DVGT-2: VISION-GEOMETRY-ACTION MODEL FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

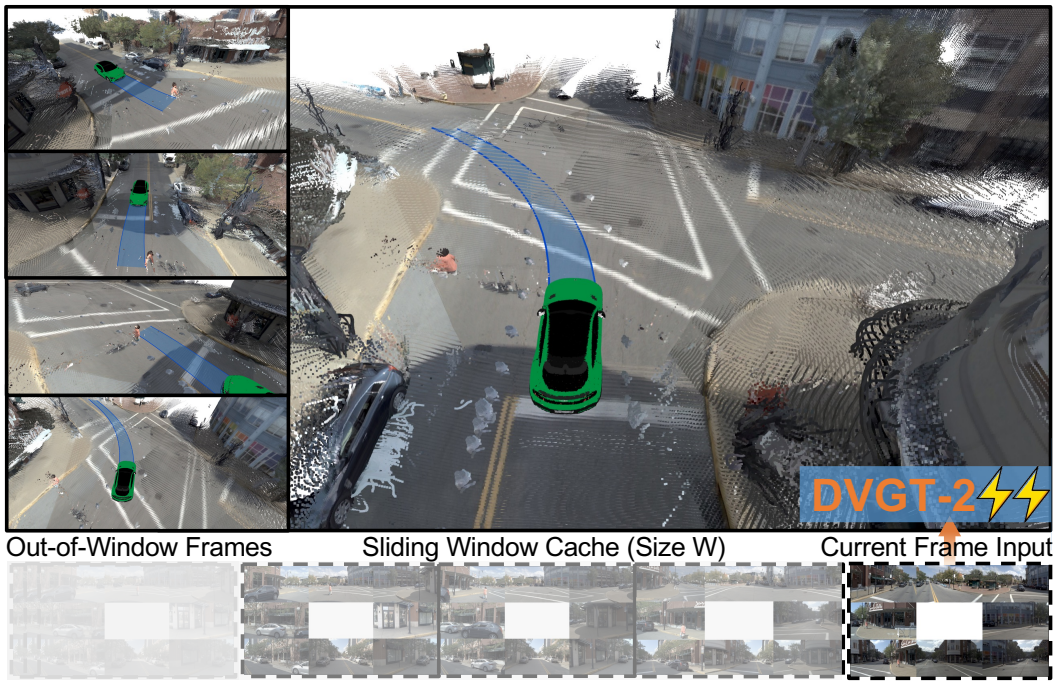

Figure 1: **DVGT-2 is a streaming visual geometry transformer specifically designed for autonomous driving**. It inputs multi-view images and jointly predicts the local 3D pointmaps, the relative ego-pose, and future ego-trajectory in an online manner.

ABSTRACT

Vision-Language-Action (VLA) and World-Action (WA) models offer promising routes to end-to-end autonomous driving. In this paper, we introduce an alternative Vision-Geometry-Action (VGA) paradigm and demonstrate that dense geometric supervision can support competitive driving planning. As vehicles operate in a 3D world, dense geometry provides rich spatial cues for decision-making. However, the computational overhead of multi-frame geometry reconstruction limits its use in online driving. To address this, we introduce a streaming Driving Visual Geometry Transformer (**DVGT-2**), which jointly predicts local geometry and future trajectories, aligning geometric reconstruction with ego-centric planning. We employ temporal causal attention and cache historical features to avoid repetitive computation. We further propose a sliding-window streaming strategy and use relative temporal encoding to support online inference with bounded per-frame computation and memory. Experiments demonstrate superior local geometric accuracy across five driving datasets and substantially improved online reconstruction efficiency. **DVGT-2** achieves strong planning performance on NAVSIM v1/v2 and remains competitive on nuScenes, while offering low inference latency compared with representative VLA and WA models.

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

1 INTRODUCTION

End-to-end autonomous driving has recently seen remarkable progress (Hu et al., 2023; Jiang et al., 2023; Liao et al., 2025). Vision-Language-Action (VLA) models leverage the semantic knowledge and reasoning capabilities of pretrained vision-language models to interpret driving scenarios and facilitate decision-making (Zhou et al., 2025b; Li et al., 2025d). Meanwhile, World-Action (WA) models integrate future scene modeling with action generation, learning environmental dynamics to support planning (Zhang et al., 2025; Shi et al., 2026). These methods highlight the importance of learning rich scene representations for trajectory planning.

As vehicles inherently operate in a 3D world, dense geometry provides rich spatial information for decision-making. This motivates us to explore an alternative Vision-Geometry-Action (VGA) paradigm for end-to-end autonomous driving. By jointly reconstructing 3D pointmaps and ego-poses and predicting future trajectories, VGA learns shared representations that capture fine-grained scene structure. Our goal is to investigate whether dense geometric supervision can provide an effective foundation for planning, with performance competitive with VLA and WA models. However, integrating high-fidelity geometry reconstruction into online planning presents significant computational challenges. Batch-processing geometry models (Wang et al., 2025a; Zuo et al., 2025) repeatedly process historical frames as new observations arrive, incurring redundant computation. Full-history streaming (Zhuo et al., 2025) avoids this recomputation but maintains a growing cache, increasing computational and memory costs during continuous driving.

To address these challenges, we introduce a streaming Driving Visual Geometry Transformer (**DVGT-2**) to align geometric reconstruction with ego-centric planning, as shown in Figure 1. It jointly predicts current-frame local geometry, relative ego-poses, and future trajectories. At each time step, an image encoder extracts tokens from the current multi-view images, and a geometry transformer relates them across views and to cached features from recent frames. Temporal causal attention reuses recent features without recomputing earlier frames, while a fixed-size sliding-window cache bounds per-frame computation and memory. Relative temporal encoding represents frame offsets within this window, supporting inference on sequences longer than those encountered during training. From the resulting representation, we employ three heads to predict local pointmaps in the current ego-coordinate system, the ego-pose relative to the preceding frame, and a future ego-trajectory. Predicting local geometry and relative motion removes the need to anchor reconstruction to the first frame, allowing older cached features to be discarded. Extensive experiments demonstrate superior local geometric accuracy across five driving datasets and substantially improved online reconstruction efficiency. On NAVSIM v1/v2 (Dauner et al., 2024; Cao et al., 2025), **DVGT-2** achieves the highest planning scores among the compared methods, while remaining competitive on nuScenes (Caesar et al., 2020). Under matched settings, **DVGT-2** achieves superior efficiency with low trajectory inference latency compared with representative VLA and WAM models (Zhou et al., 2025b; Li et al., 2025b; Xia et al., 2026; Shi et al., 2026), validating VGA as an effective and efficient alternative for autonomous driving.

2 RELATED WORK

End-to-end Autonomous Driving. End-to-end autonomous driving predicts ego-trajectories or controls from sensor observations, often learning scene representations jointly with planning (Chen et al., 2024; Liao et al., 2025; Zheng et al., 2024b). Planning-oriented frameworks connect perception, motion prediction, and planning in one model (Hu et al., 2023; Weng et al., 2024), while vectorized agents and maps provide explicit planning constraints (Jiang et al., 2023). Other efficient designs use sparse queries or 3D semantic Gaussians to represent the surrounding scene (Zhu et al., 2025; Zheng et al., 2024c). These methods show how both the scene representation and the formulation of the planning output shape end-to-end driving systems. Our model reconstructs fine-grained dense local geometry as the scene representation for trajectory prediction.

