Stable3R: Streaming 3D Reconstruction with Stable Geometric References

Jincheng Xiong[1, 2 *], **Jiarong Han**[2 *], **Hang Zhang**[2], **Mu Xu**[2], **Ming Qian**[2 †]
[1]Zhejiang University [2]AMAP, Alibaba

https://venite-xjc.github.io/Stable3R-page/

Abstract

Streaming 3D reconstruction has become a critical component for real-time applications. While recent advances have adapted large-scale offline feedforward models into streaming architectures via causal attention and KV-caching, we found they suffer from fundamental instability: the irreversible bias in early frames. This limitation stems from two coupled factors: (1) the existing KV-cache-based causal architecture strictly prohibits access to future information, resulting in information-poor early caches; and (2) current methods typically anchor the pose constraints to the initial frame, amplifying the early bias throughout the sequence. To address these challenges, we propose Stable3R, a holistic framework that establishes stable geometric references by refining historical caches and anchoring poses to prefixes rather than a single initial frame. Specifically, we first introduce a lookahead-augmented attention mechanism, integrating future observations into past caches. Second, we build on a frame-equivariant architecture and adapt relative-pose supervision to causal prefixes, alleviating the reliance on the biased initial frame. Extensive experiments across multiple benchmarks show that our method consistently boosts reconstruction quality within a strictly causal inference schedule.

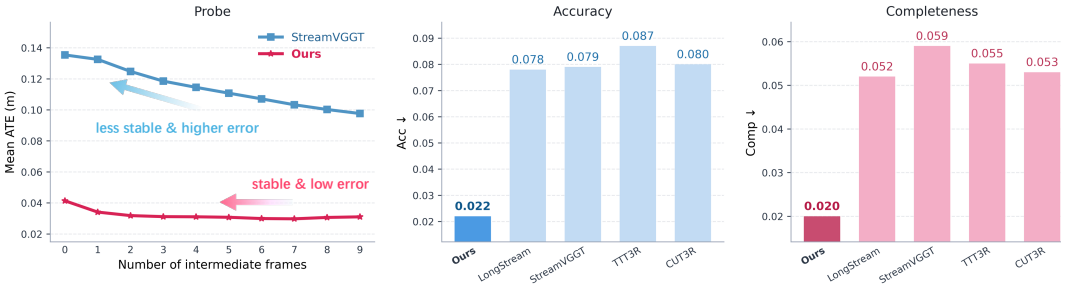
1 Introduction

Streaming 3D reconstruction is increasingly critical for real-time applications such as embodied robotics [63] and interactive scene understanding [67]. Existing approaches broadly fall into two categories. The first employs recurrent mechanisms [60, 15], compressing scenes into online updated states for continuous 3D perception, yet these methods often suffer from catastrophic forgetting. In contrast, causal streaming methods [81, 27] extend offline 3D foundation models (e.g., VGGT [59]) to streaming settings via causal attention and KV-caches, enabling more continuous and stable retention of historical information and thereby improving reconstruction quality. However, we observe a key stability bottleneck in this paradigm: an over-reliance on early frames that contain limited observations.

The current causal streaming models [81, 27] are inherently limited by two coupled factors: (i) the causal attention mechanism, which compels early frames to function on minimal geometric observations that cannot be retrospectively refined; and (ii) a "first-frame-anchored" architecture that ties global coordinate references and all pose regressions to the initial yet information-poor estimate. While first-frame-anchored designs are commonplace in offline models [62, 72, 59] with global information access, this architecture imposes a critical constraint in the streaming regime: the entire reconstruction becomes highly sensitive to the quality of the first frame. Since standard causal attention prevents later observations from refining early representations, all subsequent observations

* Equal contribution.
† Corresponding author.

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



(a) ATE of Probe Frames on NRGBD (b) 3D Reconstruction Quality on NRGBD

Figure 1: Results on NRGBD [1] dataset. (a) We introduce geometric context probes to evaluate the stability of geometric references in the KV caches. The protocol uniformly samples 30 images on each sequence, fixes the initial frame and final 20 probe frames while varying the number of intermediate frames, and measures the mean Absolute Trajectory Error (ATE) of the probe frames across sequences. We observe that the ATE of StreamVGGT [81] varies significantly with the number of intermediate frames, indicating unstable geometric references in its historical observations. Moreover, earlier frames are inherently less reliable due to limited observations. In contrast, our method achieves lower ATE by stabilizing geometric references in the historical observations. (b) Our model achieves a more accurate 3D reconstruction than prior streaming methods.

are forced to align with the frame containing the least amount of information. This mechanism inevitably leads to persistent instability. Our empirical analysis, utilizing geometric context probes as shown in Fig. 1 (a), confirms that reconstruction quality is positively correlated with the information density of early observations, demonstrating that stability is contingent upon the accumulation of rich historical context.

To overcome the aforementioned constraints, we propose Stable3R, a unified framework designed for stable streaming 3D reconstruction. We identify two key factors contributing to early-frame bias and mitigate their effects through complementary designs: First, we introduce **lookahead attention** to alleviate the information deficiency of causal caches, dynamically integrating future observations back into historical caches via a tailored parallel computation paradigm. Second, we adopt a **frame-equivariant architecture** where all frames share a unified token representation to eliminate structural asymmetry and replace absolute pose supervision with **prefix-relative pose supervision**, anchoring current estimates against preceding frames rather than a single initial noisy frame. Consequently, our model exhibits enhanced stability against historical observations (see Fig. 1 (a)). Comprehensive evaluations on various benchmarks further demonstrate that our method achieves superior performance in both accurate camera pose estimation and robust 3D reconstruction across Sparse, Mid-length, and Long-sequence settings.

2 Related Works

Classical 3D Reconstruction. 3D reconstruction, a fundamental problem in computer vision, aims to recover the geometric structure of objects or scenes from 2D observations. Traditional approaches, such as Structure-from-Motion (SfM) [25, 50] and Simultaneous Localization and Mapping (SLAM) [43, 42, 11], rely on hand-crafted features and complex pipelines. They typically involve feature extraction, matching, triangulation, and bundle adjustment to estimate camera poses and sparse keypoints, followed by Multi-View Stereo (MVS) methods [2, 6, 23, 51, 55] to recover dense geometry. These pipelines are often time-consuming and suffer from limited robustness, particularly in texture-poor or challenging scenarios, lacking reliable visual features for tracking and reconstruction.

Offline 3D Reconstruction Data-driven transformer-based models have shifted 3D reconstruction toward a feed-forward paradigm. DUSt3R [62] and its extension MASt3R [28] directly regress 3D point maps from a pair of uncalibrated RGB images, and extend to multi-view settings via global alignment. Subsequent work [72, 77] further generalizes this paradigm to multi-view inputs. Recently, VGGT [59] pioneers a simplified alternating attention architecture, demonstrating remarkable reconstruction quality via large-scale pretraining and multi-task learning. This has inspired a

wave of feed-forward methods, ranging from adaptations for dynamic scenes [14, 58] to approaches enhancing accuracy and efficiency [65, 34, 26, 70, 36]. However, these methods are inherently offline and computationally intensive, which precludes their use in real-time interactive applications.

Streaming 3D Reconstruction Unlike single-pass feed-forward models, streaming reconstruction enables continuous inference under partial observations, which is essential for real-time applications. Existing approaches are primarily built on two paradigms: recurrent/state-based models and causal streaming models. The former [60, 15, 32] compress past observations into online states for efficient inference, but often suffer from forgetting or loss of fine-grained geometry. Other related directions include chunk-wise streaming designs such as LoGeR [76], which are less aligned with strictly real-time causal inference, and hybrid methods [20, 40, 39], which combine 3D foundation models with traditional SLAM pipelines but typically rely on hand-crafted components and system-level tuning. In contrast, causal streaming methods [81, 27] extend offline 3D foundation models such as VGGT [59] to streaming settings via causal attention and KV-caches, achieving stronger reconstruction quality through direct retention of historical information. Subsequent train-free extensions [75, 38, 71] and concurrent long-horizon works, including LongStream [16] and LingBot-Map [13], further improve scalability to longer sequences. However, existing causal streaming frameworks remain constrained by strict causality and first-frame dependence, which make reconstruction highly sensitive to early, information-poor observations. Our work instead focuses on addressing this stability bottleneck throughout the full streaming process.

Causal Attention in Streaming Models Causal attention is widely used in streaming models by masking future information, but it introduces an asymmetric information flow where early tokens cannot access future evidence, leading to limited global context. This issue has been observed in large language models [78, 21, 69], vision-language models [45], and world models [80, 24]. Prior work mitigates this by augmenting causal attention with additional context, including hybrid causal-bidirectional masks [7, 47, 21], global signals such as contextual tokens [33] or decoder modifications [30], and retrieval or selection mechanisms [10, 29]. The most related work, CASTLE [53], introduces lookahead keys to inject future information into historical representations, partially alleviating the asymmetry caused by causal attention. This aligns with our observation that sparse early observations lead to biased scene estimation. However, CASTLE assumes token-level causal attention and its parallelization scheme cannot be directly applied to our frame-level setting. We therefore implement dedicated backend kernels tailored to our lookahead attention.

3 Method

We propose a streaming 3D reconstruction framework that stabilizes early scene understanding through lookahead attention, while effectively mitigating early-frame bias via frame-equivariant modeling and prefix-relative pose supervision. In the following sections, we detail the streaming architecture, the lookahead attention mechanism, and the learning objectives.

