times W}$ for frame I_t. Since the Gaussian map is pixel-aligned, the predicted dynamic probability map $P_t \in [0,1]^{H\times W}$ can be directly supervised in image space. We use the binary cross-entropy $\ell_{\mathrm{BCE}}(p,m) = -[m\log p + (1-m)\log(1-p)]$ and compute:

$$\mathcal{L}_{\mathrm{dyn}} = \frac{1}{HW}\sum_{u,v} \ell_{\mathrm{BCE}}\left(P_t(u,v), M_t(u,v)\right). \tag{14}$$

This supervision encourages the Gaussian head to assign high dynamic probability to time-varying regions and low probability to persistent scene content.

4 Experiments

4.1 Experimental Setup

Dataset. We train our model on SpatialVID(Wang et al., 2026a), Waymo Open Dataset(Sun et al., 2020), PointOdyssey(Zheng et al., 2023), DL3DV-10K(Ling et al., 2024), HM3D(Ramakrishnan et al., 2021), Replica(Straub et al., 2019), ETH3D(Schops et al., 2017), Matterport3D(Chang et al., 2017), and TartanAir(Wang et al., 2020), covering diverse real and synthetic indoor and outdoor environments with varying scene dynamics, scales, and camera motion patterns. More details about datasets are provided in the supplementary material.

Table 2: Camera tracking performance on the KITTI Odometry benchmark. We evaluate on all 11 sequences comprising 23,201 frames and report sequence-level macro-averaged trajectory errors. Lower is better for all metrics. Best results are shown in **bold**, and second-best results are underlined.

Method	Sim(3) ATE (m)↓	SE(3) ATE (m)↓	Raw ATE (m)↓	Raw RPE-t (m)↓	Scale-correctedRPE-t (m) ↓	RPE-r (deg)↓
STream3R	227.226	236.056	410.016	1.122	11.788	6.451
StreamVGGT	205.286	236.072	410.059	1.088	18.370	10.867
VGGT-SLAM	219.499	416.326	582.679	342.319	39.076	9.496
Ours	**45.627**	**142.064**	**283.220**	**0.614**	**0.255**	**0.274**

Implementation Details. We initialize the geometry transformer, depth DPT head, relative pose head with weights from VGGT(Wang et al., 2025), while the remaining layers are initialized randomly. We use AdamW as our optimizer, where β = (0.9, 0.95), weight decay 0.05, linear warm-up followed by cosine decay, BF16 mixed precision, and gradient clipping at 0.5. The training loader samples 2–14 context views with probabilities proportional to the squared view count and varies image height in multiples of 16 from 224 to 448. For training losses, we set $\lambda_1 = 1, \lambda_2 = 10, \lambda_3 = 1, \lambda_4 = 0.1, \lambda_5 = 0.05, \lambda_6 = 0.1$.

4.2 Quantitative Results

Rendering quality. We evaluate our method on both outdoor and indoor dynamic-scene benchmarks and compare our method with recent state-of-the-art feed-forward gaussian reconstruction framework. We report PSNR, SSIM and LPIPS-Alex in this experiment. As is shown in Tab. 1, our method achieve SOTA rendering quality in all metrics.