Vision-Language-Action and World-Action Models for Autonomous Driving. Vision-language models use language to support scene understanding, instruction following, and high-level driving decisions (Shao et al., 2024; Jiang et al., 2024). Building on these semantic representations, Vision-Language-Action (VLA) models connect visual-language features to trajectory prediction (Zhou et al., 2025b; Li et al., 2025d; Sheng et al., 2026). World-Action (WA) models learn predictive scene

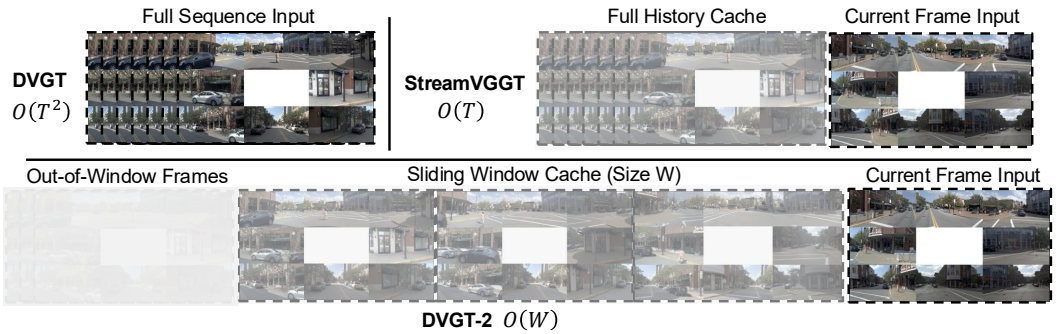

Figure 2: **Comparison of geometry reconstruction paradigms.** Batch-processing models compute pairwise relations across all frames, incurring an $\mathcal{O}(T^2)$ per-frame cost. Full-history streaming reuses cached features from all previous frames with an $\mathcal{O}(T)$ per-frame cost. Our sliding-window strategy maintains a fixed-size cache of W frames, achieving a constant $\mathcal{O}(W)$ per-frame cost.

dynamics and couple future-scene representations with planning (Zhang et al., 2025; Shi et al., 2026; Xia et al., 2026). Geometry-enhanced methods incorporate pretrained visual geometry features into VLA or WA systems (Wang et al., 2026; Yang et al., 2026). Alternatively, we use dense geometry reconstruction itself as the scene-learning objective for joint geometry prediction and planning.

Visual Geometry Models. 3D reconstruction from unposed images developed through pairwise matching models (Wang et al., 2024; Leroy et al., 2024) and expanded to multi-view geometry transformers (Wang et al., 2025a;d). DVGT (Zuo et al., 2025) adapts this capability to metric-scale reconstruction of driving scenes without post-alignment. Batch-processing models recompute features for previously observed frames during online inference, limiting efficiency for continuous driving (Wang et al., 2025a; Zuo et al., 2025). Streaming methods maintain a recurrent scene state (Wang et al., 2025b) or reuse cached historical features (Zhuo et al., 2025), although a full-history feature cache incurs growing computation and memory costs as sequences lengthen. In contrast, we combine local geometry prediction with fixed-window caching and relative temporal encoding to support continuous geometry reconstruction and planning with bounded per-frame costs.

3 Proposed Approach

3.1 Vision-Geometry-Action Model

The fundamental objective of an autonomous driving system is to predict a safe and feasible future trajectory for the ego vehicle given historical and current sensor observations. Formally, at the current time step T, given the sequence of multi-view images $\mathbf{I}_{1:T}$, a driving model $\mathcal{M}$ predicts the future ego-trajectory $\mathbf{A}_T$:

$$\mathbf{A}_T = \mathcal{M}(\mathbf{I}_{1:T}). \tag{1}$$

As planning depends on the surrounding 3D physical world, we make geometry an explicit learning target alongside trajectory prediction. We explore a Vision-Geometry-Action (VGA) framework, which uses dense geometry reconstruction as the primary scene representation learning task for autonomous driving. Given historical and current multi-view images $\mathbf{I}_{1:T}$, the model jointly predicts current-frame pointmaps $\mathbf{P}_T$, ego-pose $\mathbf{E}_T$, and the future ego-trajectory $\mathbf{A}_T$:

$$\mathbf{P}_T, \mathbf{E}_T, \mathbf{A}_T = \mathcal{M}_{\text{VGA}}(\mathbf{I}_{1:T}). \tag{2}$$

Through joint geometry reconstruction and trajectory planning, VGA learns shared scene representations that connect dense spatial understanding with driving actions. We investigate whether geometric supervision can yield effective representations for planning without additional language supervision, future-scene modeling, or object- and map-level annotations.

3.2 Ego-Centric Streaming Geometry Reconstruction

In the VGA framework, dense geometry reconstruction from multi-frame inputs is essential for comprehensive scene understanding. Recent works (Wang et al., 2025a) commonly adopt a **Batch-Processing** paradigm. Given a T-frame image sequence $\mathbf{I}_{1:T}$, these models jointly reconstruct global

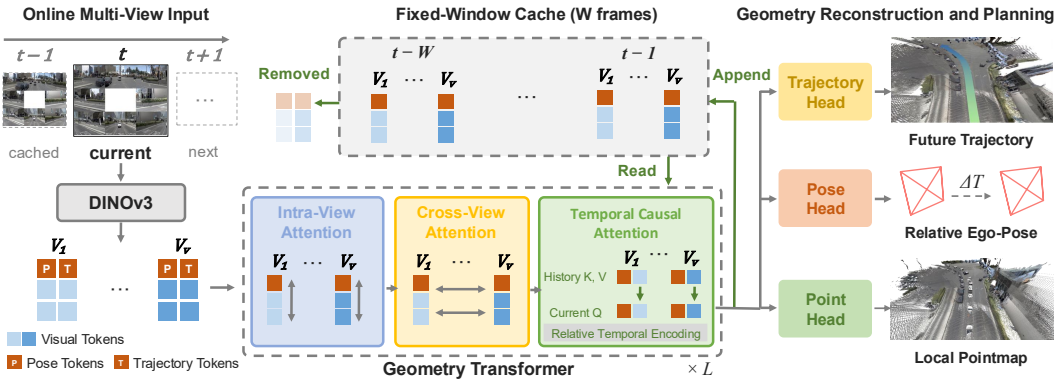

Figure 3: **Overall architecture of DVGT-2.** The image encoder extracts visual tokens from current multi-view images, which are then appended with pose and trajectory tokens. The geometry transformer integrates these tokens with cached features from the past W frames and updates the cache in a FIFO manner. Prediction heads produce local pointmaps, relative ego-pose, and future trajectory.

pointmaps and ego-poses for all frames, typically anchored to the first frame's coordinate system:

$$\mathbf{P}_{1:T}, \mathbf{E}_{1:T} = \mathcal{G}_{\text{batch}}(\mathbf{I}_{1:T}). \tag{3}$$

However, this paradigm requires computing pairwise spatial interactions across all T input frames, resulting in an $\mathcal{O}(T^2)$ computational complexity per inference step. Moreover, when processing online inputs frame by frame, the model repeatedly processes overlapping historical frames. This computational redundancy leads to high inference latency, limiting its applicability to online driving.

To alleviate this redundant computation, recent works (Zhuo et al., 2025) introduce a **Full-History Streaming** paradigm with feature caching. Upon receiving a new frame $\mathbf{I}_T$, the model computes interactions with cached historical features $\mathbf{C}_{1:T-1}$ and predicts current-frame geometry, reducing per-frame complexity from $\mathcal{O}(T^2)$ to $\mathcal{O}(T)$. The current features are then appended to the cache:

$$\mathbf{P}_T, \mathbf{E}_T, \mathbf{C}_{1:T} = \mathcal{G}_{\text{stream}}(\mathbf{I}_T, \mathbf{C}_{1:T-1}). \tag{4}$$

Although this framework avoids historical recomputation, its reconstruction remains anchored to the first frame. It retains the full historical cache to preserve this global temporal context, causing computational costs to grow linearly with sequence length and limiting continuous online driving.

To overcome these bottlenecks, we propose a **Sliding-Window Streaming** paradigm, as shown in Figure 2. The core idea is to maintain a historical feature cache with a fixed window size W. Crucially, we decouple geometry reconstruction from the first-frame coordinate system. Instead, we reconstruct local pointmaps $\mathbf{P}_T$ in the current ego-coordinate system and predict ego-pose $\mathbf{E}_T$ relative to the preceding frame. This local geometry reconstruction removes the first-frame global dependency and aligns geometry reconstruction with ego-centric planning:

$$\mathbf{P}_T, \mathbf{E}_T, \mathbf{C}_{T-W+1:T} = \mathcal{G}_{\text{window}}(\mathbf{I}_T, \mathbf{C}_{T-W:T-1}). \tag{5}$$

After each prediction, the cache is updated in a first-in-first-out manner, discarding the oldest features and appending the current ones. This preserves recent temporal context without full-cache resets and maintains constant $\mathcal{O}(W)$ per-frame complexity for long-sequence online inference.

3.3 Streaming Driving Visual Geometry Transformer

To realize the sliding-window streaming paradigm for joint geometry reconstruction and trajectory planning, we propose **DVGT-2**, a streaming Driving Visual Geometry Transformer.

