GEOACT: QUERYABLE GEOMETRIC REPRESENTATIONS FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Driving stacks need scene geometry that is metric at inference time and that downstream modules can read without re-running a reconstruction. We present GeoAct, a calibration-aware encoder that turns multi-camera image histories into one shared geometric memory. The memory is read in two ways: densely, by a DPT-style head predicting per-pixel metric depth and point maps, and sparsely, by a point query asking where the surface seen at a pixel and time lies at a requested time. The encoder has about 10^8 parameters and is trained from projected LiDAR and monocular pseudo-labels, without distilling a large 3D model. On the NAVSIM/OpenScene `test` split it reaches 0.1093 AbsRel with no test-time scale alignment, more accurate than GeoX even with a per-scene oracle scale. Behind a frozen encoder, a trained adapter and DrivoR-based planner reach 91.78 PDMS, above a frozen GeoX memory under the same planner budget (90.83). Controlled continuations from one checkpoint reveal a trade-off: dropping dense supervision improves observed-motion prediction and degrades current-frame geometry. Under our frozen-transfer protocol, these two variants show no detectable planning difference.

1 INTRODUCTION

Driving systems need a scene representation that is simultaneously *metric* (usable in meters without test-time calibration tricks), *temporal* (aware of what moved), and *queryable* (readable by downstream modules with heterogeneous interfaces without re-running a reconstruction). Feed-forward 3D models (Wang et al., 2024; 2025a) and monocular geometry models (Yang et al., 2024; Wang et al., 2025b;c) have made unposed reconstruction remarkably strong, but they are typically evaluated up to an unknown scale and on single or few frames, whereas a driving stack already owns camera intrinsics, extrinsics, and ego odometry. Conversely, planning-oriented driving stacks (Hu et al., 2023; Li et al., 2022) build task-specific bird's-eye-view features whose geometric fidelity is rarely measured directly.

Recent driving geometry models, including DVGT-2 (Zuo et al., 2026) and GeoX (Lian et al., 2026), reconstruct dense scene geometry from visual inputs. GeoAct builds on this dense-reconstruction paradigm by adding point-query geometric supervision to a shared encoder. In addition to reconstructing per-view geometry, the representation is trained to locate a surface point specified by an image pixel at a requested time. This introduces explicit point-level geometric and temporal targets without replacing the dense supervision or requiring either geometric head in the planning forward pass.

This paper asks a narrow but, we argue, decision-relevant question:

> *If we build one calibration-aware memory that answers both dense and sparse geometric queries, how do dense and sparse supervision interact, and how much of the resulting geometric quality actually reaches a planner?*

Our contributions are deliberately bounded.

1. **Dense geometry learning augmented with point queries.** A 12-stage self/temporal/cross-view encoder over DINOv3-S/16 tokens produces one geometric

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.

memory M read either by a DPT-style dense geometry head or by an 8-layer point query head. Query positions and displacements use the fixed `ego_0` frame of the last observed frame, so static points have zero ground-truth motion; dense depth and XYZ instead use each view's camera frame (Section 3). The encoder is small — about 10^8 parameters, comparable to GeoX (155M) and an order of magnitude below VGGT-Omega (1B) and DVGT-2 (1.8B) — and is trained from projected LiDAR and monocular pseudo-labels rather than by distilling one of those models.

2. **A controlled measurement of the supervision trade-off.** From one shared initialisation checkpoint, matching token list, optimizer reset, schedule and sample budget, we continue with both branches (N1), the query branch only (D0), or the dense branch only (D1). Removing the dense branch improves moving-point prediction, increasingly with displacement, and reliably degrades current-frame geometry; removing the query branch leaves current-frame geometry essentially unchanged (Section 4.4).
3. **A transfer study with an explicit negative result.** Behind a frozen encoder and an identical adapter and planner recipe, both GeoAct memories score above a frozen GeoX memory trained with the same planner budget (PDMS 91.78 and 91.87 versus 90.83, both gaps excluding zero under a Bonferroni-corrected log-clustered bootstrap), though that reference remains unmatched in input budget. Within our own matched arms, the same bootstrap over 12146 tokens does *not* detect a difference between the geometrically better N1 and the geometrically worse D0 (Section 4.5). We also show that lower depth error does not buy higher edge-contrast fidelity (Section 4.3).