3.1 Streaming 3D Reconstruction

Given an input image sequence $\{I_i\}_{i=1}^{N}$, the model ϕ processes each frame I_i sequentially and predicts the corresponding camera pose $\hat{T}_i \in \mathbb{R}^{4\times4}$ and point map $\hat{P}_i \in \mathbb{R}^{H\times W\times 3}$ using only the observed prefix $\{I_j\}_{j=1}^{i}$. Following prior streaming 3D reconstruction architectures [81, 27], we decompose ϕ into an encoder, a decoder, and a prediction head, as illustrated in Fig. 2:

$$F_i = \mathrm{Encoder}(I_i), \qquad (G_i, \mathcal{M}_i) = \mathrm{Decoder}(F_i, \mathcal{M}_{i-1}), \qquad \{\hat{T}_i, \hat{P}_i\} = \mathrm{Head}(G_i), \quad (1)$$

where $\mathcal{M}_{i-1}$ denotes the cached historical memory constructed from the observed prefix $\{I_j\}_{j=1}^{i-1}$.

The encoder extracts per-frame visual tokens using a DINOv2-based tokenizer [44]. The decoder transforms these tokens into geometry-aware representations through alternating frame-wise attention and causal temporal attention. To reduce structural asymmetry across frames, we adopt a frame-equivariant architecture following π^3 [65], where every frame is represented with the same token layout of patch tokens and register tokens. The decoder module propagates scene information across time through temporal attention with cached historical states. Finally, an MLP-based prediction head regresses the camera pose and point map for the current frame.

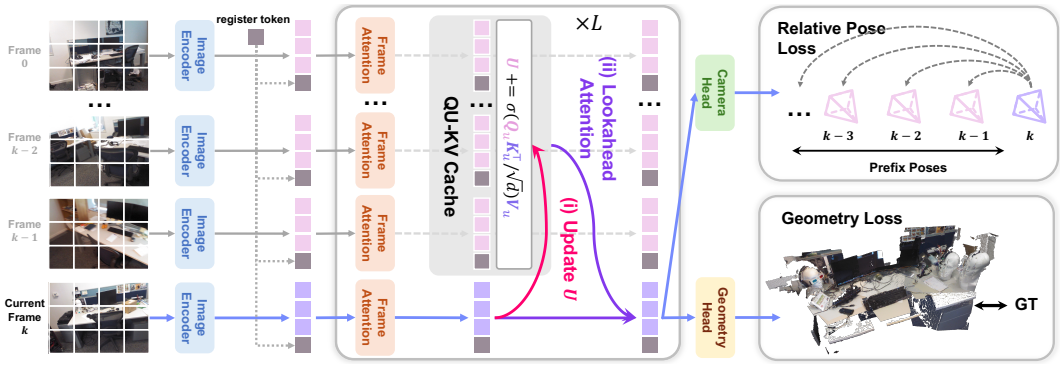


Figure 2: Overview of the proposed Stable3R framework for streaming 3D reconstruction. Pink denotes historical frames, while purple denotes the current frame. We cache the causal states $(\boldsymbol{K}_c, \boldsymbol{V}_c)$ together with the lookahead states $(\boldsymbol{Q}_u, \boldsymbol{U})$. When a new frame arrives, the accumulated lookahead representation $\boldsymbol{U}$ is first updated using the incoming observations, followed by lookahead attention computation. In practice, both stages are fused into customized Triton kernels during training.

3.2 Lookahead Attention

A critical challenge in streaming reconstruction is that early representations are derived from sparse observations, rendering them inherently noisy and incomplete. Under standard causal attention, historical memory is computed once and subsequently frozen, making early deviations difficult to rectify and adversely affecting later predictions. We overcome this limitation with lookahead attention, which iteratively refines the key-side representation of historical frames as new observations arrive, while remaining fully compatible with streaming inference: at time i, the model only accesses frames in $\{I_1, \ldots, I_i\}$. Importantly, this refinement does not trigger retrospective prediction updates; instead, it is used only to improve subsequent predictions.

Standard frame-level causal attention. We first recall standard frame-level causal attention in its parallel form. Given temporal frame tokens $\boldsymbol{X} \in \mathbb{R}^{NT \times d}$ derived from frames $\{I_1, \ldots, I_N\}$, where N is the number of frames, T is the number of tokens per frame, and d is the hidden dimension, we compute $\boldsymbol{Q}_c = \boldsymbol{X}\boldsymbol{W}_c^Q, \quad \boldsymbol{K}_c = \boldsymbol{X}\boldsymbol{W}_c^K, \quad \boldsymbol{V}_c = \boldsymbol{X}\boldsymbol{W}_c^V \in \mathbb{R}^{NT \times d_{\text{head}}}$. The causal attention output is then given by

$$\text{Attention}(\boldsymbol{X}) = \text{softmax}\left(\frac{\boldsymbol{Q}_c \boldsymbol{K}_c^\top}{\sqrt{d_{\text{head}}}} + \boldsymbol{M}_c\right)\boldsymbol{V}_c, \tag{2}$$

where the frame-wise causal mask $\boldsymbol{M}_c$ is defined as

$$\boldsymbol{M}_c(n, m) = \begin{cases} 0, & \text{if frameIdx}(m) \le \text{frameIdx}(n), \\ -\infty, & \text{otherwise.} \end{cases} \tag{3}$$

Frame-level lookahead attention. To incorporate future observations into historical memory, we introduce $\boldsymbol{Q}_u = \boldsymbol{X}\boldsymbol{W}_u^Q, \quad \boldsymbol{K}_u = \boldsymbol{X}\boldsymbol{W}_u^K, \quad \boldsymbol{V}_u = \boldsymbol{X}\boldsymbol{W}_u^V, \in \mathbb{R}^{NT \times d_{\text{head}}}$ similar to CASTLE [53] to compute time-varying lookahead keys $\boldsymbol{U} \in \mathbb{R}^{NT \times d_{\text{head}}}$. Specifically, when processing frame i, for historical token m in frame $j (j < i)$, we aggregate future information from frame $j+1$ to frame i into the lookahead key $\boldsymbol{U}^m \in \mathbb{R}^{d_{\text{head}}}$ as:

$$\boldsymbol{U}_{j \to i}^m = \sum_n \sigma\left(\frac{(\boldsymbol{Q}_u^m)^\top \boldsymbol{K}_u^n}{\sqrt{d_{\text{head}}}}\right)\boldsymbol{V}_u^n, \tag{4}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $j < \text{frameIdx}(n) \le i$. We then incorporate the lookahead keys into causal attention by modifying the attention logits:

$$\boldsymbol{O} = \text{softmax}\left(\frac{\boldsymbol{Q}_c \boldsymbol{K}_c^\top}{\sqrt{d_{\text{head}}}} - \text{SiLU}\left(\frac{\boldsymbol{Q}_c \boldsymbol{U}^\top}{\sqrt{d_{\text{head}}}}\right) + \boldsymbol{M}_c\right)\boldsymbol{V}_c, \tag{5}$$

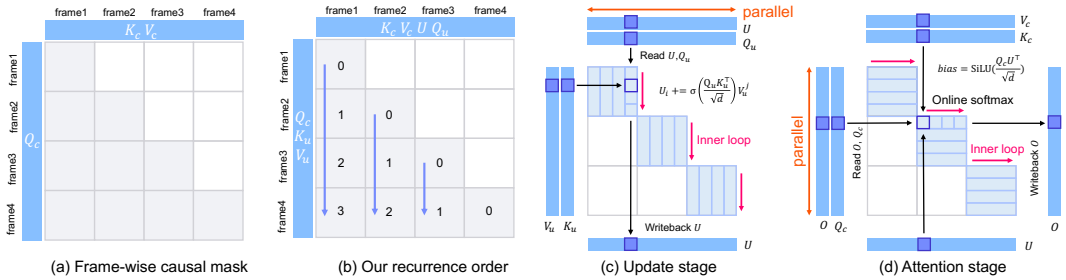


Figure 3: Efficient parallel lookahead attention under frame-causal (bidirectional + causal) attention. (a) Frame-wise causal masking induces a block-wise lower-triangular structure, enabling block-wise computation. (b) Blocks are processed in order of increasing diagonal offset (numbers indicate execution order). Blue arrows denote dependencies on the lookahead representation $\boldsymbol{U}$. Each block is computed in two stages. (c) Update stage. $\boldsymbol{Q}_u$ is tiled and processed in parallel, where each kernel instance scans $\boldsymbol{K}_u, \boldsymbol{V}_u$ to accumulate $\boldsymbol{U}$. (d) Attention stage. $\boldsymbol{Q}_c$ is tiled and processed in parallel; each kernel performs online softmax over $\boldsymbol{K}_c, \boldsymbol{V}_c$ with lookahead bias from $\boldsymbol{U}$ to compute $\boldsymbol{O}$.

where $\boldsymbol{O}$ denotes the attention output. The lookahead term introduces an adaptive bias from enriched historical memory to improve geometric stability. Following CASTLE, we adopt SiLU [22] for smoother gating and better training stability. To support streaming inference, we cache both the lookahead states $(\boldsymbol{Q}_u, \boldsymbol{U})$ and the causal states $(\boldsymbol{K}_c, \boldsymbol{V}_c)$, referred to as the **QU-KV cache**. As illustrated in Fig. 2, we first update the lookahead keys U of the history caches by new frame, then perform attention with updated U.