Overall Architecture. As shown in Figure 3, **DVGT-2** consists of an image encoder, a geometry transformer, and three task-specific prediction heads. Given the current multi-view images $\mathbf{I}_T$, a DINOv3-pretrained ViT encoder (Siméoni et al., 2025) extracts visual tokens. We append learnable pose and trajectory tokens to each view's visual tokens and feed the combined tokens into the geometry transformer for spatial-temporal reasoning. Finally, all these tokens are fed into corresponding heads to predict local pointmaps, relative ego-pose, and future ego-trajectory, respectively.

Streaming Geometry Transformer. Following DVGT (Zuo et al., 2025), we employ factorized attention to model spatial-temporal relations. Each of the L transformer blocks sequentially performs

intra-view local attention, cross-view spatial attention, and temporal causal attention. The first two operations capture token interactions within individual images and across views of the current frame, respectively. Temporal causal attention captures temporal context by using current-frame tokens as queries and cached historical features as keys and values. Each layer maintains its own cache of the preceding W frames, allowing historical features to be reused while only the current frame is processed. After processing the current frame, the cache at layer ℓ is updated as:

$$\mathbf{C}^{\ell}_{T-W+1:T} = \mathrm{FIFO}_W \left(\mathbf{C}^{\ell}_{T-W:T-1}, \hat{\mathbf{G}}^{\ell}_{T} \right), \tag{6}$$

where $\hat{\mathbf{G}}^{\ell}_{T}$ denotes the current intermediate features. The first-in-first-out operation appends these features and retains only the most recent W frames. We use $W = 3$, maintaining recent temporal context without recomputing historical features or periodically resetting the entire cache.

Relative Temporal Encoding. To support inference beyond the training sequence length, we replace absolute temporal embeddings with relative temporal encoding using MRoPE-I (Huang et al., 2025). This expresses temporal relations through relative frame offsets rather than absolute time indices. Within the fixed-size window, these offsets remain bounded as the sequence grows, reducing dependence on temporal positions unseen during training. Together with feature caching, this supports sustained long-sequence online inference.

Prediction Heads. We employ a DPT head (Ranftl et al., 2021) to decode visual features into dense local pointmaps. For pose and trajectory prediction, the corresponding tokens are aggregated across views and passed to two anchor-based diffusion heads adapted from DiffusionDrive (Liao et al., 2025). These heads use truncated diffusion to predict relative ego-poses and future trajectories, respectively. Additional architecture details are provided in Section A.

3.4 Model Training

Datasets. We conduct training and geometry evaluation on five driving datasets: nuScenes (Caesar et al., 2020), OpenScene (Peng et al., 2023), Waymo (Sun et al., 2020), KITTI (Geiger et al., 2013), and DDAD (Guizilini et al., 2020). Following DVGT (Zuo et al., 2025), we use the depth foundation model MoGe-2 (Wang et al., 2025c) to infer dense depth maps and apply threshold-based filtering to obtain high-quality geometric supervision. We further evaluate closed-loop planning performance on NAVSIM v1 and v2 (Dauner et al., 2024; Cao et al., 2025), as well as open-loop planning performance on nuScenes. Dataset details are provided in Section B.

Architecture. **DVGT-2** uses a DINOv3-pretrained ViT-L image encoder (Siméoni et al., 2025) and a 24-block streaming geometry transformer. Temporal causal attention operates on a sliding-window cache of length $W = 3$ and uses relative temporal encoding. Two anchor-based diffusion heads (Liao et al., 2025) predict relative ego-poses and future trajectories. For planning, ego status (velocity, acceleration, and driving command) is embedded into the trajectory token via an MLP.

Training. We pretrain **DVGT-2** on a mixture of the five driving datasets, starting with geometry reconstruction and then introducing streaming and joint geometry–planning training. During pretraining, we randomly sample sequences with 2 to 8 views and 2 to 24 frames per scene. We subsequently fine-tune the model on NAVSIM and nuScenes separately for planning evaluation. Training is performed on 64 H20 GPUs for about 10 days. Additional details are provided in Section A.

4 Experiments

4.1 Experimental Setup

Geometry Reconstruction. Following DVGT (Zuo et al., 2025), we evaluate global point reconstruction using Accuracy (Acc) and Completeness (Comp), with pointmaps expressed in the first-frame coordinate system. Local geometric accuracy is assessed through ray depth, defined as the distance from each 3D point to the current ego-vehicle center, using absolute relative error (Abs Rel) and threshold accuracy ($\delta < 1.25$). For global ego-pose estimation, we report AUC based on relative pose errors between frame pairs.

Planning. We evaluate open-loop planning on nuScenes using the UniAD (Hu et al., 2023) metrics: L2 displacement error and collision rate over a 3-second future horizon. Closed-loop planning is

Table 1: **Quantitative 3D geometry reconstruction results on OpenScene.** * denotes performing post-alignment with sparse LiDAR to recover the metric scale.

Method	Paradigm	Acc ↓	Comp ↓	Abs Rel ↓	δ<1.25 ↑	AUC ↑	Time
VGGT* (Wang et al., 2025a)	Full-Seq.	1.705	1.711	0.280	0.669	74.4	∼5.31s
MapAnything (Keetha et al., 2025)	Full-Seq.	3.269	4.214	0.476	0.253	69.5	∼2.28s
DVGT (Zuo et al., 2025)	Full-Seq.	**0.412**	0.491	0.048	0.971	**76.6**	∼1.88s
CUT3R* (Wang et al., 2025b)	Streaming	1.858	2.245	0.275	0.596	36.9	∼0.35s
StreamVGGT* (Zhuo et al., 2025)	Streaming	2.209	2.060	0.303	0.620	74.1	∼1.94s
Driv3R* (Fei et al., 2024)	Streaming	0.884	1.693	0.188	0.740	-	∼0.56s
DVGT-2	Streaming	0.440	**0.450**	**0.040**	**0.977**	70.3	∼**0.27**s

Table 2: **Quantitative 3D geometry reconstruction results on nuScenes.**

Method	Paradigm	Acc ↓	Comp ↓	Abs Rel ↓	δ<1.25 ↑	AUC ↑
VGGT* (Wang et al., 2025a)	Full-Seq.	1.944	2.071	0.348	0.573	83.9
MapAnything (Keetha et al., 2025)	Full-Seq.	4.447	4.860	0.562	0.271	84.9
DVGT (Zuo et al., 2025)	Full-Seq.	**0.469**	**0.508**	0.067	0.955	**86.4**
CUT3R* (Wang et al., 2025b)	Streaming	2.055	2.611	0.330	0.547	44.5
StreamVGGT* (Zhuo et al., 2025)	Streaming	2.414	2.284	0.375	0.563	86.0
Driv3R* (Fei et al., 2024)	Streaming	0.742	1.345	0.189	0.721	–
DVGT-2	Streaming	0.775	0.792	**0.055**	**0.965**	84.5

evaluated on NAVSIM v1 and v2 using PDMS (Dauner et al., 2024) and EPDMS (Cao et al., 2025), respectively. These composite scores assess safety, comfort, progress, and traffic-rule compliance. Further definitions of the geometry and planning metrics are provided in Section C.

4.2 Geometry Reconstruction

We evaluate geometry reconstruction on 16-frame multi-view sequences in an online, frame-by-frame manner, as shown in Tables 1 and 2. At each timestep t, all methods access only the current and preceding frames of the sequence. For non-streaming baselines, we rerun inference on the entire observed sequence, comprising frames 1 through t. For global evaluation, we accumulate our predicted relative ego-poses and transform local pointmaps into the first-frame coordinate system. Additional geometry results on Waymo, DDAD, and KITTI are provided in Section D.1.

Ray Depth Estimation. **DVGT-2** achieves the lowest Abs Rel and highest $\delta < 1.25$ among the compared methods on both OpenScene and nuScenes, outperforming general-purpose and driving-specific geometry models. In particular, Abs Rel decreases from 0.048 to 0.040 on OpenScene and from 0.067 to 0.055 on nuScenes compared with DVGT. The same advantage holds on Waymo, DDAD, and KITTI, showing consistently strong local geometric accuracy across all five datasets.