2 Related Work

Feed-forward geometry from images. DUSt3R (Wang et al., 2024) and VGGT (Wang et al., 2025a) regress point maps directly from unposed images, and monocular models (Ranftl et al., 2022; Yang et al., 2024; Wang et al., 2025b) produce affine- or scale-invariant geometry from a single view, while MoGe-2 (Wang et al., 2025c) additionally recovers metric scale. Unposed reconstructions are usually scored after an alignment step that consumes ground truth from the evaluated scene. GeoAct occupies a different point in the design space: because a driving platform provides intrinsics, extrinsics and odometry, we inject them as Plücker rays and predict metric coordinates directly, which removes the alignment step but also means our numbers are *not* comparable to unposed methods at equal information. We make that asymmetry explicit in every table. We use MoGe-2 as an offline dense teacher and projected LiDAR as metric supervision, and adopt the DPT decoder topology (Ranftl et al., 2021) for the dense read-out. Self-supervised visual pretraining (Oquab et al., 2024; Siméoni et al., 2025) provides the image-tokenizer initialization; GeoAct uses DINOv3-S/16.

Point tracking and 4D correspondence. Point-level tracking (Doersch et al., 2023; Karaev et al., 2024) and its 3D benchmark (Koppula et al., 2024) established the "query a pixel, get its trajectory" interface. D4RT (Zhang et al., 2026) moves this interface into dynamic 3D reconstruction, answering independent space–time point queries instead of producing a single dense reconstruction. Our point query head takes its design directly from D4RT, so *we do not claim the queryable-3D-point interface itself as a contribution.* What differs is the setting rather than the output type: queries carry a source camera identity and Plücker ray from *known* multi-camera calibration, targets live in a fixed ego frame shared across cameras and times, supervision is constructed from LiDAR surface points and annotated rigid box motion in driving scenes, and the same encoder is simultaneously supervised densely and then transferred to planning. Our contribution is that surrounding construction and what it transfers.

Geometric models for driving. End-to-end driving stacks (Hu et al., 2023; Chitta et al., 2023; Li et al., 2022) learn task-specific features whose geometry is rarely measured. A recent line instead makes geometry the representation: DVGT-2 (Zuo et al., 2026) and GeoX (Lian et al., 2026) train a visual-geometry backbone with dense reconstruction targets and then drive with it on NAVSIM. GeoAct follows this line and asks what an added point-query objective changes. Because the NAVSIM PDMS protocol (Dauner et al., 2024; Caesar et al., 2021) lets us freeze the encoder and fix the planner, we can compare geometric memories directly; our negative result in Section 4.5 is specific to that protocol.

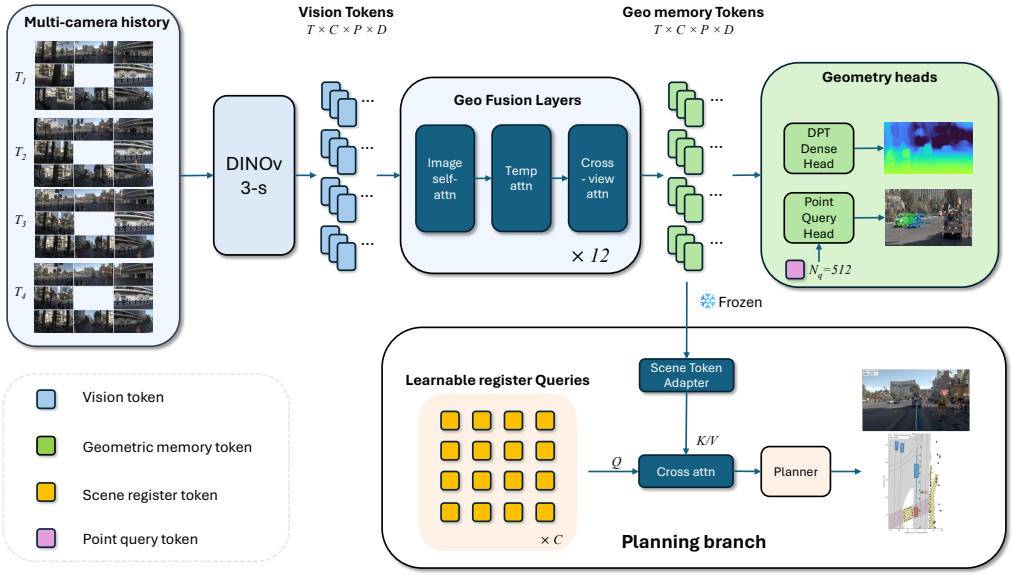


Figure 1: **Overview of GeoAct.** A calibrated multi-camera history is encoded into a shared geometric memory M. The dense head reads intermediate encoder stages; the point query head reads M. Both train the encoder. For planning, the encoder is frozen and a scene-token adapter compresses M into 128 tokens for the planner; neither geometric head runs on the planning path.

