DRIVE3R: BLOCK-WISE STREAMING GEOMETRIC FOUNDATION MODELS FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

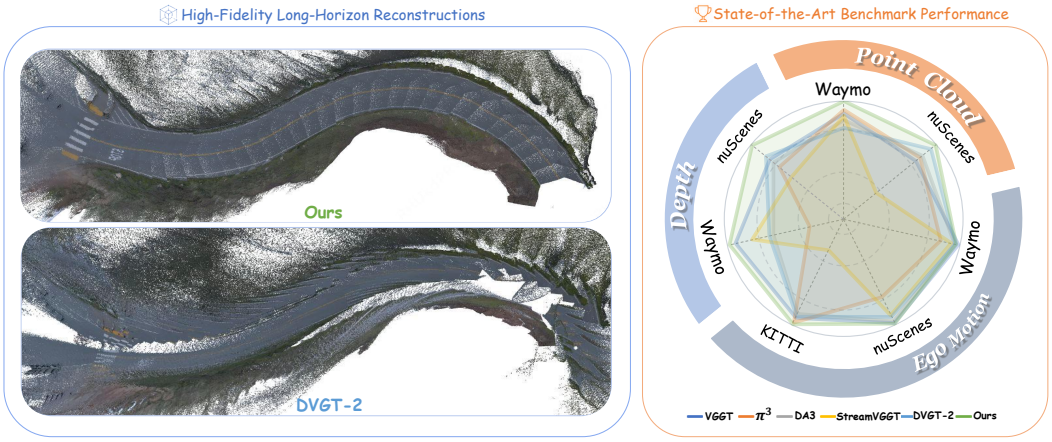

Figure 1: **Teaser of Drive3R.** Compared with the per-frame streaming baseline DVGT-2 (Zuo et al., 2026), Drive3R reconstructs cleaner and more coherent long-horizon driving scenes with better preserved road geometry. The radar chart further shows state-of-the-art performance across depth, point-cloud, and ego-motion metrics on nuScenes (Caesar et al., 2020), Waymo (Sun et al., 2019), and KITTI (Geiger et al., 2013).

ABSTRACT

High-fidelity closed-loop simulation for autonomous driving increasingly relies on dense multi-view 3D reconstruction. While recent feed-forward geometric foundation models have revolutionized per-scene optimization, applying them to continuous, long-horizon driving sequences is computationally intractable due to GPU memory limits and the quadratic complexity of global attention. Existing scaling strategies are fundamentally limited, as causal streaming suffers from compounding scale drift and trajectory deviation, while rigid "chunk-then-align" heuristics fragment global context and produce trajectory discontinuities. To address these limitations, we introduce Drive3R, a block-wise streaming geometric foundation model. Its block-wise causal attention resolves the scalability bottleneck of fully bidirectional attention while avoiding the loss of local future context in purely causal streaming. Its confidence-aware re-anchoring further mitigates long-range drift by detecting unreliable streaming states, re-initializing the KV cache, and stitching refreshed segments through $\mathrm{SE}(3)$ alignment. Extensive experiments demonstrate that Drive3R establishes a new state-of-the-art on nuScenes, KITTI, and Waymo. Specifically, on nuScenes, it improves over the recent DVGT-2 baseline from 0.1806 to 0.1081 in AbsRel, from 0.6750 to 0.5344 in point-cloud Accuracy, and from 1.5812 to 0.9092 in ATE. Code will be released.

1 INTRODUCTION

The recent leap in end-to-end autonomous driving (Hu et al., 2023; Guo et al., 2025; Zhou et al., 2025; Zhang et al., 2026a;b) demands high-fidelity, closed-loop simulation for scalable training and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

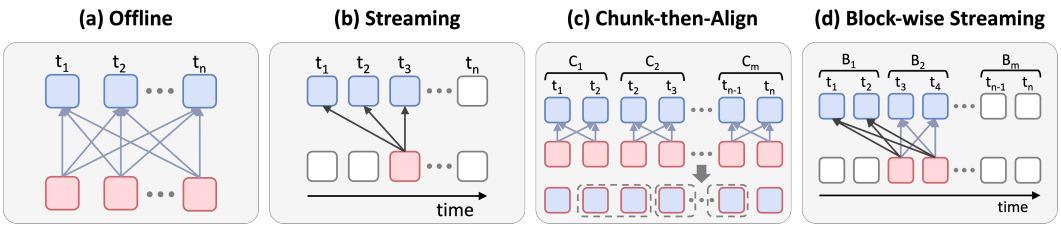

Figure 2: Paradigm comparison between previous sequence processing strategies (offline, streaming, chunk-then-align) and our proposed block-wise streaming approach, Drive3R.

evaluation. As traditional graphics-based rendering struggles to synthesize photorealistic sensor data, modern reconstruction techniques like 3D Gaussian Splatting (GS) have emerged as the de facto standard for building true-to-life digital twins (Kerbl et al., 2023; Zhou et al., 2024; Yan et al., 2024). While early pipelines rely on LiDAR for geometric initialization, its inherent sparsity severely limits the density and completeness of the reconstructed environments (Tian et al., 2023; Tonderski et al., 2024). Consequently, extracting dense and accurate 3D geometry purely from multi-view 2D images is no longer just an alternative, but a critical prerequisite for next-generation driving neural simulators (Yu et al., 2026; Shi et al., 2025).

To achieve this dense geometry recovery, traditional multi-view methods typically rely on *per-scene* global iterative optimization (Schonberger & Frahm, 2016; Kolev et al., 2009; Mildenhall et al., 2021), which is computationally prohibitive and scales poorly to large-scale applications. Geometric foundation models such as VGGT (Wang et al., 2025a; Shen et al., 2025) predict dense point clouds in a single forward pass (Wang et al., 2024; Hong et al., 2023), improving reconstruction efficiency. However, autonomous driving inherently involves continuous, long-horizon video streams, where a single driving log can easily span thousands of frames. As illustrated in Fig. 2a, when confronted with such extensive sequences, existing fully bidirectional feed-forward models hit a hard wall due to the severe constraints of GPU memory limits and the quadratic complexity of global attention mechanisms (Shen et al., 2025). Consequently, processing an entire long sequence via a single forward inference pass becomes computationally intractable.

To bypass this computational bottleneck, recent efforts to scale feed-forward models for long sequences have predominantly diverged into two distinct routes. The first route enforces a strict causal structure for per-frame streaming prediction (Fig. 2b) (Wang et al., 2025b; Chen et al., 2025; Zhuo et al., 2025; Yuan et al., 2026), while the second relies on a "chunk-then-align" (Fig. 2c) (Deng et al., 2025; Lin et al., 2025). Yet, when deployed in complex, long-range autonomous driving scenarios, both paradigms reveal fundamental flaws. By entirely discarding bidirectional context, causal streaming suffers from the continuous accumulation of minor historical errors. Over extended sequences, these errors compound, inevitably triggering severe scale drift and trajectory deviation. Conversely, while the "chunk-then-align" approach preserves bidirectional attention within isolated chunks, its rigid and coarse-grained mechanism struggles to adapt to highly dynamic environments, as predefined chunks often blindly sever continuous motions. Furthermore, the absence of coherent feature propagation between chunks forces a reliance on rigid post-processing alignment in overlapping regions, frequently producing jarring trajectory discontinuities at chunk boundaries.

In response to these fundamental limitations, we present Drive3R, a block-wise streaming geometric foundation model explicitly tailored for the long-horizon demands of autonomous driving (Fig. 2d). Its architecture is driven by two synergistic innovations, specifically *block-wise causal attention* and *confidence-aware re-anchoring*. First, to combat the scale drift inherent in pure streaming, we introduce block-wise causal attention. By allowing tokens within each block to attend bidirectionally, this mechanism establishes a robust local geometric context, effectively halting the monotonic error amplification along the temporal axis. Concurrently, strict causality is enforced across blocks. Unlike the isolated "chunk-then-align" approach that fragments global context, this design leverages cross-block KV cache reuse to continuously propagate historical geometric features, thereby maintaining long-range consistency while guaranteeing a highly scalable streaming inference complexity. Second, to address the rigidity of fixed-length chunking, we propose confidence-aware re-anchoring. Equipped with an additional lightweight confidence head, our Drive3R dynamically predicts the accumulated

drift relative to the initial KV cache state with minimal computational overhead. During online inference, if this confidence score drops below a critical threshold, indicating potential trajectory degradation, the model proactively discards the stale KV cache, backtracks a few frames to re-initialize a fresh block, and seamlessly stitches the new segment into the global trajectory via $\mathrm{SE}(3)$ alignment on overlapping frames. Ultimately, this adaptive mechanism ensures continuous, discontinuity-free geometry recovery even in highly dynamic environments.