Global Point Reconstruction. On OpenScene, **DVGT-2** achieves the best completeness of 0.450, while its accuracy of 0.440 is close to DVGT's 0.412. On nuScenes, it improves both accuracy and completeness over general-purpose models such as VGGT and StreamVGGT, although it remains behind DVGT. Unlike ray depth evaluation, global point reconstruction also depends on accumulated ego-pose estimates, making it sensitive to global pose errors in addition to local geometric accuracy.

Global Ego-Pose Estimation. **DVGT-2** achieves AUC scores of 70.3 on OpenScene and 84.5 on nuScenes, compared with 76.6 and 86.4 for DVGT, respectively. Our supervision prioritizes local geometry and relative ego-poses for ego-centric planning, rather than global reconstruction. Accumulating relative poses for global evaluation can amplify estimation errors, partly explaining the suboptimal global pose estimation performance.

4.3 Planning

Closed-loop Planning on NAVSIM v1 and v2. As shown in Table 3, **DVGT-2** achieves a PDMS of 92.5 on NAVSIM v1, surpassing the leading VLA and world-action models, including Qwen-Drive-1.0 (Zhou et al., 2026a) and DriveVA (Liu et al., 2026). On NAVSIM v2 (Table 4), **DVGT-2** achieves an EPDMS of 90.0, the highest among the compared methods, exceeding the 88.8 of ExploreVLA and the 88.9 of EponaV2. These results demonstrate that VGA can achieve leading closed-loop planning performance among the compared methods.

Table 3: **Planning results on NAVSIM v1 `navtest`.** C and L denote camera and LiDAR.

Method	Input	NC ↑	DAC ↑	TTC ↑	Comf. ↑	EP ↑	PDMS ↑
Perception-based Methods							
VADv2 (Chen et al., 2024)	C	97.2	89.1	91.6	100	76.0	80.9
UniAD (Hu et al., 2023)	C	97.8	91.9	92.9	100	78.8	83.4
Transfuser (Chitta et al., 2022)	C & L	97.7	92.8	92.8	100	79.2	84.0
DiffusionDrive (Liao et al., 2025)	C & L	98.2	96.2	94.7	100	82.2	88.1
WoTE (Li et al., 2025c)	C & L	98.5	96.8	94.9	99.9	81.9	88.3
DriveSuprim (Yao et al., 2025)	C & L	97.8	97.3	93.6	100	86.7	89.9
VLA-based Methods							
AdaThinkDrive (Luo et al., 2025)	C	98.5	94.4	94.9	100	79.9	86.2
AutoVLA (Zhou et al., 2025b)	C	98.4	95.6	98.0	99.9	81.9	89.1
DriveDreamer-Policy (Zhou et al., 2026b)	C	98.4	97.1	95.1	100	83.5	89.2
ReCogDrive (Li et al., 2025d)	C	98.2	97.8	95.2	99.8	83.5	89.6
DriveVLA-W0 (Li et al., 2025b)	C	98.7	99.1	95.3	99.3	83.3	90.2
ExploreVLA (Sheng et al., 2026)	C	98.8	98.4	96.5	99.9	83.5	90.4
Qwen-Drive-1.0-RL (Zhou et al., 2026a)	C	98.6	98.2	95.9	100	84.8	90.7
WAM-based Methods							
LAW (Li et al., 2024)	C	96.4	95.4	88.7	99.9	81.7	84.6
Epona (Zhang et al., 2025)	C	97.9	95.1	93.8	99.9	80.4	86.2
PWM (Zhao et al., 2026)	C	98.6	95.9	95.4	100	81.8	88.1
DriveLaW (Xia et al., 2026)	C	99.0	97.1	96.7	100	81.3	89.1
DriveWAM (Shi et al., 2026)	C	98.3	98.1	95.2	100	84.3	90.1
EponaV2 (Xu et al., 2026)	C	98.6	97.9	95.7	100	84.8	90.4
DriveVA (Liu et al., 2026)	C	99.2	97.5	98.7	100	83.5	90.9
VGA-based Methods							
DVGT-2	C	98.9	97.7	95.9	100	89.6	**92.5**

Table 4: **Planning results on NAVSIM v2 `navtest` split**.

Method	NC ↑	DAC ↑	DDC ↑	TL ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
Ego Status MLP	93.1	77.9	92.7	99.6	86.0	91.5	89.4	98.3	85.4	64.0
Transfuser (Chitta et al., 2022)	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7
Hydra-MDP++ (Li et al., 2025a)	97.2	97.5	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
DriveSuprim (Yao et al., 2025)	97.5	96.5	99.4	99.6	88.4	96.6	95.5	98.3	77.0	83.1
ARTEMIS (Feng et al., 2025)	98.3	95.1	98.6	99.8	81.5	97.4	96.5	98.3	-	83.1
DiffusionDrive (Liao et al., 2025)	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	84.5
DriveVLA-W0 (Li et al., 2025b)	98.5	99.1	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1
DriveLaW (Xia et al., 2026)	98.7	96.9	99.6	99.8	87.5	98.3	97.6	98.4	77.4	88.6
ExploreVLA (Sheng et al., 2026)	98.8	96.2	99.6	99.8	87.1	98.2	97.8	98.3	86.8	88.8
EponaV2 (Xu et al., 2026)	98.5	97.4	99.5	99.9	87.9	98.1	97.7	98.2	77.4	88.9
DVGT-2	98.9	97.7	99.4	99.6	92.9	98.4	97.7	97.8	71.6	**90.0**

Open-loop Planning on nuScenes. As shown in Table 5, **DVGT-2** achieves an average L2 error of 0.78 m and an average collision rate of 0.19% using the UniAD metrics. Its collision rate is the second lowest among the compared methods, while its displacement error remains competitive. Together with the NAVSIM results, these findings demonstrate competitive planning performance using dense geometry as auxiliary supervision, without explicit object- or map-level annotations.

Visualizations. Figure 4 presents joint geometry reconstruction and trajectory planning results.

4.4 Ablation Studies

For ablation studies, we conduct online evaluation on 200 sequences from the OpenScene test set, each containing more than 200 frames, as shown in Table 6. All ablation variants use the same training data and training budget. Non-streaming variants (rows a–b) use a fixed 16-frame history window to bound memory usage. Notably, PDMS here is evaluated on these sequences rather than the standard NAVSIM test split, and the values are therefore not directly comparable to Table 3. We further provide the cache-window size ablation in Section D.2.

Geometric Supervision. Adding geometric supervision encourages geometry-aware representations, increasing PDMS from 71.2 to 91.5 with modest latency and memory overhead (rows a–b).

Table 5: **Open-loop planning results on nuScenes using UniAD metrics.** Aux. Sup. denotes auxiliary supervision. Avg. averages the results at 1s, 2s, and 3s.

Method	Input	Aux. Sup.	L2 (m) ↓				Collision Rate (%) ↓			
			1s	2s	3s	Avg.	1s	2s	3s	Avg.
UniAD (Hu et al., 2023)	Camera	Map & Box & Mot. & Occ	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31
VAD-Tiny (Jiang et al., 2023)	Camera	Map & Box & Mot.	0.60	1.23	2.06	1.30	0.31	0.53	1.33	0.72
VAD-Base (Jiang et al., 2023)	Camera	Map & Box & Mot.	0.54	1.15	1.98	1.22	0.04	0.39	1.17	0.53
GenAD (Zheng et al., 2024b)	Camera	Map & Box & Mot.	0.36	0.83	1.55	0.91	0.06	0.23	1.00	0.43
OccWorld (Zheng et al., 2024a)	Camera	Occ	0.52	1.27	2.41	1.40	0.12	0.40	2.08	0.87
OccNet (Tong et al., 2023)	Camera	Map & Box & Occ	1.29	2.13	2.99	2.14	0.21	0.59	1.37	0.72
GaussianAD (Zheng et al., 2024c)	Camera	Map & Box & Mot. & Occ	0.40	0.64	**0.88**	**0.64**	0.09	0.38	0.81	0.42
Doe-1 (Zheng et al., 2024d)	Camera	Language	0.50	1.18	2.11	1.26	0.04	0.37	1.19	0.53
UniUGP (Lu et al., 2025)	Camera	Language	0.58	1.14	1.95	1.23	0.01	0.19	0.81	0.33
OpenDriveVLA (Zhou et al., 2025a)	Camera	Map & Box & Mot. & Language	**0.19**	**0.58**	1.24	0.67	0.02	0.18	0.70	0.30
AutoVLA (Zhou et al., 2025b)	Camera	Language	0.33	0.81	1.45	0.86	0.08	0.11	0.85	0.35
UniDriveVLA (Li et al., 2026)	Camera	Map & Box & Occ & Language	0.30	0.66	1.25	0.74	0.02	0.18	**0.40**	0.20
DVGT-2	Camera	Dense Geometry	0.25	0.67	1.43	0.78	**0.00**	**0.07**	0.50	**0.19**