3 METHODOLOGY

GeoAct (Figure 1) trains one encoder with two geometric read-outs — a dense head and a point query head — and then freezes it and trains a planner on its memory. Section 3.1 describes the encoder, Section 3.2 the two read-outs and their supervision, and Section 3.3 planning. Parameterisation details that do not affect the argument are in Appendix A.2.

3.1 SHARED ENCODER

The input is $T = 4$ frames from $C = 8$ cameras at 288×512. Each image is split into $P = 576$ patches, and every patch token combines appearance with where it was seen from:

$$z_{t,c,p} = \phi(I_{t,c})_p + e_{\text{pos}}(p) + e_{\text{cam}}(c) + e_{\text{time}}(\tau_t) + e_{\text{ray}}(\ell_{t,c,p}), \tag{1}$$

where ϕ is DINOv3-S/16 (Siméoni et al., 2025) and $\ell_{t,c,p}$ is the Plücker ray of the patch centre, computed from known intrinsics, extrinsics and history ego poses in the ego frame of the latest observation (`ego_0`). All terms have width $D = 384$. A 12-stage encoder $\mathcal{E}$ alternates three attentions per stage — within an image, across the T frames at the same patch, and across the other cameras at the same time — and outputs

$$M,\ \{F^{(l)}\}_{l=1}^{12} = \mathcal{E}(z), \qquad M \in \mathbb{R}^{TCP \times D}, \tag{2}$$

where M is the final *geometric memory* (a set of tokens, not a point cloud) and $F^{(l)}$ are intermediate stage features. No future image enters the encoder.

3.2 TWO GEOMETRIC READ-OUTS

Dense head. A DPT-style decoder $\mathcal{D}$ (Ranftl et al., 2021) reads four intermediate stages and predicts, for every view, a depth map and a point map in that camera's frame:

$$\big(\widehat{D}_{t,c},\ \widehat{P}_{t,c}\big) = \mathcal{D}\big(\{F^{(l)}_{t,c}\}_{l\in\{3,6,9,12\}}\big). \tag{3}$$

Supervision combines two sources. Projected LiDAR is measured and metric but covers about 10% of pixels. MoGe-2 (Wang et al., 2025c) gives a dense prediction everywhere; we re-anchor it offline

to each training image's LiDAR with one scale and shift, so both sources share LiDAR's scale and no test data is used. The dense loss is

$$\mathcal{L}_{\text{dense}} = \mathcal{L}_{\text{teacher}} + \beta_{\text{lidar}}\mathcal{L}_{\text{lidar}} + \beta_{\text{grad}}\mathcal{L}_{\nabla \log d} + \beta_{\text{normal}}\mathcal{L}_{\text{normal}}, \tag{4}$$

i.e. teacher fitting, LiDAR fitting, and two structural terms (log-depth gradients and surface normals). Only the teacher term is down-weighted by distance and by a learned uncertainty.

Point query head. Following D4RT (Zhang et al., 2026), a query asks: *where is the surface seen by camera c at pixel (u, v) and time τ_s, at time τ_q?* We write $q = (c, u, v, \tau_s, \tau_q)$; the query is also given the pixel's Plücker ray and a small RGB patch. An 8-layer decoder $\mathcal{Q}$ cross-attends from the query to M and predicts a metric position and a displacement in `ego_0`:

$$\big(\widehat{x}_q,\ \widehat{\Delta x}_q\big) = \mathcal{Q}(q;\ M). \tag{5}$$

Queries never attend to each other, so each answer is independent of which other queries are asked. $\tau_q = \tau_s$ asks for current geometry and another observed τ_q asks for tracking; training with $\tau_q > 0$ gives the future variant of Appendix A.5.