By synergizing these two designs, Drive3R elegantly unifies intra-block bidirectional precision, cross-block streaming throughput, and drift-aware long-sequence stability within a single network. Extensive experiments validate that our approach establishes a new state-of-the-art for dense geometric recovery. Notably, Drive3R currently stands as the strongest performing algorithm on the challenging nuScenes dataset (Caesar et al., 2020). It significantly surpasses both the previously dominant offline baseline (DVGT) (Zuo et al., 2025) and its per-frame streaming counterpart (DVGT-2) (Zuo et al., 2026), improving AbsRel from 0.1397/0.1806 to 0.1081, point-cloud Accuracy from 0.6158/0.6750 to 0.5344, and ATE from 1.1012/1.5812 to 0.9092.

In summary, our main contributions are as follows: ♠ **A novel block-wise streaming paradigm.** We introduce Drive3R, a geometric foundation model that resolves the fundamental conflict between sequence length and geometric consistency. By leveraging block-wise causal attention, it establishes robust local contexts bidirectionally while continuously propagating historical features via cross-block KV cache reuse, guaranteeing a highly scalable inference complexity. ♠ **Dynamic re-anchoring.** To combat scale drift and the rigidity of fixed-length chunking, we propose an adaptive re-anchoring mechanism. This module predicts temporal drift online and proactively re-initializes the KV cache upon detecting instability, seamlessly stitching new segments to ensure discontinuity-free long-range trajectory recovery. ♠ **Unprecedented state-of-the-art performance.** Drive3R currently stands as the strongest performing algorithm on multiple challenging autonomous driving datasets, including nuScenes (Caesar et al., 2020), KITTI (Geiger et al., 2013), and Waymo (Sun et al., 2019). It comprehensively surpasses previously dominant offline, per-frame streaming and chunk-then-align baselines across all core geometric metrics, including ego-pose estimation, dense depth, and point cloud reconstruction.

2 Related Work

Feed-Forward 3D Reconstruction Foundation Models. While traditional optimization-based methods achieve high-fidelity geometry recovery (Schonberger & Frahm, 2016), their computationally prohibitive iterative nature makes them difficult to scale for large-scale applications. To circumvent these limitations, a class of feed-forward 3D reconstruction Transformers has rapidly emerged, pioneered by DUSt3R (Wang et al., 2024). In a single forward pass, they directly regress pixel-aligned pointmaps, depths, and camera poses from images, effectively bypassing the arduous iterative optimization process. Subsequent works have extended this from pair-based to joint multi-view inference (Tang et al., 2025; Zhang et al., 2025). Recently, VGGT (Wang et al., 2025a; Shen et al., 2025) and π^3 (Wang et al., 2025d) further scaled this paradigm to large-scale Transformer backbones, capable of reconstructing scenes comprising dozens of frames using a single global bidirectional attention, establishing themselves as the state-of-the-art general geometric foundation models. In autonomous driving scenarios, DVGT (Zuo et al., 2025) adapts this paradigm to surround-view multi-camera inputs, directly outputting metric-scale dense geometry and ego-vehicle poses, which can serve directly as a geometric foundation for downstream scene reconstruction. However, these models architecturally rely on global bidirectional attention, requiring the entire sequence to be processed in a single forward pass. The $\mathcal{O}(N^2)$ complexity of the attention mechanism (Shen et al., 2025) makes them inherently incompatible with the continuous, long-sequence inference demanded by autonomous driving. This is precisely the bottleneck our paper aims to overcome while preserving the benefits of bidirectional reasoning.

Long-Sequence Geometric Foundation Models. To adapt feed-forward geometric reconstruction to long-duration scenarios, recent works mainly follow two trajectories. The first introduces a strict causal structure for per-frame streaming prediction. Specifically, CUT3R (Wang et al., 2025b) maintains an implicit scene state at each step, StreamVGGT (Zhuo et al., 2025) transforms VGGT into a KV-cached causal Transformer, and DVGT-2 (Zuo et al., 2026) adopts a similar sliding-window causal structure. While incremental inference avoids repeated full-sequence reconstruction, these

models completely sacrifice bidirectional context. By relying solely on historical priors without future lookahead, minute per-frame errors accumulate monotonically, inevitably leading to scale drift and trajectory deviation. The second trajectory adopts a "chunk-then-align" strategy, as seen in VGGT-Long (Deng et al., 2025) and DA3-Stream (Lin et al., 2025). These pipelines partition sequences into overlapping fixed-length chunks, perform independent offline bidirectional reconstruction on each, and subsequently align them via rigid $\mathrm{Sim}(3)$ transformations. Although this preserves local bidirectional attention, the predefined chunking often severs continuous motions in highly dynamic environments, and the absence of coherent feature propagation across chunks forces a reliance on post-processing, frequently causing trajectory discontinuities at boundaries. In contrast, Drive3R resolves these conflicting paradigms. By leveraging block-wise causal attention, it achieves highly scalable streaming without sacrificing local bidirectional reasoning. Coupled with a confidence-aware dynamic re-anchoring mechanism to correct accumulated drift online, Drive3R ensures robust, discontinuity-free long-range trajectory recovery.

3 METHOD

3.1 PROBLEM FORMULATION AND OVERVIEW

Given a long-horizon surround-view image stream in autonomous driving, we denote the input as

$$\mathcal{I}_{1:T} = \{I_{t,v} \mid t = 1, \ldots, T,\ v = 1, \ldots, V\}, \tag{1}$$

where T is the sequence length and V is the number of surround-view cameras. For each image $I_{t,v}$, the model is provided with calibrated camera intrinsics $K_{t,v}$ and the camera-to-ego extrinsics $T_{t,v}^{\text{ego}}$. Our goal is to online predict the ego pose trajectory $\{E_t\}_{t=1}^{T}$ and the dense 3D point clouds $\{P_{t,v}\}_{t=1,v=1}^{T,V}$ aligned with the multi-camera observations.

Compared with generic multi-view reconstruction, geometric modeling for autonomous driving has more specific task constraints and stronger available priors. First, the input usually comes from continuous driving logs, whose sequence length is far beyond the short clips commonly considered in generic reconstruction. Therefore, the model must scale to long-horizon streaming inference. Second, downstream simulation rely on geometry in metric space, requiring the model to preserve absolute scale consistency rather than only recover relative structure. Third, autonomous driving platforms provide stable and calibrated camera intrinsics and extrinsics. Unlike generic scenes where camera parameters are unknown or need to be estimated, these calibration parameters can be explicitly injected into the model as reliable geometric priors, helping the network reason about spatial relationships among different camera observations.

3.2 OVERALL ARCHITECTURE

Intrinsic-aware image encoding. As illustrated in Fig. 3, we build on the π^3x encoder-decoder architecture (Wang et al., 2025d) for long-horizon surround-view image streams. For each image $I_{t,v}$, a vision Transformer encoder extracts patch-level visual tokens. We further inject camera-intrinsic information by converting $K_{t,v}$ into normalized camera-ray embeddings and adding them to the visual tokens. This provides explicit projection-geometry priors for dense 3D prediction.

Spatio-temporal decoder. The intrinsic-aware tokens are then fed into a spatio-temporal Transformer decoder. The decoder adopts an alternating attention structure: intra-view spatial attention first models local context within each camera image, while inter-block spatio-temporal attention propagates information across all camera views and the temporally available blocks, jointly modeling multi-camera geometric consistency and temporal continuity. Unlike generic reconstruction models that apply fully bidirectional global attention to short input sequences, our decoder is designed for long driving logs and supports block-wise streaming inference. It preserves sufficient multi-frame and multi-camera interactions within each local temporal block, while propagating historical information along long sequences through streaming states. The block-wise streaming mechanism is introduced in Sec. 3.3.