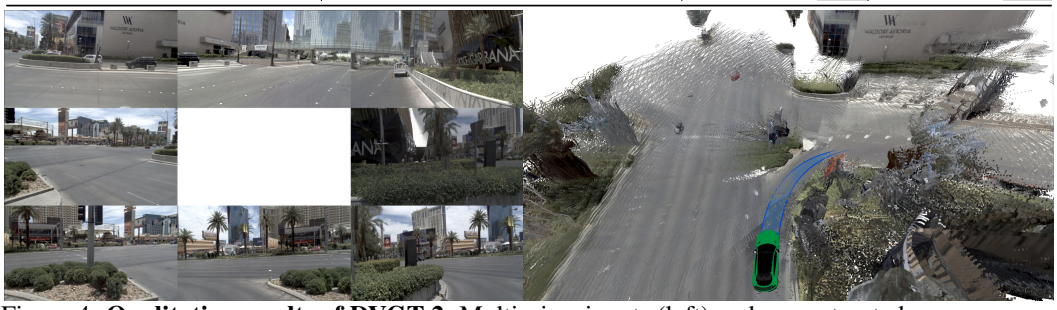

Figure 4: **Qualitative results of DVGT-2.** Multi-view inputs (left) and reconstructed scene geometry with predicted ego-trajectories (right).

Table 6: **Ablation of geometric supervision and streaming components.** Row f is the full model.

ID	Components					Performance		Efficiency	
	Geom. supervision	Streaming cache	Local coords.	Fixed cache window	Relative time enc.	Abs Rel ↓	PDMS ↑	Latency (ms) ↓	Memory (GB) ↓
a	×	×	×	×	×	–	71.2	3846	60.5
b	✓	×	×	×	×	0.060	91.5	4348	61.1
c	✓	✓	×	×	×	0.161	72.6	450	82.4
d	✓	✓	✓	×	×	0.124	69.8	450	82.4
e	✓	✓	✓	✓	×	0.115	89.4	280	17.3
f	✓	✓	✓	✓	✓	**0.047**	**92.2**	**270**	**16.8**

Streaming Components. Reusing historical features reduces latency, but the growing full-history cache must be reset as memory approaches the GPU limit, disrupting temporal context and degrading geometry and planning (row c). Local-coordinate supervision improves geometric accuracy but does not resolve this context loss (row d). A fixed-size window retains recent context without full-cache resets, recovering performance while reducing latency and memory use (row e). Relative temporal encoding further improves geometry and planning on sequences beyond the training length (row f).

Scenario Analysis. We examine whether geometric rather than semantic auxiliary supervision can support planning in semantics-sensitive scenarios. Scenarios are drawn from the OpenScene test split and grouped using its official scenario-type annotations. As shown in Table 8, **DVGT-2** achieves higher PDMS in both groups, suggesting that its geometric representation captures planning-relevant semantic cues, potentially aided by DINOv3 pretraining.

4.5 Efficiency Analysis

Geometry Reconstruction Efficiency. We measure per-frame latency, FLOPs, and GPU memory on identical eight-view sequences using a single H20 GPU with FP16 precision, following the online evaluation protocol in Section 4.2. As shown in Figure 5, VGGT and DVGT repeatedly process the observed sequence, while StreamVGGT maintains a growing history cache. Their computational and memory costs increase with sequence length, eventually leading to out-of-memory errors.

Table 7: Inference efficiency of VLA, WAM, and VGA methods on an NVIDIA A800 GPU.

Method	Resolution	Views	Latency (ms) ↓	Memory (GB) ↓	FLOPs (T) ↓
VLA-based methods					
DriveVLA-W0 (Li et al., 2025b)	144 × 256	1	1,390	19.5	209.9
DriveDreamer-Policy (Zhou et al., 2026b)	288 × 512	8	465	10.3	35.4
DynVLA (Shang et al., 2026)	288 × 512	1	1,438	23.8	104.9
AutoVLA (Zhou et al., 2025b)	280 × 504	3	12,171	7.9	18.3
WAM-based methods					
DriveVA (Liu et al., 2026)	288 × 512	1	256	23.4	25.6
Epona (Zhang et al., 2025)	288 × 512	1	374	14.5	7.0
DriveLaW (Xia et al., 2026)	288 × 512	1	589	16.4	10.2
DriveWAM (Shi et al., 2026)	288 × 512	1	1,107	26.2	19.3
VGA-based methods					
DVGT-2	288 × 512	8	256	15.8	16.5

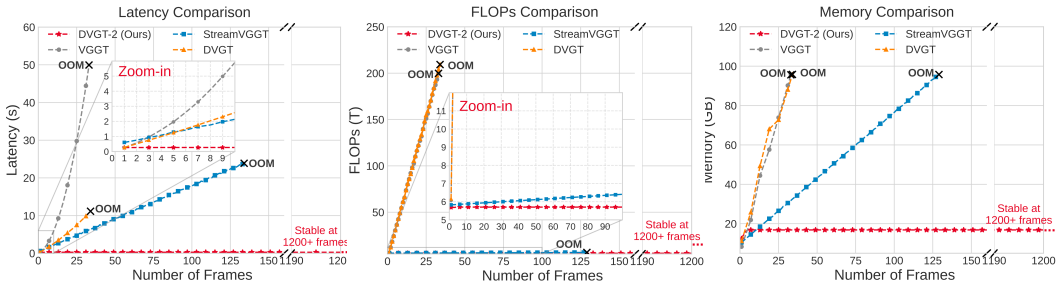

Figure 5: **Efficiency comparison of online inference.** Latency, FLOPs, and memory versus processed frames, measured on a single H20 GPU in FP16.

Table 8: **Planning performance across driving scenarios.** All scores are PDMS (↑).

Model	Paradigm	Semantics-sensitive		Geometry-sensitive	
		Traffic Sign	Traffic Lane	Following	Collision Avoidance
ELF-VLA	VLA	87.3	84.4	87.2	87.3
DVGT-2	VGA	**89.7**	**88.3**	**89.3**	**90.0**

In contrast, our fixed-size sliding-window cache maintains approximately constant per-frame computation and memory usage, with latency remaining around 260 ms over more than 1,200 frames. Notably, **DVGT-2** jointly performs geometry reconstruction and trajectory planning, whereas these baselines only reconstruct geometry. Together with the reconstruction results in Section 4.2, these measurements demonstrate a favorable accuracy-efficiency trade-off for online inference.

Planning Efficiency. We compare publicly available VLA and WAM methods on the same NVIDIA A800 GPU. All measurements use a batch size of 1, 10 warm-up runs, and 100 synchronized timed runs. We use 288 × 512 images from 8 views when supported; otherwise, we use the closest supported resolution and view count. As shown in Table 7, **DVGT-2** achieves the lowest reported latency of 256 ms with 15.8 GB memory and 16.5 T FLOPs. Together with its planning performance, these results show that **DVGT-2** can support competitive planning with high inference efficiency.

5 Conclusion

In this paper, we introduce **DVGT-2**, a streaming driving visual geometry transformer that pioneers the Vision-Geometry-Action (VGA) paradigm for end-to-end autonomous driving. By reconstructing dense 3D geometry as the foundation representation, **DVGT-2** provides comprehensive spatial and temporal cues for robust trajectory planning. To overcome the computational bottleneck of traditional multi-frame processing, we propose a sliding-window streaming strategy with temporal causal attention and feature caching, enabling efficient, on-the-fly joint prediction of geometry and trajectories. Extensive experiments demonstrate that **DVGT-2** achieves strong geometry reconstruction performance with significantly reduced latency on diverse datasets. Moreover, it exhibits strong planning capabilities across open-loop and closed-loop benchmarks. We hope this work paves the way for more efficient, geometry-aware driving systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript revision and experimental results integration. We have also used generative AI tools for LaTeX template migration and formatting. The authors reviewed and revised the AI-assisted content. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with AI assistance.

