IGGT4D: Streaming 4D Instance-Grounded Geometry Transformer

Zhengyu Zou[1,†], **Hao Li**[2], **Kuixuan Jiao**[1,†], **Liu Liu**[1,‡], **Tingyang Xiao**[1], **Xiaolin Zhou**[1], **Fangzhou Hong**[2], **Zhizhong Su**[1], **Dingwen Zhang**[3,✉], **Ziwei Liu**[2]
[1]Horizon Robotics [2]S-Lab, Nanyang Technological University
[3]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
Project Page: https://iggt4d.github.io



Figure 1: **Online geometry-instance prediction with IGGT4D.** IGGT4D processes a dynamic video stream frame by frame to incrementally build a unified 4D representation that jointly captures camera motion, 3D geometry, and temporally consistent instance features. This representation supports diverse downstream applications. We also construct **InsScene4D-147K**, a large-scale dataset with geometry-guided instance masks, to support training and evaluation.

Abstract

Real-world spatial intelligence requires agents to understand scenes from continuous video streams, where objects move, persist, disappear, and reappear over time. While recent spatial foundation models have enabled generalizable feed-forward 3D reconstruction, most streaming methods remain geometry-centric and lack temporally consistent object-level understanding. Meanwhile, existing semantic reconstruction and 3D-aware vision-language methods largely rely on externally extracted 2D semantic cues or loosely coupled geometry inputs, limiting unified geometry-instance learning in long dynamic scenes. In this paper, we propose **IGGT4D**, a streaming instance-grounded geometry Transformer for online 4D scene understanding. IGGT4D processes video frames sequentially, reuses

† Intern at Horizon Robotics ‡ Project Leader ✉ Corresponding Author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

historical context through causal spatial-temporal modeling, and incrementally updates a unified representation of camera motion, geometry, and object identity. This enables long-sequence feed-forward reconstruction with geometry-instance consistency in dynamic environments. To address the lack of high-quality 4D supervision, we further construct **InsScene4D-147K**, a large-scale dataset spanning real/synthetic and static/dynamic scenes, with RGB images, depth, poses, and temporally consistent instance masks generated by an automated geometry-guided annotation pipeline. Experiments on 3D reconstruction, pose estimation, instance spatial tracking, and open-vocabulary segmentation demonstrate that IGGT4D outperforms existing streaming baselines while supporting streaming inference on long dynamic sequences.

1 Introduction

Real-world spatial intelligence is inherently online and dynamic. During instruction-guided navigation [1, 2] or robotic manipulation [3, 4], an embodied agent receives a continuous video stream in which objects move, become occluded, leave the field of view, and reappear over time. Acting in such environments requires more than estimating camera motion and scene geometry: the agent must also maintain temporally consistent object identities. We therefore formulate 4D scene understanding as streaming geometry-instance prediction, where scene geometry and object identities are updated jointly from long, dynamic video streams.

Spatial reconstruction and scene understanding are undergoing a shift from optimization-based pipelines to data-driven feed-forward prediction. Traditional geometric pipelines, from offline SfM [5, 6] to online and neural SLAM [7–11], rely on iterative per-scene optimization. Similarly, semantic reconstruction systems, from offline scene graphs [12] to online semantic mapping [13, 14], often construct geometry and object-level cues through separate modules. These systems are effective, but their scene-specific optimization limits long-sequence efficiency, while their modular design makes temporally consistent instance reasoning difficult in dynamic scenes.

Recent spatial foundation models [15–18] address these limitations by learning generalizable priors for direct 3D prediction from images. For online long-sequence perception, this paradigm is further moving from global fixed-set inference to streaming reconstruction [19–21]. However, streaming 3D prediction alone is insufficient for embodied agents. Current streaming models remain largely geometry-centric: they estimate depth, pose, and point maps, but do not maintain temporally consistent object identities. The missing capability is a feed-forward streaming model that treats object identity as a first-class prediction target together with geometry.

Why has streaming geometry-instance understanding remained underexplored? A key reason is the lack of suitable supervision. 2D vision-language models [22–25] provide scalable open-vocabulary cues, but their predictions are view-dependent and lack metric 3D grounding. Semantic reconstruction methods [26–30] improve spatial coherence by lifting 2D semantics into 3D, but their semantic fidelity remains bounded by the capability of the external 2D predictors. 3D-aware vision-language models [31–34] incorporate geometry for spatial reasoning, but do not provide dense, temporally consistent geometry-instance supervision for training streaming models. Holi-Spatial [35] automates video geometry and object annotation through per-scene 3D reconstruction. We focus on unified, large-scale supervision of metric geometry, camera motion, and temporally consistent instances across static and dynamic scenes.

To address these gaps, we propose **IGGT4D**, a streaming instance-grounded geometry Transformer for online 4D scene understanding. IGGT4D turns geometry-instance reconstruction from a full-sequence offline problem into a causal streaming prediction problem. It processes frames sequentially, reuses historical context through causal spatial-temporal modeling, and incrementally updates camera motion, scene geometry, and instance embeddings in a unified representation. By grounding instance association in reconstructed geometry, IGGT4D maintains object identities across viewpoint changes, occlusions, and reappearance.

We further construct **InsScene4D-147K**, a large-scale dataset for geometry-instance learning in 4D scenes. It spans real/synthetic and static/dynamic sources, and provides RGB images, depth maps, camera poses, point clouds, and sequence-level consistent instance masks. To reduce annotation cost,

we design an automated geometry-guided annotation pipeline that produces multi-view consistent geometry and instance labels at scale. Our contributions are threefold:

- **Streaming geometry-instance prediction.** We formulate online 4D scene understanding as causal prediction of camera motion, scene geometry, and persistent object identities from continuous video streams.
- **Object-consistent streaming reconstruction.** We introduce IGGT4D, a feed-forward streaming Transformer that couples causal geometry modeling with geometry-grounded instance prediction and online clustering, maintaining consistent object identities without full-sequence inference.
- **Scalable 4D instance supervision.** We construct InsScene4D-147K, a large-scale real/synthetic and static/dynamic dataset with geometry-consistent instance annotations generated by a scalable reconstruction, projection, and mask-refinement pipeline.

2 Related work

Streaming Spatial Foundation Models Feed-forward spatial foundation models learn generalizable 3D priors for direct reconstruction from images [15–18, 36–40]. DUSt3R [36] formulates dense 3D prediction as point-map regression [36, 37], while VGGT-style models extend this paradigm toward large-scale and modality-augmented visual geometry estimation [15, 38]. NoDrift3R [41] further couples geometry and appearance through ray maps for drift-robust feed-forward Gaussian reconstruction. However, these models are designed for fixed image sets; applying them to a growing video stream requires reprocessing the full history or using sliding windows, causing redundant computation and weakening long-range consistency. Recent streaming variants, including Spann3R [19], MUSt3R [42], CUT3R [20], StreamVGGT [43], STream3R [21], and MoRe [44], reuse historical context through mechanisms such as causal attention and KV caching to enable incremental geometry prediction without rerunning global inference. However, these methods remain focused on geometric consistency and do not maintain persistent instance identities across frames for streaming 4D scene understanding.

Instance-Aware 3D Scene Understanding Object-level scene understanding has also progressed from 2D open-vocabulary perception and per-scene semantic lifting toward unified 3D prediction. LangSplat [45] and LangSurf [46] attach language features to optimized 3D representations for open-vocabulary querying, while language-driven segmentation models such as LSeg [23] predict semantic regions directly from images. These methods improve semantic accessibility, but they often depend on external 2D cues, per-scene optimization, or view-independent predictions, making it difficult to maintain coherent 3D object identities over time. More recent feed-forward scene understanding models aim to couple reconstruction with semantics or instances: LSM [47] predicts semantic radiance fields from unposed images, Uni3R [29] predicts semantic 3D Gaussians from arbitrary multi-view inputs, and IGGT [48] jointly predicts geometry and instance features with contrastive supervision and full-sequence offline clustering. These models move closer to joint geometry-instance learning, but they are still built around fixed image sets and global cross-view attention. For long video streams, each update must attend to or recompute historical frames, making online inference expensive and preventing causal instance updates. IGGT4D targets the intersection of these two lines: it keeps the streaming efficiency of causal spatial foundation models while jointly predicting geometry and temporally consistent instance features for online 4D scene understanding.

3 Method

IGGT4D extends the joint geometry-instance prediction of IGGT [48] to causal streaming reconstruction. Sec. 3.1 formalizes sequential feed-forward prediction. Sec. 3.2 introduces the streaming architecture, and Sec. 3.3 presents the efficient streaming clustering strategy. Sec. 3.4 describes downstream 4D scene understanding applications, followed by the training objectives in Sec. 3.5.

3.1 Problem Formulation and Notation

Given an RGB image sequence $\{I_t\}_{t=1}^N$, where each image $I_t \in \mathbb{R}^{H\times W\times 3}$ and optional camera parameters $\{\tilde{\boldsymbol{\pi}}_t\}_{t=1}^N$ with $\tilde{\boldsymbol{\pi}}_t \in \mathbb{R}^9$, IGGT4D (denoted as $\mathcal{F}_\theta$) performs feed-forward sequential

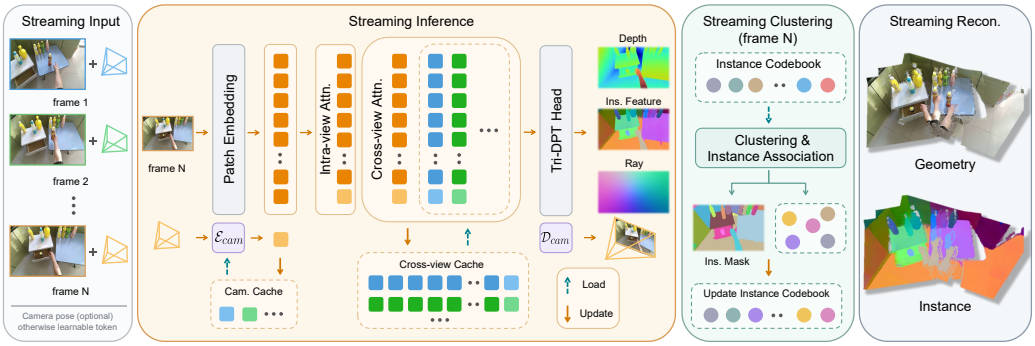

Figure 2: **Overview of IGGT4D.** Given a dynamic video sequence and optional camera poses, IGGT4D extracts a spatial-temporally consistent representation. A Tri-DPT head incrementally predicts geometry and instance features, and a streaming clustering algorithm derives instance masks for online 4D scene understanding.

inference for streaming 4D geometry and instance reconstruction:

$$\mathcal{F}_\theta : (I_t, \tilde{\boldsymbol{\pi}}_t) \mapsto (\boldsymbol{\pi}_t, R_t, D_t, S_t), \quad t = 1, \ldots, N. \tag{1}$$

For each frame, IGGT4D outputs camera parameters $\boldsymbol{\pi}_t = [\mathbf{t}_t, \mathbf{q}_t, \mathbf{f}_t] \in \mathbb{R}^9$, a ray map $R_t = [O_t, V_t] \in \mathbb{R}^{H_r \times W_r \times 6}$, a depth map $D_t \in \mathbb{R}^{H \times W}$, and an instance feature map $S_t \in \mathbb{R}^{H \times W \times 8}$. Here, $\mathbf{t}_t \in \mathbb{R}^3, \mathbf{q}_t \in \mathbb{R}^4, \mathbf{f}_t \in \mathbb{R}^2$ denote translation, rotation, and field-of-view, respectively, while $O_t \in \mathbb{R}^{H_r \times W_r \times 3}, V_t \in \mathbb{R}^{H_r \times W_r \times 3}$ represent ray origins and directions. As images are sequentially streamed in, frame-wise predictions are integrated into a spatial-temporal consistent scene representation, enabling online 4D reconstruction and understanding.