Task-specific prediction branches. The shared features produced by the spatio-temporal decoder are fed into several lightweight task-specific prediction branches, each adapting the shared representation to a particular geometric target. The dense point branch predicts per-view 3D point clouds

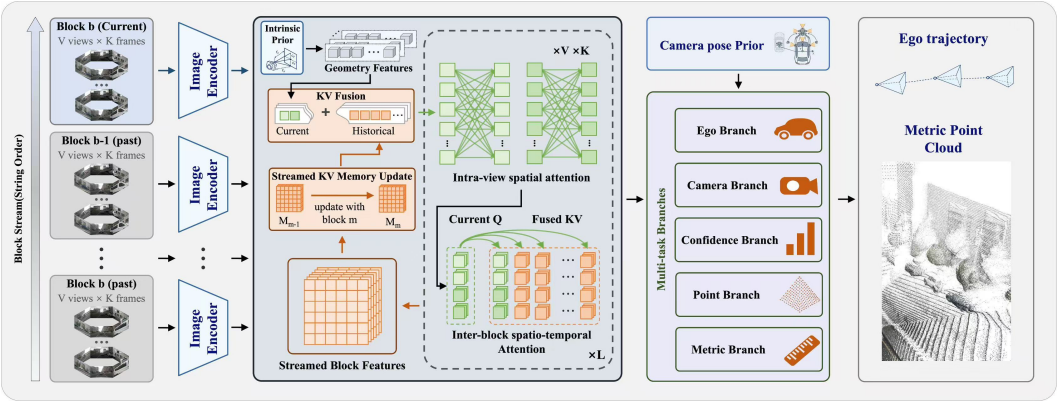

Figure 3: **The pipeline of our method.** Multi-view image streams are processed in a block-wise manner. Each image block is first encoded with intrinsic priors and then processed by a spatio-temporal decoder. By maintaining a streamed KV memory across consecutive blocks, the model continuously extracts consistent geometry features for multi-task predictions, including ego-centric motion and dense metric 3D reconstruction.

$\{P_{t,v}\}_{t=1,v=1}^{T,V}$, while a metric scale branch estimates a global scale factor that maps normalized geometry to real-world metric space. An auxiliary camera pose branch provides camera-pose supervision during training to improve geometric consistency. In addition, we introduce an ego-centric pose branch to estimate the vehicle-level trajectory $\{E_t\}_{t=1}^{T}$. This branch uses the camera-to-ego extrinsics $T_{t,v}^{\text{ego}}$ as a rig-geometry prior during multi-camera feature aggregation, enabling observations from different cameras to be fused into a unified ego-centric motion representation. Finally, a block-level confidence branch predicts the reliability of the current streaming state and provides the trigger signal for confidence-aware re-anchoring.

3.3 Block-wise Streaming Transformer

Block-wise sequence partition. To extend a globally bidirectional geometric foundation model to long-horizon autonomous driving sequences, we partition the input stream into consecutive temporal blocks. Let each block contain K frames. The b-th block is defined as

$$\mathcal{B}_b = \{(t,v) \mid t = (b-1)K+1, \ldots, \min(bK, T),\ v = 1, \ldots, V\}, \quad b = 1, \ldots, \left\lceil \frac{T}{K} \right\rceil. \tag{2}$$

This partition preserves the ordered temporal structure and the regular surround-view camera layout of driving data. Unlike frame-wise causal modeling (Zhuo et al., 2025), each block contains multiple consecutive frames and all camera views, allowing the model to exploit local multi-frame and multi-camera context for geometric reasoning.

Block-wise causal attention. During training, we process a sequence containing multiple temporal blocks in a single forward pass. To make this offline training procedure consistent with streaming inference, we introduce a block-wise causal mask in each inter-block spatio-temporal attention layer. For a query token q and a key token k, let $b(q)$ and $b(k)$ denote their corresponding temporal block indices. The attention mask is defined as

$$M(q,k) = \begin{cases} 0, & \text{if } b(k) \le b(q), \\ -\infty, & \text{if } b(k) > b(q). \end{cases} \tag{3}$$

As a result, all tokens within the same block are mutually visible, including tokens from different timestamps and camera views, while tokens from future blocks are masked out. This training mask preserves bidirectional context within each block and enforces strict causality across blocks. Therefore, training uses the same information-access pattern as KV-cache streaming inference.

Streaming inference with KV cache. During inference, the model processes the sequence block by block and maintains historical key/value caches in the inter-block spatio-temporal attention layers. For the b-th block, the current queries attend to both the key/value pairs from the current block and

the cached key/value pairs from previous blocks. After processing the block, its key/value pairs are appended to the cache for future inference. Formally, for the ℓ-th global attention layer, we have

$$K_{\leq b}^{(\ell)} = \mathrm{Concat}\left(K_{<b}^{(\ell)}, K_b^{(\ell)}\right), \quad V_{\leq b}^{(\ell)} = \mathrm{Concat}\left(V_{<b}^{(\ell)}, V_b^{(\ell)}\right). \tag{4}$$

By reusing historical features through the cache, the model avoids recomputing the entire history whenever a new block arrives. This enables efficient incremental streaming inference, while the model retains bidirectional reasoning within each block and causal streaming across blocks.

3.4 Confidence-Aware Re-Anchoring

Motivation. The block-wise streaming Transformer preserves bidirectional context within each local temporal block and propagates historical information over long sequences through KV cache. However, in long-horizon autonomous driving inference, cache propagation alone cannot completely prevent error accumulation. As the vehicle keeps moving and the scene content continuously changes, the implicit geometric state stored in the historical cache may gradually deviate from the current observations, eventually leading to ego-trajectory drift and point-cloud misalignment. To address this issue, we introduce a confidence-aware re-anchoring mechanism, enabling the model to assess streaming reliability online and reset and realign when potential drift occurs.

Block-level confidence prediction. To estimate the streaming reliability of each block, we introduce a block-level confidence branch on top of the shared spatio-temporal features. This branch first uses a lightweight decoder to extract geometry-state features related to the current block, then aggregates them over the spatial and camera dimensions, and outputs one confidence score c_b for each temporal block. A larger c_b indicates that the current block remains consistent with the existing streaming state, while a smaller c_b suggests that the current KV cache may have drifted. Unlike pixel-level confidence for dense geometry, the block-level confidence does not measure the reconstruction quality of individual points. Instead, it serves as a control signal for deciding whether the long-horizon streaming state is still reliable.

Adaptive cache reset. During inference, the model processes the input sequence block by block and reads the confidence score after each block. If $c_b \geq \tau$, where τ is a predefined threshold, the model continues to reuse the current KV cache and appends the current block output to the global result. If $c_b < \tau$, the model regards the current streaming state as potentially unreliable, discards the existing KV cache, and reinitializes a new local segment from overlapping frames near the current block. Unlike fixed-length chunk-then-align methods (Deng et al., 2025; Lin et al., 2025), the restart position is adaptively determined by the model-predicted confidence, allowing the inference process to adjust according to scene difficulty and drift risk.

SE(3) re-anchoring. To stitch the reinitialized local segment back to the existing global trajectory, we perform SE(3) alignment using the predicted poses on overlapping frames. Let Ω_b denote the set of overlapping frames between the old and new segments. For each overlapping frame i, we denote the predicted pose from the old trajectory as $(R_i^{\mathrm{old}}, t_i^{\mathrm{old}})$ and that from the new segment as $(R_i^{\mathrm{new}}, t_i^{\mathrm{new}})$. We estimate a rigid transform $T_b^\star = (R_b^\star, t_b^\star) \in \mathrm{SE}(3)$ by

$$R_b^\star = \Pi_{\mathrm{SO}(3)}\left(\frac{1}{|\Omega_b|}\sum_{i\in\Omega_b} R_i^{\mathrm{old}}\left(R_i^{\mathrm{new}}\right)^\top\right), \tag{5}$$

$$t_b^\star = \frac{1}{|\Omega_b|}\sum_{i\in\Omega_b}\left(t_i^{\mathrm{old}} - R_b^\star t_i^{\mathrm{new}}\right), \tag{6}$$

where $\Pi_{\mathrm{SO}(3)}(\cdot)$ projects a matrix to the closest valid rotation. The resulting transform is applied to the new segment's poses and point clouds, re-anchoring them into the global coordinate frame.