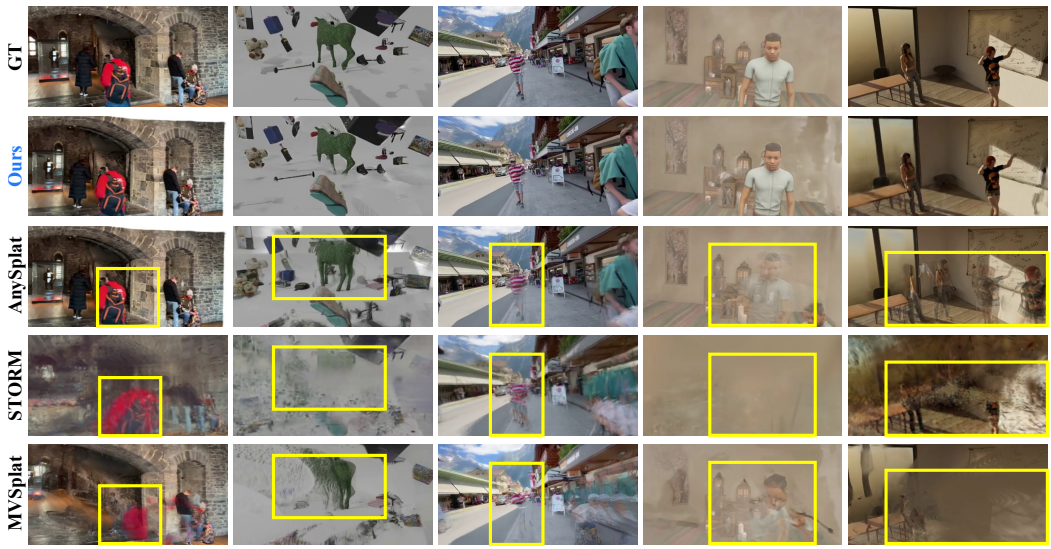


Figure 4: Qualitative comparison of novel-view synthesis (NVS) results across diverse scene types. Each column shows a scene, while each row corresponds to the ground truth or a baseline method.

Camera pose estimation. We evaluate all KITTI odometry sequences 00–10 (Geiger et al., 2012): 23,201 frames in total, with the longest sequence containing 4,661 frames and covering 5.068 km. Metrics include Sim(3)- and SE(3)-aligned absolute trajectory error, raw absolute error, and translational/rotational relative error. The results are shown in Tab.2. Compared to previous online reconstruction method, our method perform better camera pose estimation in long sequence input.

4.3 Qualitative Results

Rendering Quality. We qualitatively evaluate DynStream on large-scale driving scenes and diverse indoor and outdoor environments. As shown in Fig. 3 and Fig. 4, DynStream consistently

Table 3: Ablation of dynamic-aware scene modeling on Waymo and SpatialVID. Removing the dynamic component consistently degrades rendering quality. Best results are shown in **bold**.

	Waymo			SpatialVID		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DynStream	**26.777**	**0.8675**	**0.0972**	**23.994**	**0.8213**	**0.1034**
DynStream w/o dynamic	23.011	0.7662	0.1814	21.161	0.6529	0.1981

Table 4: Ablation of relative-pose estimation and metric-scale prediction on KITTI and VKITTI2. We report Sim(3)- and SE(3)-aligned absolute trajectory error (ATE).

	KITTI		VKITTI2	
Method	Sim(3) ATE ↓	SE(3) ATE ↓	Sim(3) ATE ↓	SE(3) ATE ↓
DynStream	**45.63**	**142.06**	**10.33**	**25.05**
DynStream w/o scale head	49.66	223.80	17.76	85.00
DynStream w/o relative pose	46.58	155.29	19.84	37.99

outperforms existing methods in both geometric fidelity and visual quality. Static methods suffer from ghosting and duplicated geometry, while dynamic methods introduce artifacts from inaccurate motion segmentation. In contrast, DynStream explicitly separates persistent and dynamic content, yielding clean geometry and faithful appearance.

Dynamic Segmentation. We further evaluate the predicted dynamic probabilities against SAM3-generated reference masks. Across both Waymo and SpatialVID, DynStream shows substantially stronger agreement with the reference foreground masks than the compared methods, particularly in foreground coverage and overall overlap. We provide the full protocol, quantitative results, threshold analysis, and fairness considerations in the supplementary material.

4.4 Ablation Study

We ablate the main components of DynStream to evaluate their contributions to dynamic-scene reconstruction and long-sequence camera estimation. Specifically, we consider three variants: *w/o dynamic*, which removes dynamic-aware scene decomposition and treats all predicted Gaussians uniformly; *w/o scale head*, which disables metric-scale prediction; and *w/o relative pose*, which removes the proposed keyframe-relative pose formulation. We report rendering quality on Waymo and SpatialVID using PSNR, SSIM, and LPIPS, and evaluate camera trajectories on KITTI and VKITTI2 using Sim(3)- and SE(3)-aligned ATE.

Dynamic-aware reconstruction. Table 3 evaluates the effect of dynamic-aware scene modeling. Removing the dynamic component substantially degrades rendering quality on both datasets, indicating that explicitly distinguishing time-varying content is important for preventing dynamic observations from being incorrectly accumulated into the persistent scene.

Relative pose and metric scale. Table 4 studies the effect of the relative-pose formulation and metric-scale prediction. Removing the scale head notably increases SE(3)-aligned ATE, especially on VKITTI2, showing that metric calibration is important for maintaining a consistent trajectory scale. Removing relative-pose prediction also degrades trajectory accuracy, confirming the benefit of keyframe-relative camera estimation for long-sequence reconstruction.

5 Conclusion

We presented **DynStream**, a framework for online 4D Gaussian reconstruction from long, unposed video streams. By incrementally reconstructing local temporal windows and aligning them into a persistent global representation, DynStream eliminates per-scene optimization while maintaining globally consistent geometry and time-varying content. Extensive experiments demonstrate state-of-the-art reconstruction quality across diverse dynamic scenes. These results establish DynStream as a scalable approach to persistent online reconstruction of dynamic environments.