3.2 Streaming Instance-Grounded Geometric Transformer

Causal Geometry-Instance Transformer. We adapt the unified geometric Transformer of DA3 [18] from fixed-set 3D prediction to causal streaming 4D understanding. Each image is encoded into image tokens and concatenated with a camera token, obtained from $\mathcal{E}_{\text{cam}}$ when camera parameters are available or from a shared learnable token otherwise. The resulting frame-level tokens are processed by 40 Transformer blocks with interleaved intra-view and cross-view attention, producing multi-scale features $\{\mathbf{F}_t^{(l)}\}_{l=1}^4$. Unlike DA3's bidirectional fixed-view reasoning, IGGT4D imposes causal masks on cross-camera and cross-view attention, so each frame only attends to current and past observations. The same constraint is used during training to simulate streaming inference. At inference, the camera and cross-view caches store historical camera and image keys/values for cross-camera attention with optional camera priors and cross-view attention, respectively. Current-frame queries attend to current and cached keys/values, after which current keys/values are appended.

Tri-DPT Geometry-Instance Head. Our DPT-based [49] Tri-DPT head jointly decodes depth D_t, ray map R_t, and instance feature map S_t from the streaming multi-scale tokens $\{\mathbf{F}_t^{(l)}\}_{l=1}^4$. All branches progressively recover spatial resolution, while the instance branch is coupled with geometric branches through geometry-aware attention. By using depth and ray features as structural priors, the head grounds instance embeddings in 3D geometry and improves object identity consistency across time.

3.3 Efficient Streaming Instance Clustering

Streaming Clustering and the Instance Codebook are used only at inference time to convert predicted instance features into temporally consistent instance masks.

Offline clustering, such as HDBSCAN [50] used in IGGT [48], incurs prohibitive quadratic complexity on long sequences. To enable online inference, we introduce a two-stage streaming clustering strategy that maintains a lightweight global instance codebook $\mathcal{C}_t = \{(\mathbf{c}_k, a_k)\}_{k=1}^{K_t}$, where $\mathbf{c}_k$ is the feature center of instance k and a_k is its accumulated pixel count.

First, intra-frame clustering extracts local masks $M_{t,i}$ and centers $\mathbf{c}_{t,i}$. For an unassigned pixel p, we group pixels with high cosine similarity ($\alpha_q = \mathbf{s}_p^\top \mathbf{s}_q \geq \tau_s$) into a core region, then expand it via

connected components over a looser threshold (τ_l) to form $M_{t,i}$. Next, we match these local centers with the global codebook using the same cosine-similarity threshold τ_s. Unmatched instances become new global entries, while unmatched background regions are naturally ignored as distinct static entities, implicitly decoupling dynamic foreground from the static background. For matches, $M_{t,i}$ inherits the global ID k as a 4D-consistent mask $M_{t,k}$, and its center is updated via area-weighted fusion:

$$\mathbf{c}_k \leftarrow \text{norm}\left(\frac{a_k \mathbf{c}_k + |M_{t,k}|\mathbf{c}_{t,i}}{a_k + |M_{t,k}|}\right), \quad a_k \leftarrow a_k + |M_{t,k}|. \tag{2}$$

This constant-time center update translates 4D-consistent instance features into high-quality tracking masks (Fig. 3), providing a lightweight mechanism for maintaining instance consistency under object motion and partial occlusion, without explicit motion modeling.

3.4 4D Scene Understanding

The 4D-consistent masks $M_{t,k}$ and temporally linked instance features $\mathbf{c}_k$ provide a reusable object-level representation for downstream scene understanding tasks. First, the persistent instance correspondences across frames enable stable spatial tracking without additional association heuristics. Second, by using $M_{t,k}$ to aggregate per-frame 2D vision-language features [22, 24], we obtain spatially and temporally consistent language features $\mathbf{f}_{t,k}^{\text{lang}}$ for open-vocabulary semantic segmentation. Finally, these 4D-consistent masks and features can be provided to Large Multimodal Models (LMMs) [51, 52] to support 4D scene grounding, including tasks such as dynamic object segmentation and out-of-view object reasoning. Additional implementation details and qualitative results are provided in the supplementary material.

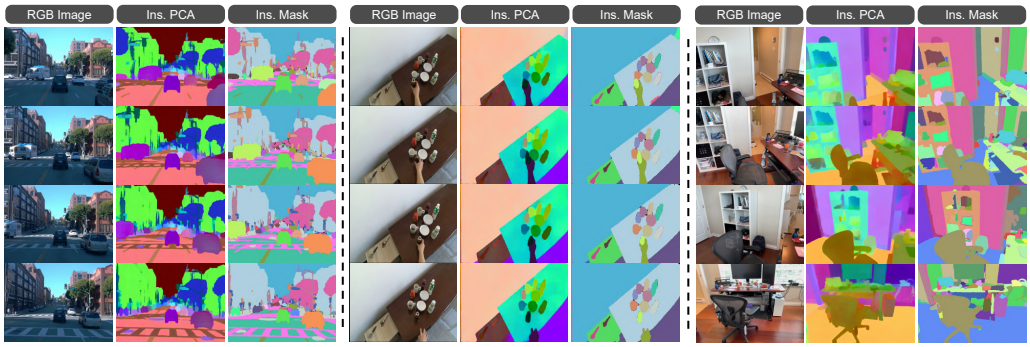

Figure 3: **Instance feature and mask visualization.** We visualize the 3D-consistent instance feature PCA results alongside the corresponding instance masks generated by our streaming clustering.

3.5 Training Objectives

First-frame Geometric Normalization. In streaming reconstruction, estimating scale from the entire sequence, as in offline settings, can introduce scale ambiguity. To avoid this ambiguity and promote spatial-temporal consistency, we apply first-frame-based geometric normalization during training: all poses are aligned to the first-frame camera coordinate system, and the first-frame point cloud $\mathcal{P}_1$ is used to compute a sequence-level scale $s = \frac{1}{|\mathcal{P}_1|}\sum_{\mathbf{p}\in\mathcal{P}_1}\|\mathbf{p}\|_2$, which is then applied to all geometric ground truth across the sequence.

Geometry Objectives. We supervise geometry with depth, ray, point, and camera losses: $\mathcal{L}_{\text{geo}} = \mathcal{L}_D + \mathcal{L}_R + \mathcal{L}_P + \mathcal{L}_\pi$. The depth loss $\mathcal{L}_D$ combines L1 regression, confidence C_t regularization, and spatial gradient $\mathcal{L}_{\text{grad}}$ consistency over valid pixels V:

$$\mathcal{L}_D = \frac{1}{|V|}\sum_{p\in V}\big(|D_t(p) - D_t^*(p)|\big(1 + \lambda_{\text{conf}} C_t(p)\big) - \lambda_{\text{conf}}\alpha \log C_t(p)\big) + \lambda_{\text{grad}}\mathcal{L}_{\text{grad}}, \tag{3}$$

where $\mathcal{L}_{\text{grad}} = \|\nabla D_t - \nabla D_t^*\|_1$ is an L1 gradient loss over spatial neighbors, V is the set of valid pixels, and α is a scaling factor. For the ray map $R_t = [O_t, V_t]$, we apply L1 loss to origins and directions: $\mathcal{L}_R = \|O_t - O_t^*\|_1 + \|V_t - V_t^*\|_1$. We further couple depth and rays by reconstructing 3D points $P_t = O_t + D_t V_t$ and minimizing $\mathcal{L}_P = \|P_t - P_t^*\|_1$. The camera loss is $\mathcal{L}_\pi = \|\boldsymbol{\pi}_t - \boldsymbol{\pi}_t^*\|_1$.

Instance Objective. Building on the contrastive supervision of IGGT [48], we optimize the L2-normalized instance features using a multi-view contrastive loss. Let M_k^v denote the ground-truth mask of instance k in view v. Its prototype μ_k^v is the L_2-normalized mean pixel feature within M_k^v. The objective enforces intra-view ($u = v$) constraints to group pixels $\mathbf{f}_p$ toward their instance center μ_k^v and repel different centers. It also applies cross-view ($u \neq v$) constraints to pull matching centers across frames and push different centers apart:

$$\begin{aligned}\mathcal{L}_{\text{ins}} = &\sum_v \Big(\lambda_{\text{pull}}^{\text{in}} \sum_k \sum_{p \in M_k^v} \left[\|\mathbf{f}_p - \mu_k^v\|_2 - \delta_{\text{pull}}^{\text{in}} \right]_+ + \lambda_{\text{push}}^{\text{in}} \sum_{k \neq j} \left[\delta_{\text{push}}^{\text{in}} - \|\mu_k^v - \mu_j^v\|_2 \right]_+ \Big) \\ &+ \sum_{u \neq v} \Big(\lambda_{\text{pull}}^{\text{cr}} \sum_k \left[\|\mu_k^u - \mu_k^v\|_2 - \delta_{\text{pull}}^{\text{cr}} \right]_+ + \lambda_{\text{push}}^{\text{cr}} \sum_{k \neq j} \left[\delta_{\text{push}}^{\text{cr}} - \|\mu_k^u - \mu_j^v\|_2 \right]_+ \Big),\end{aligned} \tag{4}$$

where $[\cdot]_+ = \max(0, \cdot)$, and δ are their respective margins. Averaging factors are omitted for brevity. The final objective is $\mathcal{L} = \mathcal{L}_{\text{geo}} + \lambda_{\text{ins}} \mathcal{L}_{\text{ins}}$. Training details and hyperparameter settings are provided in Appendix A.4.