Implementation for efficient parallel training. Parallel computation is critical for training efficiency in streaming models. CASTLE [53] proposes an efficient parallel training algorithm for lookahead attention based on strict token-level causality. However, it is incompatible with our frame-causal attention, which is combined with bidirectional intra-frame and causal inter-frame interactions. To address this, we implement a dedicated parallelization scheme tailored to hybrid lookahead attention in Triton [57]. Importantly, we observe that the inter-frame dependency remains causal, resulting in a block-wise lower-triangular structure (Fig. 3(a)) that admits block-wise decomposition [19]. For a historical frame j, the lookahead keys progressively accumulate information from future frames. When interacting with a future frame i, the lookahead keys $\boldsymbol{U}_{j\to i}$ aggregate contributions from frames $j+1$ to i, yielding the following recursive update:

$$\boldsymbol{U}_{j\to i} = \boldsymbol{U}_{j\to i-1} + \sigma\left(\frac{(\boldsymbol{Q}_u^j)^\top \boldsymbol{K}_u^i}{\sqrt{d_{\text{head}}}}\right)\boldsymbol{V}_u^i. \tag{6}$$

This recursive structure introduces dependencies across blocks along the frame dimension, which naturally leads to a diagonal scheduling order, as visualized in Fig. 3(b).

Concretely, we process the block matrix in increasing diagonal offsets, i.e., along diagonals with $i - j = 0, 1, 2, \ldots, N-1$, corresponding to the numbers in Fig. 3(b). For each block (j, i) in Fig. 3(b), we perform two stages: (i) an update stage that performs eq. (6) to accumulate information from frame i into the lookahead keys of frame j , as shown in Fig. 3(c), and (ii) an attention stage that performs eq. (5) to compute lookahead attention from frame i to frame j using the updated lookahead keys, as shown in Fig. 3(d). The update stage strictly follows a vertical dependency order, as indicated by the blue arrows in Fig. 3(b). Following FlashAttention-2 [18], we adopt query-aligned tiling in Triton [57] to reduce High Bandwidth Memory (HBM)–Static Random-Access Memory (SRAM) traffic. As the update and attention stages involve different queries $\boldsymbol{Q}_u$ and $\boldsymbol{Q}_c$, with distinct interaction directions, we decompose them into two kernels executed alternately, each parallelized along its query, leading to alternating tiling patterns.

For diagonal blocks (j, j), no future information is involved, and thus the update stage is skipped. As a result, for N frames, the forward pass requires $2N - 1$ kernel launches. The backward pass follows the reverse (anti-diagonal) order to respect the dependency structure.

3.3 Learning Objectives

To mitigate early-frame bias in pose supervision, we adopt a prefix-relative pose objective. Instead of regressing each frame against a fixed global reference, we supervise relative camera poses over all causal pairs (i, j) with $i < j$. This formulation has two advantages. First, it shifts supervision from absolute alignment to relative geometric consistency, making learning less sensitive to deviations in early frames. Second, it provides a denser supervisory signal through the entire observed prefix, resulting in tighter constraints on the model compared to using a sparse initial reference.

Following MoGe [61], we first align the predicted and ground-truth pointcloud with a sequence-level scale factor s^*. Relative transformations are derived from predicted and ground-truth camera poses as

$$\hat{T}_{ij} = \hat{T}_i^{-1}\hat{T}_j, \qquad T_{ij} = T_i^{-1}T_j, \tag{7}$$

with corresponding rotation and translation components $(\hat{R}_{ij}, \hat{t}_{ij})$ and (R_{ij}, t_{ij}). The pose loss is then defined as

$$\mathcal{L}_{\text{pose}} = \frac{1}{N_{\text{pair}}} \sum_{i<j} \left[\lambda_r \mathcal{L}_{\text{rot}}(\hat{R}_{ij}, R_{ij}) + \lambda_t \left\| s^*\hat{t}_{ij} - t_{ij} \right\|_{\text{Huber}} \right], \tag{8}$$

where N_{pair} is the number of causal frame pairs, $\mathcal{L}_{\text{rot}}$ denotes the rotation loss, and λ_r, λ_t are balancing weights.

In addition, we adopt a scale-invariant local point-map loss following π^3 [65] to encourage accurate local geometry reconstruction:

$$\mathcal{L}_{\text{points}} = \frac{1}{3NHW} \sum_{i=1}^{N} \sum_{u=1}^{H\times W} \frac{1}{z_{i,u}} \left\| s^*\hat{x}_{i,u} - x_{i,u} \right\|_1, \tag{9}$$

where $\hat{x}_{i,u}, x_{i,u} \in \mathbb{R}^3$ denote the predicted and ground-truth 3D coordinates at pixel u, and $z_{i,u}$ is the corresponding ground-truth depth used to normalize the residual.

The total training objective is a weighted sum of the pose and point-map losses:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{points}}\mathcal{L}_{\text{points}}, \tag{10}$$

where λ_{pose} and λ_{points} are balancing weights.

4 Experiments

4.1 Implementation Details

We train our model on a multi-domain dataset collection, including CO3Dv2 [48], WildRGBD [68], ARKitScenes [3], ARKitScenes-HighRes [4], ScanNet++ [74], ScanNet [17], HyperSim [49], BlendedMVS [73], MegaDepth [31], Waymo [56], VirtualKITTI2 [9], TartanAir [64], DL3DV [35], OmniObject3D [66], Spring [41], PointOdyssey [79], UASOL [5] and Matterport3D [12] datasets.

Initialized from pre-trained π^3 weights [65], our framework keeps the backbone encoder frozen. The proposed lookahead module is trained from scratch with a peak learning rate of 1×10^{-4} , whereas the rest of the model uses a peak learning rate of 1×10^{-5} . We employ the AdamW optimizer [37] for 24k steps, training on image sequences of 4 to 24 frames. The entire training process takes approximately 2 days on 32 NVIDIA H20 GPUs.

4.2 Experimental Setup

We evaluate Stable3R on two core tasks: 3D reconstruction and camera pose estimation. For 3D reconstruction, we consider three evaluation regimes: a **Sparse** setting with minimal or no frame overlap (2–5 frames), a **Mid-length** setting with uniformly sampled 25-frame sequences, and a **Long-sequence** setting with 300-frame sequences. We first report reconstruction results under the Sparse and Mid-length settings, then evaluate long-sequence generalization, and finally assess trajectory accuracy for camera pose estimation.

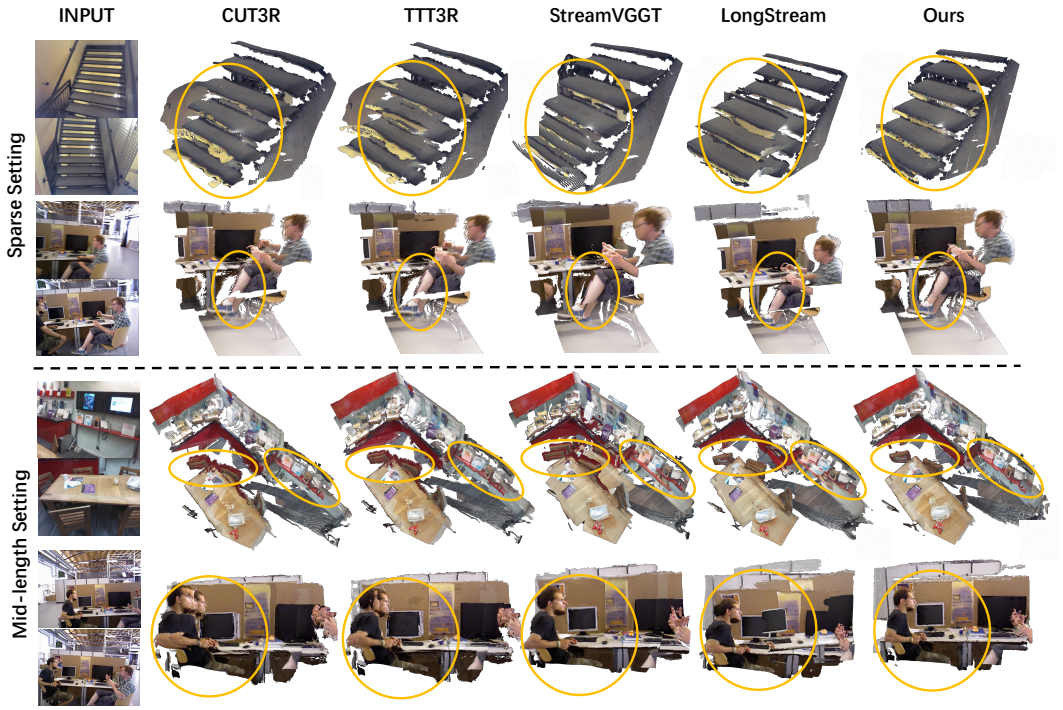


Figure 4: Qualitative Results of 3D Reconstruction.

4.3 3D Reconstruction Results

4.3.1 Sparse and Mid-length Evaluation

Following StreamVGGT and CUT3R [81, 60], we evaluate reconstruction quality on both static scenes (NRGBD [1] and 7-Scenes [52]) and the more challenging dynamic benchmark TUM-dynamics [54]. We report accuracy (Acc), completeness (Comp), and normal consistency (NC). Under the **Sparse** setting, we follow the StreamVGGT protocol to sample 2–4 frames on NRGBD and 3–5 frames on 7-Scenes, while extending the evaluation to TUM-dynamics with uniformly sampled 5 frames. Under the **Mid-length** setting, we uniformly sample 25 frames for each sequence.

We compare against strictly streaming baselines, including CUT3R [60], TTT3R [15], StreamVGGT [81], and LongStream [16]. Following the CUT3R evaluation framework [60], we benchmark all methods using their publicly available weights under a unified protocol.