AI USE STATEMENT

Generative AI tools were used for language polishing and improving the clarity and readability of this manuscript.

ETHICS STATEMENT

Reconstruction from driving and indoor imagery can expose people, locations, and private environments. The intended applications are scene understanding and mapping, not personal identification. Data, pretrained checkpoints, and baseline implementations remain subject to their licenses and access conditions; possession of a local copy does not grant redistribution rights. Any public artifact must undergo privacy, licensing, and anonymization review. Reported failures and differences in evaluation protocols are retained to avoid overstating safety, fidelity, or generalization.

REFERENCES

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

Xiaoxue Chen, Ziyi Xiong, Yuantao Chen, Gen Li, Nan Wang, Hongcheng Luo, Long Chen, Haiyang Sun, Bing Wang, Guang Chen, et al. Dggt: Feedforward 4d reconstruction of dynamic driving scenes using unposed images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1265–1276, 2026.

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*, 2026.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361. IEEE, 2012.

Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. Pf3plat: Pose-free feed-forward 3d gaussian splatting for novel view synthesis. In *Forty-second International Conference on Machine Learning*, 2025.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)*, 44(6):1–16, 2025.

Runfa Blark Li, Mahdi Shaghaghi, Keito Suzuki, Xinshuang Liu, Varun Moparthi, Bang Du, Walker Curtis, Martin Renschler, Ki Myung Brian Lee, Nikolay Atanasov, et al. Dynagslam: Real-time gaussian-splatting slam for online rendering, tracking, motion predictions of moving objects in dynamic scenes. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2434–2444. IEEE, 2026.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 343–352, 2015.

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. *arXiv preprint arXiv:2304.07193*, 2023.

Zhicheng Qiu, Jiarui Meng, Tong-an Luo, Yican Huang, Xuan Feng, Xuanfu Li, and ZHan Xu. Slarm: Streaming and language-aligned reconstruction model for dynamic scenes. *arXiv preprint arXiv:2603.22893*, 2026.

Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3260–3269, 2017.

Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5474–5483. IEEE, 2017.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2451. IEEE, 2020.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Chang Zeng, Yanxi Zhou, Xiao-Xiao Long, et al. Spatialvid: A large-scale video dataset with spatial annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42592–42603, 2026a.

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, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

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.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916. IEEE, 2020.

Jiawei Yang, Jiahui Huang, Boris Ivanovic, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Apoorva Sharma, Maximilian Igl, Peter Karkus, et al. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. In *International Conference on Learning Representations*, volume 2025, pp. 50446–50465, 2025.

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. *arXiv preprint arXiv:2601.00393*, 2026.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *International Conference on Learning Representations*, volume 2025, pp. 54009–54033, 2025.

Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. Yonosplat: You only need one model for feedforward 3d gaussian splatting. In *International Conference on Learning Representations*, volume 2026, pp. 39852–39871, 2026.

Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7287–7296, 2018.

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.

Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19798–19808. IEEE, 2023.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 88055–88072, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8da04a60948be713dc766f0c7e3a5b1f-Paper-Conference.pdf.

SUPPLEMENTARY MATERIAL

This supplementary material provides additional training details, evaluation protocols, dynamic-segmentation analysis, and limitations omitted from the main paper due to space constraints.

A DATASETS AND IMPLEMENTATION DETAILS

Training datasets. We train DynStream on SpatialVID (Wang et al., 2026a), Waymo Open Dataset (Sun et al., 2020), PointOdyssey (Zheng et al., 2023), DL3DV-10K (Ling et al., 2024), HM3D (Ramakrishnan et al., 2021), Replica (Straub et al., 2019), ETH3D (Schops et al., 2017), Matterport3D (Chang et al., 2017), and TartanAir (Wang et al., 2020). These datasets cover real and synthetic indoor and outdoor environments with diverse camera motion, scene scale, and dynamics.

Table I: Overview of the datasets used for training.