4 InsScene4D-147K Dataset

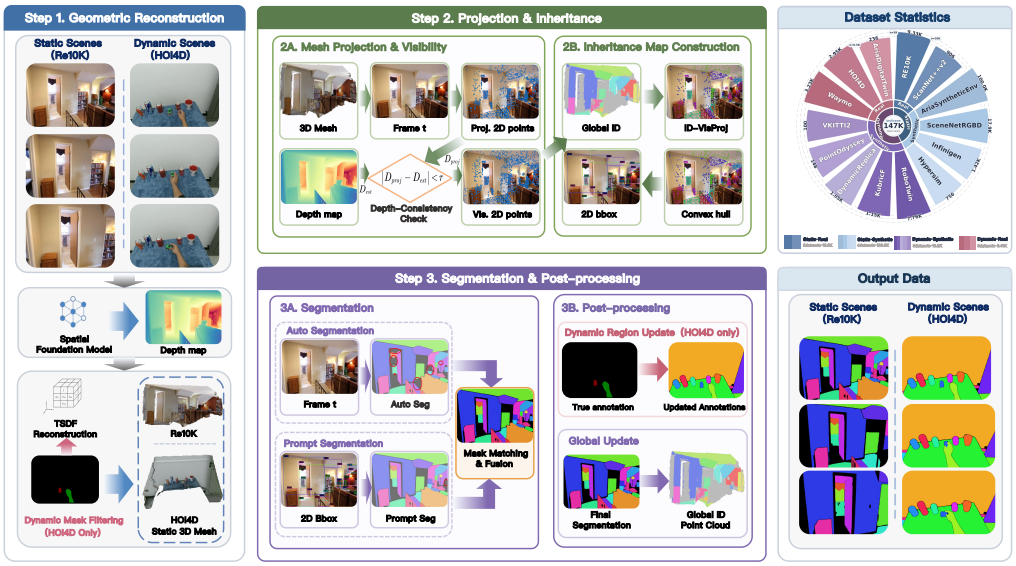

Figure 4: **InsScene4D-147K data curation pipeline.** Real/synthetic and static/dynamic sources are processed through 3D reconstruction, projection-based ID inheritance, and segmentation-based global ID refinement. The right panel summarizes the dataset splits.

We construct **InsScene4D-147K**, a large-scale dataset for online 4D scene understanding, comprising *147K* curated video sequences. The dataset spans four domains: static-real, static-synthetic, dynamic-real, and dynamic-synthetic. Specifically, the static-real split includes RealEstate10K [53] and ScanNet++ [54]; the static-synthetic split includes Aria Synthetic Environments [55], SceneNet [56], Infinigen [57], and Hypersim [58]; the dynamic-real split includes HOI4D [59], Waymo [60], and Aria Digital Twin [55]; and the dynamic-synthetic split includes RoboTwin 2.0 [61], Kubric [62], Dynamic Replica [63], PointOdyssey [64], and VKITTI2 [65]. Each sequence provides RGB images, depth maps, camera poses, point clouds, and 4D-consistent instance masks with persistent object identities across frames. Fig. 4 illustrates our geometry-guided annotation pipeline for producing temporally consistent instance supervision at scale.

For static captures, we design a curation pipeline that turns offline geometry into temporally consistent instance supervision. We first estimate multi-view consistent depth with DA3 [18], conditioned on offline ground-truth camera poses to avoid long-sequence pose drift. A static 3D mesh is then reconstructed via TSDF fusion [66], which aggregates depth predictions across views to reduce

Table 1: **Geometry benchmark on HiRoom, ETH3D, 7Scenes, and ScanNet++.** (a) Camera pose estimation (AUC@3, AUC@30). (b) 3D reconstruction (F1-score) without (w/o p.) and with (w/ p.) ground-truth camera poses. Methods are grouped into *full-attention* (offline) and *streaming* (online). Best and second-best results are highlighted in **bold** and underline, respectively.

Methods	Avg		HiRoom		ETH3D		7Scenes		ScanNet++	
(a) Camera Pose Estimation										
	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑
Full-attention (offline)										
VGGT	0.4001	0.8725	0.4927	0.8840	0.2895	0.8166	0.2293	0.8465	0.5890	0.9429
MapAnything	0.2436	0.8569	0.3069	0.8806	0.2347	0.8250	0.1683	0.8256	0.2646	0.8966
Pi3X	0.4683	0.9096	0.6662	0.9476	0.3453	0.8749	0.2526	0.8607	0.6090	0.9552
DA3	**0.6046**	**0.9352**	**0.8465**	**0.9735**	**0.4541**	**0.9184**	**0.2780**	**0.8671**	**0.8397**	**0.9818**
Streaming (online)										
CUT3R	0.1022	0.6977	0.1752	0.7438	0.1187	0.6611	0.0654	0.7177	0.0494	0.6683
StreamVGGT	0.1606	0.7579	0.1686	0.7330	0.1300	0.6866	0.1354	0.8136	0.2085	0.7985
WinT3R	0.1133	0.6937	0.2522	0.7998	0.1219	0.6693	0.0251	0.6929	0.0541	0.6129
STream3R	0.1871	0.7947	0.2510	0.8183	0.1621	0.7143	0.1313	0.8209	0.2041	0.8252
LingBot-Map	0.3060	0.8647	0.3388	0.8727	**0.2796**	**0.8417**	0.1636	0.8257	0.4419	0.9188
Ours	**0.4464**	**0.8924**	**0.7223**	**0.9533**	0.2734	0.8242	**0.2435**	**0.8534**	**0.5467**	**0.9389**
(b) 3D Reconstruction										
	F1(w/o p.)↑	F1(w/ p.)↑	F1(w/o p.)↑	F1(w/ p.)↑	F1(w/o p.)↑	F1(w/ p.)↑	F1(w/o p.)↑	F1(w/ p.)↑	F1(w/o p.)↑	F1(w/ p.)↑
Full-attention (offline)										
VGGT	0.5775	0.6477	0.5549	0.6676	0.6249	0.7198	0.4640	0.5048	0.6662	0.6985
MapAnything	0.4526	0.6407	0.4726	0.5904	0.5722	0.7596	0.4647	0.5016	0.3007	0.7112
Pi3X	0.6740	0.7756	0.7780	0.8969	0.7014	0.8331	0.4742	**0.5740**	0.7424	0.7986
DA3	**0.7430**	**0.7980**	**0.8819**	**0.9563**	**0.8098**	**0.8689**	**0.5005**	0.5653	**0.7796**	**0.8015**
Streaming (online)										
CUT3R	0.3644	0.4060	0.2795	0.1993	0.5941	0.6467	0.3221	0.3711	0.2621	0.4071
StreamVGGT	0.2703	0.3377	0.1162	0.2179	0.3730	0.4275	0.3119	0.3567	0.2800	0.3486
WinT3R	0.4260	0.4550	0.4853	0.2969	0.6167	0.7447	0.3090	0.3901	0.2929	0.3885
STream3R	0.4551	0.5504	0.4359	0.6435	0.5432	0.6400	0.4426	0.4739	0.3986	0.4443
LingBot-Map	0.5228	0.6082	0.4749	0.5973	0.6615	0.7394	0.3794	0.4726	0.5756	0.6236
Ours	**0.6678**	**0.7225**	**0.8015**	**0.8501**	**0.7186**	**0.7724**	**0.4975**	**0.5441**	**0.6535**	**0.7236**

estimation variance and suppress outlier noise. This process yields high-quality, multi-view consistent geometric pseudo-labels for subsequent instance annotation.

For each frame, we project mesh vertices onto the image plane, retain visible vertices by depth consistency, and reverse-map their global IDs to form an *inheritance map*. SAM2 [67] provides category-agnostic masks, which are filtered, de-overlapped, and matched to inherited regions by IoU. Confident matches inherit existing IDs, ambiguous matches use the best-overlap ID, and unmatched masks are initialized as new instances. To prevent stale ID propagation, an object is marked as disappeared if its projected area drops sharply for $N{=}5$ consecutive frames.

For HOI4D, DA3 estimates depth and poses. Official frame-wise 2D dynamic-instance masks exclude annotated dynamic pixels from TSDF fusion; their labels override SAM2 predictions within annotated regions during final label generation. For simulation data, the source datasets provide RGB-D sequences with instance-consistent masks, which we incorporate into the synthetic split.

5 Experiments

We evaluate our method across multiple tasks against a broad range of state-of-the-art baselines. All inference evaluations are conducted on a single NVIDIA RTX 5090 GPU with 32 GB of memory. Training configurations are provided in Appendix A.4.

5.1 Evaluation of Camera Pose Estimation and 3D Reconstruction

For static scenes, following the evaluation protocol of DA3 [18], we evaluate on HiRoom [68], ETH3D [69], 7Scenes [70], and ScanNet++ [54]. These evaluation sequences are held out from training at the scene/sequence level. For dynamic scenes, we evaluate on Waymo [60], PointOdyssey [64], Sintel [71], Bonn RGB-D Dynamic [72], and TUM-Dynamic [73].

We compare with offline full-attention models (VGGT [15], MapAnything [16], Pi3X [17], DA3) and online streaming models (CUT3R [20], StreamVGGT [43], WinT3R [74], STream3R [21], LingBot-Map [75]). We report AUC@3/AUC@30 for camera pose accuracy on both static and dynamic scenes. For static scenes, we use TSDF fusion [66] following DA3 and report 3D reconstruction F1-scores; for dynamic scenes, we evaluate depth using $\delta_{1.25}$ and AbsRel.

Table 2: **Geometry evaluation on dynamic scenes under streaming inference.** Pose averages exclude Bonn RGB-D Dynamic. Best and second-best results are in **bold** and underline, respectively.

Method	Avg.		Sintel		Bonn		TUM-Dynamic		Waymo		PointOdyssey	
(a) Camera Pose Estimation												
	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑
CUT3R	0.0892	0.5087	0.0255	0.4094	–	–	0.0368	0.3694	0.2734	0.8138	0.0209	0.4422
StreamVGGT	0.1626	0.6283	0.0625	0.4699	–	–	0.0724	0.5488	0.3046	0.7854	0.2108	0.7092
WinT3R	0.0605	0.4446	0.0059	0.2047	–	–	0.0068	0.2901	0.1884	0.7070	0.0410	0.5766
STream3R	0.2301	0.6538	0.0923	**0.5487**	–	–	0.0685	0.4861	0.4626	0.8432	0.2969	0.7372
LingBot-Map	0.2556	0.6613	**0.1043**	0.5312	–	–	**0.1306**	0.5831	0.6221	0.8792	0.1653	0.6515
Ours	**0.3346**	**0.7261**	0.0994	0.5403	–	–	0.0967	**0.6787**	**0.6928**	**0.8893**	**0.4495**	**0.7960**
(b) Depth Estimation												
	$\delta_{1.25}$ ↑	AbsRel↓	$\delta_{1.25}$ ↑	AbsRel↓	$\delta_{1.25}$ ↑	AbsRel↓	$\delta_{1.25}$ ↑	AbsRel↓	$\delta_{1.25}$ ↑	AbsRel↓	$\delta_{1.25}$ ↑	AbsRel↓
CUT3R	0.8201	0.2316	0.4979	0.7438	0.9496	0.0828	0.8632	0.1092	0.8682	0.1172	0.9217	0.1050
StreamVGGT	0.8313	0.1396	0.5728	0.2917	**0.9733**	0.0610	0.9206	0.1065	0.7097	0.1845	0.9799	0.0541
WinT3R	0.8209	0.1500	0.5435	0.3466	0.9063	0.0739	0.8005	0.1331	0.9104	0.1188	0.9438	0.0775
STream3R	0.8891	0.1072	**0.6177**	0.2665	0.9485	0.0695	0.9379	0.0940	0.9494	0.0784	**0.9919**	0.0276
LingBot-Map	0.8787	0.1166	0.5949	0.2737	0.9584	0.0617	0.9379	**0.0902**	0.9378	0.0915	0.9645	0.0657
Ours	**0.8919**	**0.1017**	0.6100	**0.2631**	0.9712	**0.0562**	**0.9395**	0.1028	**0.9516**	**0.0629**	0.9874	**0.0235**

Camera Pose Estimation. Tab. 1(a) and Tab. 2(a) show that our method achieves the best average AUC@3 and AUC@30 among streaming models on both static and dynamic benchmarks, demonstrating reliable camera motion estimation from sequential inputs. On static benchmarks, while full-attention models such as DA3 still benefit from bidirectional reasoning over the entire sequence, our method substantially reduces the gap under a causal streaming setting and even outperforms several full-attention baselines. We omit pose AUC on Bonn because its near-static cameras and small translation baselines make translation-direction angles sensitive to small errors.