As shown in Tab. 1 and Fig. 4, Stable3R consistently improves reconstruction quality under both settings. In particular, the gains in accuracy and completeness are consistent with the visual results, where our method produces more coherent structures and significantly fewer artifacts. These results suggest that lookahead attention and frame-equivariant modeling with relative supervision stabilize the streaming process by mitigating the information deficiency of causal memory. Notably, the method remains robust on dynamic scenes (e.g., the bottom row of both settings in Fig. 4), highlighting its potential for streaming 4D reconstruction in dynamic environments.

To further evaluate the generalization ability of Stable3R, we conduct experiments on the recently introduced SpatialBench [46], a cross-paradigm and domain-diverse benchmark, comparing against other streaming 3D reconstruction methods. We evaluate the sparse and medium settings, additionally reporting AbsRel for depth, AUC@30 for camera pose, and F-score for point clouds. As shown in Tab. 2, Stable3R achieves the best depth and camera pose results among streaming baselines under the sparse setting. Under the medium setting, Stable3R achieves the best results in depth, camera pose, and point cloud reconstruction, while ranking second in trajectory ATE, behind only LongStream. These results demonstrate the strong generalization ability of Stable3R across diverse domains.

Table 1: Quantitative 3D reconstruction results on NRGBD, 7-Scenes, and TUM-dynamics under Sparse and Mid-length settings. Bold and underlined values indicate the best and second-best results, respectively.

	Method	NRGBD						7-Scenes						TUM-dynamics					
		Acc↓		Comp↓		NC↑		Acc↓		Comp↓		NC↑		Acc↓		Comp↓		NC↑	
		Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
Sparse	CUT3R [60]	0.096	0.048	0.084	0.043	0.822	0.963	0.084	0.056	0.097	0.056	0.698	0.787	0.148	0.099	0.095	0.049	0.643	0.718
	TTT3R [15]	0.100	0.055	0.085	0.049	0.820	0.960	0.088	0.059	0.125	0.077	0.694	0.783	0.148	0.096	0.096	0.050	0.646	0.721
	StreamVGGT [81]	0.085	0.044	0.079	0.038	0.862	0.986	0.132	0.058	0.116	0.042	**0.749**	**0.863**	0.126	0.060	0.070	0.036	**0.714**	**0.820**
	LongStream [16]	0.091	0.063	0.106	0.078	0.814	0.956	0.068	0.045	0.069	**0.041**	0.738	0.843	0.080	**0.032**	0.060	0.031	0.709	0.816
	Ours	**0.035**	**0.020**	**0.041**	**0.023**	**0.872**	**0.987**	**0.052**	**0.032**	**0.069**	0.048	0.743	0.847	**0.079**	0.045	**0.051**	**0.028**	0.707	0.805
Mid-len.	CUT3R [60]	0.080	0.046	0.053	0.025	0.820	0.958	0.025	0.013	0.032	0.012	0.702	0.807	0.106	0.049	0.057	0.020	0.577	0.615
	TTT3R [15]	0.087	0.049	0.055	0.028	0.808	0.948	0.040	0.021	0.038	0.013	0.687	0.789	0.102	0.049	0.051	0.019	0.580	0.620
	StreamVGGT [81]	0.079	0.055	0.059	0.032	0.846	0.983	0.045	0.026	0.035	0.016	0.716	0.826	0.091	0.046	0.042	0.014	**0.633**	**0.711**
	LongStream [16]	0.078	0.050	0.052	0.024	0.836	0.975	0.030	0.019	0.027	0.013	0.720	0.832	0.072	0.024	0.045	0.012	0.615	0.682
	Ours	**0.022**	**0.014**	**0.020**	**0.010**	**0.884**	**0.987**	**0.019**	**0.010**	**0.022**	**0.010**	**0.722**	**0.834**	**0.048**	**0.015**	**0.031**	**0.011**	0.626	0.697

Table 2: Comparison on the Sparse and Medium tasks of SpatialBench [46]. Best and second-best results are in **bold** and underlined, respectively.

Method	Sparse		Medium			
	AbsRel ↓	AUC@30 ↑	AbsRel ↓	AUC@30 ↑	ATE ↓	F-Score ↑
CUT3R	0.196	0.519	0.189	0.469	2.676	0.286
TTT3R	0.202	0.469	0.179	0.493	2.343	0.294
StreamVGGT	0.154	0.598	0.171	0.562	4.940	0.397
InfiniteVGGT	0.154	0.596	0.170	0.563	4.964	0.402
LongStream	0.151	0.543	0.166	0.385	**1.188**	0.126
Ours	**0.105**	**0.628**	**0.097**	**0.642**	2.027	**0.541**

4.3.2 Long-Sequence Evaluation

Although trained only on short sequences (up to 24 frames), Stable3R generalizes effectively to long sequences. We evaluate long-sequence extrapolation under two inference settings: (i) direct inference with the original models and (ii) inference using FrameVGGT's train-free, frame-level bounded-memory strategy [71]. For the latter setting, since the original implementation of FrameVGGT is unavailable, we implement a closely related version based on the paper and apply it to both our method (denoted as Ours*) and StreamVGGT [81] (denoted as FrameVGGT*). Concretely, we use a 16-frame mid-memory budget together with 4-frame sparse anchors.

We compare against state-memory baselines, including CUT3R [60] and TTT3R [15], as well as KV-cache-based methods, including InfiniteVGGT [75], OVGGT [38], and LongStream [16]. Experiments are conducted on 7-Scenes [52] and NRGBD [1] using 300-frame samples.

As shown in Tab. 3, both inference variants of our method achieve strong geometric reconstruction performance across datasets while maintaining competitive normal consistency. In particular, our standard variant achieves the best reconstruction accuracy on both datasets, while the bounded-memory variant attains the best mean completeness on 7-Scenes. These results demonstrate strong generalization from short-sequence training to long-horizon inference, as well as compatibility with test-time optimization. Notably, LingBot-Map is specifically trained on sequences exceeding 300 frames with substantially greater training cost than ours.

4.4 Camera Pose Estimation Results

We evaluate camera pose estimation on TUM-dynamics [54] (90-frame sequences), Sintel [8] (2050 frame sequences) and ScanNet [17] (90-frame sequences), following the StreamVGGT protocol. Aligned to ground truth via Sim(3), predictions are assessed using Absolute Translation Error (ATE), Relative Translation Error (RPE trans), and Relative Rotation Error (RPE rot). Notably, Sintel and TUM-dynamics contain dynamic objects, presenting significant challenges for geometric consistency in pose estimation. As shown in Tab. 4 and Fig. 5, by providing more stable geometric references, our approach ensures superior trajectory accuracy in streaming scenarios.

Table 3: Quantitative 3D reconstruction results on 7-Scenes and NRGBD on long sequences.

Method	7-Scenes						NRGBD					
	Acc↓		Comp↓		NC↑		Acc↓		Comp↓		NC↑	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
CUT3R [60]	0.107	0.072	0.061	0.014	0.543	0.563	0.139	0.100	0.097	0.036	0.580	0.626
TTT3R [15]	0.035	0.021	0.023	0.004	0.567	0.603	0.100	0.056	0.048	0.024	0.604	0.670
LongStream [16]	0.077	0.047	0.030	0.006	0.568	0.605	0.119	0.082	0.028	0.006	0.630	0.726
InfiniteVGGT [75]	0.041	0.021	0.026	0.005	0.565	0.599	0.067	0.044	0.022	0.004	0.634	0.730
OVGGT [38]	0.028	0.012	0.020	0.004	0.576	0.617	0.033	0.023	0.011	0.003	0.641	0.738
FrameVGGT*	0.029	0.013	0.022	0.004	**0.581**	**0.626**	0.030	0.021	0.013	0.004	**0.680**	**0.807**
LingBot-Map	0.019	0.009	0.018	**0.003**	0.571	0.609	0.025	0.016	**0.007**	**0.002**	0.618	0.693
Ours*	0.020	0.010	**0.014**	**0.003**	0.572	0.610	0.029	0.018	0.009	0.003	0.627	0.707
Ours	**0.013**	**0.005**	0.015	0.004	0.572	0.611	**0.018**	**0.011**	**0.007**	**0.002**	0.645	0.738

Table 4: Camera Pose Estimation results on TUM-dynamics, Sintel, and ScanNet datasets.

Method	Sintel			TUM-dynamics			ScanNet		
	ATE ↓	RPE trans ↓	RPE rot ↓	ATE ↓	RPE trans ↓	RPE rot ↓	ATE ↓	RPE trans ↓	RPE rot ↓
CUT3R [60]	0.210	0.071	0.607	0.045	0.014	0.439	0.105	0.022	0.627
TTT3R [15]	0.209	0.082	0.683	0.029	0.013	0.380	0.072	0.021	0.805
StreamVGGT [81]	0.232	0.082	0.669	0.025	0.012	0.311	0.048	0.020	0.629
LongStream [16]	0.194	0.076	0.604	0.036	0.014	0.366	0.037	0.014	0.431
Ours	**0.116**	**0.054**	**0.425**	**0.017**	**0.009**	**0.304**	**0.034**	**0.013**	**0.374**

4.5 Ablation Study

To evaluate the effectiveness of our proposed components, we conduct ablation studies within the Sparse Setting on 3D reconstruction on the 7-Scenes and NRGBD datasets. For fair comparison, we re-finetune VGGT [59] using the identical training datasets and settings as our method, denoted as StreamVGGT*. Starting from this baseline (which employs the VGGT backbone and absolute pose supervision), we progressively incorporate our proposed designs: the frame-equivariant backbone (π^3 [65]), the prefix-relative pose supervision, and the lookahead mechanism. The detailed results are summarized in Tab. 5, which highlight the following key findings:

Table 5: Ablation study on 3D reconstruction under the Sparse Setting on NRGBD and 7-Scenes.

