Geometric Context Transformer for Streaming 3D Reconstruction

Lin-Zhuo Chen* Jian Gao* Yihang Chen Ka Leong Cheng Yipengjing Sun Liangxiao Hu
Nan Xue Xing Zhu Yujun Shen Yao Yao† Yinghao Xu†‡

*Equal Contribution †Corresponding Author ‡Project Lead

Streaming 3D reconstruction aims to recover 3D information, such as camera poses and point clouds, from a video stream, which necessitates geometric accuracy, temporal consistency, and computational efficiency. Motivated by the principles of Simultaneous Localization and Mapping (SLAM), we introduce LingBot-Map, a feed-forward 3D foundation model for reconstructing scenes from streaming data, built upon a *geometric context transformer* (GCT) architecture. A defining aspect of LingBot-Map lies in its carefully designed attention mechanism, which integrates an anchor context, a pose-reference window, and a trajectory memory to address coordinate grounding, dense geometric cues, and long-range drift correction, respectively. This design keeps the streaming state compact while retaining rich geometric context, enabling stable efficient inference at around 20 FPS on 518×378 resolution inputs over long sequences exceeding 10,000 frames. Extensive evaluations across a variety of benchmarks demonstrate that our approach achieves superior performance compared to both existing streaming and iterative optimization-based approaches.

Website: https://technology.robbyant.com/lingbot-map
Github: https://github.com/robbyant/lingbot-map

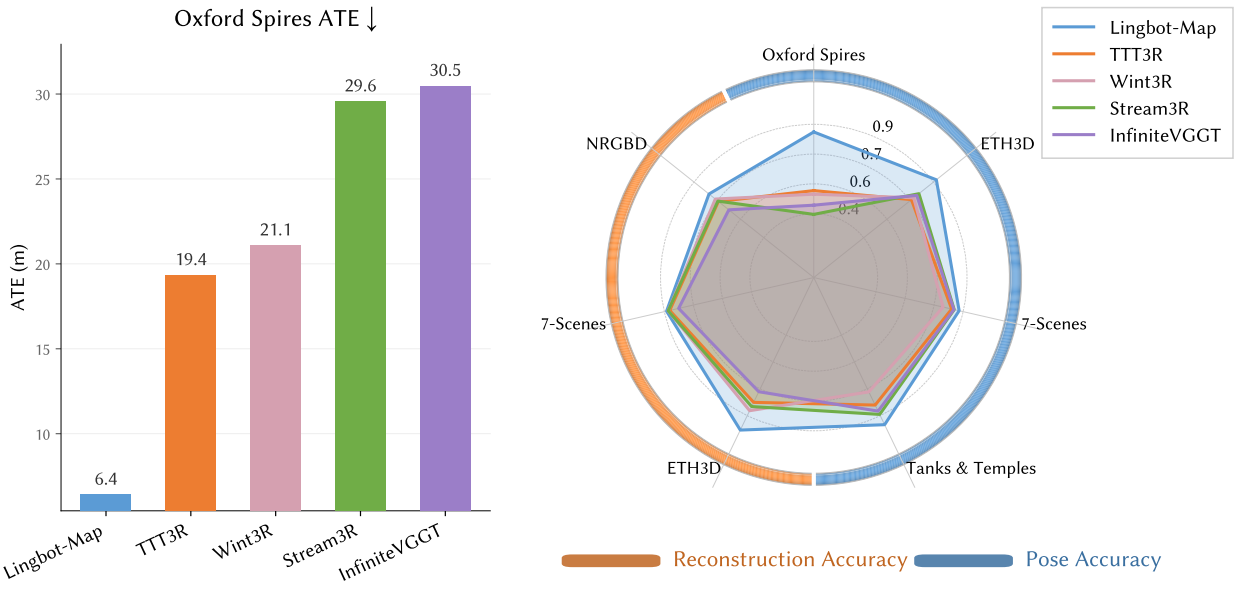


Figure 1. Comparison of LingBot-Map with State-of-the-Art streaming reconstruction methods.

1 Introduction

We perceive the world through a continuous stream of visual input, yet our spatial memory is not a faithful recording of every moment: it is sparse, structured, and efficient. Rather than retaining every observation, human spatial cognition selectively preserves only the most essential cues, enabling coherent navigation through large-scale environments over extended periods. Can we build machines that operate similarly, performing streaming 3D reconstruction from continuous visual input, selectively and efficiently?

In recent years, 3D foundation models have advanced rapidly, with methods such as VGGT [74] and Depth Anything 3 [36] directly predicting camera poses, depth maps, and dense point maps from multiple images in a single feed-forward pass. However, these successes are largely confined to offline settings, where the full image set is available and processed globally. Recent efforts [27, 32, 78, 102] have begun to adapt these capabilities to streaming reconstruction, but existing approaches still exhibit limited robustness to long sequences and complex scenes. The core difficulty is selective context management: balancing rich geometric context for long-term consistency with a compact state for efficient inference.

Existing methods adopt different strategies to manage streaming context, each involving distinct trade-offs. CUT3R [78] maintains a persistent recurrent-style state, but its aggressive compression can lead to state forgetting and weak retention of essential geometric priors. In contrast, StreamVGGT [102] and Stream3R [27] adopt causal attention and caching, yet retain near-complete history without explicit selection, mixing useful geometry with redundancy and causing memory and computation to grow rapidly. A third line of work, such as VGGT-SLAM [41] and MASt3R-SLAM [45], integrates learned 3D models with classical SLAM backends. While these methods incorporate structured context management through keyframe selection and pose-graph maintenance, such selection relies on hand-crafted heuristics rather than learned priors, and iterative optimization further limits real-time applicability. These observations point to a key design principle: the streaming state should selectively retain *what* matters most, not merely how much, and this selection should be grounded in geometric priors yet learned end-to-end from data.

To this end, we introduce `LingBot-Map`, a streaming foundation model built around Geometric Context Attention (GCA) that realizes this principle within a unified attention framework. The design draws on a key insight from classical SLAM systems: robust real-time reconstruction requires maintaining distinct types of spatial context—a reference frame for coordinate grounding, a local window for dense local geometry estimation, and a global map for drift correction. Accordingly, GCA explicitly maintains three complementary types of context: an *anchor context* for coordinate and scale grounding, a local *pose-reference window* that retains dense visual features from recent frames for accurate local geometry estimation, and a *trajectory memory* that compresses the full observation history into compact per-frame tokens for global consistency. While the context structure is motivated by classical reconstruction principles, GCA replaces hand-crafted optimization with end-to-end learned attention that adaptively weights, encodes, and compresses information within each context type. This structured yet learned representation ensures stable and efficient inference even over arbitrarily long sequences, with nearly constant memory and computation per frame, as context beyond the local window is compressed into compact per-frame tokens.

To scale up training effectively, we employ a progressive training strategy combined with context parallelism [20], along with a relative loss formulation that facilitates stable optimization on long sequences. This allows `LingBot-Map` to learn efficiently from diverse, large-scale 3D datasets. We evaluate `LingBot-Map` on comprehensive benchmarks, including Oxford Spires, 7-Scenes, Tanks and Temples, ETH3D, and show consistent improvements over existing streaming methods in both camera pose estimation and dense 3D reconstruction quality.

Our contributions are summarized as follows:

- We introduce `LingBot-Map`, a streaming 3D foundation model built around Geometric Context Attention (GCA), which maintains three complementary context types – anchor, pose-reference window, and trajectory memory – for efficient and consistent long-sequence streaming inference.
- We propose an efficient training recipe based on progressive training and context parallelism with a relative loss formulation for stable long-sequence optimization.
- We demonstrate that `LingBot-Map` achieves state-of-the-art performance on multiple benchmarks (Oxford Spires, Tanks and Temples, ETH3D, and 7-Scenes), significantly outperforming existing streaming approaches in reconstruction quality and inference speed.

(a) Real-World Captured Aerial Video (b) World Model Generated Cartoon Video

(c) Long-sequence Multi-Room Traversal Video

(d) Long-sequence Outdoor Driving video

Figure 2. Visualization results of `LingBot-Map` in VO mode on varying scenes: Given a continuous video stream, `LingBot-Map` performs *efficient* **streaming** 3D reconstruction with accurate camera pose estimation and high-quality point cloud reconstruction. Examples include Real-World Captured Aerial Video (*top-left*), World Model Generated Cartoon Video (*top-right*), Long-sequence Multi-Room Traversal Video (*middle*), and Long-sequence Outdoor Driving video (*bottom*).

2 Related Work

Traditional 3D Reconstruction. Traditional 3D reconstruction methods mainly include Structure-from-Motion (SfM) [47, 55, 61], Simultaneous Localization and Mapping (SLAM) [4, 43, 44], and Multi-View Stereo (MVS) [15, 56, 90]. SfM and SLAM recover camera poses and scene geometry from multi-view observations, where SfM typically operates offline on unordered image collections, while SLAM processes video streams online. These systems are usually complex and highly modular, typically centered around optimization-based bundle adjustment for camera pose estimation. In contrast, MVS focuses on dense reconstruction given known camera poses. Over the past decade, many works have explored replacing individual components of these pipelines with deep learning modules, particularly for feature extraction [12] and matching [53, 64]. More recently, several approaches attempt to implement SfM, SLAM, or MVS in an end-to-end manner, such as VGGTSfM [75], DROID-SLAM [68], and MVSNet [90, 91].

3D Foundation Model. DUSt3R [80] represents a paradigm shift in feed-forward 3D reconstruction. Given a collection of unposed images, DUSt3R directly regresses a dense 3D reconstruction of the scene without explicit geometric modeling. However, DUSt3R [80] only supports two-view input and requires aligning all results via optimization for additional views. To support more than two views and improve reconstruction quality, VGGT [74] uses an advanced transformer architecture that includes cross-view attention layers, achieving state-of-the-art performance on standard benchmarks. Crucially, VGGT demonstrates that leveraging large-scale data together with powerful model architectures can significantly improve reconstruction quality. Building on this foundation, numerous subsequent works have advanced feed-forward reconstruction across various dimensions, including improving reconstruction accuracy [36, 74, 82, 99], enhancing computational efficiency [38, 89], handling dynamic scenes [6, 13, 23, 39, 63, 73, 85, 97], enabling novel view synthesis [8, 17, 22, 28, 35, 60, 66, 77, 87, 88, 93], and incorporating multi-modal inputs [21, 24, 40]. However, these methods are primarily designed for offline processing and do not address the unique challenges of streaming 3D reconstruction, such as maintaining long-term consistency and managing computational resources over extended sequences.

Streaming 3D Reconstruction. Driven by the need for online applications, streaming 3D reconstruction can be broadly categorized into hybrid SLAM-based approaches and end-to-end feed-forward methods. Hybrid methods typically integrate 3D foundation models with traditional SLAM pipelines [11, 41, 45], aiming to leverage the strengths of both paradigms. However, these approaches often rely on hand-crafted components and careful parameter tuning to achieve optimal performance, lacking the benefits of a fully end-to-end learning framework. In contrast, recent feed-forward streaming methods [27, 32, 72, 79, 102] extend the offline paradigm to streaming settings by employing Recurrent Neural Network (RNN)-based architectures or by combining caching mechanisms with causal attention. Specifically, CUT3R [78] maintains a persistent state that is updated recurrently via RNN architectures. To mitigate state forgetting, TTT3R [7] adopts a test-time training strategy. Meanwhile, StreamVGGT [102], Stream3R [27], and Wint3R [32] adapt the more advanced VGGT architecture using causal attention and caching strategies. Despite these advances, existing streaming methods often struggle to maintain performance over long input sequences and in complex environments. Common failure modes include significant trajectory drift, degraded reconstruction accuracy, and prohibitive growth in memory and computational requirements. We attribute these limitations to the absence of a principled mechanism for effectively retaining essential geometric context during the streaming process. Concurrently, LoGeR [98] and Scal3R [86] explore scaling 3D reconstruction to long sequences via chunk-based processing with hybrid memory mechanisms. LoGeR combines sliding window attention for local alignment with test-time training (TTT) for global consistency, while Scal3R extends the TTT paradigm with chunking and visual place recognition for large-scale scenes. However, both methods operate in an offline setting and rely on test-time parameter updates, which introduces additional computational overhead. In contrast, our `LingBot-Map` is a purely feed-forward streaming model that requires no test-time training or post-optimization, achieving real-time inference through a compact geometric context design.

3 Method

In Sec. 3.1, we present an overview of our method and the problem formulation. We then introduce Geometric Context Attention (GCA) (Sec. 3.2), a mechanism specifically designed for streaming 3D reconstruction. Sec. 3.3 describes the network architecture and the training procedure. Finally, Sec. 3.4 presents the inference pipeline for efficient inference.