3D Reconstruction and Depth Estimation. Tab. 1(b) reports static-scene 3D reconstruction F1-scores with predicted poses (w/o p.) and ground-truth poses (w/ p.). MapAnything, Pi3X, DA3, and our method accept ground-truth poses during inference, while the other methods use them only for evaluation-time fusion. In both settings, our method outperforms all streaming baselines, showing strong 3D consistency under sequential inference. With ground-truth poses, it further improves reconstruction quality, suggesting that the model can effectively exploit streaming camera inputs. Fig. 5 shows qualitative comparisons with LingBot-Map and STream3R using TSDF-fused point clouds, with invalid ground-truth depth regions masked out. On dynamic scenes, our method also achieves the highest average $\delta_{1.25}$ and lowest average AbsRel among streaming models (Tab. 2(b)). Full-attention results on dynamic scenes are reported in Appendix A.6.

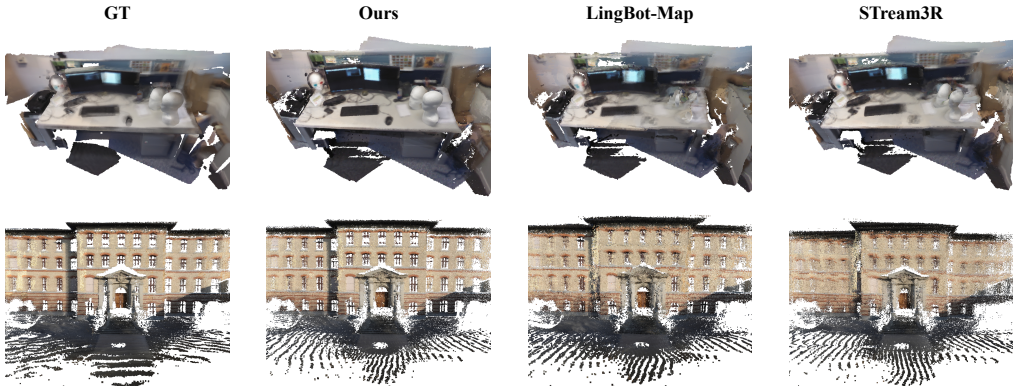

Figure 5: **Qualitative comparison of 3D reconstruction on the 7Scenes and ETH3D datasets.**

5.2 Evaluation of Instance Spatial Tracking

For instance spatial tracking and open-vocabulary semantic segmentation, all evaluation sequences are sampled from held-out splits that are disjoint from the training data at the scene or sequence level. Our evaluation is conducted on three dynamic datasets (HOI4D [59], Waymo [60], and PointOdyssey [64]) and one static dataset (ScanNet++). Following the protocol in IGGT [48], we compare against SpaTrackerV2 [76]+SAM [25], SAM2 [67], and IGGT itself, and report Temporal mIoU (T-mIoU) and Temporal Success Rate (T-SR). We evaluate the methods under both long- and short-sequence settings. For long sequences, all datasets consist of 100 frames, except for

Table 3: **Instance spatial tracking on HOI4D, Waymo, ScanNet++, and PointOdyssey.** Evaluated under two settings: (a) long sequences and (b) short sequences. Best and second-best results are highlighted in **bold** and underline, respectively.

Methods	Avg		HOI4D		Waymo		ScanNet++		PointOdyssey	
(a) Long sequences (∼100 frames, HOI4D: 40 frames)										
	T-mIoU↑	T-SR↑	T-mIoU↑	T-SR↑	T-mIoU↑	T-SR↑	T-mIoU↑	T-SR↑	T-mIoU↑	T-SR↑
SpaTrackerV2+SAM	43.73	59.28	64.40	74.89	42.13	53.04	21.86	39.49	46.52	69.68
SAM2	45.38	58.65	65.41	78.34	47.62	56.93	12.88	23.24	55.62	76.07
IGGT	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
Ours	**58.84**	**85.41**	**78.44**	**95.60**	**59.35**	**88.04**	**33.19**	**61.35**	**64.36**	**96.65**
(b) Short sequences (∼16 frames)										
	T-mIoU↑	T-SR↑	T-mIoU↑	T-SR↑	T-mIoU↑	T-SR↑	T-mIoU↑	T-SR↑	T-mIoU↑	T-SR↑
SpaTrackerV2+SAM	46.84	63.19	61.94	71.20	47.57	57.92	33.73	59.47	44.12	64.17
SAM2	51.75	66.03	65.87	78.11	60.65	71.37	21.25	35.08	59.21	79.57
IGGT	51.68	84.65	63.02	87.14	49.59	88.56	32.97	**68.07**	61.13	94.83
Ours	**59.25**	**86.28**	**78.27**	**95.42**	**61.08**	**89.34**	**34.18**	63.47	**63.45**	**96.88**



Figure 6: **Qualitative visualization results of instance spatial tracking.**

HOI4D, which uses 40 frames. As shown in Tab. 3, IGGT suffers from out-of-memory (OOM) issues when processing long sequences. In contrast, benefiting from the proposed streaming inference and clustering mechanisms, our method can scale to 4D long sequences. We present the qualitative visualization results of Instance Spatial Tracking in Fig. 6.

5.3 Evaluation of Open-Vocabulary Semantic Segmentation

We evaluate on Waymo and ScanNet++. Baselines include 2D vision-language models (OpenSeg [24], LSeg [23]) and 3D semantic reconstruction methods (per-scene Feature-3DGS [30] and feed-forward IGGT [48]). For semantic label assignment, we average pixel-wise OpenSeg or LSeg features over mask regions sharing a global instance ID across views, match the pooled feature to candidate-category text embeddings, and assign the best-matching label to all masks with that ID.

As shown in Tab. 4, our method achieves the highest mIoU and mAcc, demonstrating stronger semantic segmentation performance under both long- and short-sequence settings in both dynamic outdoor and complex static indoor scenes. Fig. 7 shows qualitative visualization results.

5.4 Ablation Study and Streaming Clustering Efficiency

We conduct the ablation study on ScanNet++ using DA3-Large, as shown in Tab. 5. Removing Geometry-Aware Attention (Geo-Attn) degrades instance and semantic performance, demonstrating that instance features benefit from geometric priors. Furthermore, without First-Frame Geometric Normalization (FF-Norm), severe scale ambiguity arises during streaming reconstruction. This disrupts the unified geometry-instance representation, leading to a decline across all metrics. Additional

Table 4: **Open-vocabulary semantic segmentation on Waymo and ScanNet++.** Evaluated under two settings: (a) long sequences and (b) short sequences. Best and second-best results are highlighted in **bold** and underline, respectively.

Methods	(a) Long sequences (∼100 frames)						(b) Short sequences (∼16 frames)					
	Avg		Waymo		ScanNet++		Avg		Waymo		ScanNet++	
	mIoU↑	mAcc↑	mIoU↑	mAcc↑	mIoU↑	mAcc↑	mIoU↑	mAcc↑	mIoU↑	mAcc↑	mIoU↑	mAcc↑
OpenSeg	19.16	45.77	15.14	44.10	23.17	47.43	19.41	46.80	15.03	43.87	23.78	49.73
LSeg	31.95	48.23	35.52	56.55	28.37	39.90	32.06	50.54	34.82	57.04	29.29	44.03
Feature-3DGS	31.94	48.15	35.51	56.49	28.37	39.81	26.31	44.89	34.81	56.88	17.80	32.90
IGGT	OOM	OOM	OOM	OOM	OOM	OOM	31.16	61.90	29.44	62.42	32.88	61.38
Ours	**40.08**	**73.84**	**41.63**	**78.07**	**38.52**	**69.61**	**39.11**	**72.25**	**40.68**	**75.51**	**37.53**	**68.98**

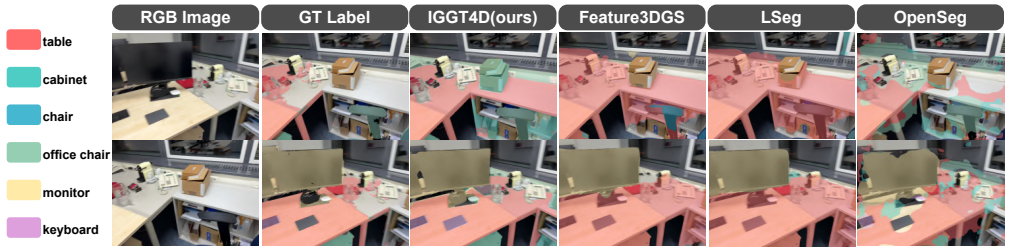

Figure 7: **Qualitative visualization results of open-vocabulary semantic segmentation.**

ablations on training strategy, instance feature dimensionality, and clustering thresholds are reported in Appendices A.7 and A.8.

Streaming Clustering Efficiency. Tab. 6 compares the time and GPU memory consumption of our streaming clustering against the offline HDBSCAN [50] used in IGGT at a resolution of 504×336. Over the evaluated sequence lengths, the clustering module uses approximately 0.7 GB of GPU memory and its runtime grows approximately linearly.

Table 5: **Ablation study on ScanNet++ using DA3-Large.** Geo-Attn: Geometry-Aware Attention. FF-Norm: First-Frame Geometric Normalization.

Model	Geometry		Instance		Semantic	
	AUC@3 ↑	F1-score ↑	mIoU ↑	Match Rate ↑	mIoU ↑	mAcc ↑
Full model	0.2984	0.4790	0.3098	0.6271	0.3666	0.6853
w/o Geo-Attn	0.2902	0.4778	0.2970	0.6074	0.3559	0.6796
w/o FF-Norm	0.2334	0.3542	0.2873	0.6019	0.3480	0.6768

Table 6: **Clustering efficiency comparison.** Ours: Streaming clustering. IGGT: HDBSCAN. N denotes the number of frames. Measurements cover the clustering module only.