Dataset	Scene Type	Dynamic	Real / Synthetic	# Frames	# Scenes
SpatialVID Wang et al. (2026a)	Indoor / Outdoor	✓	Real	7,089 h	2.7M clips
Waymo Sun et al. (2020)	Outdoor	✓	Real	390K	2,030
PointOdyssey Zheng et al. (2023)	Indoor / Outdoor	✓	Synthetic	∼200K	159
DL3DV-10K Ling et al. (2024)	Indoor / Outdoor	✗	Real	51.2M	10,510
HM3D Ramakrishnan et al. (2021)	Indoor	✗	Real	–	1,000
Replica Straub et al. (2019)	Indoor	✗	Real	–	18
ETH3D Schops et al. (2017)	Indoor / Outdoor	✗	Real	454	25
Matterport3D Chang et al. (2017)	Indoor	✗	Real	194.4K	90
TartanAir Wang et al. (2020)	Indoor / Outdoor	✓	Synthetic	>1M	30

Training details. We initialize the geometry transformer, DPT depth head, and relative-pose head from VGGT (Wang et al., 2025); the remaining prediction layers are initialized randomly. We use AdamW with $\beta = (0.9, 0.95)$ and weight decay 0.05, followed by linear warm-up and cosine learning-rate decay. Training uses BF16 mixed precision and gradient clipping at 0.5.

The training loader samples 2–14 input views, with sampling probability proportional to the squared number of views, and varies the image height from 224 to 448 in multiples of 16. The loss weights are $\lambda_1 = 1$, $\lambda_2 = 10$, $\lambda_3 = 1$, $\lambda_4 = 0.1$, $\lambda_5 = 0.05$, and $\lambda_6 = 0.1$. Metric scale supervision is applied only to samples with metric-scale annotations.

B ADDITIONAL EVALUATION PROTOCOLS

Rendering evaluation. We report PSNR, SSIM, and LPIPS-Alex for rendering quality. All methods are evaluated on the same input sequences for each benchmark. No additional image enhancement or post-processing is applied to the rendered outputs.

Camera pose evaluation. We evaluate KITTI odometry sequences 00–10 (Geiger et al., 2012), containing 23,201 frames in total. We report Sim(3)-aligned ATE, SE(3)-aligned ATE, raw ATE, translational RPE, and rotational RPE. Sim(3) alignment removes global scale, while SE(3) alignment preserves scale error and therefore additionally reflects the quality of metric-scale estimation.

Long-sequence setting. All sequences are processed incrementally. The reconstruction network operates on the current bounded chunk rather than jointly processing the complete sequence. Previously reconstructed scene content and camera poses are maintained in the global scene state.

C ADDITIONAL ANALYSIS OF DYNAMIC SEGMENTATION

The Gaussian head predicts a dynamic probability for every pixel-aligned Gaussian. We further evaluate these predictions against automatically generated SAM3 reference masks.

Table II: Consistency with SAM3-generated foreground reference masks.

Dataset	Method	Scenes / Frames	IoU ↑	Coverage ↑	Precision ↑	Dice ↑	AP ↑	Sec./Frame ↓
Waymo	DynStream	100 / 1400	72.00	97.74	72.97	83.22	96.58	0.0243
Waymo	DGGT	100 / 1400	34.88	39.13	76.68	46.69	63.37	0.0329
Waymo	NeoVerse	100 / 1400	23.02	35.40	48.15	33.09	30.51	0.0386
SpatialVID	DynStream	100 / 1400	58.82	95.79	59.91	72.69	91.86	0.0251
SpatialVID	NeoVerse	100 / 1400	28.25	44.85	47.72	39.16	37.77	0.0462
SpatialVID	MoVieS	100 / 1400	12.40	51.48	18.68	19.70	36.04	0.0980

Table III: Dynamic segmentation using a common probability threshold of 0.5.

Dataset	Method	Scenes	IoU ↑	Coverage ↑	Precision ↑	Dice ↑
Waymo	DynStream	100	83.31	88.90	92.74	90.61
Waymo	DGGT	100	35.45	40.67	74.15	47.52
SpatialVID	DynStream	100	74.93	82.46	87.28	84.32

Reference masks. We generate SAM3 masks using the prompts *vehicle*, *person*, and *cyclist*. These masks describe semantic foreground rather than ground-truth physical motion: for example, parked vehicles may be included, while moving vegetation or water may be excluded. Our dynamic head is also trained with automatically generated labels from the same source. Therefore, this experiment measures consistency with a common foreground reference rather than independent zero-shot motion segmentation.

Protocol. We evaluate 100 Waymo scenes and 100 SpatialVID scenes. For each scene, 14 frames are uniformly sampled at 224×448 resolution, resulting in 1,400 frames per dataset. All methods receive the same RGB inputs, without ground-truth cameras or additional sky masks. Scenes are selected before inference using a fixed random seed and foreground amount, without filtering according to method performance.