3.1 Overview

Given a continuous stream of images $\mathcal{I} = \{I_1, I_2, \ldots\}$, LingBot-Map processes each new frame I_t upon arrival and estimates its camera pose $\hat{P}_t$ and depth map $\hat{D}_t$ using only the current and previously observed frames $\{I_1, \ldots, I_t\}$, without access to future observations. Building on the architecture of recent feed-forward 3D models [74], we design a streaming variant where each frame is encoded by a ViT backbone and processed through alternating layers of frame-wise attention and Geometric Context Attention (GCA), before task-specific heads predict the camera pose and depth map (see Fig. 4). The key to enabling efficient streaming inference is GCA, which maintains three complementary geometric contexts: anchor, pose-reference window, and trajectory memory, thereby balancing long-term consistency with compact state representation. We detail GCA in Sec. 3.2, the overall architecture and training strategy in Sec. 3.3, and the inference pipeline in Sec. 3.4.

3.2 Geometric Context Attention

The central challenge of streaming 3D reconstruction is managing geometric context: the model must retain sufficient long-range context to ensure global consistency, yet keep its streaming state compact enough for efficient inference. Classical SLAM and SfM systems provide structural insights into this trade-off by decomposing the streaming state into three distinct types of spatial context, each serving a complementary role: a reference frame for coordinate and scale grounding, a local window of recent observations for dense geometry estimation, and a global map for correcting accumulated drift. Drawing on this principle, GCA decomposes the streaming context into three complementary learned attention mechanisms, replacing hand-crafted optimization with end-to-end differentiable attention: *anchor context*, a local *pose-reference window*, and *trajectory memory*. We describe each below.

Anchor Context. Monocular reconstruction is inherently scale-ambiguous, so a consistent coordinate system and absolute scale must be established before streaming begins. Offline methods such as DUSt3R [80] and VGGT [74] resolve this by normalizing with respect to the *global* point cloud, but this requires access to all frames and is therefore incompatible with causal streaming inference. Instead, we designate the first n images ($n \ll N$) as anchor frames and use them to fix the scale. We apply full attention among these frames and augment their image tokens with a learnable anchor token, enabling the network to identify and distinguish them from subsequent streaming frames. After initialization, the anchor and image tokens for these frames are retained in the attention context, and all subsequent frames attend to them as fixed references. During training, we normalize all ground-truth annotations to a canonical scale derived from the anchor frames: we compute $s = \frac{1}{|\bar{\mathcal{X}}^{\text{anchor}}|} \sum_{\mathbf{x} \in \bar{\mathcal{X}}^{\text{anchor}}} \|\mathbf{x}\|_2$ as the mean distance of the ground-truth point cloud $\bar{\mathcal{X}}^{\text{anchor}}$ from the coordinate origin, and divide all ground-truth depths and camera translations by s.

Local Pose-Reference Window. Accurately registering each new frame requires dense visual overlap with nearby observations, context that the distant anchor frames alone cannot provide. To address this, we maintain a sliding window of the k most recent frames during inference, retaining their *full* image tokens. This dense local context provides essential relative pose cues from immediate visual connections, enabling the network to accurately register new frames into the global trajectory. To further encourage geometric consistency within the local window, we apply a relative pose loss between frames in this window, as detailed in Sec. 3.3.

Trajectory Memory. The anchor context and local window together provide a fixed global reference and dense recent observations, but without any record of intermediate frames, pose errors accumulate unchecked over long sequences, causing the estimated trajectory to drift. To mitigate this, we retain a compact trajectory context that summarizes the full observation history. Specifically, for frames that fall outside both the anchor set and the active sliding window, we retain only the camera, anchor, and register tokens (i.e., 6 context tokens per frame) while discarding the memory-intensive image tokens (M tokens per frame). Additionally, we incorporate video temporal positional encodings [71] into the retained tokens to impose temporal ordering on the global trajectory. By maintaining this lightweight yet temporally ordered record of all past observations, the trajectory memory provides long-range cues that help correct accumulated drift and ensure global consistency.

Attention Mask Design. Fig. 3 compares different attention patterns for streaming inference. Global attention (a) attends to all frames but cannot operate in a streaming fashion. Causal attention (b) enables streaming but causes memory and computation to grow linearly with sequence length. Sliding window attention (c) bounds compute but sacrifices long-term context. Our GCA (d) combines the anchor context, trajectory memory, and local window into a structured attention mask that retains long-range consistency with bounded per-frame cost.

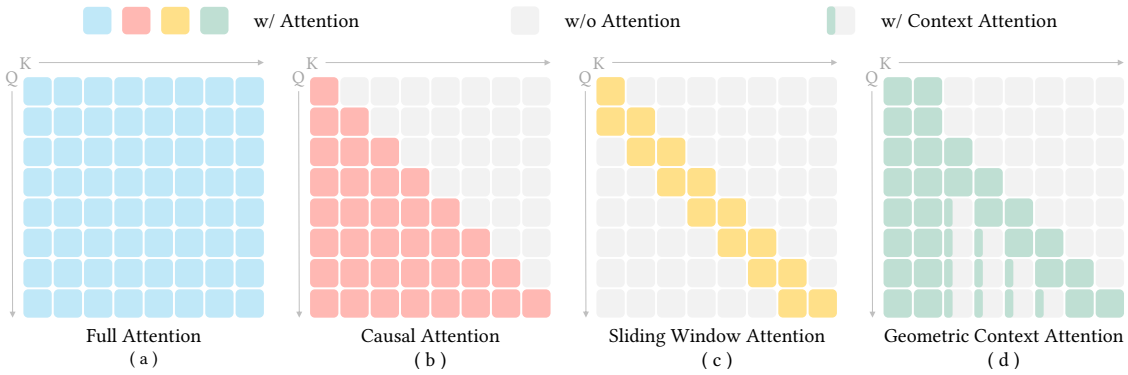


Figure 3. Comparison of attention masks. Each square block represents one frame's tokens, consisting of a small leading segment of context tokens and a larger segment of image tokens. (a) Full attention attends to all frames. (b) Causal attention enables streaming but grows linearly with sequence length. (c) Sliding-window attention bounds cost but loses long-range context. (d) GCA partitions the streaming context into *anchors* (n=2), a *local window* (k=2), and *trajectory memory*, retaining rich long-range context while keeping cost nearly constant as the sequence length increases.

Complexity Analysis. For a T-frame sequence, the per-frame attention context in GCA comprises n anchor frames with full tokens ($n \cdot (M+6)$), k window frames with full tokens ($k \cdot (M+6)$), and $(T-n-k)$ trajectory frames with compact tokens (6 each). Since n and k are fixed constants, the total context simplifies to $(n+k) \cdot M + 6T$, where the first term is constant and the second grows at a rate of just 6 tokens per frame. In contrast, causal attention retains $T \cdot (M+6) = MT + 6T$ tokens, sharing the same $6T$ term but incurring an additional MT that grows with the full token count. Since each new frame adds $(M+6)$ tokens under causal attention but only 6 under GCA, the per-frame growth rate is reduced by roughly $80\times$ for typical values ($M \approx 500$). Concretely, with n=3, k=16, and T=10,000, causal attention accumulates $\sim 5\times10^6$ tokens, while GCA retains only $\sim 7\times10^4$, yielding nearly constant memory and computation per frame.

3.3 Geometric Context Transformer Framework

Architecture. The overall architecture is illustrated in Fig. 4. Each input image is first encoded by a Vision Transformer (ViT) backbone initialized from DINOv2 [46] to produce M image tokens per frame. These image tokens are augmented with a camera token $\mathbf{c} \in \mathbb{R}^C$, four register tokens $\mathbf{r}_j \in \mathbb{R}^C$ ($j = 1, \ldots, 4$), and a learnable anchor token $\mathbf{a} \in \mathbb{R}^C$. The augmented tokens are then processed through multiple alternating layers of Frame Attention and GCA. Frame Attention operates independently within each frame, enabling per-frame feature refinement, while GCA operates across frames according to the structured attention mask described in Sec. 3.2, enabling cross-frame geometric reasoning. Finally, a camera head takes the camera token to predict the absolute camera pose $\hat{P}_t$, and a depth head takes the image tokens to predict the corresponding depth map $\hat{D}_t$.

Loss Function. We train `LingBot-Map` using a composite loss function consisting of depth, absolute pose, and relative pose terms:

$$\mathcal{L} = \lambda_{\text{depth}}\,\mathcal{L}_{\text{depth}} + \lambda_{\text{abs-pose}}\,\mathcal{L}_{\text{abs-pose}} + \lambda_{\text{rel-pose}}\,\mathcal{L}_{\text{rel-pose}}. \tag{1}$$

The depth loss ($\mathcal{L}_{\text{depth}}$) and absolute pose loss ($\mathcal{L}_{\text{abs-pose}}$) follow the definitions in VGGT [74]: $\mathcal{L}_{\text{depth}} = \sum_{i=1}^{N} \left\| \Sigma_i^D \odot (\hat{D}_i - D_i) \right\| + \left\| \Sigma_i^D \odot (\nabla \hat{D}_i - \nabla D_i) \right\| - \alpha \log \Sigma_i^D$, $\mathcal{L}_{\text{abs-pose}} = \sum_{i=1}^{N} \left\| \hat{\mathbf{P}}_i - \mathbf{P}_i \right\|_\epsilon$. Here, Σ_i^D represents the predicted uncertainty map, and $\odot$ denotes element-wise multiplication. Unlike VGGT [74], we supervise the network using camera-to-world transformations rather than world-to-camera ones. In the world-to-camera parameterization, rotation and translation are inherently coupled, making translation estimation highly sensitive to rotation errors, particularly in long sequences. Inspired by π^3 [82], we incorporate a relative pose loss over all frame pairs within the sliding window:

$$\mathcal{L}_{\text{rel-pose}} = \frac{1}{k(k-1)} \sum_{\substack{i \neq j \\ i,j \in \{1,\ldots,k\}}} \left(\mathcal{L}_{\text{rot}}(i,j) + \lambda_{\text{trans}} \mathcal{L}_{\text{trans}}(i,j) \right), \tag{2}$$

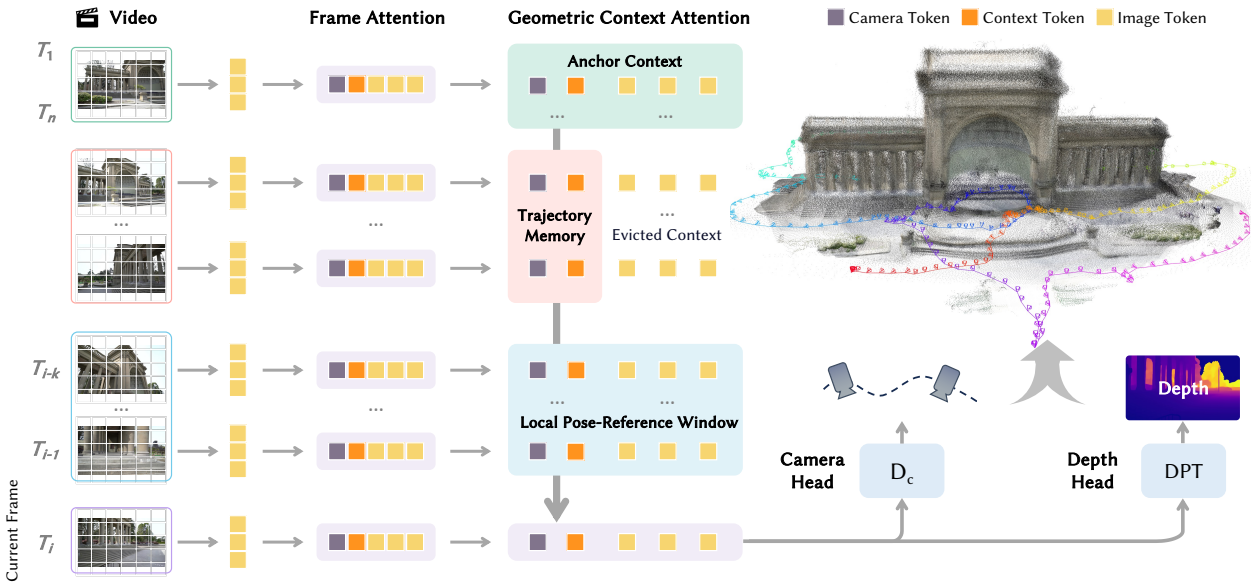


Figure 4. Pipeline of the proposed `LingBot-Map`**.** The framework processes the current view T_i relative to an initialization set $[T_1, T_n)$. A DINO backbone extracts image features, which are then refined through alternating layers of Frame Attention and GCA. Within the GCA module, the input view aggregates information from the *Anchor Context*, *Local Pose-Reference Window* $[T_{i-k}, T_i)$, and *Trajectory Memory Context*. Finally, task-specific heads predict camera pose and depth maps, enabling robust, memory-efficient streaming 3D reconstruction for long-range sequences.

where $\mathcal{L}_{\text{rot}}(i, j)$ and $\mathcal{L}_{\text{trans}}(i, j)$ denote the geodesic rotation error and ℓ_1 translation error of the relative pose between frames i and j, respectively. Because the window comprises only already-observed frames, this loss is inherently causal and encourages local trajectory consistency.

Progressive View Training. Training directly on long sequences is challenging: early-stage pose errors propagate along the trajectory and destabilize the loss landscape, leading to slow or divergent optimization. To address this, we adopt a progressive training strategy that starts with short subsequences and gradually increases the number of views over training. This curriculum enables the network to first acquire reliable local geometry estimation from short clips before learning to maintain global consistency across progressively longer trajectories.