Method	$N = 8$		$N = 16$		$N = 32$		$N = 100$	
	Time (s)	Mem (GB)	Time (s)	Mem (GB)	Time (s)	Mem (GB)	Time (s)	Mem (GB)
Ours	0.78	0.7	1.60	0.7	2.98	0.7	7.43	0.7
IGGT	105.8	13.2	418.3	25.85	OOM	OOM	OOM	OOM

6 Conclusion

We presented IGGT4D, a streaming geometry-instance framework for online 4D scene understanding. IGGT4D jointly predicts camera motion, scene geometry, and temporally consistent instance features via causal spatial-temporal modeling and geometry-grounded instance prediction. We also introduced InsScene4D-147K, a large-scale dataset with geometry-consistent instance annotations across real/synthetic and static/dynamic scenes. Experiments on reconstruction, pose estimation, instance tracking, and open-vocabulary segmentation show that IGGT4D improves object-level consistency while supporting streaming inference.

Limitations. InsScene4D-147K scales 4D instance supervision, but broader data coverage is needed for stronger robustness and generalization. IGGT4D remains mainly supervised, motivating future work on self-supervised learning and larger curated supervision. Our evaluations are also perception-centric; integrating embodied actions and interaction feedback is a key step toward embodied scene understanding. IGGT4D does not explicitly model object motion or non-rigid deformation; "4D" here refers to streaming reconstruction of dynamic scenes with temporally consistent instance identities. Full-history KV-cache memory grows approximately linearly with sequence length, and per-frame computation also increases, preventing unbounded inference under a fixed resource budget.

Acknowledgments and Disclosure of Funding

We thank the research community for openly sharing the datasets, models, and prior work that made this study possible. This work was supported by the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center Project under Grant (24KT004). The authors declare no financial conflicts of interest.

References

[1] Diqi He, Xuehao Gao, Hao Li, Junwei Han, and Dingwen Zhang. Strider: Navigation via instruction-aligned structural decision space optimization. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 26700–26720. Curran Associates, Inc., 2025.

[2] Kailing Li, Tianwen Qian, Lijin Yang, Yuqian Fu, Jingyu Gong, Xiaoling Wang, and Liang He. Bridging the 2d-3d gap: A hierarchical semantic-geometric map for vision language navigation. *arXiv preprint arXiv:2606.00095*, 2026.

[3] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[4] Yufei Guo, Yinan Wu, Haoran Duan, Guiguang Ding, and Jungong Han. Phaselora: Control-regime-conditioned low-rank adaptation for continuous-action vision-language-action policies. *arXiv preprint arXiv:2608.15285*, 2026.

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

[6] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European conference on computer vision*, pages 501–518. Springer, 2016.

[7] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

[8] Tingyang Xiao, Xiaolin Zhou, Liu Liu, Wei Sui, Wei Feng, Jiaxiong Qiu, Xinjie Wang, and Zhizhong Su. Geoflow-slam: A robust tightly-coupled rgbd-inertial and legged odometry fusion slam for dynamic legged robotics, 2025.

[9] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

[10] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022.

[11] Tingyang Xiao, Liu Liu, Wei Feng, Zhengyu Zou, Xiaolin Zhou, Wei Sui, Hao Li, Dingwen Zhang, and Zhizhong Su. Iris-slam: Unified geo-instance representations for robust semantic localization and mapping, 2026.

[12] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

[13] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.

[14] Tomas Berriel Martins, Martin R Oswald, and Javier Civera. Open-vocabulary online semantic mapping for slam. *IEEE Robotics and Automation Letters*, 2025.

[15] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[16] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.

[17] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.

[18] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

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

[20] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025.

[21] Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. Stream3r: Scalable sequential 3d reconstruction with causal transformer. *arXiv preprint arXiv:2508.10893*, 2025.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[23] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.

[24] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European conference on computer vision*, pages 540–557. Springer, 2022.

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[26] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19729–19739, 2023.

[27] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–824, 2023.

[28] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631*, 2023.

[29] Xiangyu Sun, Haoyi Jiang, Liu Liu, Seungtae Nam, Gyeongjin Kang, Xinjie Wang, Wei Sui, Zhizhong Su, Wenyu Liu, Xinggang Wang, et al. Uni3r: Unified 3d reconstruction and semantic understanding via generalizable gaussian splatting from unposed multi-view images. *arXiv preprint arXiv:2508.03643*, 2025.

[30] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.

[31] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

[32] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024.

[33] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction, 2025.

[34] Haoyi Jiang, Liu Liu, Xinjie Wang, Yonghao He, Wei Sui, Zhizhong Su, Wenyu Liu, and Xinggang Wang. Spa3r: Predictive spatial field modeling for 3d visual reasoning, 2026.

[35] Yuanyuan Gao, Hao Li, Yifei Liu, Xinhao Ji, Yuning Gong, Yuanjun Liao, Fangfu Liu, Manyuan Zhang, Yuchen Yang, Dan Xu, et al. Holi-spatial: Evolving video streams into holistic 3d spatial intelligence. *arXiv preprint arXiv:2603.07660*, 2026.

[36] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024.

[37] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

[38] Haosong Peng, Hao Li, Yalun Dai, Yushi Lan, Yihang Luo, Tianyu Qi, Zhengshen Zhang, Yufeng Zhan, Junfei Zhang, Wenchao Xu, and Ziwei Liu. Omnivggt: Omni-modality driven visual geometry grounded transformer. *arXiv preprint arXiv:2511.10560*, 2025.

[39] Xianze Fang, Jingnan Gao, Zhe Wang, Zhuo Chen, Xingyu Ren, Jiangjing Lyu, Qiao-Mu Ren, Zhonglei Yang, Xiaokang Yang, Yichao Yan, et al. Dens3r: A foundation model for 3d geometry prediction. In *International Conference on Learning Representations*, volume 2026, pages 80604–80634, 2026.

[40] Jingnan Gao, Zhe Wang, Xianze Fang, Xingyu Ren, Zhuo Chen, Shengqi Liu, Yuhao Cheng, Jiangjing Lyu, Xiaokang Yang, and Yichao Yan. More: 3d visual geometry reconstruction meets mixture-of-experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14680–14691, 2026.

[41] Xiangyu Sun, Liu Liu, Seungkwon Yang, Jingbing Han, Seungtae Nam, Zhizhong Su, and Eunbyung Park. Nodrift3r: Raymap-guided coupling for drift-robust unposed feed-forward 3d reconstruction. In *European Conference on Computer Vision*, pages 440–457. Springer, 2026.

[42] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1050–1060, 2025.

[43] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025.

[44] Juntong Fang, Zequn Chen, Weiqi Zhang, Donglin Di, Xuancheng Zhang, Chengmin Yang, and Yu-Shen Liu. More: Motion-aware feed-forward 4d reconstruction transformer. *arXiv preprint arXiv:2603.05078*, 2026.

[45] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[46] Hao Li, Minghan Qin, Zhengyu Zou, Diqi He, Xinhao Ji, Bohan Li, Bingquan Dai, Dingewn Zhang, and Junwei Han. Langsurf: Language-embedded surface gaussians for 3d scene understanding, 2026.

[47] Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, et al. Large spatial model: End-to-end unposed images to semantic 3d. *Advances in neural information processing systems*, 37:40212–40229, 2024.

[48] Hao Li, Zhengyu Zou, Fangfu Liu, Xuanyang Zhang, Fangzhou Hong, Yukang Cao, Yushi Lan, Manyuan Zhang, Gang Yu, Dingwen Zhang, et al. Iggt: Instance-grounded geometry transformer for semantic 3d reconstruction. *arXiv preprint arXiv:2510.22706*, 2025.

[49] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[50] Leland McInnes, John Healy, Steve Astels, et al. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.

[51] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[52] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[53] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[54] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023.

[55] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20133–20143, 2023.

[56] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *Proceedings of the IEEE international conference on computer vision*, pages 2678–2687, 2017.

[57] Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, et al. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21783–21794, 2024.

[58] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021.

[59] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022.

[60] Jieru Mei, Alex Zihao Zhu, Xinchen Yan, Hang Yan, Siyuan Qiao, Liang-Chieh Chen, and Henrik Kretzschmar. Waymo open dataset: Panoramic video panoptic segmentation. In *European Conference on Computer Vision*, pages 53–72. Springer, 2022.

[61] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[62] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022.

[63] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023.

[64] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023.

[65] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020.

[66] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[67] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[68] SVLightVerse: Large-scale photorealistic indoor dataset with spatially-varying hdri lighting. `https://jerrypiglet.github.io/SVLightVerse/`, 2025. Project page.

[69] 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*, pages 3260–3269, 2017.

[70] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013.

[71] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012.

[72] 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)*, pages 7855–7862. IEEE, 2019.

[73] 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*, pages 573–580. IEEE, 2012.

[74] 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. *arXiv preprint arXiv:2509.05296*, 2025.

[75] 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*, 2026.

[76] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. *arXiv preprint arXiv:2507.12462*, 2025.

[77] 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.

[78] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1316–1326. IEEE, 2023.

[79] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3151–3161. IEEE, 2024.

[80] Shuangrui Ding, Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Yuwei Guo, Dahua Lin, and Jiaqi Wang. Sam2long: Enhancing sam 2 for long video segmentation with a training-free memory tree. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13614–13624. IEEE, 2025.

[81] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2432–2443. IEEE, 2017.

[82] Shuai Yuan, Yantai Yang, Xiaotian Yang, Xupeng Zhang, Zhonghao Zhao, Lingming Zhang, and Zhipeng Zhang. Infinitevggt: Visual geometry grounded transformer for endless streams. *arXiv preprint arXiv:2601.02281*, 2026.

A Technical Appendices and Supplementary Material

A.1 Tri-DPT Geometry-Instance Head.

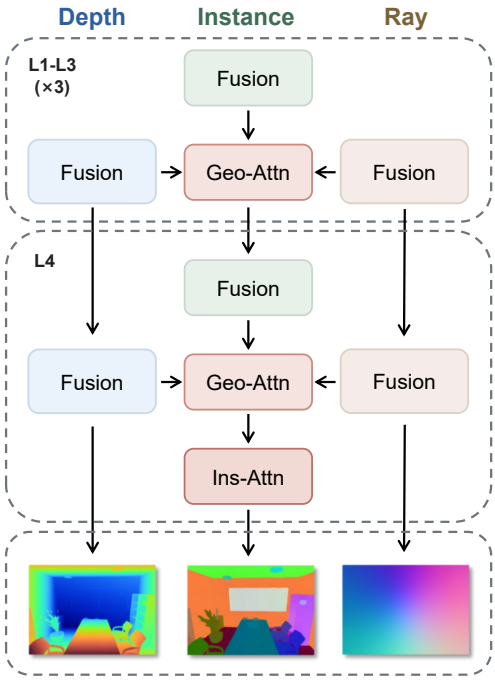

Figure 8: **Tri-DPT head.** The head jointly decodes depth, ray, and instance features, with geometry-aware attention injecting depth and ray priors into the instance branch.

We design a Tri-DPT head to jointly decode geometry and instance representations from streaming multi-scale tokens $\{\mathbf{F}_t^{(l)}\}_{l=1}^4$. Rather than treating instance prediction as an independent add-on, Tri-DPT uses a unified DPT-style decoder [49] with three coordinated output branches for the depth map D_t, ray map R_t, and instance feature map S_t. The multi-scale tokens are reassembled and fused progressively to recover spatial resolution, allowing dense geometric predictions and instance embeddings to be produced in the same spatially aligned representation.