For SpatialVID, we use the validation split produced by the current backbone data loader. Since a historical fixed training-scene list is unavailable, we cannot completely exclude possible overlap introduced by earlier preprocessing changes.

Metrics. We report IoU, foreground coverage, precision, Dice, and average precision (AP). Coverage is defined as $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, while precision is $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$. Metrics are first computed after aggregating the 14 frames of each scene and then averaged across scenes. AP is computed from continuous dynamic scores and does not require selecting a binary threshold.

As shown in Table II, DynStream obtains substantially higher IoU, Dice, and AP on both datasets. Its high foreground coverage is particularly useful for reconstruction, since missing dynamic regions may cause time-varying Gaussians to be incorrectly accumulated into the persistent scene.

Threshold analysis. The main reconstruction pipeline uses $p_{\mathrm{dyn}} > 0.05$, favoring foreground recall to reduce dynamic false negatives. Since different methods expose different dynamic signals, their native thresholds are not directly comparable. We therefore additionally compare compatible probability outputs using the fixed threshold 0.5.

DGGT uses its native dynamic-logit threshold of 0.5. NeoVerse uses a velocity threshold of 2×10^{-4} together with cross-view motion verification and therefore cannot be directly compared under a probability threshold. For MoVieS, we estimate the required cameras using an independent official VGGT model from the same RGB inputs and classify points as dynamic when the maximum normalized 3D displacement over the 14 target times exceeds 0.01.

Runtime includes synchronized model inference and dynamic-mask extraction but excludes model loading, image decoding, export, and metric computation. The additional VGGT preprocessing required by MoVieS is included.

Table IV: Average end-to-end runtime across all available online chunk experiments. Runtime is measured as milliseconds per input frame. Lower is better.

	End-to-end runtime under different chunk numbers ↓			
Method	**2**	**3**	**5**	**100**
DGGT	300.4	308.8	405.2	Failed
MVSplat	410.2	505.2	568.6	Failed
NeoVerse	1482.3	320.3	1313.2	Failed
Ours	753.4	444.6	639.3	315.5

D Model Efficiency

DynStream achieves a favorable trade-off between reconstruction quality and computational efficiency while remaining scalable to arbitrarily long video streams. As shown in Table IV, its end-to-end runtime remains manageable as the number of input chunks increases, reaching 315.5 ms per frame even with 100 chunks. In contrast, all competing feed-forward methods fail to process the 100-chunk stream, highlighting their limited scalability to long-term online reconstruction. This scalability stems from our incremental chunk-wise processing, which incorporates new observations without reprocessing previously reconstructed content, enabling DynStream to operate continuously on unbounded streaming inputs.

E Additional Qualitative Results

We provide additional qualitative results for both driving scenes and diverse indoor and outdoor environments. Consistent with the main paper, static reconstruction methods often produce ghosting or duplicated geometry around moving objects, while imperfect dynamic separation can leave residual artifacts. DynStream better preserves persistent background geometry while maintaining time-specific foreground content.

F Detailed Ablation Studies

We ablate the main components of DynStream: causal temporal aggregation, keyframe-relative pose prediction, metric scale estimation, and dynamic-aware scene integration.

The causal-attention ablation measures the effect of temporal information exchange under the online constraint. The relative-pose and scale ablations evaluate cross-chunk camera and geometry consistency. Finally, removing dynamic-aware integration tests whether treating all Gaussians as persistent static content introduces artifacts in time-varying regions.

G Failure Cases and Limitations

DynStream still has several limitations. First, global camera poses are obtained by propagating relative transformations across chunks, so pose and scale errors may accumulate over very long trajectories without global bundle adjustment or loop closure.

Second, dynamic integration depends on the predicted dynamic probability. False negatives may incorrectly fuse moving content into the persistent scene, while false positives may prevent static geometry from being accumulated correctly. The SAM3-based supervision also emphasizes the selected semantic foreground categories rather than arbitrary physical motion.

Finally, bounded temporal context limits network inference cost but does not bound the size of the accumulated Gaussian scene. Memory consumption can therefore continue to increase when processing very long sequences containing large unexplored regions or extensive dynamic content.

H BROADER IMPACTS

Online 4D reconstruction can benefit robotics, autonomous systems, simulation, and embodied intelligence by providing continuously updated representations of dynamic environments. However, reconstruction of driving and indoor imagery may contain people, vehicles, locations, or private spaces. The intended use of DynStream is scene understanding and mapping rather than personal identification. Dataset licenses, privacy requirements, and anonymization procedures should therefore be respected when releasing reconstructed scenes or derived data.