Context Parallel. As the number of training views grows, GPU memory becomes the primary bottleneck due to the quadratic cost of cross-frame attention. To address this, we employ the Ulysses [20] context-parallelism strategy, which distributes different views across multiple GPUs to enable parallel attention computation via efficient all-to-all collective communication.

3.4 Inference System Design

Like autoregressive LLMs, our causal model caches the key-value (KV) states of previously processed frames to avoid redundant recomputation. However, with naive causal attention, the KV cache scales linearly with the number of frames, increasing memory consumption and per-frame latency. GCA addresses this by keeping the per-frame context compact (see Sec. 3.2), but the sliding-window and trajectory-eviction logic still requires frequent cache updates (appending new entries and discarding old ones), which incur overhead due to repeated memory reallocation under a standard contiguous layout. We eliminate this overhead with a paged KV-cache layout [26], in which updates affect only newly appended tokens rather than the entire cached sequence.

We implement the runtime on FlashInfer [94], which provides native support for paged KV-cache management, as well as optimized attention kernels for paged and sparse KV layouts. In the 518×378 setting with video sequences up to 1000 frames and a sliding window of 64 frames, our FlashInfer-based implementation achieves $\sim$20 FPS, compared to $\sim$10.5 FPS for an otherwise identical PyTorch baseline with contiguous KV-cache updates. To support robust long-sequence inference, we select a key frame every m frames to be retained in the KV cache, when the input views

are more than our max training views in training.

4 Training & Inference

Training a streaming 3D reconstruction model end-to-end on long sequences is challenging: pose errors in early frames propagate along the trajectory and destabilize the loss landscape, making direct optimization on hundreds of views impractical. To address this, we adopt a two-stage training curriculum. The first stage trains an offline base model on short, diverse multi-view data to establish robust geometric priors (Sec. 4.1). The second stage introduces our proposed Geometric Context Attention (GCA) and progressively scales to long sequences, transferring the base model's geometric foundations to the streaming setting (Sec. 4.2). We describe the training data curation in Sec. 4.3 and the inference pipeline in Sec. 4.4.

4.1 Base Model Training

Model Initialization. We initialize the ViT backbone from DINOv2 [46] with a patch size of 14 pixels, followed by 24 alternating blocks of frame attention and cross-frame attention, following the architecture of VGGT [74]. At this stage, we use standard global attention rather than GCA: since the training data includes both unordered multi-view collections and temporally ordered video sequences, global attention imposes no temporal structure and can fully exploit both data types. The number of input views per training sample is randomly sampled between 2 and 24, matching the diverse scale of available datasets.

Optimization. We use AdamW with a base learning rate of 2×10^{-4} and weight decay of 0.05. The learning rate follows a linear warmup from 10^{-8} to the base rate over the first 5% of training, followed by cosine annealing back to 10^{-8} over the remaining 95%. Training runs for 160K iterations.

Data Augmentation. Images are resized to a maximum dimension of 518 pixels. To improve robustness to appearance variation across datasets with diverse capture conditions, we apply aggressive photometric augmentation: random color jittering (brightness, contrast, saturation ± 0.5; hue ± 0.1) with probability 0.9, random grayscale conversion with probability 0.05, and random spatial rescaling in $[0.8\times, 1.2\times]$ with aspect ratios sampled from $[0.33, 1.0]$. Additionally, we apply *co-jittering*—applying an identical color transform to all frames within a scene—with probability 0.3, and independent per-frame transforms otherwise. This encourages the model to rely on geometric cues rather than appearance shortcuts when frames share similar photometric characteristics, while the independent transforms build robustness to inter-frame appearance variation.

Distributed Training. Training requires approximately 21,500 GPU hours, using fully sharded data parallelism (FSDP) with gradient checkpointing and bfloat16 mixed precision to manage memory consumption.

4.2 Streaming Model Training

Initialization from Base Model. We initialize the streaming model from the pretrained base model weights and replace global attention with GCA. Since the query, key, and value projections in GCA share the same parameterization as global attention, the pretrained weights transfer directly, providing a strong initialization that accelerates convergence.

Optimization and Progressive Curriculum. We train for 160K iterations with a base learning rate of 5×10^{-4}, using the same warmup and cosine annealing schedule as the first stage. To stabilize training on progressively longer sequences, we adopt a view curriculum: the number of training views increases linearly from 24 to 320 over the course of training. The starting count of 24 views matches the maximum used in the base stage, and the upper limit of 320 is set by the GPU memory budget under context parallelism. Similarly, the local pose reference-window size k of GCA is randomly sampled from 16 to 64 during training, exposing the model to varying receptive fields and improving robustness at inference time when different window sizes may be used.

Context Parallelism. As the number of views grows, GPU memory becomes the primary bottleneck due to the quadratic cost of cross-frame attention. We employ the Ulysses [20] context-parallelism strategy with a parallelism dimension of 16, distributing different views across GPUs and computing attention via all-to-all collective communication. Our implementation builds on TorchTitan [33] and Magi Attention. Training requires approximately 15,360 GPU hours.

4.3 Training Data

Dataset Composition. We curate a training corpus of 29 datasets covering indoor, outdoor, object-centric, synthetic, and real-world scenarios. The full list, together with data format, scene type, and per-stage sampling ratios, is given in Tab. 1. At the coarsest level, the datasets fall into two categories: *multi-view collections*, where frames are unordered and may lack temporal continuity, and *video sequences*, where frames follow a continuous camera trajectory. This distinction drives the different sampling strategies used in each training stage.

Stage 1: Diverse Short-Sequence Data. The first stage aims to build general geometric priors from a broad distribution of scenes. We draw from all 29 datasets with roughly balanced sampling ratios. The multi-view collections include BlendedMVS [92], HyperSim [52], MegaDepth [30], MVS Synth [19], GTA SFM [76], CO3D [51], Objaverse [10], and Texverse [100]. The video datasets include Unreal4K [69], WildRGBD [84], TartanAir [81], TartanAirV2 [81], TartanAirGround [49], PointOdyssey [101], VirtualKITTI [3], Kubric [16], DL3DV [37], Replica [62], SceneRGBD [42], Mapfree [1], Aria Synthetic Environments [48], ADT [48], ScanNet [9], ScanNet++ [95], MatrixCity [29], MidAir [14], and our internal game dataset. Each iteration samples 2 to 24 frames per scene, with a dynamic batch sampler that packs at most 48 images per GPU. Frame selection relies on a *nearby sampler*: a reference frame is chosen at random, and the remaining frames are drawn from a spatial window around it, without enforcing any temporal order. This unordered sampling is well-suited to the mixed-modality data in this stage, where many datasets have no natural frame ordering.

Stage 2: Long-Trajectory Video Data. The second stage shifts the distribution toward long, temporally coherent sequences needed for streaming reconstruction. We significantly increase the sampling weights for datasets with extended trajectories and multi-scene coverage, including TartanAir [81], TartanAirV2, TartanAirGround [49], MidAir [14], MatrixCity [29], Waymo [65], VirtualKITTI [3], KITTI-360 [34], ScanNet++ [95], ScanNet [9], and our internal game datasets, while down-weighting or dropping multi-view-only datasets that lack temporal structure (see Tab. 1 for exact ratios).

Foldback Video Sampler. To produce temporally coherent training subsequences from long videos, we replace the spatial nearby sampler with a *foldback video sampler*. The sampler starts at a random frame and advances with a random stride. Upon reaching a sequence boundary, it reverses direction and draws a new stride (distinct from the previous one) to avoid degenerate oscillation. This mechanism yields subsequences with naturally varying frame rates and no forward-time bias, providing the model with diverse temporal contexts during training.

4.4 Inference Modes

`LingBot-Map` supports two inference modes, Direct Output and Visual Odometry (VO), that share a common keyframe selection mechanism.

Keyframe Selection. When the input sequence exceeds the maximum training length, we employ an adaptive keyframe selection strategy to control KV-cache growth. For each incoming frame, the model first estimates its depth map and camera pose, then computes the optical flow relative to the most recent keyframe using the predicted pose and depth. If the flow magnitude exceeds a predefined threshold, the frame is designated as a new keyframe, whose features are appended to the KV cache; otherwise, it is discarded. This mechanism is shared by both inference modes.

Direct Output Mode. Direct mode is the default inference setting. The model processes frames causally through GCA, with the full three-level context (anchor, trajectory memory, and local window) accumulating continuously without reset. Each frame directly outputs an absolute camera pose and a dense depth map. In this mode, prediction errors accumulate solely from the model's frame-by-frame inference, without introducing any additional error from external alignment steps. Although the model is trained on sequences of up to 320 views, we empirically find that the Direct mode with

Table 1. Dataset composition, data format, scene type, and sampling proportions across the two training stages. Blank entries indicate that the dataset was not used in that stage.

Dataset	Format	Scene Type	Stage 1 Ratio (%)	Stage 2 Ratio (%)
BlendedMVS [92]	Multi-view	Single	1.1	1.9
HyperSim [52]	Multi-view	Single	5.8	5.7
MegaDepth [30]	Multi-view	Multi	3.9	-
Unreal4K [69]	Video	Single	1.0	0.9
WildRGBD [84]	Video	Single	5.8	1.9
TartanAir [81]	Video	Multi	3.9	7.6
TartanAirV2 [81]	Video	Multi	5.8	10.8
TartanGround [49]	Video	Multi	5.8	10.8
Waymo [65]	Video	Multi	-	0.9
PointOdyssey [101]	Video	Single	0.3	0.9
VirtualKITTI [3]	Video	Multi	0.8	1.9
Kubric [16]	Video	Single	0.3	0.9
DL3DV [37]	Video	Multi	11.0	5.7
MVS-Synth [19]	Multi-view	Single	1.7	-
GTA-SFM [76]	Multi-view	Single	1.7	-
CO3D [51]	Multi-view	Single	5.5	-
SceneRGBD [42]	Video	Single	7.3	5.7
Mapfree [1]	Video	Multi	3.9	1.5
Aria Synthetic [48]	Video	Single	7.3	5.7
ADT [48]	Video	Single	1.0	-
Objaverse [10]	Mesh	Single	2.7	-
Texverse [100]	Mesh	Single	2.7	-
ScanNet++ [95]	Video	Single	3.9	2.8
ScanNet [9]	Video	Single	1.9	2.8
MatrixCity [29]	Video	Multi	1.7	7.6
MidAir [14]	Video	Multi	2.9	5.7
KITTI-360 [34]	Video	Multi	-	3.8
Internal Game	Video	Multi	10.6	10.8
Gibson [83]	Mesh	Multi	-	2.6
Matterport3D [5]	Mesh	Multi	-	2.6
HM3D [50]	Mesh	Multi	-	2.6

keyframe selection remains stable for approximately $10\times$ the training length ($\sim$3,000 frames), beyond which prediction quality gradually degrades.

Visual Odometry (VO) Mode. For sequences that far exceed the Direct mode's effective range (e.g., tens of thousands of frames), we switch to VO mode. The input is partitioned into overlapping local windows. Within each window, an initial subset of frames is processed jointly to establish a robust local scale and coordinate system, and the remaining frames are processed causally via GCA with keyframe selection. At the end of each window, the model state is reset. To fuse successive windows into a single global trajectory, we compute a Sim(3) alignment between the overlapping regions of consecutive windows, recovering the relative scale, rotation, and translation. This enables `LingBot-Map` to process arbitrarily long sequences with bounded memory, at the cost of additional drift introduced at each window boundary: unlike Direct mode, VO mode incurs extra alignment error that compounds with the number of windows.

Trade-offs. Direct mode produces more accurate trajectories by avoiding inter-window alignment error, and is the preferred choice when the sequence length stays within $\sim$3,000 frames. For inputs that substantially exceed this range, VO mode generalizes more effectively at the cost of accumulated alignment drift at the window boundaries. In practice, the choice is determined by the sequence length and the required level of global consistency.

Default Inference Configuration. Unless otherwise specified, all experiments in this report use Direct Output Mode with a local pose-reference window size $k = 64$ and keyframe interval $m = 1$, at a resolution of 518×378 with bfloat16 precision. For the large-scale demo videos that span city-scale environments or extremely long sequences, we use VO mode.

5 Evaluation Benchmark

We establish a comprehensive evaluation benchmark covering camera pose estimation and 3D reconstruction across diverse indoor, outdoor, and large-scale environments.

5.1 Datasets

Streaming 3D reconstruction must generalize across a wide spectrum of scenarios, from object-centric captures to room-scale interiors and city-scale outdoor trajectories, under varying sequence lengths, camera motions, and scene complexities. To thoroughly evaluate this, our benchmark is built on five complementary datasets: Oxford Spires [67], ETH3D [57], 7-Scenes [59], Tanks and Temples [25], and NRGBD [2], which collectively cover indoor and outdoor environments, object-centric and multi-scene trajectories, short sequences (tens of frames) to long sequences (thousands of frames), and both synthetic-grade and real-world capture conditions. Below we describe each dataset and our evaluation configuration.