As shown in Fig. 8, instead of simply appending an independent instance head, the instance branch explicitly leverages multi-scale geometric features from the depth and ray branches. At the four fusion stages, depth features and ray features are fused as geometric priors and injected into the instance branch through geometry-aware attention. At the final stage, instance attention is further introduced to refine the instance features. As a result, the predicted instance features are not only driven by appearance cues, but are also tightly coupled with the underlying geometric structure. This design allows instance features to benefit from spatial-temporally consistent geometric constraints even under frame-by-frame streaming inputs, enabling stable 4D scene-level instance awareness.

A.2 Explanation of First-Frame Geometric Normalization

In our streaming training framework, we utilize the first-frame point cloud to normalize the geometric ground truth for the entire sequence. This design is crucial for preventing scale ambiguity. Consider an extreme scenario where global sequence-level normalization is applied instead. Suppose the model processes two different sequences that share an identical first frame. In the first sequence, the second frame captures a close-up of the ground, whereas in the second sequence, it captures an expansive open plaza. Under global normalization, the overall geometric scales of these two sequences would be drastically different.

Consequently, when the streaming model processes the identical first frame in both cases, it is forced to regress entirely different scale values. From the model's perspective, observing only the first

frame provides insufficient information to correctly infer the global scale of the yet-unseen future frames. This fundamentally leads to scale ambiguity and optimization conflicts. By anchoring the geometric normalization strictly to the first frame, we ensure that the scale is uniquely and consistently determined by the initial observation, completely eliminating this ambiguity and ensuring stable causal streaming training. Fig. 9 compares streaming predictions *without* versus *with* first-frame geometric normalization (FF-Norm).

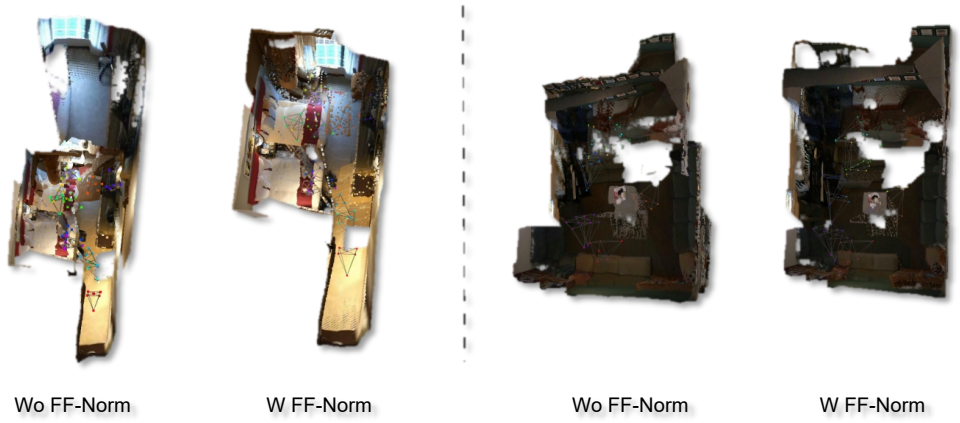

Figure 9: **First-frame geometric normalization (FF-Norm).** We compare streaming predictions *without* versus *with* first-frame geometric normalization.

A.3 Visualization of the Automated Geometry-Guided Annotation Pipeline

To demonstrate the effectiveness of our proposed automated geometry-guided annotation pipeline, Fig. 10 visualizes the generated annotations across the three diverse data sources. Furthermore, for the RealEstate10K (Re10K) scenes, we provide a qualitative comparison between our refined annotations and the prior vanilla annotations from InsScene-15K [48].

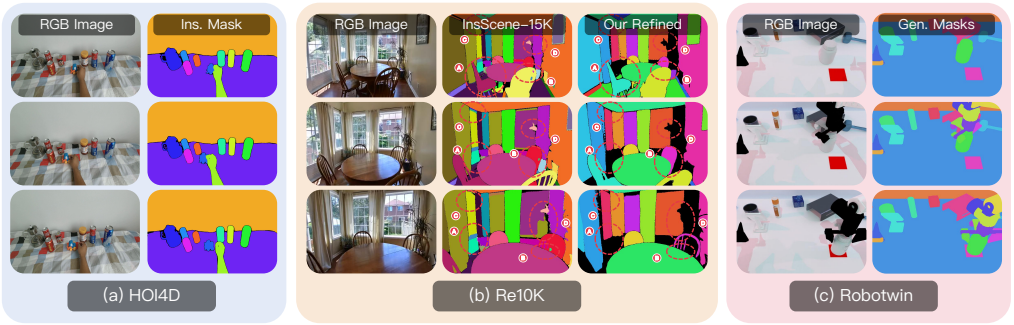

Figure 10: Qualitative evaluation of the automated geometry-guided annotation pipeline across three distinct sources (HOI4D [59], Re10K [53], and RoboTwin 2.0 [61]). For the Re10K scenes, the comparison illustrates how our approach overcomes the common failure modes observed in the vanilla InsScene-15K annotations: (a) over-segmentation, (b) false-positive segmentation, (c) identity switching, and (d) background leakage.

A.4 Training Details

Our model is initialized from the DA3-Giant [18] architecture. We train IGGT4D on our constructed **InsScene4D-147K** dataset using 16 NVIDIA H20 GPUs. The training process is divided into two stages: the first 6 epochs focus exclusively on optimizing the streaming geometry, while the subsequent 9 epochs jointly optimize the geometry and instance branches. During training, the

input image resolution varies between 504×504 and 504×280. To simulate streaming sequences of varying lengths, each training iteration contains a fixed total of 24 frames. The batch size is dynamically selected from $B \in \{1, 2, 3, 4, 6, 8, 12\}$, with the corresponding sequence length given by $L = 24/B$. The maximum training sequence length is therefore 24 frames. The initial learning rate is set to 2×10^{-4} for the instance branch and 1×10^{-5} for the geometry. Both learning rates are decayed following a cosine annealing schedule. Additionally, we apply a 20% probability of injecting ground-truth camera poses during the training. For all ablation studies, we adopt the DA3-Large architecture and train the models using 16 NVIDIA RTX 5090 GPUs.

Loss hyperparameters. For geometry supervision, we set $\lambda_{\text{conf}} = 1$, $\lambda_{\text{grad}} = 1$, and $\alpha = 0.2$. The overall instance loss weight is $\lambda_{\text{ins}} = 0.4$. For the instance contrastive loss, we use $\lambda^{\text{in}}_{\text{pull}} = \lambda^{\text{in}}_{\text{push}} = \lambda^{\text{cr}}_{\text{push}} = 1$ and $\lambda^{\text{cr}}_{\text{pull}} = 2$. The corresponding margins are $\delta^{\text{in}}_{\text{pull}} = 0.15$, $\delta^{\text{cr}}_{\text{pull}} = 0.30$, $\delta^{\text{in}}_{\text{push}} = 1.10$, and $\delta^{\text{cr}}_{\text{push}} = 1.00$. These margins apply to the Euclidean distances in the instance objective, computed using L_2-normalized pixel features and their instance prototypes.

Inference hyperparameters. For Streaming Clustering, we set the cosine-similarity thresholds to $\tau_s = 0.95$ for core-region formation and global codebook matching, and $\tau_l = 0.90$ for within-view region expansion.

A.5 Experiments Details

For *instance spatial tracking*, we compare with SpaTrackerV2 [76]+SAM [25], SAM2 [67], and IGGT [48]. Following the protocol in IGGT, we integrate SAM into SpaTrackerV2 by utilizing tracking points as prompts for dense segmentation, and modify SAM2 to support multi-view dense segmentation and tracking. We use HOI4D [59], Waymo [60], ScanNet++ [54], and PointOdyssey [64]. HOI4D contains about 40 frames per scene and is manually annotated by us. Waymo and ScanNet++ each contain about 100 frames per scene; we use the official instance annotations and manually remove several small objects. For PointOdyssey, we use scenes with about 100 frames from the official test set, manually select a subset of masks, and merge them when needed.

For *camera pose estimation and 3D reconstruction*, we compare our method with offline full-attention models (VGGT [15], MapAnything [16], Pi3X [17], DA3 [18]) and online streaming models (CUT3R [20], StreamVGGT [43], WinT3R [74], STream3R [21], LingBot-Map [75]). Among them, MapAnything, Pi3X, DA3, and our method support the injection of ground-truth camera poses. Among the streaming methods, our method, CUT3R, StreamVGGT, and STream3R perform frame-by-frame inference; WinT3R utilizes overlapping sliding windows with a window size of 4 and a stride of 2; and LingBot-Map initially applies full attention to the first 8 frames, subsequently reconstructing the remaining frames in a streaming manner.

Our evaluation datasets (HiRoom [68], ETH3D [69], 7Scenes [70], and ScanNet++ [54]) and protocol follow the DA3 benchmark. To enable streaming inference, we reorder each sequence to ensure that every frame shares visual overlap with at least one preceding frame. We adopt the robust evaluation pipeline from DA3: predicted poses are aligned to the ground truth using RANSAC-based `evo` alignment [77], and the aligned poses alongside predicted depths are integrated via TSDF fusion to assess 3D consistency. For camera pose estimation, we report the Area Under the Curve (AUC) at error thresholds of 3° (AUC@3) and 30° (AUC@30), where accuracy is determined by the minimum of the Relative Rotation Accuracy (RRA) and Relative Translation Accuracy (RTA). For 3D reconstruction, we evaluate the Chamfer Distance (CD) and F1-score. Let $\mathcal{P}_{\text{pred}}$ and $\mathcal{P}_{\text{gt}}$ denote the predicted and ground-truth point clouds, respectively:

$$d(\mathbf{x}, \mathcal{S}) = \min_{\mathbf{y} \in \mathcal{S}} \|\mathbf{x} - \mathbf{y}\|_2,$$

$$\text{Accuracy} = \frac{1}{|\mathcal{P}_{\text{pred}}|} \sum_{\mathbf{p} \in \mathcal{P}_{\text{pred}}} d(\mathbf{p}, \mathcal{P}_{\text{gt}}),$$

$$\text{Completeness} = \frac{1}{|\mathcal{P}_{\text{gt}}|} \sum_{\mathbf{g} \in \mathcal{P}_{\text{gt}}} d(\mathbf{g}, \mathcal{P}_{\text{pred}}),$$

$$\text{CD} = \frac{\text{Accuracy} + \text{Completeness}}{2},$$

$$\text{Precision} = \frac{1}{|\mathcal{P}_{\text{pred}}|} \sum_{\mathbf{p} \in \mathcal{P}_{\text{pred}}} \left[d(\mathbf{p}, \mathcal{P}_{\text{gt}}) < \tau \right],$$

$$\text{Recall} = \frac{1}{|\mathcal{P}_{\text{gt}}|} \sum_{\mathbf{g} \in \mathcal{P}_{\text{gt}}} \left[d(\mathbf{g}, \mathcal{P}_{\text{pred}}) < \tau \right],$$

$$\text{F1-score} = \frac{2 \, \text{Precision} \, \text{Recall}}{\text{Precision} + \text{Recall}}.$$

We report the Chamfer Distance (CD) for 3D reconstruction in Tab. 7. Our method achieves the best average performance among streaming methods.