Config	Settings			NRGBD				7-Scenes			
	Backbone	Pose Loss	Lookahead	Acc↓		Comp↓		Acc↓		Comp↓	
				Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
StreamVGGT [81]	VGGT	Absolute	-	0.085	0.044	0.079	0.038	0.132	0.058	0.116	0.042
StreamVGGT*	VGGT	Absolute	-	0.070	0.043	0.070	0.041	0.101	0.063	0.106	0.067
w/o Rel.& Look.	π^3	Absolute	-	0.063	0.043	0.071	0.042	0.064	0.040	0.083	0.059
w/o Look.	π^3	Prefix-Relative	-	0.055	0.033	0.071	0.025	0.054	0.032	0.069	0.048
Ours (Full)	π^3	Prefix-Relative	✓	0.035	0.020	0.041	0.023	0.052	0.032	0.069	0.048

Frame-equivariant backbone. Comparing Rows 2 and 3, we keep the supervision unchanged and only replace the VGGT [59] backbone and pretrained initialization, which uses a dedicated first-frame reference, with the frame-equivariant π^3 [65] backbone. This alone leads to a substantial improvement across both datasets. The result indicates that a frame-equivariant pretrained model is better matched to the causal streaming setting, where early frames typically contain the least geometric information. Unlike a first-frame-anchored design, it does not impose a privileged dependence on the earliest frame, thereby substantially mitigating the initial-frame bias discussed in section 1.

Prefix-Relative Supervision: As shown in Row 4, switching from absolute to prefix-relative pose loss further improves reconstruction accuracy. This demonstrates that shifting the focus to relative geometric consistency imposes tighter constraints on the model, making learning less sensitive to deviations in early frames.

Lookahead Mechanism: The full model (Row 5) achieves the best performance by integrating the lookahead mechanism. The results confirm that the Lookahead module explicitly alleviates the challenge of information asymmetry in the streaming model by leveraging future context to refine history estimates.



Figure 5: Pose visualization comparison on the TUM dataset.

5 Conclusion

In this paper, we present Stable3R, a holistic framework designed to address the inherent instability in streaming 3D reconstruction caused by early-frame bias. By identifying the critical bottleneck of over-reliance on information-poor initial estimates, we develop two complementary designs: a lookahead attention mechanism to enrich historical caches with future context, and a frame-equivariant architecture with prefix-relative pose supervision to stabilize geometric references. Comprehensive experiments across various benchmarks demonstrate that our method achieves superior reconstruction quality and camera pose estimation accuracy. Notably, Stable3R exhibits robust generalization, maintaining high performance even in challenging settings such as sparse views and long-sequence inference.

Acknowledgments and Disclosure of Funding

This work was conducted during Jincheng Xiong's internship at AMAP, Alibaba. We thank Linzhe Shi and Xin Zhang for their valuable discussions and feedback.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *CVPR*, pages 6290–6301, 2022.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM TOG*, 28(3):24, 2009.

[3] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

[4] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

[5] Zuria Bauer, Francisco Gomez-Donoso, Edmanuel Cruz, Sergio Orts-Escolano, and Miguel Cazorla. Uasol, a large-scale high-resolution outdoor stereo dataset. *Scientific data*, 6(1):162, 2019.

[6] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *BMVC*, volume 11, pages 1–11, 2011.

[7] Nicolas Boizard, Théo Deschamps-Berger, Hippolyte Gisserot-Boukhlef, Céline Hudelot, and Pierre Colombo. Bidirlm: From text to omnimodal bidirectional encoders by adapting and composing causal llms. *arXiv preprint arXiv:2604.02045*, 2026.

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

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

[10] Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, et al. Mixture of contexts for long video generation. *arXiv preprint arXiv:2508.21058*, 2025.

[11] 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 Trans. on robotics*, 37(6):1874–1890, 2021.

[12] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *3DV*, pages 667–676, 2017.

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

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

[15] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3r: 3d reconstruction as test-time training. In *ICLR*, 2026. URL `https://openreview.net/forum?id=aMs6FtNaY5`.

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

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

[18] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024.

[19] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022.

[20] 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.

[21] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022.

[22] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

[23] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, pages 873–881, 2015.

[24] Jian Han, Jinlai Liu, Jiahuan Wang, Bingyue Peng, and Zehuan Yuan. Generative refinement networks for visual synthesis, 2026. URL `https://arxiv.org/abs/2604.13030`.

[25] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[26] Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T Barron, Noah Snavely, and Aleksander Holynski. Zipmap: Linear-time stateful 3d reconstruction via test-time training. *arXiv preprint arXiv:2603.04385*, 2026.

[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 *ICLR*, 2026.

[28] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, pages 71–91. Springer, 2024.

[29] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Selective attention improves transformer. *arXiv preprint arXiv:2410.02703*, 2024.

[30] Xianming Li and Jing Li. Bellm: Backward dependency enhanced large language model for sentence embeddings. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 792–804, 2024.

[31] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018.

[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 *ICLR*, 2026. URL `https://openreview.net/forum?id=PjviszIZf1`.

[33] Ailiang Lin, Zhuoyun Li, Kotaro Funakoshi, and Manabu Okumura. Causal2vec: Improving decoder-only llms as versatile embedding models. *arXiv preprint arXiv:2507.23386*, 2025.

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

[35] 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 *CVPR*, pages 22160–22169, 2024.

[36] Changkun Liu, Bin Tan, Zeran Ke, Shangzhan Zhang, Jiachen Liu, Ming Qian, Nan Xue, Yujun Shen, and Tristan Braud. PLANA3r: Zero-shot metric planar 3d reconstruction via feed-forward planar splatting. In *NeurIPS*, 2026. URL `https://openreview.net/forum?id=YTwRZP8mNO`.

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[38] Si-Yu Lu, Po-Ting Chen, Hui-Che Hsu, Sin-Ye Jhong, Wen-Huang Cheng, and Yung-Yao Chen. Ovggt: O (1) constant-cost streaming visual geometry transformer. *arXiv preprint arXiv:2603.05959*, 2026.

[39] Dominic Maggio and Luca Carlone. Vggt-slam 2.0: Real-time dense feed-forward scene reconstruction, 2026. URL `https://arxiv.org/abs/2601.19887`.

[40] Dominic Rosario Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: Dense RGB SLAM optimized on the SL(4) manifold. In *NeurIPS*, 2026. URL `https://openreview.net/forum?id=63ljkCGMhE`.

[41] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *CVPR*, pages 4981–4991, 2023.

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

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

[44] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[45] Xiaohuan Pei, Tao Huang, YanXiang Ma, and Chang Xu. Rethinking causal mask attention for vision-language inference. *arXiv preprint arXiv:2505.18605*, 2025.

[46] Haosong Peng, Hao Li, Jiaqi Chen, Yuhao Pan, Runmao Yao, Yalun Dai, Fushuo Huo, Fangzhou Hong, Zhaoxi Chen, Haozhao Wang, et al. Spatialbench: Is your spatial foundation model an all-round player? *arXiv preprint arXiv:2605.27367*, 2026.

[47] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020.

[48] 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 *ICCV*, pages 10901–10911, 2021.

[49] 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 *ICCV*, pages 10912–10922, 2021.

[50] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016.

[51] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, pages 501–518. Springer, 2016.

[52] 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 *CVPR*, pages 2930–2937, 2013.

[53] Zhuoqing Song, Peng Sun, Huizhuo Yuan, and Quanquan Gu. Causal attention with lookahead keys. *arXiv preprint arXiv:2509.07301*, 2025.

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

[55] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE TPAMI*, 25(7):787–800, 2003.

[56] 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 *CVPR*, pages 2446–2454, 2020.

[57] Philippe Tillet, Hsiang-Tsung Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pages 10–19, 2019.

[58] Haonan Wang, Hanyu Zhou, Haoyue Liu, and Luxin Yan. 4d-vggt: A general foundation model with spatiotemporal awareness for dynamic scene geometry estimation. *arXiv preprint arXiv:2511.18416*, 2025.

[59] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, pages 5294–5306, 2025.

[60] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, pages 10510–10522, 2025.

[61] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *CVPR*, pages 5261–5271, 2025.

[62] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, pages 20697–20709, 2024.

[63] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, et al. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *CVPR*, pages 19757–19767, 2024.

[64] 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 *IROS*, pages 4909–4916. IEEE, 2020.

[65] 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. In *ICLR*, 2026. URL `https://openreview.net/forum?id=DTQIjngDta`.

[66] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *CVPR*, pages 803–814, 2023.

[67] Yuqi Wu, Wenzhao Zheng, Sicheng Zuo, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Embodiedocc: Embodied 3d occupancy prediction for vision-based online scene understanding. In *ICCV*, pages 26360–26370, 2025.

[68] 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 *CVPR*, pages 22378–22389, 2024.

[69] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *ICLR*, 2024. URL `https://openreview.net/forum?id=NG7sS51zVF`.

[70] Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, et al. Scal3r: Scalable test-time training for large-scale 3d reconstruction. *arXiv preprint arXiv:2604.08542*, 2026.