Oxford Spires [67] is a large-scale outdoor and indoor dataset captured across the historic Oxford campus, featuring complex scene transitions, revisits, and significant scale variation within each sequence. The dataset provides ground-truth camera trajectories from a high-precision LiDAR-inertial SLAM system. We select 13 scenes with available ground-truth trajectories: Keble College 02–05, Observatory Quarter 01–02, Blenheim Palace 01–02 and 05, Christ Church 01–03 and 05, and Bodleian Library 02. We use camera 0 as the viewpoint and evaluate under two settings: *sparse* (320 frames, sampled every 12 frames) to test single-pass reconstruction within our training range, and *dense* (3,840 frames) to stress-test long-sequence streaming capabilities.

ETH3D [57] provides high-resolution indoor and outdoor images with ground-truth depth maps acquired from a laser scanner. The scenes span offices, lecture rooms, relief structures, facades, playgrounds, terraces, and botanical gardens, covering both small-scale indoor environments and large-scale outdoor settings. We follow the evaluation configurations established in DA3 [36], using all available frames. We use a threshold of $d = 0.1$ m for the F1 reconstruction metric.

7-Scenes [59] is a widely-used indoor RGB-D dataset consisting of 7 scenes (Chess, Fire, Heads, Office, Pumpkin, Kitchen, Stairs) captured with a Kinect sensor at 640×480 resolution. It is a challenging real-world dataset: the images contain significant motion blur, and the scenes feature textureless surfaces and repetitive structures that are difficult for pose estimation. We downsample the number of frames for each scene by a stride of 5 to reduce redundancy from the high-framerate Kinect capture while retaining sufficient viewpoint coverage.

Tanks and Temples [25] is a large-scale outdoor dataset providing high-resolution images, depth maps from LiDAR, and ground-truth 3D shapes for benchmarking multi-view reconstruction. We select 6 scenes covering diverse structures: Barn, Caterpillar, Church, Ignatius, Meeting Room, and Truck, with all images included.

NRGBD [2] is a dataset designed for neural RGB-D surface reconstruction, containing indoor scenes with high-quality ground-truth depth from structured-light sensors. The scenes include cluttered room-scale environments with fine geometric details. We sample images with a stride of 5 and evaluate dense reconstruction quality using the F1 metric.

5.2 Metrics

Camera Pose Estimation. We evaluate camera pose accuracy using the Area Under the Curve (AUC) of the relative pose error at angular thresholds of 3° and 30°. For trajectory-level evaluation, we report the Absolute Trajectory Error (ATE) in meters, which measures global trajectory consistency after Sim(3) alignment, as well as Relative Pose Error for translation (RPE-trans) and rotation (RPE-rot), which capture local frame-to-frame accuracy.

3D Reconstruction. We evaluate reconstruction quality using the F1 score, Accuracy (Acc), and Completeness (Comp). The reconstructed and ground-truth point clouds are first aligned using the Umeyama [70] method, followed by ICP refinement (threshold 0.1). For ETH3D, which features large-scale scenes with relatively sparse ground-truth points, we

adopt an F1 threshold of 0.25 with a voxel size of 0.039m. For 7-Scenes and NRGBD, which contain many frames and dense depth projections producing tens of millions of points, we first downsample both point clouds to a uniform voxel grid (voxel size $4.0/512$) before ICP, and set the F1 threshold to 0.05.

6 Experiments

In this section, we compare LingBot-Map with state-of-the-art methods for camera pose estimation and 3D reconstruction, analyze efficiency, and conduct ablation studies. Training details and data are described in Sec. 4, the evaluation benchmark is defined in Sec. 5, and inference configuration is specified in Sec. 4.4.

6.1 Baseline Methods

We compare LingBot-Map against three categories of methods:

Offline Feed-Forward Models. These methods process all input frames simultaneously using bidirectional attention, producing globally consistent reconstructions in a single forward pass but requiring access to the full sequence. We include VGGT [74], DA3 [36], Fast3R [89], FastVGGT [58], and Pi3 [82].

Optimization-Based Methods. These methods iteratively refine camera poses and geometry through explicit optimization objectives such as reprojection error minimization or bundle adjustment. They typically achieve high accuracy but at significant computational cost. We include DroidSLAM [68], MegaSAM [31], and VIPE [18].

Streaming Methods. These methods process frames causally in a streaming fashion, predicting poses and depth without access to future frames. This is the same setting as LingBot-Map. We include StreamVGGT [102], SLAM3R [38], InfiniteVGGT [96], Spann3R [72], Stream3R [27], CUT3R [78], TTT3R [7], and Wint3R [32]. For fair comparison, all streaming methods are evaluated without resetting their internal state throughout each sequence.

6.2 Camera Pose Estimation

Large-Scale Trajectory Estimation on Oxford Spires. Oxford Spires is one of the most challenging benchmarks for streaming pose estimation: sequences span complex indoor-outdoor environments with abrupt scene transitions (e.g., outdoor courtyards to dark staircases), revisits to previously observed areas after long temporal gaps, and large scale variation across the trajectory. These properties demand both accurate local pose estimation and long-range global consistency, which are precisely the capabilities GCA is designed to provide. We evaluate under two settings to test both aspects.

In the *sparse* setting (320 frames, sampled every 12 frames), all method categories can run on our hardware, enabling a fair comparison across offline, optimization-based, and online approaches. As shown in Tab. 2, LingBot-Map achieves the best results on nearly all metrics. Despite operating in a streaming online manner, our method surpasses the strongest offline baselines by a large margin. LingBot-Map achieves an AUC@15 of 61.64, substantially exceeding the best offline method DA3 [36] (49.84) and more than doubling VGGT [74] (23.84). On trajectory-level accuracy, LingBot-Map reduces ATE from 12.87 (DA3) and 24.78 (VGGT) to 6.42. We attribute this to a fundamental limitation of existing offline methods: they are trained on datasets with a small number of viewpoints where consecutive frames remain close to each other and largely observe the same local region. When confronted with the complex scene transitions and large viewpoint changes in Oxford Spires, the learned priors fail to transfer. Compared with optimization-based methods, which explicitly minimize reprojection error across frames, LingBot-Map still achieves superior performance. LingBot-Map outperforms the best optimization-based approach VIPE [18] on both pose accuracy (AUC@15: 61.64 vs. 45.35) and trajectory consistency (ATE: 6.42 vs. 10.52). VIPE relies on iterative bundle adjustment and is computationally expensive; in contrast, LingBot-Map achieves this accuracy in a single forward pass. Among online streaming methods, the performance gap is even more pronounced. All competing approaches suffer from severe memory forgetting: as the sequence progresses, they lose track of previously observed geometry, leading to accumulated drift. The best online competitor, CUT3R [78], achieves an AUC@15 of only 5.98 and ATE of 18.16, while LingBot-Map achieves 61.64 and 6.42 respectively, a $10\times$ improvement in pose accuracy and $2.8\times$ reduction in trajectory error (see Fig. 5(a) for qualitative comparison).

Table 2. Pose and trajectory accuracy comparison on Oxford Spires (sparse setting). Our method achieves the best performance on most metrics among prior offline, optimization-based, and online methods. The **best** and second best results are marked.

Methods	Type	AUC@15 ↑	AUC@30 ↑	ATE ↓	RPE-trans ↓	RPE-Rot ↓
Fast3R [89]	offline	1.20	2.99	34.80	8.21	59.51
VGGT [74]	offline	23.84	35.09	24.78	8.87	22.79
DA3 [36]	offline	49.84	56.68	12.87	3.22	16.17
FastVGGT [58]	offline	21.68	34.64	22.43	7.25	16.12
Pi3 [82]	offline	38.64	48.65	14.03	2.58	10.33
DroidSLAM [68]	optim	8.58	21.41	21.84	1.02	6.90
MegaSAM [31]	optim	15.91	28.03	13.80	0.76	7.24
VIPE [18]	optim	45.35	51.88	10.52	**0.43**	5.98
StreamVGGT [102]	online	10.91	17.04	28.41	6.35	16.28
SLAM3R [38]	online	1.67	5.10	29.69	7.57	27.50
InfiniteVGGT [96]	online	10.25	16.33	30.49	5.72	15.01
Spann3R [72]	online	2.06	5.09	32.12	3.54	14.37
Stream3R-w [27]	online	6.56	11.03	33.03	4.73	16.79
Stream3R [27]	online	9.67	15.21	29.58	6.67	16.90
CUT3R [78]	online	5.98	14.95	18.16	1.17	7.18
TTT3R [7]	online	13.92	25.90	19.35	2.28	13.30
Wint3R [32]	online	11.61	23.42	21.10	1.62	6.27
LingBot-Map (Ours)	online	**61.64**	**75.16**	**6.42**	1.01	**3.70**

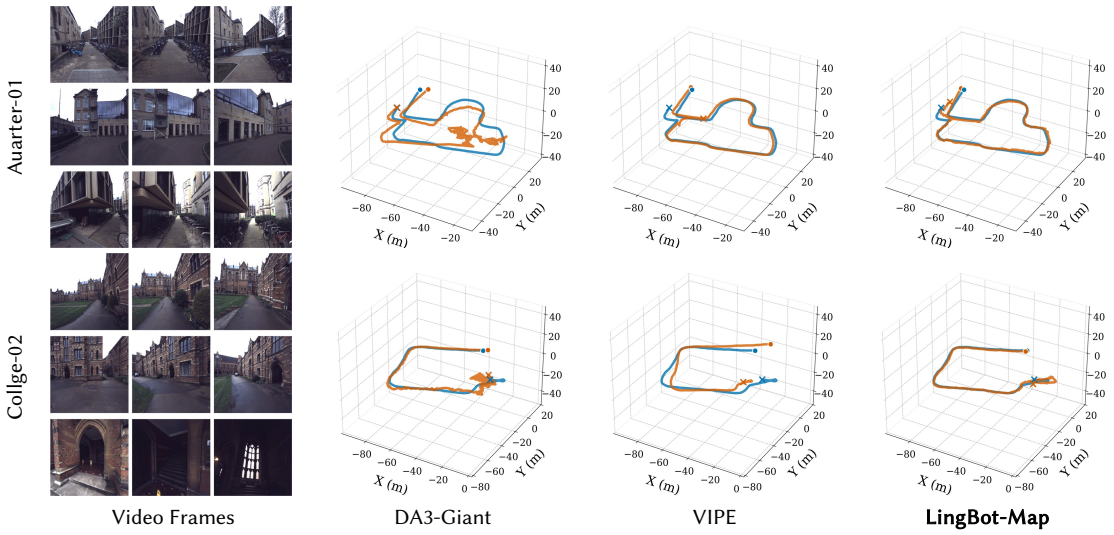
Table 3. Sparse vs. dense trajectory accuracy on Oxford Spires. We compare ATE under the sparse (320 frames) and dense (3,840 frames) settings. ΔATE measures the degradation from sparse to dense. LingBot-Map maintains near-constant accuracy while competing methods degrade significantly. The **best** and second best results are marked.

	CUT3R	TTT3R	Wint3R	Inf.-VGGT	Stream3R-w	LingBot-Map
ATE_{sparse} ↓	18.16	19.35	21.10	30.49	33.03	**6.42**
ATE_{dense} ↓	32.47 (+14.31)	25.05 (+5.70)	32.90 (+11.80)	31.75 (+1.26)	33.73 (+0.70)	**7.11** (+0.69)
FPS ↑	**29.21**	28.97	3.88	7.78	13.66	20.29

In the *dense* setting (full 3,840 frames), we stress-test long-sequence streaming capabilities (Tab. 3). This setting is particularly revealing: as the trajectory length increases from 320 to 3,840 frames, most feed-forward methods degrade dramatically due to accumulated drift. CUT3R's ATE rises from 18.16 to 32.47 (1.8× increase), and Wint3R degrades from 21.10 to 32.90. In contrast, LingBot-Map maintains consistently low error (6.42 → 7.11), with only a marginal increase of 0.69 over a 12× longer sequence. This demonstrates that the three-level context structure of GCA (anchor, trajectory memory, and local window) effectively preserves long-range geometric consistency without explicit optimization or loop closure. Notably, LingBot-Map achieves competitive inference speed at 20.29 FPS while maintaining the best trajectory accuracy among all streaming methods.

Generalization on Diverse Benchmarks. To verify that the strong performance on Oxford Spires is not specific to large-scale outdoor trajectories, we evaluate on three additional benchmarks that cover fundamentally different scales and scene types: ETH3D [57] (mixed indoor and outdoor scenes with laser-scanned ground-truth depth), 7-Scenes [59] (room-scale RGB-D sequences with textureless surfaces and significant motion blur), and Tanks and Temples [25] (outdoor multi-view captures of large structures). As shown in Tab. 4, LingBot-Map consistently outperforms all competing streaming methods by a substantial margin across all three datasets and all metrics.