Table 7: **3D reconstruction Chamfer distance (CD) on HiRoom, ETH3D, 7Scenes, and ScanNet++.** Evaluated without (w/o p.) and with (w/ p.) ground-truth camera poses. Lower is better. Methods are grouped into *full-attention* (offline) and *streaming* (online). Best and second-best results are highlighted in **bold** and underline, respectively.

Methods	Avg		HiRoom		ETH3D		7Scenes		ScanNet++	
	CD(w/o p.)↓	CD(w/ p.)↓	CD(w/o p.)↓	CD(w/ p.)↓	CD(w/o p.)↓	CD(w/ p.)↓	CD(w/o p.)↓	CD(w/ p.)↓	CD(w/o p.)↓	CD(w/ p.)↓
Full-attention (offline)										
VGGT	0.2220	0.1838	0.0968	0.0646	0.5669	0.4664	0.1446	0.1269	0.0798	0.0771
MapAnything	0.2472	0.1883	0.0863	0.0750	0.6478	0.4919	0.1263	0.1111	0.1283	0.0754
Pi3X	0.1889	0.1458	0.0449	0.0307	0.5326	0.3990	**0.1078**	**0.0897**	0.0702	**0.0640**
DA3	**0.1688**	**0.1457**	**0.0355**	**0.0202**	**0.4499**	**0.3893**	0.1216	0.1080	**0.0680**	0.0651
Streaming (online)										
CUT3R	0.4691	0.4235	0.7885	0.7858	0.7618	0.6409	0.1586	0.1454	0.1674	0.1219
StreamVGGT	0.3827	0.3410	0.2806	0.2036	0.9274	0.8623	0.1573	0.1574	0.1655	0.1407
WinT3R	0.3131	0.2629	0.2644	0.2410	0.6525	0.5368	0.1675	0.1413	0.1680	0.1328
STream3R	0.3223	0.2569	0.1278	0.0911	0.7271	0.5542	0.3046	0.2609	0.1298	0.1217
LingBot-Map	0.2431	0.2158	0.1019	0.0796	0.6451	0.5772	**0.1352**	0.1203	0.0904	0.0860
Ours	**0.1852**	**0.1556**	**0.0403**	**0.0354**	**0.4822**	**0.3976**	0.1370	**0.1154**	**0.0814**	**0.0740**

A.6 Detailed Geometry Evaluation on Dynamic Scenes

Table 8: **Geometry evaluation on dynamic scenes.** (a) Camera pose estimation. (b) Depth estimation. Bonn denotes Bonn RGB-D Dynamic; pose averages exclude Bonn. Best and second-best results are marked in **bold** and underline within each inference group.

Method	Avg.		Sintel		Bonn		TUM-Dynamic		Waymo		PointOdyssey	
(a) Camera Pose Estimation												
	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑	AUC@3↑	AUC@30↑
Full-attention (offline)												
VGGT	0.4249	0.7656	0.2157	0.6133	–	–	**0.2428**	**0.7465**	0.7166	0.8983	0.5245	0.8044
MapAnything	0.2066	0.5917	0.0548	0.4310	–	–	0.0842	0.5130	0.5864	0.8651	0.1009	0.5578
Pi3X	**0.4537**	0.7726	**0.2804**	0.6520	–	–	0.2226	0.7418	**0.8578**	**0.9193**	0.4539	0.7771
DA3	0.4351	0.7617	0.2065	0.6497	–	–	0.1956	0.6652	0.7192	0.8923	**0.6192**	**0.8397**
Ours	0.4432	**0.7923**	0.2130	**0.7142**	–	–	0.2308	0.7190	0.7414	0.9018	0.5874	0.8342
Streaming (online)												
CUT3R	0.0892	0.5087	0.0255	0.4094	–	–	0.0368	0.3694	0.2734	0.8138	0.0209	0.4422
StreamVGGT	0.1626	0.6283	0.0625	0.4699	–	–	0.0724	0.5488	0.3046	0.7854	0.2108	0.7092
WinT3R	0.0605	0.4446	0.0059	0.2047	–	–	0.0068	0.2901	0.1884	0.7070	0.0410	0.5766
STream3R	0.2301	0.6538	0.0923	**0.5487**	–	–	0.0685	0.4861	0.4626	0.8432	0.2969	0.7372
LingBot-Map	0.2556	0.6613	**0.1043**	0.5312	–	–	**0.1306**	0.5831	0.6221	0.8792	0.1653	0.6515
Ours	**0.3346**	**0.7261**	0.0994	0.5403	–	–	0.0967	**0.6787**	**0.6928**	**0.8893**	**0.4495**	**0.7960**
(b) Depth Estimation												
	$\delta_{1.25}$ ↑	AbsRel↓	$\delta_{1.25}$ ↑	AbsRel↓	$\delta_{1.25}$ ↑	AbsRel↓	$\delta_{1.25}$ ↑	AbsRel↓	$\delta_{1.25}$ ↑	AbsRel↓	$\delta_{1.25}$ ↑	AbsRel↓
Full-attention (offline)												
VGGT	0.8961	0.0967	0.6148	0.2679	0.9694	0.0571	0.9370	0.0887	0.9661	0.0528	**0.9931**	**0.0168**
MapAnything	0.8657	0.1612	0.5533	0.4479	0.9736	0.0609	0.9382	0.1020	0.9475	0.0892	0.9159	0.1061
Pi3X	**0.9143**	**0.0802**	**0.6980**	**0.2284**	**0.9750**	**0.0415**	**0.9584**	**0.0528**	**0.9771**	**0.0386**	0.9632	0.0397
DA3	0.8949	0.1034	0.6392	0.2596	0.9727	0.0521	0.9289	0.1019	0.9450	0.0837	0.9886	0.0197
Ours	0.9098	0.0920	0.6717	0.2410	0.9717	0.0543	0.9520	0.0893	0.9647	0.0569	0.9890	0.0184
Streaming (online)												
CUT3R	0.8201	0.2316	0.4979	0.7438	0.9496	0.0828	0.8632	0.1092	0.8682	0.1172	0.9217	0.1050
StreamVGGT	0.8313	0.1396	0.5728	0.2917	**0.9733**	0.0610	0.9206	0.1065	0.7097	0.1845	0.9799	0.0541
WinT3R	0.8209	0.1500	0.5435	0.3466	0.9063	0.0739	0.8005	0.1331	0.9104	0.1188	0.9438	0.0775
STream3R	0.8891	0.1072	**0.6177**	0.2665	0.9485	0.0695	0.9379	0.0940	0.9494	0.0784	**0.9919**	0.0276
LingBot-Map	0.8787	0.1166	0.5949	0.2737	0.9584	0.0617	0.9379	**0.0902**	0.9378	0.0915	0.9645	0.0657
Ours	**0.8919**	**0.1017**	0.6100	**0.2631**	0.9712	**0.0562**	**0.9395**	0.1028	**0.9516**	**0.0629**	0.9874	**0.0235**

Tab. 8 reports detailed camera pose and depth results on the five dynamic-scene benchmarks under full-attention and streaming inference, excluding Bonn from pose evaluation. Despite causal training, full-attention IGGT4D outperforms DA3 on all four average dynamic-scene metrics. The streaming accuracy drop suggests that missing future context limits bidirectional matching and global geometric reasoning, trading accuracy for online inference.

A.7 Training Strategy and Instance Feature Dimensionality

Tab. 9 evaluates two-stage training and instance feature dimensionality on ScanNet++ using DA3-Large. Joint training from the first epoch degrades all metrics, especially geometry, supporting two-stage training. Using 4-D features largely preserves geometry but degrades instance and semantic results; 16-D offers no consistent gain, favoring compact 8-D features.

Table 9: **Ablation of two-stage training and instance feature dimensionality on ScanNet++ using DA3-Large.** Best and second-best results are highlighted in **bold** and underline, respectively.

Model	Geometry		Instance		Semantic	
	AUC@3 ↑	F1-score ↑	mIoU ↑	Match Rate ↑	mIoU ↑	mAcc ↑
Full model (8-D)	**0.2984**	0.4790	**0.3098**	**0.6271**	**0.3666**	0.6853
w/o Two-stage Training	0.2224	0.3885	0.2937	0.6103	0.3548	0.6626
4-D Instance Features	0.2949	**0.4799**	0.2481	0.5496	0.3225	0.6437
16-D Instance Features	0.2967	0.4774	0.3088	0.6231	0.3638	**0.6911**

A.8 Mask Generation and Clustering Sensitivity

Mask generation. Direct mask prediction requires jointly optimizing pixel-level segmentation, instance separation, and cross-frame identity consistency as the number of instances and input views evolves. Maintaining high-resolution mask logits for all instances also increases memory and computational costs. We therefore learn compact 8-D geometry-aware instance features end to end and use Streaming Clustering to generate masks and maintain global identities.

Clustering thresholds. Tab. 10 evaluates the two inference-time clustering thresholds. The default setting, $\tau_s = 0.95$ and $\tau_l = 0.90$, achieves the best average T-mIoU and T-SR among the tested settings. Reducing τ_s, shared by core-region formation and global codebook matching, causes a clear performance drop, whereas varying the region-expansion threshold τ_l produces smaller changes. For L_2-normalized features, $\tau_s = 0.95$ corresponds to a Euclidean distance of $\sqrt{2 - 2\tau_s} \approx 0.316$, close to the cross-view pull margin $\delta_{\text{pull}}^{\text{cr}} = 0.30$.

Table 10: **Average spatial instance tracking performance under different clustering thresholds.**

τ_s	τ_l	T-mIoU ↑	T-SR ↑
0.95	0.90	**59.25**	**86.28**
0.95	0.85	58.83	85.89
0.95	0.80	58.17	85.04
0.90	0.85	54.22	79.07
0.85	0.80	48.86	72.25

A.9 Clustering Quality Comparison

We compare clustering quality on spatial instance tracking by replacing Streaming Clustering with HDBSCAN [50] while keeping the predicted instance features and evaluation protocol unchanged. As shown in Tab. 11, HDBSCAN increases the average T-mIoU/T-SR from 59.25/86.28 to 60.02/90.34, but requires full-sequence offline processing and incurs substantially higher runtime and memory costs (Tab. 6). Streaming Clustering therefore provides a practical balance between tracking quality, efficiency, and online inference.

Table 11: **Clustering quality comparison.** The two Ours variants use identical predicted features and evaluation settings; IGGT is included as a reference baseline.

Method	T-mIoU ↑	T-SR ↑
IGGT (HDBSCAN)	51.68	84.65
Ours (Streaming)	<u>59.25</u>	<u>86.28</u>
Ours (HDBSCAN)	**60.02**	**90.34**

A.10 Visualization of Streaming Clustering Masks

We further visualize instance-aware representations and the instance masks produced by our efficient streaming clustering. Fig. 11 shows PCA projections of the predicted instance features together with the corresponding temporally consistent masks, illustrating stable identity tracking across frames without explicit motion modeling.