REFERENCES

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 *CVPR*, pp. 11621–11631, 2020.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *TPAMI*, 45 (11):12878–12895, 2022.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *NeurIPS*, 37:28706–28719, 2024.

Xin Fei, Wenzhao Zheng, Yueqi Duan, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Jiwen Lu. Driv3r: Learning dense 4d reconstruction for autonomous driving. *arXiv preprint arXiv:2412.06777*, 2024.

Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *arXiv preprint arXiv:2504.19580*, 2025.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32(11):1231–1237, 2013.

Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, pp. 2485–2494, 2020.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *EMNLP*, pp. 4246–4253, 2020.

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 *CVPR*, pp. 17853–17862, 2023.

Jie Huang, Xuejing Liu, Sibo Song, Ruibing Hou, Hong Chang, Junyang Lin, and Shuai Bai. Revisiting multimodal positional encoding in vision-language models. *arXiv preprint arXiv:2510.23095*, 2025.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *ICCV*, pp. 8340–8350, 2023.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 2024.

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

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

Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025a.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. *arXiv preprint arXiv:2406.08481*, 2024.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. *arXiv preprint arXiv:2510.12796*, 2025b.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. *arXiv preprint arXiv:2504.01941*, 2025c.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025d.

Yongkang Li, Lijun Zhou, Sixu Yan, Bencheng Liao, Tianyi Yan, Kaixin Xiong, Long Chen, Hongwei Xie, Bing Wang, Guang Chen, et al. Unidrivevla: Unifying understanding, perception, and action planning for autonomous driving. *arXiv preprint arXiv:2604.02190*, 2026.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *CVPR*, pp. 12037–12047, 2025.

Mengmeng Liu, Diankun Zhang, Jiuming Liu, Jianfeng Cui, Hongwei Xie, Guang Chen, Hangjun Ye, Michael Ying Yang, Francesco Nex, and Hao Cheng. DriveVA: Video action models are zero-shot drivers. *arXiv preprint arXiv:2604.04198*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Hao Lu, Ziyang Liu, Guangfeng Jiang, Yuanfei Luo, Sheng Chen, Yangang Zhang, and Ying-Cong Chen. Uniugp: Unifying understanding, generation, and planing for end-to-end autonomous driving. *arXiv preprint arXiv:2512.09864*, 2025.

Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, et al. Adathinkdrive: Adaptive thinking via reinforcement learning for autonomous driving. *arXiv preprint arXiv:2509.13769*, 2025.

Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–824. IEEE, 2023.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pp. 12179–12188, 2021.

Shuyao Shang, Bing Zhan, Yunfei Yan, Yuqi Wang, Yingyan Li, Yasong An, Xiaoman Wang, Jierui Liu, Lu Hou, Lue Fan, et al. Dynvla: Learning world dynamics for action reasoning in autonomous driving. *arXiv preprint arXiv:2603.11041*, 2026.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *CVPR*, pp. 15120–15130, 2024.

Zihao Sheng, Xin Ye, Jingru Luo, Sikai Chen, and Liu Ren. ExploreVLA: Dense world modeling and exploration for end-to-end autonomous driving. *arXiv preprint arXiv:2604.02714*, 2026.

Chen Shi, Jinrui Xu, Shaoshuai Shi, Kehua Sheng, Bo Zhang, and Li Jiang. DriveWAM: Video generative priors enable scalable world-action modeling for autonomous driving. *arXiv preprint arXiv:2605.28544*, 2026.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

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*, pp. 2446–2454, 2020.

Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8406–8415, 2023.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, pp. 32–42, 2021.

Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *ICCV*, pp. 9773–9783, 2023.

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

Jie Wang, Guang Li, Zhijian Huang, Chenxu Dang, Hangjun Ye, Yahong Han, and Long Chen. Vggdrive: Empowering vision-language models with cross-view geometric grounding for autonomous driving. *arXiv preprint arXiv:2602.20794*, 2026.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 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*, 2025c.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 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.

Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *CVPR*, pp. 15449–15458, 2024.

Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, et al. DriveLaW: Unifying planning and video generation in a latent driving world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39701–39712, 2026.

Jiawei Xu, Zhizhou Zhong, Zhijian Shu, Mingkai Jia, Mingxiao Li, Jia-Wang Bian, Qian Zhang, Kaicheng Zhang, Jin Xie, Jian Yang, et al. EponaV2: Driving world model with comprehensive future reasoning. *arXiv preprint arXiv:2605.14696*, 2026.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *NeurIPS*, 37:21875–21911, 2024.

Pengxuan Yang, Ben Lu, Zhongpu Xia, Chao Han, Yinfeng Gao, Teng Zhang, Kun Zhan, XianPeng Lang, Yupeng Zheng, and Qichao Zhang. Worldrft: Latent world model planning with reinforcement fine-tuning for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11649–11657, 2026.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M Alvarez, and Zuxuan Wu. Drivesuprim: Towards precise trajectory selection for end-to-end planning. *arXiv preprint arXiv:2506.06659*, 2025.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, et al. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27220–27230, 2025.

Zhida Zhao, Talas Fu, Yifan Wang, Lijun Wang, and Huchuan Lu. From forecasting to planning: Policy world model for collaborative state-action prediction. *Advances in Neural Information Processing Systems*, 38:134585–134611, 2026.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *ECCV*, pp. 55–72. Springer, 2024a.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *ECCV*, pp. 87–104. Springer, 2024b.

Wenzhao Zheng, Junjie Wu, Yao Zheng, Sicheng Zuo, Zixun Xie, Longchao Yang, Yong Pan, Zhihui Hao, Peng Jia, Xianpeng Lang, et al. Gaussianad: Gaussian-centric end-to-end autonomous driving. *arXiv preprint arXiv:2412.10371*, 2024c.

Wenzhao Zheng, Zetian Xia, Yuanhui Huang, Sicheng Zuo, Jie Zhou, and Jiwen Lu. Doe-1: Closed-loop autonomous driving with large world model. *arXiv preprint arXiv:2412.09627*, 2024d.

Xin Zhou, Zongchuang Zhao, Zhibo Yang, Mingsheng Li, Humen Zhong, Shuai Bai, Du Chu, Ruizhe Chen, Zhaohai Li, Jun Tang, et al. Qwen-drive-1.0: An initial step towards a vision-language foundation model for autonomous driving. *arXiv preprint arXiv:2609.00111*, 2026a.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, and Alois C Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. *arXiv preprint arXiv:2503.23463*, 2025a.

Yang Zhou, Xiaofeng Wang, Hao Shao, Letian Wang, Guosheng Zhao, Jiangnan Shao, Jiagang Zhu, Tingdong Yu, Zheng Zhu, Guan Huang, et al. DriveDreamer-Policy: A geometry-grounded world-action model for unified generation and planning. *arXiv preprint arXiv:2604.01765*, 2026b.

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*, 2025b.

Runwen Zhu, Jianbo Zhao, Diankun Zhang, Guoan Wang, Xiwu Chen, Siyu Zhang, Jiahao Gong, Qibin Zhou, Wenyuan Zhang, Ningzi Wang, et al. Sparsead: Sparse query-centric paradigm for efficient end-to-end autonomous driving. *IEEE Transactions on Artificial Intelligence*, 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.

Appendix

A Additional Implementation Details

Architecture. DVGT-2 comprises approximately 1.8 billion parameters. The geometry transformer consists of 24 blocks with a feature dimension of 1024 and 16 attention heads. To ensure training stability, we apply QKNorm (Henry et al., 2020) and LayerScale (Touvron et al., 2021) (initialized at 0.01) in each attention layer. For dense pointmap prediction, following Yang et al. (2024), we feed intermediate tokens from the 4th, 11th, 17th, and 23rd blocks into a DPT (Ranftl et al., 2021) head. For ego-pose prediction and trajectory planning, we augment the visual tokens of each view and frame with one pose token and eight trajectory tokens to aggregate global context. Two anchor-based diffusion heads adapted from DiffusionDrive (Liao et al., 2025) predict relative ego-poses and future trajectories, using four self-attention layers for inter-frame interactions and two cross-attention layers for diffusion decoding. We employ 20 anchors for both ego-poses and trajectories, which are pre-computed by clustering the training data.