(a) Comparison with SOTA offline and optimization-based methods on Oxford-Spires



(b) Comparison with SOTA streaming methods across diverse scenes

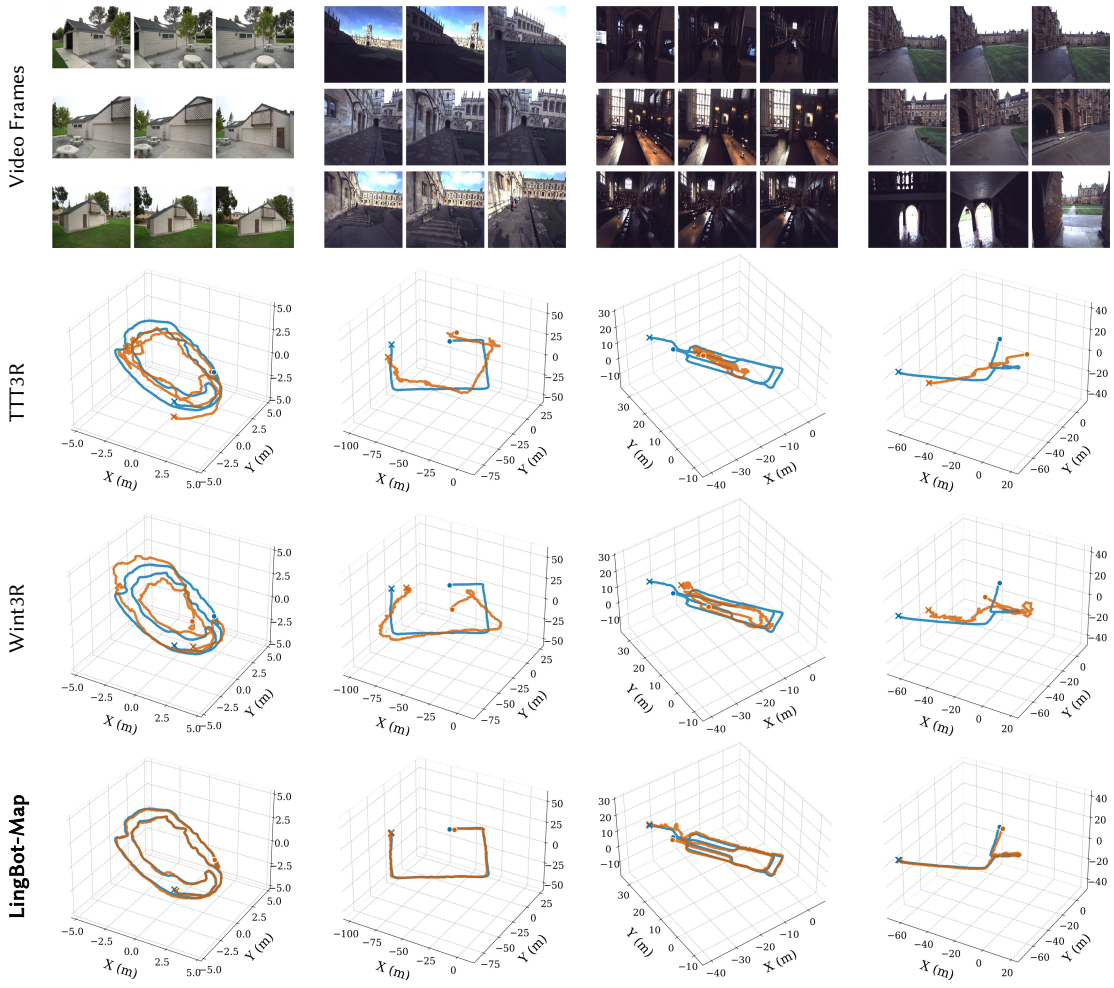


Figure 5. Trajectory comparison. (a) On Oxford-Spires, LingBot-Map even outperforms bidirectional (DA3-Giant), and optimization-based (ViPE) methods, accurately preserving trajectories during complex outdoor-to-indoor transitions and dark staircases. (b) Across Tanks and Temples and additional Oxford-Spires scenes, our method consistently produces accurate trajectories while competing streaming methods suffer from severe drift. Ground truth in blue, predicted in orange; start •, end ×.

Table 4. Pose and trajectory accuracy comparison on ETH3D, 7-Scenes, and Tanks & Temples. Results on ETH3D, 7-Scenes, and Tanks & Temples show that our method achieves the best performance across all datasets. The **best** and second best results are marked.

Methods	Type	ETH3D			7-Scenes			Tanks & Temples		
		Auc3 ↑	Auc30 ↑	ATE ↓	Auc3 ↑	Auc30 ↑	ATE ↓	Auc3 ↑	Auc30 ↑	ATE ↓
SLAM3R [38]	online	1.46	22.95	1.98	4.79	66.92	0.11	2.87	47.92	1.42
Spann3R [72]	online	1.13	23.02	2.10	2.79	57.87	0.20	2.22	32.22	2.11
InfiniteVGGT [96]	online	12.00	62.20	1.46	8.45	73.40	0.12	21.69	77.76	1.00
CUT3R [78]	online	10.63	57.77	1.43	1.50	42.44	0.29	1.71	25.19	1.79
TTT3R [7]	online	9.98	56.12	1.22	5.52	71.23	0.10	9.01	71.30	0.66
Wint3R [32]	online	11.31	58.71	0.86	2.74	63.02	0.12	3.84	57.85	0.88
Stream3R [27]	online	13.73	64.76	1.67	9.31	73.70	0.10	39.27	81.33	0.76
Stream3R-w [27]	online	9.00	58.69	1.71	7.47	61.70	0.25	24.69	72.30	1.22
LingBot-Map (Ours)	online	**27.79**	**86.20**	**0.22**	**12.63**	**78.59**	**0.08**	**45.80**	**92.80**	**0.20**

On Tanks and Temples, LingBot-Map achieves an AUC@30 of 92.80 and ATE of 0.20, improving over the runner-up Stream3R (AUC@30: 81.33, ATE: 0.76) by $+11.47$ AUC points and $3.8\times$ lower ATE. On ETH3D, our ATE of 0.22 is nearly $4\times$ lower than the second-best Wint3R (0.86), indicating that our model handles both the precise indoor geometry and the broader outdoor structures in this dataset. On 7-Scenes, LingBot-Map achieves the lowest ATE of 0.08, confirming robust performance even on room-scale indoor scenes, where the main challenges are textureless walls, repetitive structures, and heavy motion blur rather than trajectory length. Taken together, these results demonstrate that LingBot-Map is not a specialist for any particular scenario but a general-purpose streaming pose estimator that scales from small rooms to city-scale environments.

Qualitative trajectory comparisons are shown in Fig. 5. In part (a), on Oxford Spires, LingBot-Map accurately tracks the camera through complex outdoor-to-indoor transitions and dark staircases, while DA3-Giant and ViPE both exhibit significant trajectory drift. In part (b), across Tanks and Temples and additional Oxford Spires scenes, competing streaming methods (TTT3R, Wint3R) produce trajectories that progressively diverge from the ground truth, whereas LingBot-Map consistently maintains close alignment throughout the full sequence.

6.3 3D Reconstruction

We evaluate the 3D reconstruction quality of LingBot-Map on ETH3D, 7-Scenes, and NRGBD [2]. The evaluation protocol and metrics (Accuracy, Completeness, F1) are described in Sec. 5.2. Quantitative results are summarized in Tab. 5.

Since reconstruction quality depends directly on pose accuracy and depth estimation, the improvements reported in the previous section translate into substantial gains in 3D reconstruction. LingBot-Map achieves the best F1 score on all three datasets. On ETH3D, LingBot-Map reaches an F1 of 98.98, outperforming the runner-up Wint3R (77.28) by +21.70 points. The improvement comes from both better accuracy (0.09 vs. 0.28) and completeness (0.03 vs. 0.21), indicating that our reconstructions are not only more precise but also more complete in their coverage of the scene. On 7-Scenes, LingBot-Map achieves an F1 of 80.39 with the best accuracy (0.02) and completeness (0.07), matching or exceeding all baselines. The gains here are more modest because the room-scale scenes have limited trajectory length, and most methods already perform reasonably well; nonetheless, LingBot-Map still ranks first. On NRGBD, the advantage is most pronounced: LingBot-Map achieves an F1 of 64.26, improving over Wint3R (56.96) by +7.30 points. NRGBD contains cluttered indoor environments with fine geometric details, where accumulated pose drift in competing methods leads to blurred or duplicated surfaces in the reconstruction. Our drift-resistant trajectory estimation directly benefits reconstruction fidelity in these scenarios.

A qualitative comparison is shown in Fig. 6. In simpler scenes (top rows), all methods produce reasonable reconstructions, but TTT3R and Wint3R already show noticeable misalignment at building edges. As scene complexity increases (middle rows), competing methods begin to produce duplicated structures and blurred surfaces, a direct consequence of accumulated pose drift causing the same geometry to be projected to different locations. In the most challenging multi-building outdoor scenes (bottom rows), the gap becomes striking: TTT3R and Wint3R lose

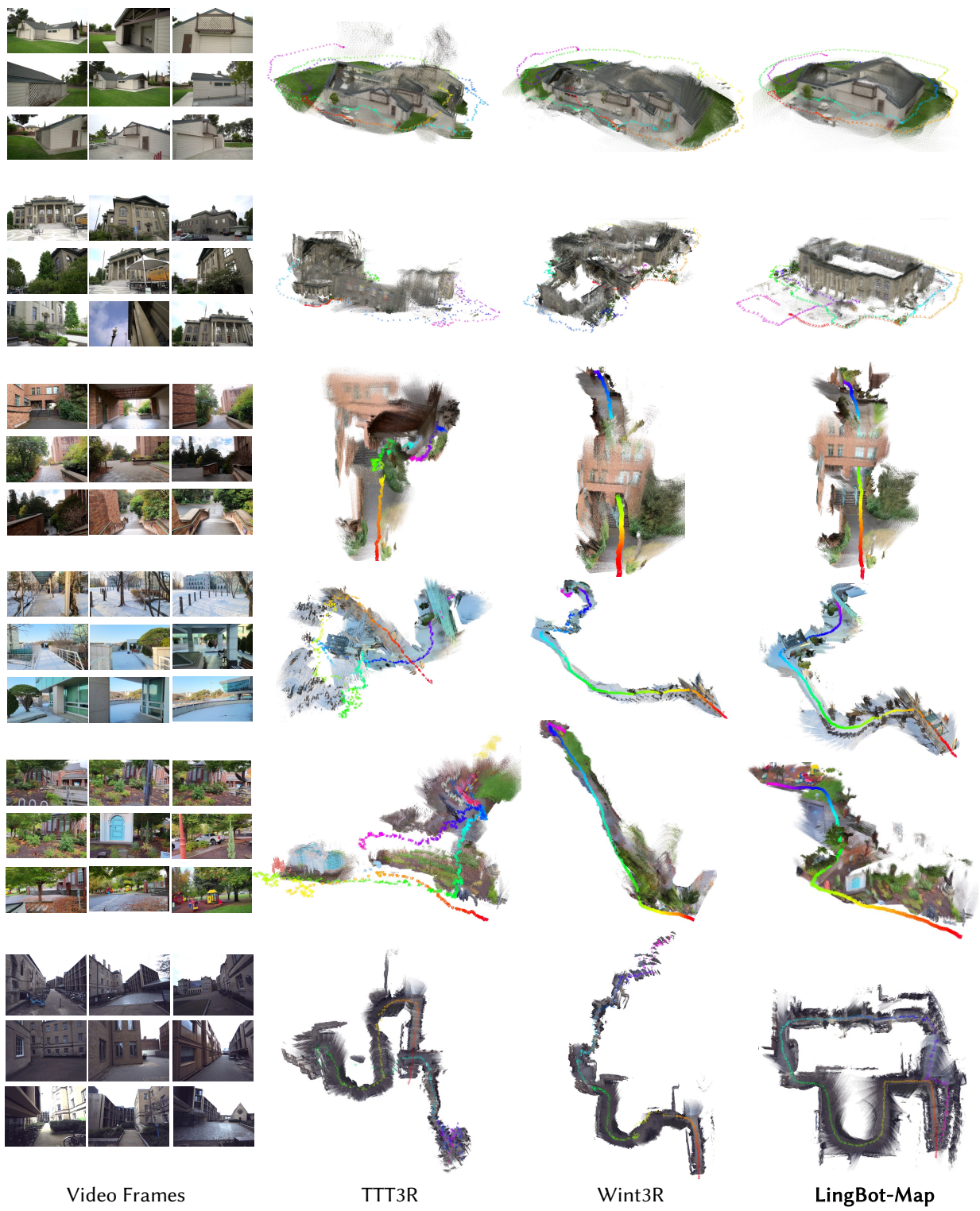


Figure 6. Qualitative comparison of point cloud reconstruction. LingBot-Map produces geometrically coherent and temporally stable reconstructions, preserving sharp building edges and continuous surfaces even over long sequences with revisits. TTT3R and Wint3R suffer from trajectory drift that causes fragmented geometry and duplicated structures, with degradation most severe in complex multi-building scenes (bottom rows).

Table 5. Point cloud reconstruction comparison on ETH3D, 7-Scenes, and NRGBD. Our method achieves the best results on Accuracy, Completeness, and F1 score. The **best** and second best results are marked.