4 Experiments

We evaluate the proposed block-wise streaming framework on Waymo (Sun et al., 2019), nuScenes (Caesar et al., 2020), and KITTI (Geiger et al., 2013). Waymo (Sun et al., 2019) and nuScenes (Caesar et al., 2020) are used for dense geometry and ego-motion evaluation, while KITTI (Geiger et al., 2013) is used for long-sequence ego-motion estimation. We report depth, point-cloud, and

Table 1: Quantitative comparison on Waymo (Sun et al., 2019) under the 15-frame, full-view setting. Best results are **bolded**, and second best are underlined. * indicates models predicting relative scale that are aligned to recover the metric scale.

Model	Depth		Point Cloud		Ego-Motion	
	AbsRel ↓	$\delta < 1.25$ ↑	Acc. ↓	Comp. ↓	AUC@30 ↑	ATE ↓
Offline						
VGGT* (Wang et al., 2025a)	0.2386	0.7657	0.9033	1.4851	86.78	**3.2108**
π^3x (Wang et al., 2025d)	0.5005	0.3273	0.7513	1.5304	77.04	6.9732
DA3-Nested (Lin et al., 2025)	0.3970	0.3629	1.0020	2.0973	79.20	3.3885
DVGT (Zuo et al., 2025)	0.4698	0.4183	0.9653	1.6218	85.58	3.6566
Per-frame Streaming						
StreamVGGT* (Zhuo et al., 2025)	0.3253	0.5616	1.0984	0.9806	82.72	4.9199
LongStream (Cheng et al., 2026)	0.2065	0.7849	0.9836	4.7566	40.88	9.7395
DVGT-2 (Zuo et al., 2026)	0.4356	0.4774	1.4894	0.9539	84.73	3.8579
Block-wise Streaming						
Ours	**0.1947**	**0.8429**	**0.5941**	**0.5894**	**86.84**	3.9387

Table 2: Quantitative comparison on nuScenes (Caesar et al., 2020) with 15 frames and all views.

Model	Depth		Point Cloud		Ego-Motion	
	AbsRel ↓	$\delta < 1.25$ ↑	Acc. ↓	Comp. ↓	AUC@30 ↑	ATE ↓
Offline						
VGGT* (Wang et al., 2025a)	0.1967	0.6835	0.8315	1.5516	85.47	0.9151
π^3x (Wang et al., 2025d)	0.2651	0.6672	0.9515	0.7946	78.88	3.1063
DA3-Nested (Lin et al., 2025)	0.2261	0.6778	0.8521	0.9007	86.86	1.1709
DVGT (Zuo et al., 2025)	0.1397	0.8211	0.6158	0.6013	**88.64**	1.1012
Per-frame Streaming						
StreamVGGT* (Zhuo et al., 2025)	0.2623	0.4928	1.1090	2.4368	80.98	1.5748
LongStream (Cheng et al., 2026)	0.3026	0.4517	1.1155	0.9305	39.29	1.8030
DVGT-2 (Zuo et al., 2026)	0.1806	0.7285	0.6750	0.6230	85.08	1.5812
Block-wise Streaming						
Ours	**0.1081**	**0.9077**	**0.5344**	**0.5374**	86.92	**0.9092**

trajectory metrics, including AbsRel, $\delta < 1.25$, Accuracy, Completeness, ATE, and AUC@30. We further conduct ablation and block-wise analyses to validate our design choices. Unless otherwise specified, all methods use the same input protocol and metric computation pipeline.

4.1 Dense Geometry and Ego-Motion Estimation

We first evaluate the proposed method on Waymo (Sun et al., 2019) and nuScenes (Caesar et al., 2020) under the 15-frame, full-view setting. By utilizing all available camera views, this setting jointly measures depth estimation, point-cloud reconstruction, and ego-motion estimation.

Quantitative Evaluation. As shown in Tabs. 1 and 2, our method obtains the best depth and point-cloud reconstruction results on Waymo (Sun et al., 2019), with an AbsRel of 0.1947, $\delta < 1.25$ of 0.8429, Accuracy of 0.5941, and Completeness of 0.5894, while also achieving the highest AUC@30. On nuScenes (Caesar et al., 2020), it achieves the best results in five out of six metrics, including AbsRel, $\delta < 1.25$, Accuracy, Completeness, and ATE, showing consistent gains in both dense reconstruction and trajectory estimation.

Qualitative Evaluation Fig. 4 illustrates the geometric degradation in per-frame streaming baselines when bidirectional context is entirely discarded. Driven by the continuous accumulation of historical errors under a strict causal structure, StreamVGGT (Zhuo et al., 2025) produces noisy and incomplete geometry. Similarly, while DVGT-2 (Zuo et al., 2026) preserves the coarse scene layout, it still blurs object boundaries and fine-grained structures. In contrast, by establishing bidirectional local contexts

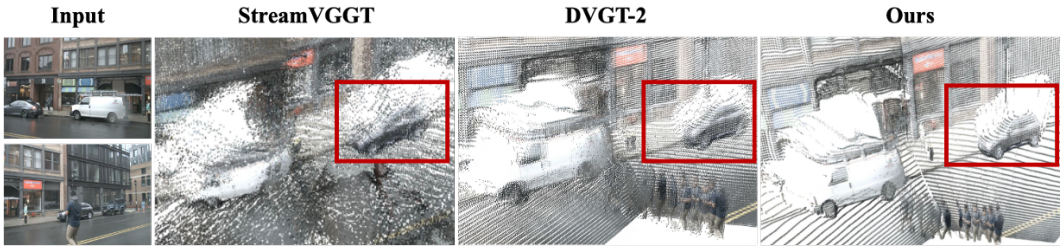

Figure 4: Qualitative comparison with per-frame streaming baselines. Red boxes highlight representative local regions where our method recovers cleaner geometry and sharper object structures.

Table 3: Comparison on KITTI (Geiger et al., 2013) Stereo 2Hz ($V = 2$). We report $\mathrm{Sim}(3)$-aligned ATE. †Average over completed sequences; missing StreamVGGT entries indicate OOM.

Methods	**KITTI Stereo 2Hz** (ATE↓)											**Avg.↓**
	00 $T = 909$ 3.7km	01 $T = 221$ 2.5km	02 $T = 933$ 5.1km	03 $T = 161$ 0.6km	04 $T = 55$ 0.4km	05 $T = 553$ 2.2km	06 $T = 221$ 1.2km	07 $T = 221$ 0.7km	08 $T = 815$ 3.2km	09 $T = 319$ 1.7km	10 $T = 241$ 0.9km	
Per-frame Streaming												
CUT3R	181.0	692.4	271.6	51.3	16.2	147.5	133.3	84.8	256.1	155.6	63.0	186.6
TTT3R	181.1	68.6	243.2	16.2	6.3	138.8	127.5	64.9	126.3	110.4	43.3	102.4
DVGT-2	148.4	<u>29.6</u>	252.9	15.6	2.8	59.2	**12.1**	**5.2**	181.2	105.5	18.2	75.5
StreamVGGT	OOM	680.6	OOM	154.2	79.8	OOM	123.4	79.3	OOM	208.0	157.2	211.8†
Chunk-then-Align												
DA3-Streaming	125.2	113.2	215.8	6.2	**0.8**	124.7	50.5	35.1	100.2	127.1	46.1	85.9
VGGT-Long	181.6	50.1	189.9	<u>4.9</u>	<u>1.7</u>	<u>43.3</u>	33.3	21.8	96.8	120.9	21.0	69.6
π-Long	**71.7**	167.1	**129.2**	7.5	2.4	**33.2**	<u>19.6</u>	<u>7.3</u>	**61.4**	<u>27.7</u>	<u>15.0</u>	<u>49.3</u>
Block-wise Streaming												
Ours	<u>89.8</u>	**25.5**	<u>135.8</u>	**4.6**	2.5	52.4	44.2	7.8	<u>65.6</u>	**17.5**	**12.7**	**41.7**

Baseline references: CUT3R (Wang et al., 2025b); TTT3R (Chen et al., 2025); DVGT-2 (Zuo et al., 2026); StreamVGGT (Zhuo et al., 2025); DA3-Streaming (Lin et al., 2025); VGGT-Long (Deng et al., 2025); π-Long (Wang et al., 2025d).

to halt temporal error amplification, our block-wise streaming method reconstructs cleaner road surfaces, sharper vehicle shapes, and more consistent foreground structures, demonstrating superior geometric coherence.