Training. We pretrain **DVGT-2** on the mixture of five driving datasets, first for 160K iterations with geometry reconstruction supervision, followed by 80K iterations of streaming training with joint geometry and planning supervision. We then fine-tune the model separately on NAVSIM and nuScenes for 40K iterations each. The peak learning rate is 10^{-4} during pretraining and 10^{-5} during fine-tuning. Across all stages, we use AdamW (Loshchilov & Hutter, 2019) with a cosine learning rate schedule and an 8K-iteration linear warmup. We apply gradient norm clipping at 1.0, bfloat16 precision, and gradient checkpointing for training stability and efficiency. All pretraining and fine-tuning stages are conducted on 64 NVIDIA H20 GPUs.

Input Sampling. During pretraining, we randomly sample 2–8 camera views and 2–24 frames per sequence, with the sequence length and batch size constrained by a budget of 48 images per GPU. For fine-tuning, we use 4-frame sequences with 8 views on NAVSIM and 6 views on nuScenes, with a per-GPU batch size of 1.

Image Preprocessing. We resize the image long edge to 512 pixels while preserving the aspect ratio. During pretraining, we sample a target aspect ratio from $[1.6, 3.3]$ and center-crop the short edge accordingly. The crop size is rounded down to a multiple of 16, resulting in short-edge lengths of 144–320 pixels. Fine-tuning uses fixed image dimensions. During training, we apply per-frame color jittering, Gaussian blur, and grayscale augmentation.

Training Objectives. We follow DVGT (Zuo et al., 2025) for geometric reconstruction and adapt its pose regression loss to relative ego-poses during initial pretraining. During joint streaming training and subsequent fine-tuning, both pose and trajectory heads use focal classification and L_1 regression losses, following DiffusionDrive (Liao et al., 2025). The joint objective is

$$\mathcal{L} = \mathcal{L}_{\text{geo}} + 5\mathcal{L}_{\text{pose}} + 5\mathcal{L}_{\text{traj}}, \tag{7}$$

where $\mathcal{L}_{\text{pose}}$ and $\mathcal{L}_{\text{traj}}$ each sum their classification and regression terms across decoder outputs. During initial pretraining, the trajectory loss is omitted and the pose loss contains only relative pose regression, with the same task weights.

Evaluation Checkpoints. We use dataset-specific fine-tuned checkpoints for the NAVSIM and nuScenes planning benchmarks and the driving-scenario analysis. All other evaluations use the pretrained checkpoints without dataset-specific fine-tuning.

B Dataset Details

Following DVGT (Zuo et al., 2025), we use five driving datasets for pretraining and geometry evaluation: nuScenes, OpenScene, Waymo, KITTI, and DDAD. Their statistics are summarized in Table A.1. For each pretraining batch, we select a dataset using sampling weights of $6 : 77 : 6 : 5 : 6$ in the above order.

The frame counts in Table A.1 describe segmented clips rather than the full continuous logs. For ablation and long-sequence evaluation, we randomly select 200 continuous sequences from the original OpenScene test logs, each exceeding 200 frames and 100 s at 2 Hz. The efficiency measurements be-

Table A.1: **Detailed statistics of the datasets used in our experiments.** Frame counts refer to segmented clips sampled at 2 Hz.

Dataset	Train Scenes	Test Scenes	Min Frames	Max Frames	Avg Frames	Avg Aspect Ratio	Num of Views
nuScenes	700	150	32	41	40	1.77	6
OpenScene	19376	2026	1	41	38	1.77	8
Waymo	798	202	34	40	40	1.77	5
KITTI	138	13	2	1033	62	3.31	2
DDAD	150	50	10	20	17	1.59	6

yond 1,200 frames likewise use long sequences from the original continuous logs in the OpenScene test split.

C Detailed Evaluation Metrics

Point Reconstruction. Following DVGT (Zuo et al., 2025), we evaluate point reconstruction using Accuracy (Acc) and Completeness (Comp). Given the valid predicted and ground-truth point sets $\mathcal{P}$ and $\mathcal{G}$, these metrics measure bidirectional nearest-neighbor distances:

$$\begin{aligned} \text{Acc} &= \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{g}\in\mathcal{G}} \|\mathbf{p}-\mathbf{g}\|_2, \\ \text{Comp} &= \frac{1}{|\mathcal{G}|} \sum_{\mathbf{g}\in\mathcal{G}} \min_{\mathbf{p}\in\mathcal{P}} \|\mathbf{g}-\mathbf{p}\|_2. \end{aligned} \tag{8}$$

Both metrics are reported in meters. For global evaluation, pointmaps are expressed in the first-frame coordinate system; for **DVGT-2**, this requires accumulating predicted relative ego-poses to transform local pointmaps. The cache-window ablation also reports global Acc. In contrast, the long-sequence evaluation reports local Acc, using the same distance metric in the current ego-coordinate system.

Ray Depth Estimation. Ray depth measures the Euclidean distance from a reconstructed point to the current ego-vehicle center. Let $\hat{d}_u$ and d_u denote predicted and ground-truth ray depths at valid pixel $u \in \Omega$. We compute absolute relative error and threshold accuracy as:

$$\begin{aligned} \text{Abs Rel} &= \frac{1}{|\Omega|} \sum_{u\in\Omega} \frac{|\hat{d}_u - d_u|}{d_u}, \\ \delta_{1.25} &= \frac{1}{|\Omega|} \sum_{u\in\Omega} \mathbb{I}\left[\max\left(\frac{\hat{d}_u}{d_u}, \frac{d_u}{\hat{d}_u}\right) < 1.25\right], \end{aligned} \tag{9}$$

where $\mathbb{I}[\cdot]$ is the indicator function. The latter is reported as $\delta < 1.25$ in the tables.

Global Ego-Pose Estimation. Following Wang et al. (2023), we compute relative rotation and translation-direction errors between frame pairs from the predicted and ground-truth global ego-poses. The error of each pair is the maximum of its rotation and translation angular errors. We report the normalized area under the accuracy–threshold curve up to 30° (AUC@30), expressed as a percentage. This metric is abbreviated as AUC in the geometry tables.

Open-loop Planning. On nuScenes, we follow the UniAD (Hu et al., 2023) evaluation protocol and report L2 displacement error in meters and collision rate in percent at 1, 2, and 3 seconds. L2 measures the Euclidean distance between predicted and ground-truth ego positions, while collision evaluation checks the ego-vehicle footprint against obstacle occupancy.

Closed-loop Planning. We use the official NAVSIM evaluation protocols, reporting PDMS on NAVSIM v1 (Dauner et al., 2024) and EPDMS on NAVSIM v2 (Cao et al., 2025). PDMS combines no at-fault collisions (NC), drivable area compliance (DAC), time-to-collision within bound (TTC), comfort (Comf.), and ego progress (EP). EPDMS additionally considers driving direction compliance (DDC), traffic light compliance (TL), and lane keeping (LK), with comfort assessed through history comfort (HC) and extended comfort (EC).

Table A.2: **Quantitative 3D geometry reconstruction results on Waymo.** * denotes performing post-alignment with sparse LiDAR to recover the metric scale.

Method	Paradigm	Acc ↓	Comp ↓	Abs Rel ↓	δ<1.25 ↑	AUC ↑
VGGT* (Wang et al., 2025a)	Full-Seq.	3.202	3.390	0.333	0.567	84.3
MapAnything (Keetha et al., 2025)	Full-Seq.	9.903	8.365	0.492	0.217	82.3
DVGT (Zuo et al., 2025)	Full-Seq.	1.752	2.263	0.102	0.922	**86.0**
CUT3R* (Wang et al., 2025b)	Streaming	3.369	4.192	0.290	0.563	50.9
StreamVGGT* (Zhuo et al., 2025)	Streaming	3.207	2.960	0.302	0.604	85.6
Driv3R* (Fei et al., 2024)	Streaming	**0.800**	**1.311**	0.168	0.770	–
DVGT-2	Streaming	1.238	1.367	**0.073**	**0.949**	85.9

Table A.3: **Quantitative 3D geometry reconstruction results on DDAD.**

Method	Paradigm	Acc ↓	Comp ↓	Abs Rel ↓	δ<1.25 ↑	AUC ↑
VGGT* (Wang et al., 2025a)	Full-Seq.	2.322	2.879	0.798	0.395	86.3
MapAnything (Keetha et al., 2025)	Full-Seq.	7.442	7.928	1.836	0.207	87.4
DVGT (Zuo et al., 2025)	Full-Seq.	**0.751**	**1.017**	0.145	0.848	**95.3**
CUT3R* (Wang et al., 2025b)	Streaming	2.827	4.724	0.875	0.317	48.6
StreamVGGT* (Zhuo et al., 2025)	Streaming	2.655	2.731	0.810	0.419	91.9
Driv3R* (Fei et al., 2024)	Streaming	0.950	1.259	0.185	0.740	–
DVGT-2	Streaming	1.770	1.837	**0.093**	**0.919**	92.5