[71] Zhisong Xu and Takeshi Oishi. Framevggt: Geometry-aligned frame-level memory for bounded streaming vggt. *arXiv preprint arXiv:2603.07690*, 2026.

[72] 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 *CVPR*, pages 21924–21935, 2025.

[73] 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 *CVPR*, pages 1790–1799, 2020.

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

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

[76] 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.

[77] 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 *CVPR*, pages 21936–21947, 2025.

[78] Siyue Zhang, Yilun Zhao, Liyuan Geng, Arman Cohan, Luu Anh Tuan, and Chen Zhao. Diffusion vs. autoregressive language models: A text embedding perspective. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4273–4303, 2025.

[79] 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 *ICCV*, pages 19855–19865, 2023.

[80] Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026.

[81] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *ICLR*, 2026. URL `https://openreview.net/forum?id=5APgTKsnx8`.

A Additional Experiments and Analysis

In this section, we provide additional experiments and analyses to further investigate the stability of geometric references, the behavior of lookahead attention, and the efficiency and scalability of Stable3R. We first extend the early-frame bias analysis in the main paper to different datasets and perturbation settings. We then analyze the lookahead mechanism from the perspectives of update range, geometric interpretation, and long-horizon stability. Finally, we report detailed runtime and memory measurements, additional long-sequence experiments, and comparisons with concurrent methods.

A.1 Additional Analysis of Early-Frame Bias

A.1.1 Stability across Prefix Lengths and Datasets

We first extend the geometric context probe introduced in Fig. 1 to NRGBD, 7-Scenes, TUM-dynamics, and Sintel. Following the same protocol, we fix the initial and probe frames while varying the number of intermediate observations from 0 to 8. We report the mean Absolute Trajectory Error (ATE) of the probe frames together with the ATE range across different prefix configurations.

As shown in Tab. 6, StreamVGGT exhibits noticeable variation as the number of intermediate observations changes, particularly on NRGBD, 7-Scenes, and Sintel. In contrast, Stable3R consistently achieves lower probe-frame ATE and substantially smaller variation across prefix lengths. These results indicate that the early-frame bias observed in Fig. 1 generalizes across datasets, and that Stable3R provides more stable geometric references under varying amounts of historical context.

Table 6: Geometric context probe across different datasets and prefix lengths. We fix the initial and probe frames while varying the number of intermediate frames. Lower ATE and a smaller ATE range indicate more stable historical geometric references.

Dataset	Method	0 frame	2 frames	4 frames	6 frames	8 frames	ATE Range↓
NRGBD	StreamVGGT	0.13539	0.12477	0.11450	0.10707	0.10022	0.03517
	Stable3R	**0.04137**	**0.03182**	**0.03105**	**0.02992**	**0.03064**	**0.01145**
TUM-dynamics	StreamVGGT	0.01676	0.01556	0.01513	0.01507	0.01496	0.00180
	Stable3R	**0.01378**	**0.01300**	**0.01317**	**0.01300**	**0.01325**	**0.00078**
7-Scenes	StreamVGGT	0.12108	0.12250	0.09757	0.08873	0.07615	0.04635
	Stable3R	**0.07418**	**0.07319**	**0.07335**	**0.05919**	**0.05821**	**0.01597**
Sintel	StreamVGGT	0.14975	0.17687	0.14751	0.13530	0.13171	0.04516
	Stable3R	**0.06194**	**0.06104**	**0.06255**	**0.06512**	**0.06416**	**0.00408**

A.1.2 Sensitivity to Initial-Frame Quality

We next directly evaluate the sensitivity to the quality of the initial observation. Specifically, we apply Gaussian blur only to the first frame while keeping all subsequent observations unchanged, and measure the resulting increase in probe-frame ATE. This perturbation isolates the influence of the initial observation without modifying the remaining sequence.

As shown in Tab. 7, degrading the first frame consistently leads to a larger ATE increase for StreamVGGT across all four datasets. Stable3R is substantially less affected by the same perturbation, indicating reduced dependence on the quality of the initial estimate. This provides direct evidence that alleviating first-frame anchoring improves streaming stability.

A.1.3 Sensitivity to Historical Frame Overlap

Besides prefix length and initial-frame quality, we further study whether geometric stability depends on the overlap of historical observations. We fix the initial and probe frames while varying the intermediate frames. Overlap is quantified according to the visibility of scene points in the probe view, and the samples are divided into three equally sized groups corresponding to low, medium, and high overlap.

Table 7: Sensitivity to the quality of the initial frame. We apply Gaussian blur only to the first frame and report the increase in probe-frame ATE. Lower values indicate weaker dependence on the initial observation.

Dataset	Method	2 frames	4 frames	6 frames	8 frames
NRGBD	StreamVGGT	0.01642	0.01138	0.00741	0.00708
	Stable3R	**0.00006**	**0.00160**	**0.00237**	**0.00198**
TUM-dynamics	StreamVGGT	0.00651	0.00495	0.00363	0.00235
	Stable3R	**0.00081**	**0.00037**	**0.00029**	**0.00056**
7-Scenes	StreamVGGT	0.00531	0.00599	0.00861	0.00694
	Stable3R	**0.00040**	**0.00239**	**0.00098**	**0.00037**
Sintel	StreamVGGT	0.01992	0.03024	0.03649	0.03766
	Stable3R	**0.01097**	**0.00095**	**0.00202**	**0.00292**

As reported in Tab. 8, Stable3R achieves lower probe-frame ATE across all overlap levels and generally exhibits smaller variation among the three groups. This indicates that Stable3R is less sensitive to the composition of intermediate observations and provides more stable historical geometric references.

Table 8: Sensitivity to the overlap of intermediate observations. Lower ATE and a smaller ATE range indicate greater stability under changes in historical frame composition.

Dataset	Method	Low	Medium	High	ATE Range↓
NRGBD	StreamVGGT	0.1341	0.1346	0.1315	0.0031
	Stable3R	**0.0363**	**0.0367**	**0.0363**	**0.0004**
TUM-dynamics	StreamVGGT	0.0162	0.0168	0.0168	0.0006
	Stable3R	**0.0138**	**0.0136**	**0.0135**	**0.0003**
7-Scenes	StreamVGGT	0.1167	0.1030	0.1009	0.0158
	Stable3R	**0.0710**	**0.0657**	**0.0570**	**0.0140**
Sintel	StreamVGGT	0.1414	0.1381	0.1461	0.0080
	Stable3R	**0.0634**	**0.0666**	**0.0626**	**0.0040**

A.2 Analysis of Lookahead Attention

We next analyze the behavior of lookahead attention in greater detail. In particular, we study how much historical context needs to be updated, whether the learned lookahead representation correlates with geometric properties, and whether its recursive accumulation remains stable over long sequences.

A.2.1 Sensitivity to Lookahead Update Range

We first study how many historical frames need to be refined by lookahead attention. In addition to the default setting that updates the full history, we restrict the lookahead update to only the 10 most recent historical frames, or disable historical updates entirely.

As shown in Tab. 9, updating only the most recent 10 frames results in nearly identical reconstruction quality to full-history refinement on both NRGBD and 7-Scenes. In contrast, disabling historical refinement leads to a clear performance drop, particularly on NRGBD. This suggests that most of the benefit of lookahead attention is captured by refining relatively recent historical representations, while increasingly distant frames provide only limited additional gains.

Table 9: Sensitivity to the number of historical frames updated by lookahead attention under the Mid-length setting.

Dataset	Method	Acc Mean↓	Acc Med.↓	Comp Mean↓	Comp Med.↓	NC Mean↑	NC Med.↑
7-Scenes	Stable3R	0.019	0.010	0.022	0.010	0.722	0.834
	Stable3R (update 10)	0.019	0.010	0.022	0.010	0.721	0.834
	Stable3R (update 0)	0.023	0.012	0.023	0.011	0.721	0.835
NRGBD	Stable3R	0.022	0.014	0.020	0.010	0.884	0.987
	Stable3R (update 10)	0.023	0.014	0.021	0.010	0.884	0.986
	Stable3R (update 0)	0.034	0.022	0.029	0.013	0.863	0.983

A.2.2 Geometric Interpretation of Lookahead Attention

We further investigate whether the learned lookahead representation reflects meaningful geometric properties of the historical observations. We analyze it from two complementary perspectives: frame-level co-visibility and token-level geometric reliability.

Co-visibility. For each current–historical frame pair, we estimate co-visibility through point-cloud projection and divide all pairs into equally sized low-, medium-, and high-co-visibility groups. We then measure the proportion of historical attention weights that increase after applying the lookahead term.

As shown in Tab. 10, highly co-visible historical frames are more likely to receive increased relative attention than low-overlap frames. This trend is consistent on both 7-Scenes and NRGBD, suggesting that lookahead preferentially preserves geometrically relevant historical observations.

Table 10: Relationship between frame co-visibility and lookahead attention. We report the proportion of historical attention weights that increase after applying the lookahead term.

Dataset	Low co-vis.	Medium co-vis.	High co-vis.
7-Scenes	31.1%	33.4%	39.5%
NRGBD	26.4%	32.3%	39.2%

Geometric reliability. For each historical token, we compute its predicted point-cloud error and mean lookahead bias. We divide all tokens into five equally sized groups according to their geometric error and report the average bias relative to the lowest-error group.

As shown in Tab. 11, the lookahead bias increases consistently with geometric error. Since the lookahead term is subtracted from the causal attention logits in eq. (5), historical tokens with larger geometric errors receive stronger suppression. Together with the co-visibility analysis above, these results show that the learned lookahead representation correlates with both geometric relevance and reliability.