4.2 Long-sequence Ego-Motion Estimation

We next evaluate long-sequence ego-motion estimation on KITTI (Geiger et al., 2013) using complete driving sequences. This setting stresses offline global-attention models, whose memory cost prevents single-pass inference over complete driving logs. For KITTI (Geiger et al., 2013) stereo evaluation, we disable loop closure for methods whose loop module assumes a single-view temporal stream, since stereo inputs interleave synchronized left/right images rather than forming a purely monocular sequence. This setting therefore evaluates the long-range stability of each method under a consistent feed-forward odometry protocol. Tab. 3 reports the ATE on all 11 KITTI (Geiger et al., 2013) sequences. Our method achieves the best average ATE of 41.7, reducing the error by 44.8% compared with DVGT-2 (Zuo et al., 2026). The improvement suggests that block-wise streaming better preserves long-range temporal consistency than per-frame streaming and chunk-then-align baselines. This result also highlights the benefit of confidence-aware re-anchoring. Unlike StreamVGGT (Zhuo et al., 2025), whose cached history grows monotonically and can lead to OOM on long KITTI sequences, Drive3R detects unreliable streaming states and reinitializes the cache before stale history dominates the prediction. The refreshed segment is then stitched back through overlap alignment, yielding more stable long-range ego-motion estimation. The qualitative results in Fig. 5 further show that our estimated trajectories are more consistent with the ground truth on sequences 09 and 10.

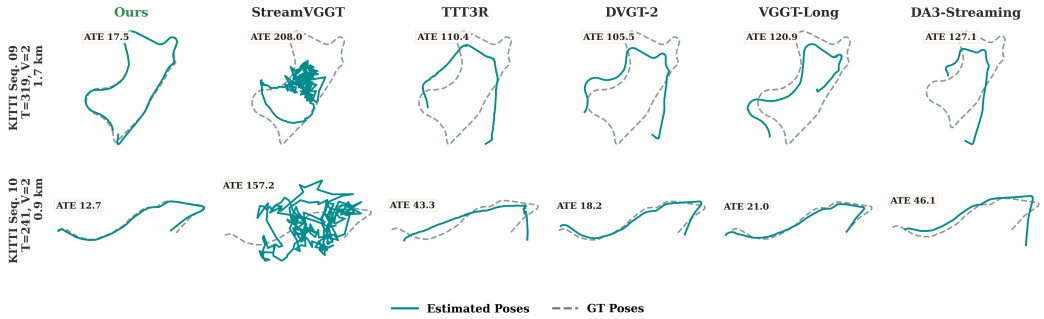

Figure 5: Qualitative trajectory comparison on KITTI (Geiger et al., 2013) sequences 09 and 10. Dashed gray and solid teal curves show ground-truth and estimated trajectories, respectively. Drive3R achieves lower ATE.

Table 4: Ablation study on confidence-aware re-anchoring on nuScenes (Caesar et al., 2020).

Setting	Depth		Point Cloud		Ego-Motion	
	AbsRel ↓	$\delta < 1.25$ ↑	Acc. ↓	Comp. ↓	AUC@30 ↑	ATE ↓
Ours w/o Conf.	0.1730	0.8274	0.8320	0.7918	70.66	18.1769
Ours w/ Conf.	**0.1063**	**0.9039**	**0.7129**	**0.6137**	**88.16**	**4.1422**

4.3 Experimental Analysis

Ablation on confidence-aware re-anchoring. We first ablate the confidence-aware re-anchoring strategy on 36-frame, 6-view nuScenes (Caesar et al., 2020) sequences. The efficacy of confidence-aware re-anchoring is demonstrated in Tab. 4. Removing this strategy triggers a significant performance drop, with ATE escalating from 4.1422 to 18.1769. This gap highlights the necessity of dynamic anchor selection. By identifying and re-anchoring to reliable temporal states, Drive3R effectively suppresses scale drift and trajectory deviation in conventional streaming architectures.

Analysis on block length. We further analyze the influence of block length in Fig. 6. Here, CD denotes the sum of Accuracy and Completeness errors. Increasing the block size from 1 to 3 consistently improves Accuracy, Completeness, and CD, indicating that moderate temporal context benefits local geometry. Further increasing the block length improves Accuracy but hurts Completeness, suggesting a trade-off between local precision and scene coverage. Considering this trade-off, we choose a block size of 3 for our system. The OOM result of the offline setting further supports the necessity of bounded block-wise processing.

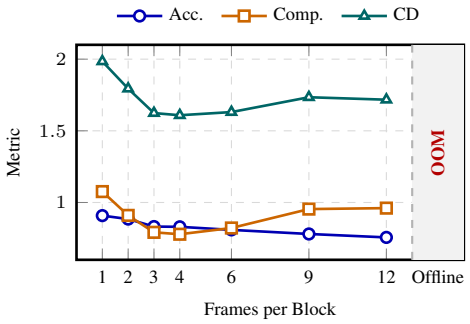

Figure 6: Block-length analysis on nuScenes (Caesar et al., 2020) with 36 input frames.

5 Conclusion

We present Drive3R, a block-wise streaming geometric foundation model for long-horizon autonomous driving reconstruction. Drive3R combines local multi-frame and multi-camera bidirectional reasoning with cross-block KV cache reuse and confidence-aware re-anchoring to mitigate long-range drift. Experiments show strong performance in dense depth, point-cloud reconstruction, and ego-motion estimation. Looking forward, Drive3R can serve as a foundation for online Gaussian scene representations and closed-loop autonomous driving simulation.

AI USE STATEMENT

Generative AI tools were used to assist with writing and language polishing, retrieval and discovery of related literature, and conversion of the manuscript to the ICLR 2027 LaTeX template, including reference formatting. The authors are responsible for the final content of the paper, including its text, citations, methodology, and scientific claims.

REFERENCES

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020. URL https://arxiv.org/abs/2001.10773.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025.

Chong Cheng, Xianda Chen, Tao Xie, Wei Yin, Weiqiang Ren, Qian Zhang, Xiaoyuang Guo, and Hao Wang. LongStream: Long-Sequence Streaming Autoregressive Visual Geometry. *arXiv preprint arXiv:2602.13172*, February 2026.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.11929.

A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, September 2013. doi: 10.1177/0278364913491297.

Mingzhe Guo, Zhipeng Zhang, Yuan He, Ke Wang, Liping Jing, and Haibin Ling. End-to-end autonomous driving without costly modularization and 3d manual annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17853–17862, 2023.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Kalin Kolev, Maria Klodt, Thomas Brox, and Daniel Cremers. Continuous global optimization in multiview 3D reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009. doi: 10.1007/s11263-009-0233-1.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2507.10496.

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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

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

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

Chen Shi, Shaoshuai Shi, Xiaoyang Lyu, Chunyang Liu, Kehua Sheng, Bo Zhang, and Li Jiang. Unisplat: Unified spatio-temporal fusion via 3d latent scaffolds for dynamic driving scene reconstruction. *arXiv preprint arXiv:2511.04595*, 2025.

Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. *arXiv preprint arXiv:1708.07120*, 2017. URL https://arxiv.org/abs/1708.07120.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo Open Dataset. *arXiv preprint arXiv:1912.04838*, 2019. URL https://arxiv.org/abs/1912.04838.

Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5283–5293, 2025.

Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *Advances in Neural Information Processing Systems*, 36:64318–64330, 2023.

Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14895–14904, 2024.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021. URL https://arxiv.org/abs/2103.17239.

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

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

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details. *arXiv preprint arXiv:2507.02546*, July 2025c.

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

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-Equivariant Visual Geometry Learning. *arXiv preprint arXiv:2507.13347*, 2025d.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, January 2023. doi: 10.48550/ARXIV.2301.00493.

Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pp. 156–173. Springer, 2024.

Haibao Yu, Kuntao Xiao, Jiahang Wang, Ruiyang Hao, Yuxin Huang, Guoran Hu, Haifang Qin, Bowen Jing, Yuntian Bo, and Ping Luo. Recondrive: Fast feed-forward 4d gaussian splatting for autonomous driving scene reconstruction. *arXiv preprint arXiv:2603.07552*, 2026.

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

Jiaru Zhang, Manav Gagvani, Can Cui, Juntong Peng, Ruqi Zhang, and Ziran Wang. Efficient and explainable end-to-end autonomous driving via masked vision-language-action diffusion. *arXiv preprint arXiv:2602.20577*, 2026a.

Wei Zhang, Yihang Wu, Songhua Li, Wenjie Ma, Xin Ma, Qiang Li, and Qi Wang. Review of feed-forward 3d reconstruction: From dust3r to vggt. *arXiv preprint arXiv:2507.08448*, 2025.

Yiwei Zhang, Xuesong Chen, Jin Gao, Hanshi Wang, Fudong Ge, Weiming Hu, Shaoshuai Shi, and Zhipeng Zhang. Onedrive: Unified multi-paradigm driving with vision-language-action models. *arXiv preprint arXiv:2604.17915*, 2026b.

Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21634–21643, 2024.

Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025.

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

Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Shengyin Jiang, Long Chen, Zhi-Xin Yang, and Jiwen Lu. Dvgt: Driving visual geometry transformer. *arXiv preprint arXiv:2512.16919*, 2025.

Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Hanbing Li, Long Chen, Zhi-Xin Yang, and Jiwen Lu. Dvgt-2: Vision-geometry-action model for autonomous driving at scale. *arXiv preprint arXiv:2604.00813*, 2026.

A Implementation Details

A.1 Architecture

Our implementation follows the π^3x (Wang et al., 2025d) architecture and initializes the model from pretrained π^3x (Wang et al., 2025d) weights. We use a ViT-L/14 image encoder (Dosovitskiy et al., 2021) with register tokens (Darcet et al., 2024), followed by a geometry Transformer composed of 36 attention blocks. Each block uses feature dimension 1024, 16 attention heads, MLP ratio 4, RoPE positional encoding (Su et al., 2021), QKNorm, and LayerScale (Touvron et al., 2021) initialized to 0.01. All prediction branches are implemented as lightweight task-specific decoders followed by compact output heads on top of the shared geometry features. They consume the concatenation of the penultimate and final geometry features and predict local point maps, pixel confidence, per-view camera poses, ego pose, metric scale, and block-level confidence. Dense branches use Transformer decoders followed by MoGe-style convolutional upsampling heads (Wang et al., 2025c), while pose, scale, and block-confidence branches use compact Transformer decoders followed by MLP heads.

A.2 Calibration Prior Injection

Intrinsic injection. For each image token, we inject the calibrated camera intrinsics as an explicit projection-geometry prior. Given a pixel location $u = (x, y)$, we first map it to normalized camera coordinates

$$r_{t,v}(u) = \Pi_{xy}\left(K_{t,v}^{-1}[x + 0.5, y + 0.5, 1]^\top\right), \tag{7}$$

where $\Pi_{xy}(\cdot)$ keeps the first two coordinates. The resulting two-channel ray field is embedded with the same patch stride as the image encoder and added to the corresponding visual tokens. This intrinsic-aware tokenization provides explicit projection-geometry priors for visual features, reducing the burden of implicitly learning camera imaging geometry when recovering dense 3D point clouds from surround-view observations.

Extrinsic injection. For the ego decoder, we use the calibrated camera-to-ego transformation $T_{t,v}^{\mathrm{ego}}$ as a rig-geometry prior for ego-centric motion estimation. The ego decoder alternates per-view self-attention with same-frame cross-view attention. After the first cross-view layer, we apply a pose-aware injection block to the patch tokens:

$$\tilde{F}_{t,v} = F_{t,v} + \mathrm{PoseInject}\left(F_{t,v}, \left(T_{t,v}^{\mathrm{ego}}\right)^{-1}\right), \tag{8}$$

where $F_{t,v}$ denotes the ego-decoder feature for camera v at time t. Inside $\mathrm{PoseInject}(\cdot)$, the inverse transformation is used as an ego-to-camera view matrix for projective positional encoding in attention (Li et al., 2025). This makes the ego decoder aware of the fixed surround-view rig layout, allowing observations from different cameras to be fused into a unified vehicle-coordinate motion representation.

B Training Details

B.1 Training Data

We train on a mixture of autonomous driving datasets, including Argoverse 2 (Wilson et al., 2023), Waymo (Sun et al., 2019), nuScenes (Caesar et al., 2020), and Virtual KITTI (Cabon et al., 2020). These datasets cover diverse real-world urban driving scenes as well as synthetic environments with dense supervision. We use a mixed sampling strategy in which Argoverse 2 (Wilson et al., 2023), Waymo (Sun et al., 2019), nuScenes (Caesar et al., 2020), and Virtual KITTI (Cabon et al., 2020) are sampled with weights $35 : 30 : 25 : 10$. After selecting a dataset, each batch is formed by randomly sampling synchronized multi-camera clips at 2Hz.

We use grouped multi-view batches so that samples in the same batch share the same number of frames and cameras. This avoids excessive padding and keeps the per-device memory usage predictable. Each GPU processes at most 60 input images. For input processing, we use dynamic multi-resolution training. The image resolution is sampled from a resolution pool whose aspect ratio lies in $[0.5, 2.0]$ and whose pixel count lies in $[100\mathrm{K}, 200\mathrm{K}]$. To improve robustness to real-world image degradation, we apply color jittering, JPEG degradation, and image blurring during training.

B.2 Dense Pseudo-Label Generation

Real-world driving datasets often provide accurate but sparse LiDAR measurements rather than dense metric depth and point maps. For these datasets, we generate dense pseudo supervision using MoGe-2 (Wang et al., 2025c) and align the predictions to sparse LiDAR depth. Specifically, we first run MoGe-2 (Wang et al., 2025c) on each camera image to obtain dense depth and point-map predictions. We then project LiDAR points into each camera view using the calibrated camera intrinsics and extrinsics, producing sparse metric depth samples at valid projected pixels.

To put the MoGe-2 (Wang et al., 2025c) prediction into the metric scale of the driving scene, we estimate an affine depth alignment between the predicted dense depth and the sparse LiDAR depth. Let $d^{\mathrm{M}}(u)$ denote the MoGe-2 (Wang et al., 2025c) depth at pixel u and $d^{\mathrm{L}}(u)$ denote the projected LiDAR depth. Over valid pixels where both depths are available, we fit scale and shift parameters (a, b) such that

$$a^\star, b^\star = \arg\min_{a,b} \sum_{u \in \Omega} \left\| a d^{\mathrm{M}}(u) + b - d^{\mathrm{L}}(u) \right\|_1, \tag{9}$$

where Ω is the set of valid projected LiDAR pixels. The aligned dense depth is then used together with the adjusted point map as dense pseudo ground truth.

We discard low-quality pseudo labels before training. The filtering criteria include insufficient projected LiDAR points, poor spatial coverage, low valid overlap with the MoGe-2 (Wang et al., 2025c) mask, abnormal affine scale or shift, and poor post-alignment depth accuracy on the sparse LiDAR samples. This procedure provides dense metric supervision for real-world driving data while reducing the impact of unreliable pseudo labels. We will release this dense pseudo-label generation pipeline together with the code to facilitate reuse on other autonomous driving datasets.

B.3 Training Losses

Following π^3x (Wang et al., 2025d), we adopt standard geometric supervision for dense reconstruction, metric scale prediction, and auxiliary camera pose estimation. The dense geometry loss $\mathcal{L}_{\mathrm{geo}}$ supervises predicted point maps and depths on valid pixels, while the metric scale loss $\mathcal{L}_{\mathrm{scale}}$ encourages the predicted geometry to recover metric-space scale. We also include an auxiliary camera pose loss $\mathcal{L}_{\mathrm{cam}}$ to improve the geometric consistency of the shared representation. In addition, the ego pose branch is supervised by $\mathcal{L}_{\mathrm{ego}}$. Similar to camera pose supervision, $\mathcal{L}_{\mathrm{ego}}$ combines translation error and geodesic rotation error for vehicle-level trajectory estimation.