Table A.4: **Quantitative 3D geometry reconstruction results on KITTI.**

Method	Paradigm	Acc ↓	Comp ↓	Abs Rel ↓	δ<1.25 ↑	AUC ↑
VGGT* (Wang et al., 2025a)	Full-Seq.	1.154	1.294	0.158	0.801	**96.91**
MapAnything (Keetha et al., 2025)	Full-Seq.	1.807	**1.006**	0.184	0.727	90.51
DVGT (Zuo et al., 2025)	Full-Seq.	**0.846**	1.468	0.136	0.849	87.63
CUT3R* (Wang et al., 2025b)	Streaming	0.973	2.054	0.217	0.660	51.82
StreamVGGT* (Zhuo et al., 2025)	Streaming	3.393	2.181	0.365	0.467	95.57
Driv3R* (Fei et al., 2024)	Streaming	0.864	1.083	0.164	0.784	-
DVGT-2	Streaming	1.615	3.238	**0.087**	**0.942**	80.36

Long-Sequence Evaluation. Planning accuracy is measured by the mean L1 trajectory error over the future 4 seconds:

$$\mathrm{L1}_{4\mathrm{s}} = \frac{1}{K}\sum_{k=1}^{K}\left(|\hat{x}_k - x_k| + |\hat{y}_k - y_k|\right), \tag{10}$$

where K is the number of future waypoints within the 4-second horizon, and $(\hat{x}_k, \hat{y}_k)$ and (x_k, y_k) are predicted and ground-truth ego positions. The error is reported in meters and averages over the entire horizon rather than only its endpoint. At each frame index in Table A.6, we average current-frame measurements across the same 200 sequences, rather than cumulatively averaging over preceding frames.

D Additional Experiments

D.1 Additional Geometry Reconstruction Results

Tables A.2 to A.4 report geometry reconstruction results on Waymo, DDAD, and KITTI. **DVGT-2** achieves the lowest Abs Rel and highest $\delta < 1.25$ across all three datasets, demonstrating consistent advantages in local geometry reconstruction. On Waymo, it also improves global Acc and Comp over DVGT with comparable ego-pose accuracy, while global reconstruction and pose estimation remain behind DVGT on DDAD and KITTI. Our model prioritizes local geometry and relative ego-poses for ego-centric planning, whereas global evaluation requires accumulation of predicted relative ego-poses, which can propagate errors into both global pose and point reconstruction. The relatively small training shares of DDAD and KITTI, together with differences in their ego-motion distributions, may further contribute to these performance gaps.

Table A.5: **Cache window size ablation.** W denotes the number of cached historical frames, excluding the current frame. The default setting ($W = 3$) is in bold.

Window size W	Geometry Acc ↓	Latency (ms) ↓	Memory (GB) ↓	FLOPs (T) ↓
1	0.433	263	14.4	5.72
3	**0.376**	**278**	**16.8**	**5.72**
5	0.369	286	19.6	5.73
7	0.369	294	22.2	5.73
9	0.369	303	24.8	5.74
11	0.369	303	27.4	5.74

Table A.6: **Long-sequence evaluation.** Each of the 200 sequences exceeds 200 frames and 100 s.

Frame index	Local geometry		Planning	Efficiency		
	Acc ↓	Abs Rel ↓	L1 (4 s) ↓	Latency (ms) ↓	Memory (GB) ↓	FLOPs (T) ↓
40	0.370	0.043	1.51	267	**16.8**	**5.72**
80	0.362	0.040	1.54	270	**16.8**	**5.72**
120	**0.353**	**0.039**	**1.31**	267	**16.8**	**5.72**
160	0.401	0.046	1.67	267	**16.8**	**5.72**
200	0.363	0.044	1.66	**266**	**16.8**	**5.72**

Table A.7: **L2 error of trajectory planning across five datasets.**

Dataset	L2 @ 1 s (m) ↓
nuScenes	0.56
OpenScene	0.20
Waymo	0.78
KITTI	2.12
DDAD	2.00

D.2 Cache Window Size Ablation

Table A.5 evaluates the accuracy–efficiency trade-off across cache window sizes during streaming inference. A larger window provides more temporal context for inter-frame geometry modeling, reducing global Acc from 0.433 at $W = 1$ to 0.376 at $W = 3$. Increasing the window to $W = 5$ yields a smaller improvement to 0.369, with no further gains at larger window sizes. Meanwhile, retaining and attending to more historical features increases memory usage and generally raises latency, while total FLOPs change only slightly. We therefore use $W = 3$ to capture most of the geometric benefit while maintaining a compact cache and low inference cost.

D.3 Long Sequences.

We evaluate whether geometry and planning performance can be maintained during long-sequence online inference. Across 200 sequences exceeding 200 frames, Table A.6 shows stable local geometry and planning performance at successive frame indices. Together with the bounded computational and memory costs in Figure 5, these results support sustained long-sequence online inference with consistent prediction quality and efficiency.

D.4 Planning on All Datasets

Table A.7 evaluates the jointly pretrained **DVGT-2** on the validation sets of all five datasets, without dataset-specific fine-tuning. We report the L2 displacement error at the 1-second future timestep. The model achieves errors of 0.20 m on OpenScene, 0.56 m on nuScenes, and 0.78 m on Waymo, while errors are higher on KITTI and DDAD, at 2.12 m and 2.00 m, respectively. These differences may partly reflect the OpenScene-dominated training mixture and variations in ego-motion distributions across datasets. We therefore fine-tune **DVGT-2** separately on NAVSIM and nuScenes for planning evaluation, validating its use as an effective and efficient geometry-supervised foundation for downstream planning.

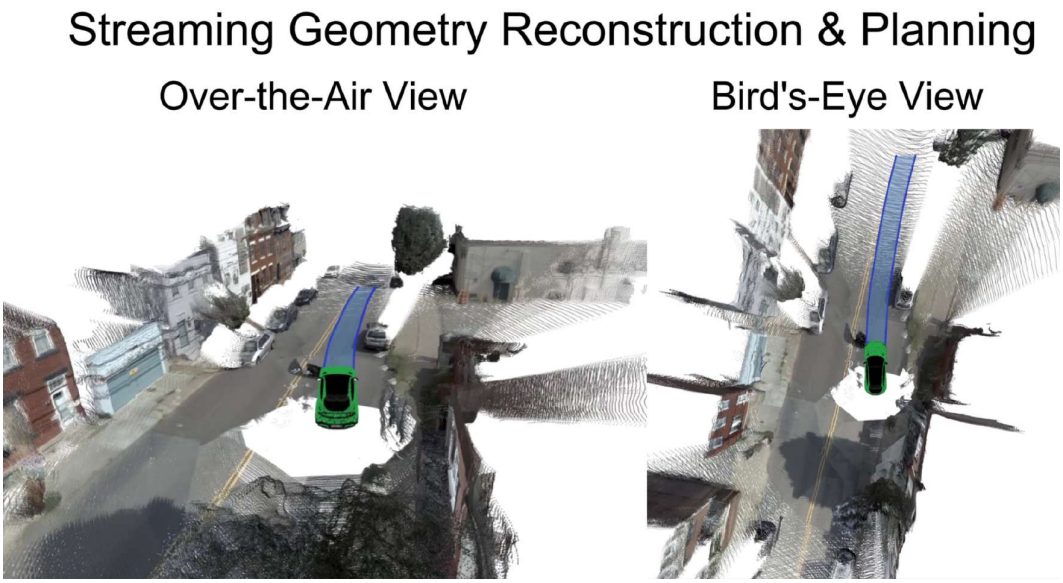

Figure A.1: **Video demonstration** of the geometry reconstruction and trajectory planning based on online multi-view inputs on the validation set.

E VIDEO DEMONSTRATION

We provide a supplementary video demonstrating online geometry reconstruction and trajectory planning on validation sequences. The video shows how the reconstructed geometry and planned trajectory are updated as new multi-view frames arrive. Figure A.1 presents a representative frame from the video.