Table 11: Relationship between historical-token geometric error and lookahead bias. Tokens are equally divided into five groups according to point-cloud error. The relative bias is normalized by the mean bias of the lowest-error group.

Dataset	Metric	Group 1	Group 2	Group 3	Group 4	Group 5
7-Scenes	Point-cloud error	3.92	5.72	7.27	9.13	16.87
	Relative bias	100.0%	104.5%	107.9%	113.2%	123.5%
NRGBD	Point-cloud error	2.80	4.09	5.15	6.72	15.43
	Relative bias	100.0%	105.8%	109.8%	120.3%	133.3%

A.2.3 Long-Horizon Stability of Lookahead States

The recursive formulation in eq. (6) continuously accumulates newly observed information into the lookahead state. We therefore analyze whether this accumulation leads to an increasingly dominant or unstable attention bias over long sequences.

We track the RMS norm of the first-frame lookahead state U and the attention assigned to the first frame as the sequence length increases from 4 to 128 frames. As shown in Tab. 12, the magnitude of the lookahead state grows approximately linearly with sequence length. However, its effect on normalized attention remains stable: lookahead consistently reduces the attention assigned to the first frame by approximately 20–24% across all evaluated lengths.

These results indicate that although the internal lookahead representation accumulates over time, its influence on normalized attention does not become increasingly dominant within the evaluated sequence range.

Table 12: Long-horizon behavior of the first-frame lookahead state. We report its RMS norm together with the attention assigned to the first frame with and without lookahead.

Seq. Length	U RMS	Attn. w/ Lookahead	Attn. w/o Lookahead	Relative Change
4	183	19.371%	25.239%	-23.2%
8	424	9.734%	12.744%	-23.6%
16	900	4.804%	6.238%	-23.0%
32	1,849	2.406%	3.102%	-22.4%
64	3,806	1.271%	1.608%	-21.0%
128	7,608	0.666%	0.835%	-20.2%

A.2.4 Replaying Earlier Frames with Refined Lookahead States

Stable3R refines historical cache representations as new observations arrive, but does not revise previously emitted predictions. To further examine this design, we perform a replay experiment in which the lookahead states obtained after a complete online pass are retained and used to reprocess earlier frames.

As shown in Tab. 13, replaying earlier frames with the enriched states does not improve reconstruction and instead degrades performance. We attribute this to a mismatch between enriched historical states and early-frame queries that is not encountered during training. This result further clarifies that Stable3R is designed to improve subsequent online predictions through refined historical representations, rather than to retroactively revise previous outputs.

Table 13: Reconstruction results when replaying the sequence with lookahead states enriched during the first online pass.

Dataset	Setting	Acc↓	Acc Med.↓	Comp↓	Comp Med.↓	NC↑	NC Med.↑
7-Scenes	Stable3R	0.019	0.010	0.022	0.010	0.722	0.834
	Enriched-U replay	0.022	0.012	0.022	0.009	0.721	0.834
NRGBD	Stable3R	0.022	0.014	0.020	0.010	0.884	0.987
	Enriched-U replay	0.032	0.023	0.026	0.014	0.860	0.981

A.3 Streaming Efficiency and Scalability

Since lookahead attention augments the conventional KV cache with additional states, we provide a detailed analysis of its computational and memory overhead. All efficiency experiments in this section are measured on a single NVIDIA A100-SXM4-80GB GPU.

A.3.1 Runtime and Memory Overhead of Lookahead Attention

The QU-KV cache stores two additional lookahead states compared with standard KV-cached causal attention, introducing approximately 50 MiB of additional memory per frame. The sequential execution of lookahead update and attention also incurs additional latency.

We compare Stable3R with Stable3R without lookahead and StreamVGGT in Tab. 14. As expected, removing lookahead reduces both runtime and memory consumption. Nevertheless, the full Stable3R model remains more efficient than StreamVGGT across all evaluated sequence lengths. These results characterize the accuracy–efficiency trade-off introduced by lookahead refinement.

Table 14: Runtime and memory comparison for unbounded-memory streaming inference. All results are measured on a single A100-SXM4-80GB GPU.

Method	Metric	50 frames	150 frames	200 frames	300 frames
Stable3R	FPS↑	7.98	3.84	3.08	2.17
	Latency (ms)↓	189.35	463.78	594.34	859.40
	Mem. (GiB)↓	9.54	19.49	24.53	34.43
Stable3R w/o Lookahead	FPS↑	11.75	7.01	5.93	3.85
	Latency (ms)↓	104.70	234.23	302.46	432.31
	Mem. (GiB)↓	7.05	12.07	15.04	20.78
StreamVGGT	FPS↑	5.10	2.56	2.28	2.01
	Latency (ms)↓	259.93	604.13	780.55	1133.37
	Mem. (GiB)↓	13.08	25.65	31.93	40.74

A.3.2 Efficiency of the Triton Implementation

We further evaluate the practical benefit of the customized Triton implementation introduced in section 3.2. We compare it with a straightforward PyTorch implementation of the same lookahead computation and report forward latency and peak GPU memory.

As shown in Tab. 15, the Triton implementation substantially reduces both runtime and memory consumption, with the benefit becoming increasingly pronounced for longer sequences. At 64 frames, the forward latency decreases from 1058.8 ms to 342.0 ms, while peak memory decreases from 9.13 GiB to 1.06 GiB. These results validate the effectiveness of the block-wise parallelization and memory-efficient tiling strategy.

Table 15: Efficiency comparison between the PyTorch and Triton implementations of lookahead attention.

Metric	Impl.	4	8	16	32	64
Forward time (ms)↓	PyTorch	5.66	18.58	67.74	262.25	1058.80
	Triton	**1.96**	**4.60**	**17.16**	**73.00**	**342.00**
Peak memory (GiB)↓	PyTorch	0.53	1.10	2.25	4.54	9.13
	Triton	**0.07**	**0.13**	**0.27**	**0.53**	**1.06**

A.3.3 Bounded-Memory Long-Sequence Inference

For practical long-sequence deployment, we additionally combine Stable3R with the frame-level bounded-memory retention strategy used in [71]. We use a fixed 20-frame memory budget consisting of 16 recent frames and 4 sparse anchors. For brevity, we denote this frame-level retention mechanism as FERM in the following tables.

As shown in Tab. 16, direct inference exhibits approximately linear growth in cache size, GPU memory, and latency as the sequence becomes longer. In contrast, FERM keeps the cache size fixed at approximately 2.61 GiB. Across sequences from 25 to 300 frames, peak GPU memory remains below 8.68 GiB and latency remains between 109.7 and 140.1 ms.

We further compare the bounded-memory efficiency of Stable3R against OVGGT, FrameVGGT, and InfiniteVGGT. As shown in Tab. 17, Stable3R achieves throughput comparable to OVGGT while exhibiting a flatter GPU-memory growth curve. FrameVGGT and InfiniteVGGT require substantially more GPU memory and achieve lower throughput under the evaluated setting.

A.4 Additional Long-Sequence Evaluation

A.4.1 Effect of the Frame-Level Retention Strategy

The long-sequence evaluation in the main paper adopts FERM to control inference cost. To distinguish the contribution of Stable3R itself from that of the external memory-retention mechanism,

Table 16: Scaling behavior of direct and bounded-memory inference as the sequence length increases.

Frames	Direct				FERM			
	Latency (ms)↓	FPS ↑	Mem. (GiB)↓	Cache (GiB)↓	Latency (ms)↓	FPS ↑	Mem. (GiB)↓	Cache (GiB)↓
25	117.36	10.891	7.09	2.49	109.7	9.09	8.01	2.61
50	189.35	7.981	9.54	4.98	111.4	8.98	8.25	2.61
100	322.31	5.228	14.54	9.96	114.8	8.77	8.33	2.61
150	463.78	3.839	19.49	14.93	132.1	7.57	8.34	2.61
200	594.34	3.075	24.53	19.91	136.6	7.32	8.43	2.61
250	733.64	2.531	29.45	24.89	140.0	7.11	8.51	2.61
300	859.40	2.169	34.43	29.87	140.1	7.14	8.68	2.61

Table 17: Efficiency comparison with bounded-memory streaming baselines.

Method	Metric	50 frames	150 frames	200 frames	300 frames
Stable3R (w/ FERM)	FPS↑	8.98	7.57	7.32	7.14
	Latency (ms)↓	111.4	132.1	136.6	140.1
	Mem. (GiB)↓	8.25	8.34	8.43	8.68
OVGGT	FPS↑	8.49	8.20	8.18	8.13
	Latency (ms)↓	117.8	122.0	122.3	123.0
	Mem. (GiB)↓	8.65	9.57	10.03	10.96
FrameVGGT	FPS↑	6.14	5.81	5.94	5.80
	Latency (ms)↓	162.9	172.1	168.4	172.4
	Mem. (GiB)↓	15.07	17.39	18.24	19.17
InfiniteVGGT	FPS↑	5.53	3.91	3.68	3.56
	Latency (ms)↓	180.8	255.8	271.7	280.9
	Mem. (GiB)↓	15.21	16.28	16.77	17.71

we additionally evaluate Stable3R by directly processing the same 300-frame sequences without bounded-memory retention.

As shown in Tab. 18, direct inference achieves better reconstruction accuracy on most metrics than the bounded-memory variant. This indicates that the long-sequence reconstruction gains primarily originate from Stable3R rather than from FERM. The retention mechanism mainly serves as an efficient extension that substantially reduces memory and computation while preserving comparable reconstruction quality.