Methods	Type	ETH3D			7-Scenes			NRGBD		
		Acc ↓	Comp ↓	F1 ↑	Acc ↓	Comp ↓	F1 ↑	Acc ↓	Comp ↓	F1 ↑
StreamVGGT [102]	online	0.64	0.34	58.11	0.04	0.11	69.44	0.13	0.05	45.08
InfiniteVGGT [96]	online	0.65	0.35	57.69	0.04	0.11	68.53	0.13	0.05	42.27
CUT3R [78]	online	0.57	0.50	67.63	0.07	0.10	58.98	0.25	0.15	32.22
TTT3R [7]	online	0.41	0.22	68.48	0.03	0.08	77.25	0.16	0.06	53.55
Wint3R [32]	online	0.28	0.21	77.28	0.03	**0.07**	78.81	0.09	0.04	56.96
Stream3R [27]	online	0.44	0.28	72.87	**0.02**	0.09	78.79	0.21	0.07	54.07
Stream3R-w [27]	online	0.58	0.37	67.09	0.04	0.15	71.94	0.20	0.06	53.74
LingBot-Map (Ours)	online	**0.09**	**0.03**	**98.98**	**0.02**	**0.07**	**80.39**	**0.07**	**0.03**	**64.26**

spatial coherence entirely, producing collapsed and fragmented point clouds where individual buildings are no longer distinguishable. In contrast, LingBot-Map maintains clean geometry with sharp structural edges and continuous wall surfaces throughout, demonstrating that the long-range consistency provided by GCA directly translates to high-fidelity 3D reconstruction.

6.4 Ablation Study

We conduct ablation studies to analyze the contributions of each component in GCA. We use TartanAir and TartanGround as the ablation testbed for their high-quality ground-truth annotations, complex scene geometry, and long trajectories (up to thousands of frames), which provide a controlled yet challenging setting that isolates the effect of each component. We initialize the model from the first-stage checkpoint and fine-tune on these datasets, evaluating on the TartanGround validation set with sequences of 320 frames sampled at a stride of 8 (spanning $\sim$2,400 frames in temporal extent). We use a learning rate of 5×10^{-5} and follow the same progressive view curriculum as the main streaming training (Sec. 4.2). Each ablation experiment requires approximately 3,840 GPU hours.

Results are shown in Tab. 6. Starting from a baseline with only the relative pose loss (row 1), we progressively add each component and measure its impact.

Anchor Initialization. Adding anchor initialization (row 1 $\rightarrow$ row 2) boosts AUC@3 from 9.80 to 13.63 (+3.83) and reduces ATE from 8.59 to 7.88. Monocular streaming reconstruction is inherently scale-ambiguous: without an explicit geometric reference, the model must implicitly infer both the coordinate origin and the absolute scale from the data, which becomes increasingly unreliable as the sequence grows. The anchor context resolves this by designating the first n frames as fixed references that establish the scale and coordinate system before streaming begins. The improvement in AUC@3 (+3.83) confirms that the anchor mechanism improves not just global trajectory consistency but also local pose accuracy, as each new frame can now be registered against a well-defined geometric reference.

Context Tokens (Trajectory Memory). Adding context tokens on top of anchor initialization (row 2 $\rightarrow$ row 4) further improves AUC@3 from 13.63 to 15.75 and reduces ATE from 7.88 to 7.46. The anchor context and local window together provide a fixed global reference and dense recent observations, but without any record of intermediate frames, pose errors accumulate unchecked between the anchor and the current window. Context tokens address this by retaining a compact 6-token summary for each evicted frame, preserving the key geometric cues of the full observation history at minimal memory cost. The consistent improvement across all metrics (AUC@3: +2.12, AUC@30: +1.21, ATE: -0.42) validates that even a lightweight trajectory memory meaningfully reduces long-range drift.

Relative Pose Loss. Comparing row 3 (without Rel. Loss) and row 4 (with Rel. Loss) isolates the effect of relative pose supervision. Without it, RPE-rot degrades sharply from 2.26 to 5.35 (2.4$\times$ worse) and ATE increases from 7.46 to 8.25. The relative pose loss supervises all frame pairs within the local pose-reference window, directly constraining the frame-to-frame relative motion. This is complementary to the absolute pose loss: while the absolute loss anchors each frame's pose in the global coordinate system, the relative loss enforces local geometric consistency within the sliding

Table 6. Ablation study on long-sequence pose estimation and trajectory accuracy. All components contribute significantly to the final performance. The **best** results are marked.

Rel. Loss	A. Init.	Co. Tok.	V. RoPE	AUC@3 ↑	AUC@30 ↑	ATE ↓	RPE-trans ↓	RPE-rot ↓
✓				9.80	65.84	8.59	1.62	2.57
✓	✓			13.63	68.71	7.88	1.60	2.90
	✓	✓		13.91	68.25	8.25	1.67	5.35
✓	✓	✓		15.75	69.92	7.46	1.48	2.26
✓	✓	✓	✓	**16.39**	**71.87**	**5.98**	**1.33**	**1.93**

Table 7. Pose-reference window (size 64) vs. full causal attention. The bounded window achieves better trajectory accuracy with 1.7× higher speed and 2.7× lower memory. The **best** and second best results are marked.

Window Size	ATE ↓	RPE-trans ↓	RPE-rot ↓	FPS↑	Mem (GB)↓
64	**5.98**	**1.33**	1.93	**20.29**	**13.28**
Full	6.60	1.50	**1.71**	11.87	36.06

window, preventing the small per-frame errors that compound into trajectory drift over long sequences. Notably, the rotation degradation (+3.09 in RPE-rot) is far more severe than the translation degradation, suggesting that rotation estimation is particularly sensitive to the lack of local pairwise supervision.

Video RoPE. The final component, Video RoPE (row 4 → row 5), yields the single largest ATE improvement: 7.46 → 5.98 (−1.48), along with gains across all other metrics (AUC@3: +0.64, AUC@30: +1.95, RPE-trans: −0.15). Without temporal positional encoding, the trajectory memory tokens carry geometric information but lack any notion of when each frame was observed. Video RoPE injects temporal ordering directly into the attention computation, enabling the model to reason about the sequential structure of the trajectory: how far apart two frames are in time, and in which direction the camera has been moving. The disproportionately large ATE improvement (−1.48 vs. the −0.42 from context tokens alone) suggests that temporal ordering is the missing ingredient that allows the trajectory memory to realize its full potential for correcting long-range drift.

Pose-Reference Window vs. Full Attention. Finally, we compare GCA's bounded pose-reference window (size 64) against full causal attention that retains all historical tokens (Tab. 7). The pose-reference window yields a 1.7× speedup (11.87 → 20.29 FPS) and a 2.7× memory reduction (36.06 → 13.28 GB). More importantly, the bounded window also improves trajectory accuracy: ATE decreases from 6.60 to 5.98 and RPE-trans from 1.50 to 1.33. This counterintuitive result can be explained by the fact that retaining all historical image tokens introduces noise from distant, less relevant frames that can confuse the attention computation. GCA's design, which evicts image tokens but preserves compact context tokens for the full trajectory, retains the essential geometric cues while filtering out redundant information. The only metric where full attention leads is RPE-rot (1.71 vs. 1.93), suggesting that dense historical tokens provide slightly richer rotational cues at the local level, but this marginal benefit is far outweighed by the efficiency gains and the improved global trajectory consistency. As sequence length grows further, the gap widens: full attention's memory and compute scale quadratically with the number of frames, while GCA's cost remains nearly constant (6 tokens per evicted frame), making it the only viable option for streaming at scale.

7 Conclusion and Discussion

We have presented `LingBot-Map`, a streaming foundation model for long-range 3D reconstruction from continuous visual input. At its core is Geometric Context Attention (GCA), which decomposes the streaming state into three complementary context types — anchor, local pose-reference window, and trajectory memory — inspired by the structure of classical SLAM systems but learned end-to-end. This design reduces per-frame context growth by roughly 80× compared to causal attention, enabling stable inference over arbitrarily long sequences at around 20 FPS. Extensive evaluations across multiple benchmarks demonstrate that `LingBot-Map` achieves state-of-the-art performance among

streaming methods, and even surpasses offline and optimization-based approaches on large-scale datasets such as Oxford Spires. By enabling accurate, real-time dense 3D reconstruction from continuous visual streams, LingBot-Map opens the door to a wide range of applications, including autonomous navigation, augmented reality, and, most notably, embodied AI systems that require persistent, on-the-fly spatial understanding to interact with the physical world.

Limitations. While LingBot-Map demonstrates strong performance on diverse benchmarks, several limitations remain. First, the model currently does not incorporate explicit loop-closure detection, which could further reduce accumulated drift when revisiting previously observed regions. Second, the trajectory memory compression into a fixed number of tokens per frame may lose fine-grained geometric details that could be beneficial for very long sequences spanning tens of thousands of frames. Third, like other feed-forward methods, our approach does not perform test-time optimization, which could further refine the reconstruction quality in challenging scenarios.

Future Directions. A promising direction is to incorporate bundle-adjustment-like refinement and explicit loop-closure detection into the attention mechanism, further closing the gap with classical SLAM backends while retaining end-to-end differentiability. Additionally, extending LingBot-Map to handle dynamic scenes with moving objects, integrating multi-modal inputs such as LiDAR or IMU data, and exploring the model as a backbone for downstream applications such as novel view synthesis and navigation are exciting avenues for future work.

Acknowledgements

We thank Shangzhan Zhang, Jianyuan Wang, Yudong Jin, Christian Rupprecht, and Xun Cao for their helpful discussions and support.

References

[1] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Aron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *Eur. Conf. Comput. Vis.*, pages 690–708. Springer, 2022.

[2] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6290–6301, 2022.

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

[4] 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.

[5] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. In *International Conference on 3D Vision (3DV)*, 2017.

[6] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3r: Estimating disentangled motion from dust3r without training. In *Int. Conf. Comput. Vis.*, pages 9158–9168, 2025.

[7] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *Int. Conf. Learn. Represent.*, 2026.

[8] Zhuoguang Chen, Minghui Qin, Tianyuan Yuan, Zhe Liu, and Hang Zhao. Long3r: Long sequence streaming 3d reconstruction. In *Int. Conf. Comput. Vis.*, pages 5273–5284, 2025.

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5828–5839, 2017.

[10] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13142–13153, 2023.

[11] Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. Vggt-long: Chunk it, loop it, align it–pushing vggt's limits on kilometer-scale long rgb sequences. *arXiv preprint arXiv:2507.16443*, 2025.

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[13] Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *Int. Conf. Comput. Vis.*, pages 8503–8513, 2025.

[14] Michael Fonder and Marc Van Droogenbroeck. Mid-air: A multi-modal dataset for extremely low altitude drone flights. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, June 2019.

[15] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

[16] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3749–3761, 2022.

[17] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. In *Int. Conf. Learn. Represent.*, 2024.

[18] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025.

[19] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2821–2830, 2018.

[20] Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.

[21] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3r: Empowering unconstrained 3d reconstruction with camera and scene priors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1071–1081, 2025.

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

[23] Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4D: Learning How Things Move in 3D from Internet Stereo Videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

[24] 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.

[25] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4), 2017.

[26] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[27] 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. In *Int. Conf. Learn. Represent.*, 2026.

[28] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3D with sparse-view generation and large reconstruction model. In *Int. Conf. Learn. Represent.*, 2024.

[29] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Int. Conf. Comput. Vis.*, pages 3205–3215, 2023.

[30] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2041–2050, 2018.

[31] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10486–10496, 2025.

[32] Zizun Li, Jianjun Zhou, Yifan Wang, Haoyu Guo, Wenzheng Chang, Yang Zhou, Haoyi Zhu, Junyi Chen, Chunhua Shen, and Tong He. Wint3r: Window-based streaming reconstruction with camera token pool. In *Int. Conf. Learn. Represent.*, 2026.

[33] Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, Andrew Gu, Chien-Chin Huang, Iris Zhang, Wei Feng, Howard Huang, Junjie Wang, Sanket Purandare, Gokul Nadathur, and Stratos Idreos. Torchtitan: One-stop pytorch native solution for production ready LLM pretraining. In *Int. Conf. Learn. Represent.*, 2025.

[34] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(3):3292–3310, 2022.

[35] Chin-Yang Lin, Cheng Sun, Fu-En Yang, Min-Hung Chen, Yen-Yu Lin, and Yu-Lun Liu. Longsplat: Robust unposed 3d gaussian splatting for casual long videos. In *Int. Conf. Comput. Vis.*, pages 27412–27422, 2025.

[36] Haotong Lin, Sili Chen, Jun Hao 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.

[37] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 22160–22169, 2024.

[38] Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16651–16662, 2025.

[39] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3r: Aligned monocular depth estimation for dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22820–22830, 2025.

[40] Yuanxun Lu, Jingyang Zhang, Tian Fang, Jean-Daniel Nahmias, Yanghai Tsin, Long Quan, Xun Cao, Yao Yao, and Shiwei Li. Matrix3d: Large photogrammetry model all-in-one. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11250–11263, 2025.

[41] Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *Adv. Neural Inform. Process. Syst.*, 39, 2025.

[42] 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 *Int. Conf. Comput. Vis.*, 2017.

[43] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[44] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[45] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16695–16705, 2025.

[46] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[47] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pages 58–77. Springer, 2024.