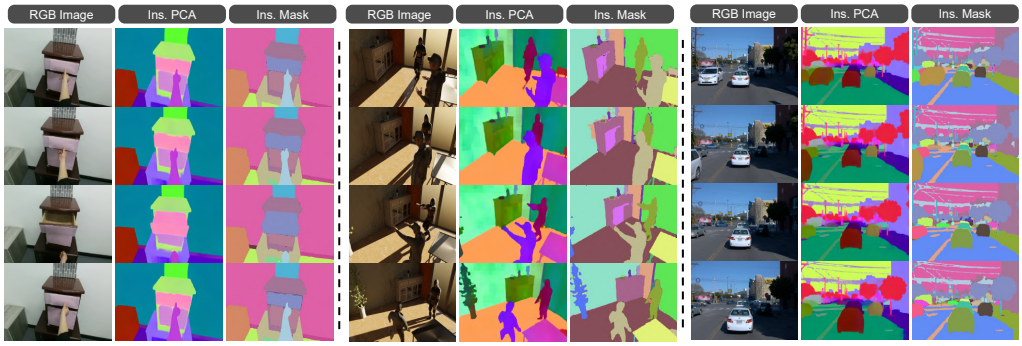

Figure 11: **Streaming clustering masks.** PCA visualization of instance features alongside masks obtained by our streaming clustering, demonstrating 4D-consistent instance segmentation in dynamic sequences.

A.11 Class-Agnostic Multi-Instance Video Segmentation

We evaluate class-agnostic multi-instance video segmentation on Waymo [60], comparing IGGT4D with SAM 2 [67], SpaTrackerV2 [76]+SAM [25], DEVA [78], Cutie [79], and SAM2Long [80]. Tab. 12 reports the joint region and contour score $\mathcal{J}\&\mathcal{F}$, region similarity $\mathcal{J}$, and contour accuracy $\mathcal{F}$. IGGT4D outperforms all compared methods across the three metrics.

Table 12: **Class-agnostic multi-instance video segmentation on Waymo.**

Method	$\mathcal{J}\&\mathcal{F}$ ↑	$\mathcal{J}$ ↑	$\mathcal{F}$ ↑
SAM 2	26.12	22.71	29.53
SpaTrackerV2+SAM	28.58	25.12	32.04
DEVA	<u>34.99</u>	<u>28.64</u>	<u>41.34</u>
Cutie	25.03	21.89	28.17
SAM2Long	25.99	23.03	28.95
Ours	**48.21**	**39.86**	**56.56**

A.12 Comparison with Online Semantic SLAM

We compare our method with OVO-ORB-SLAM3 [14] on the five-scene ScanNetv2 [81] subset used in OVO (`scene0011_00`, `scene0050_00`, `scene0231_00`, `scene0378_00`, and `scene0518_00`) using contiguous 150-frame subsequences and ScanNet20 labels. Following its evaluation protocol, we report Sim(3)-aligned camera trajectory error and 3D semantic segmentation metrics. mIoU and

mAcc average per-class IoU and recall, respectively; f-mIoU and f-mAcc weight classes by their ground-truth vertex frequencies.

Table 13: **Quantitative comparison with online semantic SLAM for camera pose estimation and 3D semantic segmentation.**

Method	ATE RMSE (cm) ↓	mIoU (%) ↑	mAcc (%) ↑	f-mIoU (%) ↑	f-mAcc (%) ↑
OVO-ORB-SLAM3 w/o loop clos.	5.76	32.60	52.10	47.00	62.30
OVO-ORB-SLAM3	<u>5.25</u>	<u>34.20</u>	<u>52.70</u>	<u>48.50</u>	<u>63.40</u>
Ours	**4.89**	**37.24**	**59.17**	**56.17**	**68.51**

As shown in Tab. 13, our method outperforms OVO-ORB-SLAM3 with and without loop closure. Moreover, our method requires only RGB input, whereas OVO-ORB-SLAM3 additionally requires depth, demonstrating the advantage of feed-forward streaming prediction without iterative optimization.

Nevertheless, long-sequence scalability remains a limitation of current feed-forward reconstruction methods, while optimization-based SLAM can reduce accumulated drift through loop closure and global optimization.

A.13 Long-Sequence Streaming: Efficiency, Limitations, and Extensions

Inference efficiency. Tab. 14 compares full-attention and streaming inference on a single NVIDIA GeForce RTX 5090 GPU with 32 GB of memory and images resized to 504×336. Full-attention inference processes all N frames in one offline forward pass, whereas streaming inference processes them sequentially.

Table 14: **Model-forward runtime and GPU memory at different sequence lengths.** Each entry reports total model-forward time for the sequence (s) / peak GPU memory (GB), excluding Streaming Clustering and Instance Codebook updates. Parentheses show KV-cache memory (GB), which is included in peak memory. OOM denotes out of memory.

Method	$N = 32$	$N = 64$	$N = 100$	$N = 180$	$N = 200$	$N = 250$
Ours (Full-attention)	1.48 / 16.75	3.55 / 19.60	6.56 / 22.77	16.00 / 29.89	OOM	OOM
StreamVGGT (Streaming)	2.42 / 9.76 (2.55)	6.60 / 12.81 (5.11)	13.51 / 16.23 (7.98)	36.87 / 23.85 (14.37)	44.65 / 25.75 (15.96)	OOM
Ours (Streaming)	2.80 / 11.04 (2.22)	7.17 / 13.32 (4.44)	14.04 / 16.51 (6.93)	36.48 / 23.81 (12.47)	43.83 / 25.63 (13.86)	71.11 / 30.19 (17.32)

Full-attention inference runs out of memory at $N = 200$, while our streaming implementation processes $N = 250$ frames using 30.19 GB of peak GPU memory, including 17.32 GB for the KV cache. Retaining the full history causes KV-cache memory to grow approximately linearly and per-frame computation to increase with sequence length, limiting inference under a fixed resource budget.

Overlapping sliding-window inference. Overlapping windows could bound the KV cache by the window size. Adjacent 3D predictions could be aligned using shared frames, with Streaming Clustering and the Instance Codebook updating instance features across windows. However, discarding historical context may cause alignment drift under large camera motion or insufficient overlap and weaken instance association under occlusion or scene revisitation.

Selective KV-cache pruning. InfiniteVGGT [82], a training-free extension of StreamVGGT, prunes redundant KV-cache tokens while retaining informative historical features under a fixed cache budget. Combining this strategy with IGGT4D's persistent global instance memory is a direction for future work.

A.14 4D QA Scene Grounding

To demonstrate the capability of IGGT4D in supporting complex spatial-temporal reasoning, we evaluate 4D QA scene grounding using Large Multimodal Models (LMMs) [51, 52]. When provided with only RGB video streams, LMMs often struggle to track moving objects under challenging visual conditions. For example, as shown in Fig. 12, when a picked-up bottle moves from left to right, its appearance closely blends with the background. Consequently, the LMMs erroneously infer that the bottle has been moved out of view based on its motion trajectory. However, when augmented with our 4D-consistent instance features or masks, the LMMs can robustly track the object, correctly determine that it remains within the field of view, and accurately segment the bottle.

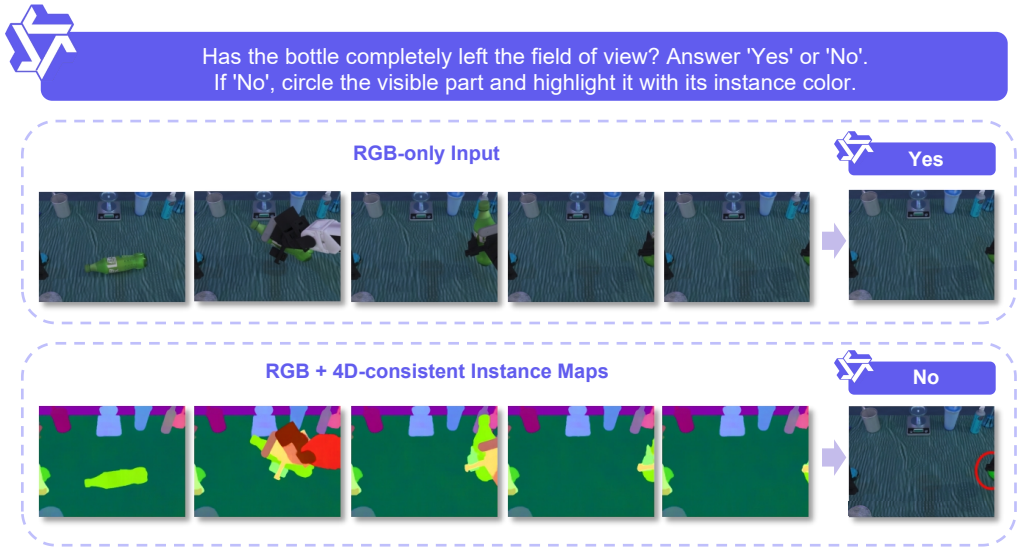

Figure 12: **4D QA scene grounding.** Comparison of spatial-temporal reasoning capabilities using LMMs. Given only RGB frames (top), the LMM fails to track the moving bottle as it blends into the background, erroneously concluding it has left the view. In contrast, by incorporating our 4D-consistent instance features (bottom), the LMM successfully tracks the object across frames, accurately grounds its location, and correctly answers the 4D semantic query despite challenging visual conditions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our contributions: a streaming geometry-instance prediction framework (IGGT4D), an online clustering strategy, and a large-scale 4D dataset (InsScene4D-147K). These are supported by experiments in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 6, including growth in KV-cache memory and per-frame computation with sequence length, data coverage, the reliance on supervised learning, and the current perception-centric focus which lacks embodied interaction. The appendix provides model-forward resource measurements and discusses possible long-sequence extensions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper presents a methodological and empirical study and does not claim formal theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed problem formulations (Sec 3.1) , architecture details (Sec 3.2) , and training objectives (Sec 3.5).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will provide open access to the InsScene4D-147K dataset and our code as described in the supplementary material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed problem formulations (Sec 3.1) , architecture details (Sec 3.2) , and training objectives (Sec 3.5).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report benchmark averages and per-dataset results, but do not report error bars, confidence intervals, or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The training configuration details appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses cited 3D vision datasets and models under their original licenses/access terms and does not introduce new human-subject experiments or deployment decisions affecting individuals.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: This work has potential positive societal impacts by improving spatial reasoning capabilities for applications such as embodied AI, robotics, AR/VR, and assistive visual understanding.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release a high-risk generative model, language model, or scraped internet dataset; released assets are benchmark and 3D vision evaluation resources, subject to the original dataset licenses and access terms.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets, models, and tools used in this paper are properly cited throughout the text.

    Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce the InsScene4D-147K dataset, detailed in Section 4 and Figure 4.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not conduct new crowdsourcing or human-subject studies; any human activity data comes from previously released, cited datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A].

Justification: The work does not recruit participants or collect new human-subject data; existing datasets are cited and used under the original creators' terms.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We specify that IGGT4D's representation can support downstream 4D scene grounding when used with Large Multimodal Models (LMMs) as mentioned in Section 3.4.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.