For the block-level confidence branch, we derive the confidence target from trajectory drift errors. For the ego track, we take the first frame of the sequence segment as the reference frame and compute both predicted and ground-truth poses relative to this frame. We then measure their discrepancy using a weighted combination of translation error and rotation error, average it over frames within the b-th temporal block to obtain a block-level error e_b. The soft confidence target is defined as

$$y_b = \exp(-\alpha_{\mathrm{conf}} e_b), \tag{10}$$

where α_{conf} is a fixed temperature coefficient, set to 25 in our experiments. Thus, a block with small relative trajectory error receives a confidence target close to 1, whereas a block with larger drift receives a lower confidence target. Given the predicted confidence logit s_b, we use binary cross-entropy for supervision:

$$\mathcal{L}_{\mathrm{conf}} = \mathrm{BCEWithLogits}(s_b, y_b). \tag{11}$$

B.4 Optimization Schedule

Training follows a two-stage schedule. In the first stage, we train the geometric reconstruction model for 8000 iterations using 64 H20 GPUs. This stage optimizes the dense geometry, camera pose, metric scale, and ego-motion objectives described in Appendix B.3. In the second stage, we freeze the pretrained geometric backbone and train only the block-level confidence branch using 16 H20 GPUs. This fine-tuning stage is substantially lighter than the first stage and does not alter the learned geometry representation.

In both stages, the temporal block size K is randomly sampled from 2 to 6 during training. This exposes the model to different streaming granularities and prevents the confidence branch from

Table 5: Depth and point-cloud reconstruction after training from scratch on nuScenes (Caesar et al., 2020). Best results are bold; second best are underlined. †: CUT3R (Wang et al., 2025b) depth is scale-aligned; Drive3R and DVGT-2 (Zuo et al., 2026) predict metric depth.

Model	AbsRel ↓	$\delta < 1.25$ ↑	Acc. ↓	Comp. ↓
Per-frame Streaming				
DVGT-2 (Zuo et al., 2026)	0.8746	<u>0.1541</u>	<u>1.6782</u>	<u>5.2002</u>
CUT3R† (Wang et al., 2025b)	<u>0.6444</u>	0.1284	2.1214	11.4841
Block-wise Streaming				
Ours	**0.2873**	**0.5661**	**1.2006**	**1.9167**

Table 6: Trajectory, depth, and point-cloud results on 300-frame closed-course driving sequences with a new camera rig. Mean results are reported; best values are bold; second best are underlined. ATE uses $\mathrm{Sim}(3)$ alignment.

Model	$\mathrm{ATE}_{\mathrm{Sim}(3)}$ ↓	AbsRel ↓	$\delta < 1.25$ ↑	PC F1@0.5 ↑	Chamfer ↓
Per-frame Streaming					
CUT3R (Wang et al., 2025b)	9.833	0.374	0.386	0.304	7.667
TTT3R (Chen et al., 2025)	<u>3.172</u>	0.292	0.542	0.457	5.716
StreamVGGT (Zhuo et al., 2025)	33.057	85.422	0.002	0.181	30.336
DVGT-2 (Zuo et al., 2026)	17.880	24.074	0.021	0.097	18.496
Chunk-then-Align					
DA3-Streaming (Lin et al., 2025)	3.189	**0.228**	<u>0.733</u>	<u>0.637</u>	<u>2.013</u>
VGGT-Long (Deng et al., 2025)	5.229	0.405	0.394	0.520	3.529
π-Long (Wang et al., 2025d)	4.108	0.396	0.532	0.441	12.464
Block-wise Streaming					
Ours	**2.677**	<u>0.229</u>	**0.736**	**0.656**	**1.987**

overfitting to a fixed chunk length. We optimize the model with AdamW (Loshchilov & Hutter, 2019). The base learning rate is 2.5×10^{-5}, and the encoder learning rate is set to 2.5×10^{-6}. We use a cosine-annealed OneCycle learning-rate schedule (Smith & Topin, 2017), bfloat16 mixed precision, and gradient norm clipping with threshold 1.0.

C Additional Experiments

C.1 Training from Scratch

We investigate whether Drive3R's reconstruction advantage persists when different streaming models learn directly from the same driving data. We compare block-wise streaming (Drive3R), sliding-window KV caching (DVGT-2 (Zuo et al., 2026)), and implicit state propagation (CUT3R (Wang et al., 2025b)). Each model is trained from scratch for $10{,}000$ iterations using only the nuScenes (Caesar et al., 2020) training split and evaluated on the held-out validation split.

Drive3R achieves the best performance across all four metrics under the same from-scratch training protocol (Tab. 5). In particular, it reduces depth AbsRel and point-cloud completeness error by approximately 67% and 63%, respectively, compared with DVGT-2 (Zuo et al., 2026), demonstrating substantial gains in both depth estimation and point-cloud reconstruction.

C.2 Long-Sequence Reconstruction across Camera Rigs

We examine whether Drive3R can sustain accurate trajectory estimation and scene reconstruction over longer sequences captured with a new camera configuration. For this evaluation, we use 300-frame clips from an internal closed-course driving dataset with a camera rig distinct from those in the training data. Each sequence is processed with the intrinsics and extrinsics of the evaluation rig. We compare Drive3R with streaming and chunk-then-align methods on trajectory, depth, and point-cloud metrics to assess both motion estimation and geometric reconstruction in this setting.

Table 7: Trajectory, depth, and point-cloud performance under calibration perturbations. Clean denotes the original calibration; extrinsic perturbations list rotation and translation magnitudes.

Perturbation	ATE ↓	$\text{ATE}_{\text{Sim}(3)}$ ↓	Racc@5 ↑	Tacc@5 ↑	AbsRel ↓	PC F1@0.5 ↑
Clean	4.15	1.05	88.4	89.0	0.157	0.757
Intrinsics: $\pm 5\%$	4.24	1.22	88.6	85.2	0.160	0.761
Extrinsics: 0.5°, 1 cm	4.02	1.07	89.0	89.0	0.156	0.756
Extrinsics: 1°, 2 cm	4.17	1.05	88.4	89.0	0.157	0.754
Extrinsics: 2°, 5 cm	4.16	1.05	88.4	89.0	0.157	0.738

Table 8: Confidence and drift on 220 nuScenes (Caesar et al., 2020) validation blocks. Left: correlations, with bars denoting scene-wise means. Right: mean error by confidence group.

Statistic	Value
Pearson correlation (c_b, e_b)	-0.54
Spearman correlation (c_b, e_b)	-0.58
Pearson correlation (c_b, y_b)	$+0.64$
Scene-level Pearson correlation $(\bar{c}, \bar{e})$	-0.78

Confidence group	Mean e_b
Low tercile	0.47
Middle tercile	0.48
High tercile	0.10
$c_b \leq 0.9$	0.44
$c_b > 0.9$	0.07

Tab. 6 shows that Drive3R achieves the lowest trajectory error, with an ATE of 2.677, compared with 3.172 for TTT3R (Chen et al., 2025) and 3.189 for DA3-Streaming (Lin et al., 2025). It also produces the strongest point-cloud results, reaching an F1 score of 0.656 and a Chamfer distance of 1.987. For depth estimation, Drive3R obtains an AbsRel of 0.229 and the highest $\delta < 1.25$ accuracy of 0.736. These results demonstrate accurate trajectory estimation and scene reconstruction over long driving sequences with a new camera configuration.

C.3 Robustness to Calibration Perturbations

Drive3R incorporates camera calibration as a geometric prior, making robustness to calibration errors important for its use across driving platforms. We test whether trajectory and reconstruction quality remain stable when these inputs are perturbed at inference. The evaluation uses 10 scenes of 36 frames each, with intrinsic perturbations of $\pm 5\%$ and camera-to-ego extrinsic perturbations up to 2° in rotation and 5 cm in translation.

As shown in Tab. 7, Drive3R maintains stable overall performance under calibration perturbations. Despite small fluctuations in individual metrics, depth AbsRel remains within 0.156–0.160, aligned ATE within 1.05–1.22, and point-cloud F1 within 0.738–0.761. These results demonstrate robustness to the tested intrinsic and extrinsic calibration errors, preserving both trajectory estimation and reconstruction quality with imperfect calibration.