[48] 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 *Int. Conf. Comput. Vis.*, pages 20133–20143, 2023.

[49] Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. Tartanground: A large-scale dataset for ground robot perception and navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 20524–20531. IEEE, 2025.

[50] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI. In *Adv. Neural Inform. Process. Syst.*, 2021.

[51] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Int. Conf. Comput. Vis.*, pages 10901–10911, 2021.

[52] 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 *Int. Conf. Comput. Vis.*, pages 10912–10922, 2021.

[53] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[54] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Int. Conf. Comput. Vis.*, pages 9339–9347, 2019.

[55] 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.

[56] 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.

[57] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3260–3269, 2017.

[58] You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.

[59] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2930–2937, 2013.

[60] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024.

[61] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, 2006.

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

[63] Edgar Sucar, Zihang Lai, Eldar Insafutdinov, and Andrea Vedaldi. Dynamic point maps: A versatile representation for dynamic 3d reconstruction. In *Int. Conf. Comput. Vis.*, pages 7295–7305, 2025.

[64] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[65] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2446–2454, 2020.

[66] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large multi-view gaussian model for high-resolution 3D content creation. In *Eur. Conf. Comput. Vis.*, 2024.

[67] Yifu Tao, Miguel Ángel Muñoz-Bañón, Lintong Zhang, Jiahao Wang, Lanke Frank Tarimo Fu, and Maurice Fallon. The oxford spires dataset: Benchmarking large-scale lidar-visual localisation, reconstruction and radiance field methods. *International Journal of Robotics Research*, 2025.

[68] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Adv. Neural Inform. Process. Syst.*, 34:16558–16569, 2021.

[69] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8942–8952, 2021.

[70] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376–380, 2002.

[71] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[72] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *3DV*, pages 78–89. IEEE, 2025.

[73] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2026.

[74] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

[75] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024.

[76] Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020.

[77] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. PF-LRM: Pose-free large reconstruction model for joint pose and shape prediction. In *Int. Conf. Learn. Represent.*, 2024.

[78] Qianqian Wang*, Yifei Zhang*, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

[79] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10510–10522, 2025.

[80] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 20697–20709, 2024.

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

[82] 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.

[83] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[84] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 22378–22389, 2024.

[85] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Iurii Makarov, Bingyi Kang, Xin Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *Int. Conf. Comput. Vis.*, 2025.

[86] Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, and Xiaowei Zhou. Scal3r: Scalable test-time training for large-scale 3d reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2026.

[87] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. GRM: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *Eur. Conf. Comput. Vis.*, 2024.

[88] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Kai Zhang. DMV3D: Denoising multi-view diffusion using 3D large reconstruction model. In *Int. Conf. Learn. Represent.*, 2024.

[89] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21924–21935, 2025.

[90] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

[91] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5525–5534, 2019.

[92] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1790–1799, 2020.

[93] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *Int. Conf. Learn. Represent.*, 2025.

[94] Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, and Luis Ceze. Flashinfer: Efficient and customizable attention engine for llm inference serving. *arXiv preprint arXiv:2501.01005*, 2025.

[95] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Int. Conf. Comput. Vis.*, pages 12–22, 2023.

[96] 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.

[97] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *Int. Conf. Learn. Represent.*, 2025.

[98] Junyi Zhang, Charles Herrmann, Junhwa Hur, Chen Sun, Ming-Hsuan Yang, Forrester Cole, Trevor Darrell, and Deqing Sun. Loger: Long-context geometric reconstruction with hybrid memory. *arXiv preprint arXiv:2603.03269*, 2026.

[99] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21936–21947, 2025.

[100] Yibo Zhang, Li Zhang, Rui Ma, and Nan Cao. Texverse: A universe of 3d objects with high-resolution textures. *arXiv preprint arXiv:2508.10868*, 2025.

[101] 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 *Int. Conf. Comput. Vis.*, pages 19855–19865, 2023.

[102] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. In *Int. Conf. Learn. Represent.*, 2026.

A Data Processing Pipeline

Training a general-purpose streaming 3D reconstruction model requires large-scale, diverse data with accurate ground-truth camera poses and depth maps. Our training corpus (Section 4.3) spans 29 datasets from heterogeneous sources, each with different data formats, coordinate conventions, depth representations, and quality characteristics. To unify these into a consistent training pipeline, we develop a multi-stage data processing strategy: (1) *Existing data processing* (Sec. A.1): standardizing publicly available datasets by converting coordinate systems, normalizing depth scales, filtering corrupted frames, and unifying metadata formats; (2) *Synthetic data generation* (Sec. A.2): rendering additional training data from 3D environments with controlled camera trajectories and pixel-perfect ground truth; (3) *Gaming data processing* (Sec. A.3): extracting high-quality pose and depth annotations from internal game engine captures, which provide long trajectories through large-scale, visually rich environments; (4) *MatrixCity data sequencing* (Sec. A.4): reorganizing the MatrixCity aerial and street-level data into temporally continuous sequences via random walks on spatial topologies, enabling its use for sequential training; (5) *Cross-scene traversal data* (Sec. A.5): rendering long-range cross-room RGBD video sequences from large-scale 3D scene reconstructions using Habitat-Sim, providing the multi-room traversal signals that existing indoor datasets lack. Below we describe each pipeline in detail.

A.1 Existing Data Processing

To facilitate multi-dataset training, we standardize all 29 publicly available datasets into a unified format through a series of preprocessing steps.

Coordinate System Unification. Datasets store camera poses using different conventions. We convert all poses to a consistent camera to world representation. For datasets with non-standard axis conventions, such as UnReal4K [69], an additional rotation matrix is applied to align the coordinate system with the OpenCV standard. Intrinsic parameters are either extracted from per-frame calibration files (*e.g.*, .npz) or set to known constant values for datasets with a fixed camera model, such as TartanAir [81].

Depth Scale Normalization. The raw depth maps are provided in diverse formats and units. For instance, ScanNet and ScanNet++ store depth in millimeters as 16-bit PNG files, which are converted by dividing by 1000. VirtualKITTI2 uses centimeters stored in 16-bit color images, requiring division by 100. Other datasets, including DL3DV, HyperSim, TartanAir, and BlendedMVS, provide raw float .npy arrays in meters, while Waymo utilizes OpenEXR floating-point maps. All depth values are uniformly converted to meters and stored as float32.

Corrupted Frame Filtering. We apply several validation checks to remove corrupted or degenerate frames. First, scenes are discarded if the number of RGB images, depth maps, and camera pose files is inconsistent. Second, sequences with fewer than a minimum threshold of valid frames are excluded. Third, invalid depth values such as NaN or Inf are set to zero. Fourth, for datasets with noisy depth like DL3DV and MVS Synth, values exceeding the 98th percentile or an absolute threshold (*e.g.*, 1000m) are clamped to zero. Fifth, for outdoor datasets such as DL3DV and Mapfree, sky regions are identified using pre-computed masks or depth saturation thresholds and their corresponding depth is set to zero. Finally, for DL3DV and Mapfree, pre-computed outlier masks are used to further remove geometrically inconsistent depth estimates.

Metadata Format Unification. All datasets are converted into a common metadata structure, which is serialized as pickle files. This structure contains scene lists, per-frame index mappings (sceneids, id ranges), paths to images and depth maps, intrinsic matrices, and camera trajectories as 4×4 pose matrices. This unified format is generated and cached during the initial execution, which enables efficient composition of multiple datasets at training time without requiring repeated parsing of the raw data.

A.2 Generation Data Settings

We render multi-view images of 3D assets in Objaverse [10] and Texverse [100] using Blender Cycles. Each scene is normalized to fit within $[-0.9, 0.9]^3$. A pinhole camera with $\theta_h \sim \mathcal{U}(40^\circ, 70^\circ)$ and focal length $f = w/(2\tan(\theta_h/2))$ captures 512×512 images. Camera positions are sampled on a sphere around the origin at multiple elevations, all oriented toward the scene center. We apply HDR environment lighting and render RGBA color and metric depth in OpenEXR format. Images are tonemapped using percentile-based normalization with $\gamma{=}1/2.2$ correction.

A.3 Gaming Data Processing Pipeline

We introduce a runtime acquisition pipeline that captures dense visual and geometric annotations from modern game engines. This provides a scalable source of large-scale, accurately annotated, and stylistically diverse long-sequence 3D data that is otherwise difficult to obtain from real-world captures alone. All sequences maintain high visual quality with smooth camera motion and sufficient inter-frame overlap, free of motion blur, over-exposure, or under-exposure artifacts. The dataset covers a wide variety of viewpoints including forward, lateral, and vertical gaze directions, and incorporates both intra-sequence and inter-sequence field-of-view variations to improve robustness to focal-length changes. Cutscenes, UI overlays, and other non-gameplay frames are excluded, and all graphics settings remain constant within each game to ensure consistent annotation quality across the entire dataset.

To ensure sufficient diversity in scene types and camera behaviors, we establish a standardized collection protocol organized into two primary categories based on environment type:

- **Indoor scenes:** Covers navigation within enclosed environments such as buildings, rooms, and corridors.
  - *Free roaming:* Stochastic exploration along random trajectories, with frequent room transitions and floor changes;

  - *Loop roaming:* Departing from and returning to the same location to form closed-loop sequences through interconnected rooms;
  - *Transition navigation:* Traversing significant scene boundaries such as moving between distinct interiors or exiting to outdoor areas via doors and elevators.
- **Outdoor scenes:** Covers a broader range of navigation and observation modes in open environments.
  - *Free roaming:* Stochastic exploration along random trajectories, with diverse headings and frequent lateral viewpoint changes;
  - *Loop roaming:* Departing from and returning to the same location to form closed-loop sequences across open terrain;
  - *Transition navigation:* Traversing significant scene boundaries such as entering or exiting buildings and crossing between distinct outdoor regions;
  - *Dynamic sightseeing:* Touring scenes containing moving elements such as pedestrians, vehicles, or animals;
  - *Object orbiting:* Circling around objects to capture multi-view observations.

A.4 MatrixCity Data Sequencing

The MatrixCity dataset provides aerial and street-level multi-view images with known camera poses. However, the original data is organized for spatial coverage rather than temporal continuity—aerial images are sampled on a regular grid, while street images are stored as independent road segments. This is incompatible with downstream tasks that require temporally continuous input, such as video generation and sequential 3D reconstruction. We reorganize both data types into view-continuous sequences by performing random walks on their respective spatial topologies, without modifying the original images or poses.

A.4.1 Aerial Data

Data layout. Aerial data is arranged on an $H \times W$ regular grid in row-major order (Fig. S1 a). Each scene may contain multiple *grid loops*, each corresponding to a complete grid sampling pass. The frame index i and grid coordinate (r, c) satisfy

$$r = \lfloor i/W \rfloor, \quad c = i \bmod W. \tag{S1}$$

Consecutive frames are row-wise neighbors on the grid, but spatial discontinuities occur at row boundaries (*e.g.*, the last frame of one row and the first frame of the next row are far apart), so the raw storage order does not form a natural camera trajectory.

Sequencing method. We treat the grid as a graph with 8-connected adjacency and perform random walks to produce spatially continuous trajectories (Fig. S1 b). To suppress immediate backtracking, each step excludes the previous position from the candidate set. Grid coordinates are mapped to global frame indices via

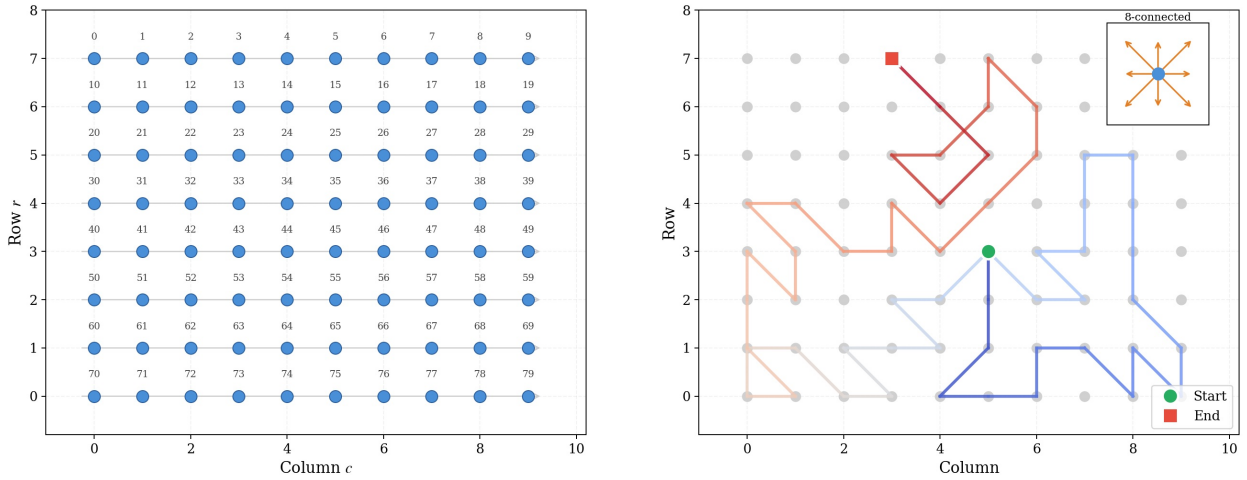
$$\text{idx} = l \cdot N_{\text{loop}} + r \cdot W + c, \tag{S2}$$

where l is the grid loop index and N_{loop} is the number of frames per loop. The number of sequences is proportional to the data size (by default, 5 sequences per 1,000 frames). The full procedure is given in Algorithm 1.