Table 18: Effect of the frame-level retention strategy on 300-frame reconstruction. Direct inference processes the full history, while FERM uses bounded memory.

Method	7-Scenes						NRGBD					
	Acc↓		Comp↓		NC↑		Acc↓		Comp↓		NC↑	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
Stable3R (w/ FERM)	0.020	0.010	**0.014**	**0.003**	**0.572**	0.610	0.029	0.018	0.009	0.003	0.627	0.707
Stable3R (w/o FERM)	**0.013**	**0.005**	0.015	0.004	**0.572**	**0.611**	**0.018**	**0.011**	**0.007**	**0.002**	**0.645**	**0.738**

A.4.2 Evaluation on Long3D

We additionally evaluate Stable3R on the Long3D benchmark using the same bounded-memory inference strategy. As shown in Tab. 19, Stable3R improves reconstruction accuracy from 2.09 to 1.72 and normal consistency from 0.52 to 0.54 compared with InfiniteVGGT.

Stable3R obtains worse completeness under this evaluation. We observe that the two methods use different effective fields of view: Stable3R crops image borders during inference, which removes

part of the reconstructed point cloud and affects the completeness metric. Nevertheless, the results demonstrate that Stable3R can be extended to substantially longer sequences without numerical instability.

Table 19: Long-sequence reconstruction results on the Long3D benchmark.

Method	Acc↓	Comp↓	NC↑
InfiniteVGGT	2.09	**0.85**	0.52
Stable3R (w/ FERM)	**1.72**	1.84	**0.54**

B Technical appendices and supplementary material

Qualitative Results We provide additional results on the **mid-length** and **long-sequence** setting, as shown in Fig. 6 and Fig. 7. Our method produces more consistent geometric reconstruction results.

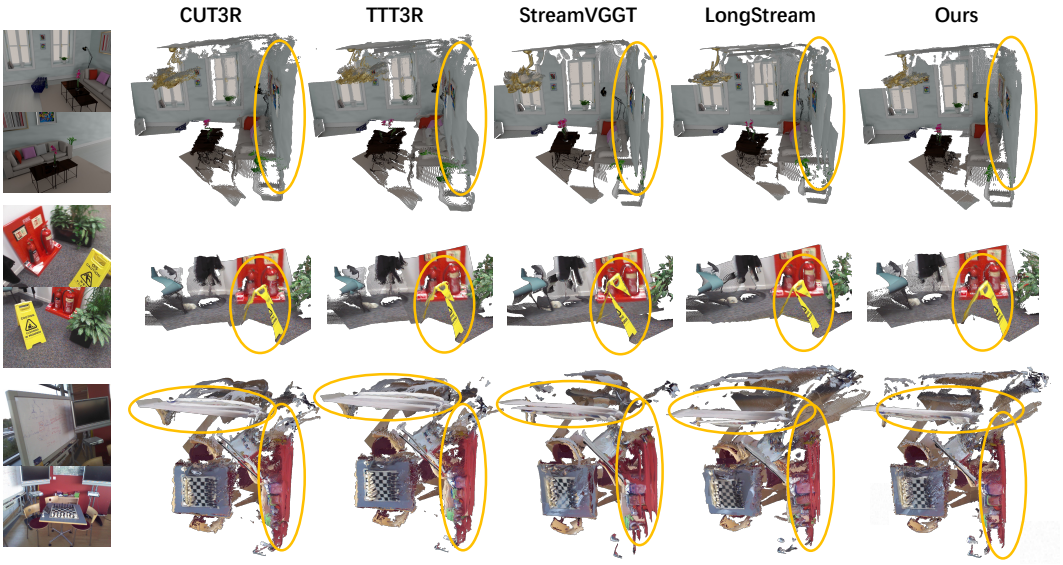


Figure 6: Qualitative Reconstruction Results on the NRGBD and 7-scenes datasets on mid-length settings.

C Limitations and Future Work

While Stable3R achieves strong performance across multiple streaming reconstruction benchmarks, several directions remain for future work. First, our current study focuses on short-sequence training (up to 24 frames). Nevertheless, with a training-free extension strategy, our model generalizes well to 300-frame sequences and already achieves strong performance in long-horizon reconstruction. At the same time, concurrent works such as LongStream [16] and LingBot-Map [13] have shown that training on longer sequences can further benefit long-horizon reconstruction. Extending our framework to longer-sequence training is therefore a promising direction that we hope to explore as more computational resources become available.

Second, although our current framework is designed for stable streaming reconstruction, it does not explicitly model temporal correspondence over long horizons. Incorporating stronger temporal modeling and tracking mechanisms could further improve consistency across frames and broaden the applicability of the framework to more dynamic streaming scenarios.

Finally, an important future direction is to extend Stable3R toward dynamic-static decomposition, so that static scene geometry and moving objects can be reconstructed separately. We believe such

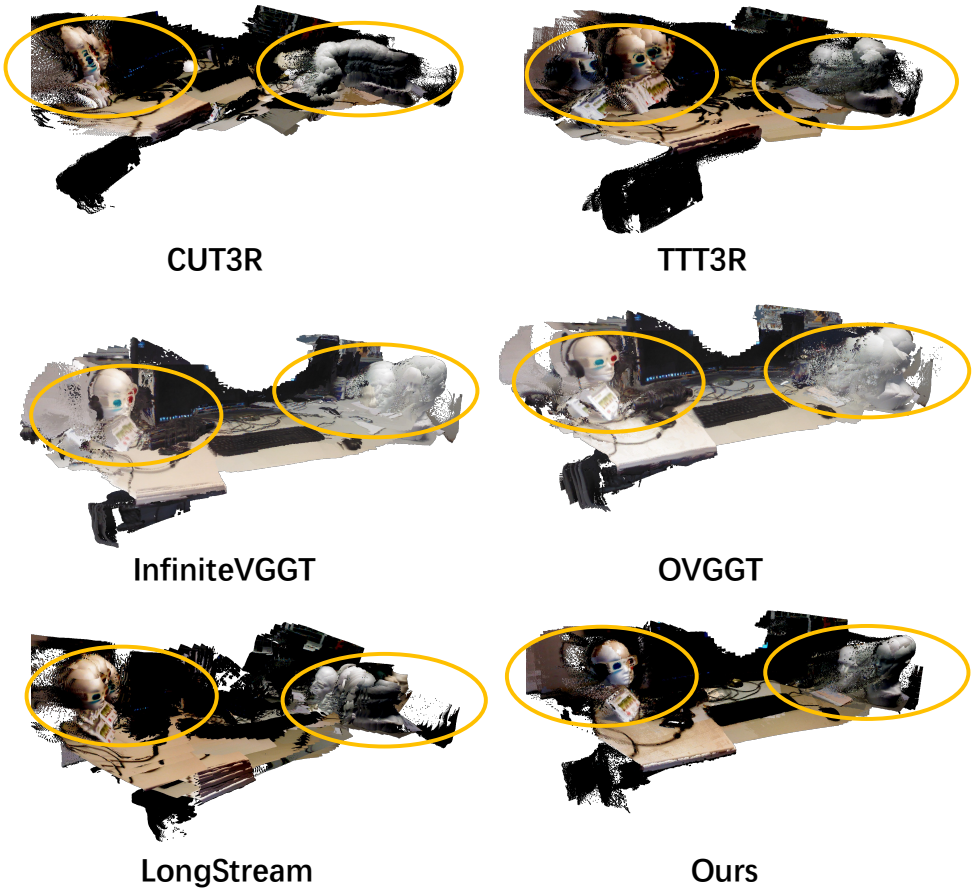


Figure 7: Qualitative Reconstruction Results on the 7-scenes dataset on long-sequence settings.

a formulation would be especially valuable for dynamic environments and could serve as a step toward more general streaming 4D reconstruction.

D Societal Impacts

Positive Social Impact. The advancement of our stable streaming 3D reconstruction technology can lead to several societal benefits: (1) Assistive Navigation: Reliable real-time 3D reconstruction can significantly enhance navigation aids for individuals with visual impairments, providing detailed spatial awareness of their surroundings. (2) Autonomous Systems: Improved stability in pose estimation and reconstruction is crucial for the safe operation of autonomous vehicles and embodied robots, potentially reducing accidents in complex environments. (3) Cultural Preservation: The technology can be utilized for the rapid and accurate digital archiving of historical sites and artifacts, preserving cultural heritage for future generations.

Potential Negative Impact. We also acknowledge the following risks associated with our technology: (1) Privacy Invasion: High-fidelity 3D reconstruction capabilities could potentially be misused for unauthorized surveillance. (2) Resource Consumption: Training and deploying deep learning models for 3D reconstruction require substantial computational resources, contributing to energy consumption. (3) Misuse in Deception: The reconstructed 3D models could theoretically be exploited to generate highly realistic synthetic data, which might be used to deceive other AI systems.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: See Abstract and section 1.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See section C in appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have provided comprehensive details regarding our network architecture, datasets, evaluation protocols, and experimental implementation. Additional information can be found in the supplementary materials.

   Guidelines:

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

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

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

Answer: [No]

Justification: The datasets employed in this research are all publicly accessible. Our code has not been released yet. We will release our code as soon as possible in the future.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: See section 4.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: See section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See section 4.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This paper conducted in the paper conforms with the NeurIPS Code of ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: See section D.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our model was trained on public open-source datasets and no risk for misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We already cited all public datasets we used in this paper. See section 4

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The details of the datasets/code/model used in the paper are well documented.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: This paper only used LLM for writing and editing.

Guidelines:

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