C.4 Block Confidence and Measured Drift

For confidence to guide re-anchoring, it should reflect accumulated trajectory drift and distinguish reliable streaming states from unreliable ones. We test this relationship on 20 nuScenes (Caesar et al., 2020) validation scenes with $T = 36$ and $V = 6$, yielding 220 evaluated blocks. Re-anchoring is disabled so that confidence and drift can evolve without intervention. We compare predicted confidence with the block-level tracking error e_b used in the confidence-training objective (Appendix B.3), and examine error across confidence groups.

Tab. 8 shows a negative association between predicted confidence and measured drift at both block and scene levels. The high-confidence tercile has a mean error of 0.10, compared with 0.47 and 0.48 for the low and middle terciles. Grouping at 0.9 similarly separates blocks with mean errors of 0.44 ($c_b \leq 0.9$) and 0.07 ($c_b > 0.9$). This separation shows that low confidence identifies blocks with greater drift, supporting its use as a signal for re-anchoring.

Table 9: Design choices for re-anchoring: (a) reset policy, (b) confidence threshold, and (c) overlap length. Trajectory errors use $\mathrm{Sim}(3)$ alignment. Best results within each panel are bold.

(a) Reset policy

Reset policy	$\mathrm{ATE}_{\mathrm{Sim}(3)} \downarrow$	AbsRel $\downarrow$	PC F1@0.5 $\uparrow$
No reset	11.57	0.144	0.54
Fixed interval (interval/4)	1.05	**0.066**	0.76
Confidence-based ($\tau = 0.9$)	**0.57**	**0.066**	**0.79**

(b) Confidence threshold

τ	$\mathrm{ATE}_{\mathrm{Sim}(3)} \downarrow$	Racc@5 $\uparrow$	Tacc@5 $\uparrow$
0.3	1.84	66.6	74.4
0.5	1.35	70.2	82.2
0.7	1.14	78.8	86.7
0.9	**0.57**	**96.7**	**96.0**

(c) Overlap length

Overlap	$\mathrm{ATE}_{\mathrm{Sim}(3)} \downarrow$	$\mathrm{RPE}_{t,\mathrm{Sim}(3)} \downarrow$	AbsRel $\downarrow$
1	**0.57**	0.41	**0.064**
2	**0.57**	**0.36**	0.066

C.5 Design Choices for Confidence-Aware Re-anchoring

Having established the association between confidence and drift in Appendix C.4, we evaluate how to use this signal for re-anchoring. We test whether confidence-based resets improve accuracy over a fixed schedule, how the confidence threshold affects this benefit, and whether segment alignment is sensitive to the overlap length. The experiments use 10 scenes with 36 frames each and measure trajectory and reconstruction accuracy.

Selecting reset locations. Tab. 9(a) compares confidence-based re-anchoring with a fixed schedule and no reset. Re-anchoring reduces aligned ATE from 11.57 without resets to 1.05 with the fixed schedule and 0.57 with confidence-based resets. Relative to the fixed schedule, confidence-based re-anchoring also improves point-cloud F1 from 0.76 to 0.79 while maintaining the same AbsRel of 0.066. These gains support using predicted reliability to select segment boundaries.

Setting the confidence threshold. The threshold τ controls the reliability required to continue reusing the current cache. Raising τ from 0.3 to 0.9 reduces aligned ATE from 1.84 to 0.57 and improves both rotation and translation accuracy (Tab. 9(b)). The consistent gains across these metrics favor a stricter reliability criterion, with $\tau = 0.9$ yielding the best accuracy in the sweep.

Choosing the overlap length. The two overlap settings yield similar results, indicating that overlap length has little effect on overall performance within the tested range (Tab. 9(c)). We use an overlap length of 1 by default.

C.6 Inference Latency and Peak Memory

We evaluate the computational cost of multi-view reconstruction by measuring inference latency and peak GPU memory under a shared nuScenes (Caesar et al., 2020) setting. All methods use $T = 36$ frames, $V = 6$ cameras, an image resolution of 336×518, and bf16 precision. Each model is benchmarked separately, with only one model running at a time.

Tab. 10 shows that Drive3R has the lowest peak memory among the evaluated methods, using 15.57 GB. Compared with the offline methods DVGT (Zuo et al., 2025) and VGGT (Wang et al., 2025a), which require 55.62 and 48.12 GB, respectively, Drive3R reduces peak memory by approximately 72% and 68% under this setting. Inference takes 21.10 s, about 2.2–$4.5\times$ faster than DA3 (Lin et al., 2025), VGGT (Wang et al., 2025a), and StreamVGGT (Zhuo et al., 2025).

DVGT (Zuo et al., 2025) and DVGT-2 (Zuo et al., 2026) achieve lower latency. Their speed benefits from factorized spatial-temporal attention, which separates cross-view spatial attention from temporal attention and reduces the number of tokens involved in each attention operation (Zuo et al., 2025; 2026). This factorization restricts direct joint interactions across views and time. The attention ablation in DVGT likewise reports faster inference at the cost of reconstruction accuracy (Zuo et al., 2025, Sec. 4.4, Tab. 6). In our experiments, adopting a similar factorization in Drive3R reduced

Table 10: Inference latency and peak GPU memory on nuScenes (Caesar et al., 2020). Best results are bold; second best are underlined.

Model	Latency (s) ↓	Peak memory (GB) ↓
Offline		
DVGT (Zuo et al., 2025)	**12.56**	55.62
DA3 (Lin et al., 2025)	46.67	30.46
VGGT (Wang et al., 2025a)	50.93	48.12
Per-frame Streaming		
DVGT-2 (Zuo et al., 2026)	<u>12.64</u>	<u>16.55</u>
StreamVGGT (Zhuo et al., 2025)	94.91	28.71
Block-wise Streaming		
Ours	21.10	**15.57**

reconstruction quality; we therefore retain joint cross-view and temporal interactions in the block-wise decoder.

Efficiency gains from re-anchoring. To assess the effect of cache resets on inference cost, we compare Drive3R with and without re-anchoring on 10 scenes with $T = 36$ and $V = 6$, using $\tau = 0.9$ when re-anchoring is enabled. Re-anchoring reduces inference latency from 28.6 to 21.1 s (approximately 26%) and peak memory from 24.2 to 15.9 GB, saving 8.3 GB. Clearing stale KV entries reduces the cost of retaining and attending to an expanding history. The $\mathrm{SE}(3)$ stitching step takes only 0.39 ms per reset, so this efficiency gain is achieved with little alignment overhead.

D Limitations

Recent geometric foundation models are trained under heterogeneous data regimes, with different data mixtures, annotations, and preprocessing pipelines. Retraining all compared methods on exactly the same large-scale dataset is challenging due to the substantial computational cost. We select representative baselines whose training data includes autonomous-driving scenarios and evaluate all methods under the same input protocol, datasets, and metric computation pipeline. We additionally conduct a smaller-scale controlled experiment in Appendix C.1, training Drive3R, DVGT-2 (Zuo et al., 2026), and CUT3R (Wang et al., 2025b) from scratch on the same nuScenes (Caesar et al., 2020) split for $10{,}000$ iterations. Extending this controlled comparison to a shared large-scale training dataset would further clarify the contributions of model design and training data.

E Existing Assets and Licenses

We use several existing datasets, pretrained models, and baseline implementations. All existing assets are cited in the main paper and are used according to their official licenses or terms of use. Specifically, we use Argoverse 2 (Wilson et al., 2023), Waymo (Sun et al., 2019), nuScenes (Caesar et al., 2020), KITTI (Geiger et al., 2013), and Virtual KITTI (Cabon et al., 2020) for training and evaluation, MoGe-2 (Wang et al., 2025c) for dense pseudo-label generation, and π^3x (Wang et al., 2025d) for model initialization. For datasets with dataset-specific access agreements or terms of use, we do not redistribute the raw data. Reproduction should obtain the data from the corresponding official sources and follow their terms. For existing code, models, and baselines, we follow the licenses and usage terms provided by their original authors.