A.4.2 Street Data

Data layout. Street data is captured per-street. Each street is a linear trajectory with N evenly-spaced camera positions, each photographed from 5 viewpoints (viewpoints 0–3 are horizontal; viewpoint 4 is zenith-facing), yielding $5N$ frames per street. Frames are stored in viewpoint-first order: all N positions of viewpoint 0 appear first, followed by viewpoints 1–4 in sequence. Different streets may physically intersect (*e.g.*, at crossroads), but are stored independently without explicit connectivity information.

Sequencing proceeds in three stages (Fig. S2): street identification, intersection detection, and graph-based random walk.



(a). Row-major grid layout. **(b).** 8-connected random walk.

Figure S1. Aerial data sequencing. **(a)** Each blue node is a camera position labeled by its row-major index (index $= r \times W + c$); gray arrows show the storage scan order. **(b)** A random walk on the grid; color gradient (blue $\rightarrow$ red) encodes temporal progression. The inset shows the 8-connected neighborhood. Each step excludes the previous position to suppress backtracking.

Algorithm 1 Aerial Grid Random Walk

Input: Grid size $H \times W$, sequence length T, loop index l
Output: Frame index sequence $\mathcal{S}$

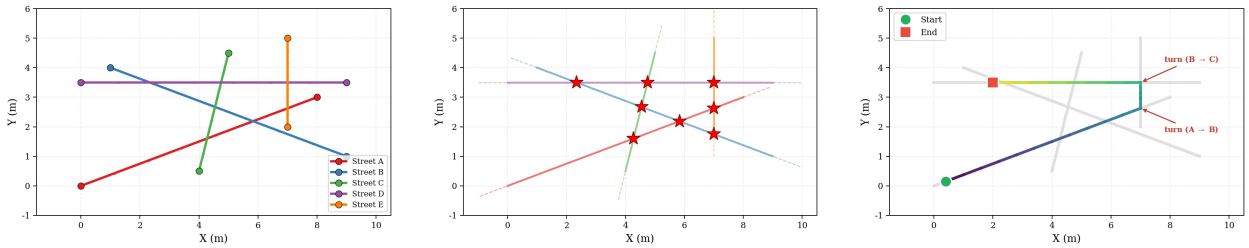
Sample random start position $(r, c) \in [0, H) \times [0, W)$
prev $\leftarrow$ null
for $t = 1$ **to** T **do**
  $\mathcal{S}$.append$(l \cdot N_{\text{loop}} + r \cdot W + c)$
  $\mathcal{N} \leftarrow \{(r+\Delta r,\ c+\Delta c) \mid (\Delta r, \Delta c) \in \mathcal{D}_8\} \cap \mathcal{G} \ \setminus \ \{\text{prev}\}$
  if $\mathcal{N} = \emptyset$ **then**
    $\mathcal{N} \leftarrow$ all valid 8-connected neighbors ▷ corner fallback
  prev $\leftarrow (r, c)$
  $(r, c) \leftarrow$ RANDOMCHOICE$(\mathcal{N})$
return $\mathcal{S}$

Stage 1: Street identification. We exploit the viewpoint-first storage order to automatically segment street boundaries via *position repetition detection* (Fig. S2 a). Frames are traversed sequentially, and camera positions are extracted. A set of unique positions $\mathcal{U}$ is maintained; when a new position falls within $\epsilon = 0.01$ m of any existing point in $\mathcal{U}$, it is flagged as a duplicate, indicating that viewpoint 0 traversal is complete. At this point $|\mathcal{U}| = N$ gives the number of unique positions for that street, and the subsequent $5N$ frames are partitioned into 5 viewpoint segments.

Stage 2: Intersection detection and graph construction. To establish topological connectivity, each street is simplified to its endpoint-to-endpoint 2D line segment (projected onto the XY plane), and all segment pairs are tested for intersection (Fig. S2 b). To handle streets that are physically close but do not precisely intersect (*e.g.*, T-junctions), each segment is optionally extended by a configurable distance d_{ext} at both ends before testing. Detected intersections define an undirected street connectivity graph $G = (V, E)$, where vertices V are streets and edges E connect intersecting pairs.

Stage 3: Graph-based random walk. We generate cross-street continuous sequences by performing random walks on G (Alg. 2, Fig. S2 c). Three key design choices ensure sequence quality:

1. **Intersection decisions.** When the walker approaches an intersection (distance $< d_{\text{prox}}$), it switches to a connecting street with probability $p_{\text{turn}} = 0.5$.



(a). Street identification. **(b).** Intersection detection. **(c).** Graph-based random walk.

Figure S2. Street data sequencing pipeline. **(a)** Stage 1: independently identified street segments (A–E) with camera positions (dots). **(b)** Stage 2: red stars mark detected intersections; dashed lines show optional segment extensions for near-miss junctions. **(c)** Stage 3: a random walk traversing Street A $\rightarrow$ B $\rightarrow$ C via intersections; color gradient encodes temporal order.

2. **Viewpoint continuity.** Upon switching streets, the viewpoint whose camera forward direction (third column of the rotation matrix) best matches the current one is selected:

$$v^* = \arg \max_{v \in \{0,\ldots,4\}} \mathbf{f}_{\text{curr}} \cdot \mathbf{f}_v. \tag{S3}$$

3. **Endpoint handling.** At dead ends (degree ≤ 1), the walking direction is reversed.

A.5 Cross-Scene Traversal Data Construction

Long-range streaming 3D reconstruction requires the model to handle continuously expanding observation sequences in which the camera moves from one room through a corridor into the next, encountering substantial changes in geometry and appearance along the way. Existing indoor RGBD datasets (*e.g.*, ScanNet [9]) are largely confined to single rooms or small regions and therefore lack the cross-region, long-range traversal signals needed for training. To address this limitation, we render cross-room continuous RGBD video sequences from large-scale real-world 3D scene reconstructions using Habitat-Sim [54].

Scene sources. We draw scenes from three complementary indoor 3D datasets: Gibson [83] ($\sim$450 building-scale scans covering residential and office environments with high geometric fidelity), Matterport3D [5] (90 large residential and commercial spaces, each typically containing 10 to 30 rooms with rich furniture layouts and multi-story structures), and HM3D [50] (900 semantically diverse indoor environments spanning a wide range of scene scales and complexities). The three datasets differ in capture devices, reconstruction quality, and scene types, and their joint use broadens the training distribution while mitigating overfitting to any single data source.

Trajectory generation. For each scene, we first compute a navigation mesh and then randomly sample $N+1$ waypoints on walkable surfaces, requiring a minimum geodesic distance of 2 m between consecutive waypoints to ensure sufficient spatial coverage. The number of segments N is drawn from a piecewise mixture distribution (40% in 5 to 8, 45% in 8 to 12, and 15% in 12 to 16), corresponding to local exploration, cross-functional-area walking, and full-scene traversal, respectively. Adjacent waypoints are connected via the shortest geodesic path on the navigation mesh, allowing trajectories to naturally pass through doorways and corridors for cross-room traversal. The concatenated path is then linearly interpolated, smoothed with a double-pass moving average, and projected back onto the navigation mesh, yielding a dense, smooth, and physically plausible trajectory. A single trajectory typically comprises 500 to 3,000 frames (roughly 15 to 100 s at 30 fps).

Motion control. To produce realistic egocentric videos, we design a multi-stage motion controller that converts the trajectory path into per-frame camera poses. Translation is governed by a first-order low-pass filter (α=0.18) combined with navigation-mesh collision correction. Yaw is steered toward a lookahead target 10 frames ahead through a $\tanh$-saturated velocity model that provides natural acceleration and deceleration during turns. Pitch follows terrain elevation

Algorithm 2 Street Network Random Walk

Input: Street graph G, sequence length T, turn probability p_{turn}, proximity threshold d_{prox}
Output: Frame sequence $\mathcal{S}$

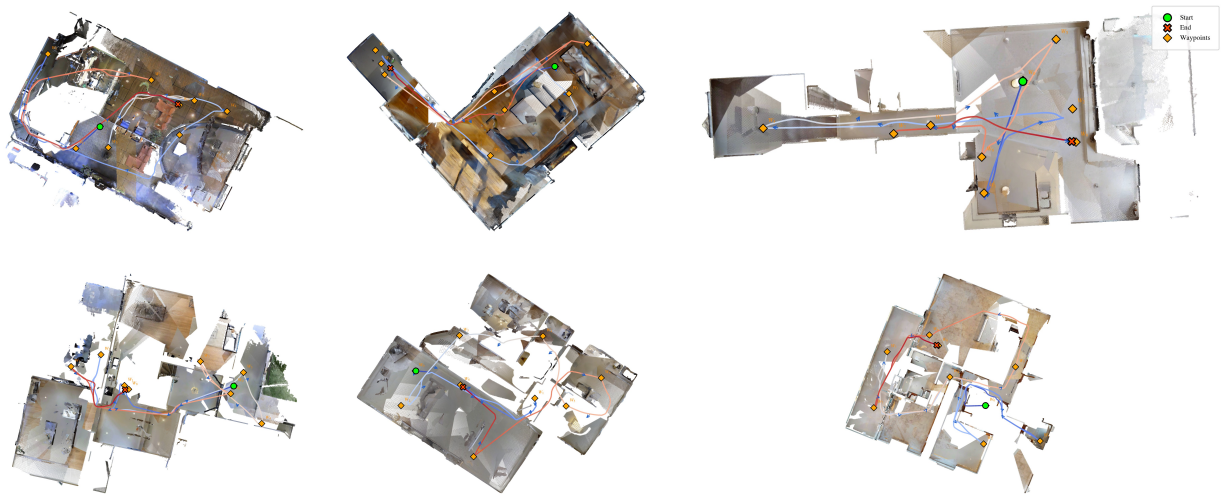
Randomly select start street e, position index i, viewpoint $v \in \{0..3\}$, direction $d \in \{+1, -1\}$
for $t = 1$ **to** T **do**
  $\mathcal{S}$.append$\big($FRAME$(e, i, v)\big)$
  $i_{\text{next}} \leftarrow i + d$
  if i_{next} out of bounds **then** ▷ street endpoint
    if $\deg(e) \leq 1$ **then**
      $d \leftarrow -d$ ▷ dead end: reverse
    else
      $(e, i, v, d) \leftarrow$ SWITCHSTREET(e, i, v)
  else if $\exists$ intersection within d_{prox} **and** RAND$() < p_{\text{turn}}$ **then**
    $(e, i, v, d) \leftarrow$ SWITCHSTREET(e, i, v)
  else
    $i \leftarrow i_{\text{next}}$ ▷ advance along current street
return $\mathcal{S}$

function SWITCHSTREET(e, i, v)
  $e' \leftarrow$ random neighbor of e in G
  $i' \leftarrow$ nearest position on e' to intersection point
  $v' \leftarrow \arg\max_{v''} \mathbf{f}_{\text{curr}} \cdot \mathbf{f}_{v''}$ ▷ viewpoint continuity
  $d' \leftarrow$ direction away from intersection
  return (e', i', v', d')

changes and is smoothed by exponential decay. On top of this base motion, IIR-filtered Gaussian noise is added to both position and orientation to simulate handheld camera shake, and stochastic glance events, *i.e*. brief lateral and vertical head turns following a half-cosine envelope, are triggered at random to reproduce the natural gaze shifts that occur during walking.

Parameter diversification. Camera intrinsics and motion parameters are independently sampled per sequence from piecewise mixture uniform distributions to cover a broad range of real-world conditions: horizontal field of view from 40° to 100°, sensor height from 1.35 to 1.80 m, walking speed from 0.3 to 1.8 m/s, and several tiers of jitter intensity and glance frequency. Walking speed is sampled independently per trajectory segment, producing natural speed variations within a single sequence. All randomized parameters and waypoint coordinates are recorded in a configuration file for exact reproducibility.

Rendering and output. We employ a pinhole camera model and render at 640×480 resolution and 30 fps, producing per-frame RGB images and metric depth maps together with 6-DoF camera poses (position and quaternion) and intrinsics (f_x, f_y, c_x, c_y). In total, we generate approximately 2,800 sequences from the three datasets, each containing 1k to 5k frames, amounting to 14.4 TB of data. Figure S3 visualizes the top-down point clouds and sampled trajectories for several representative scenes.



Matterport3D Gibson HM3D

Figure S3. Top-down visualizations of rendered point clouds and sampled camera trajectories across six scenes from three datasets (Matterport3D, Gibson, and HM3D). Each point cloud is colored using per-pixel RGB values from the rendered frames. The trajectory color varies from blue (start) to red (end), indicating temporal progression. Green circles and red crosses mark the start and end positions, respectively. Orange diamonds denote randomly sampled waypoints on the navigation mesh, and blue arrows indicate the camera viewing direction at regular intervals.