Query targets. Queries are anchored on real LiDAR points x_s^*. A point on a moving object is carried along by its annotated box $B(\tau)$; a static point stays put:

$$x_q^* = B(\tau_q)\, B(\tau_s)^{-1}\, x_s^*, \qquad \Delta x_q^* = x_q^* - x_s^*. \tag{6}$$

Box-rigid motion is an approximation for pedestrians and cyclists. Boxes and track identities build targets only and never enter the model. The query loss fits position and displacement and asks the two to agree:

$$\mathcal{L}_{\text{query}} = \ell\big(\widehat{x}_q, x_q^*\big) + \lambda_\Delta\, \ell\big(\widehat{\Delta x}_q, \Delta x_q^*\big) + \lambda_c\, \rho\big(\widehat{x}_s + \widehat{\Delta x}_q - \widehat{x}_q\big), \tag{7}$$

where ρ is smooth-L1, ℓ is ρ with learned uncertainty weighting (Kendall & Gal, 2017), and $\widehat{x}_s$ is the answer to the same pixel queried at τ_s.

Joint objective. Both heads train the same encoder:

$$\mathcal{L}_{\text{geom}} = \mathcal{L}_{\text{query}} + \lambda_d\mathcal{L}_{\text{dense}} + \lambda_{qd}\mathcal{L}_{\text{query-dense}} + \lambda_{rp}\mathcal{L}_{\text{reproj}}. \tag{8}$$

The two small coupling terms make the query answer agree with the dense point map at the same pixel, and make a predicted point reproject onto its pixel in another observed view (Appendix A.1). Removing the dense head (D0) removes $\mathcal{L}_{\text{dense}}$ and $\mathcal{L}_{\text{query-dense}}$; removing the query head (D1) removes the query terms.

3.3 Planning on the Frozen Memory

For planning the encoder is frozen and neither geometric head is run. A scene-token adapter $\mathcal{A}$ gives each camera 16 learnable queries that cross-attend to that camera's tokens in M, yielding 128 scene tokens S. The DrivoR planner (Kirby et al., 2026) proposes K trajectories from S and the ego state e, scores them, and returns the best:

$$S = \mathcal{A}(M) \in \mathbb{R}^{128\times 256}, \qquad \widehat{\mathcal{T}} = \arg\max_{\mathcal{T}_k \in \mathcal{P}(S,e)} \mathcal{S}\big(S, e, \mathcal{T}_k\big). \tag{9}$$

Only $\mathcal{A}$, $\mathcal{P}$ and $\mathcal{S}$ are trained, with the backend's own imitation and scoring losses and one recipe for every encoder.

4 Experiments

4.1 Setup

Training. N1 (dense + query), D0 (query only) and D1 (dense only) continue from one dual-head warm-up checkpoint, trained from DINOv3 on about a fifth of the NAVSIM training corpus. Each arm then trains 12000 further steps at global batch 64 on the same frozen list of 17004 scenes; the arms differ only in which heads remain. For planning every encoder is frozen, and the adapter and planner are trained with one recipe (20 epochs, global batch 256, final-epoch weights). Configurations are in Appendix A.1.

Table 1: **Planning on NAVSIM v1 NAVTEST, non-reactive evaluation.** 0–100 scale, higher is better; C and L denote camera and LiDAR inputs at inference. Literature rows are transcribed from Table 6 of LiAuto-GeoX (Lian et al., 2026) (arXiv:2606.05774v2), with citations to the original papers; they are not our reproductions. GeoAct uses the frozen N1 encoder with a trainable scene-token adapter and DrivoR-based planner, on all 12146 official `navtest` tokens with zero failures. NC: no-at-fault collision; DAC: drivable-area compliance; TTC: time-to-collision; Comf.: comfort; EP: ego progress. **Bold**: best per column; comfort saturates and is not bolded.

Method	Input	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Traditional end-to-end methods							
UniAD (Hu et al., 2023)	C	97.8	91.9	92.9	100	78.8	83.4
TransFuser (Prakash et al., 2021)	C&L	97.7	92.8	92.8	100	79.2	84.0
Hydra-MDP (Li et al., 2024)	C&L	98.3	96.0	94.6	100	78.7	86.5
GoalFlow (Xing et al., 2025)	C&L	98.3	93.8	94.3	100	79.8	85.7
ARTEMIS (Feng et al., 2025)	C&L	98.3	95.1	94.3	100	81.4	87.0
DiffusionDrive (Liao et al., 2025)	C&L	98.2	96.2	94.7	100	82.2	88.1
VADv2 (Chen et al., 2026)	C	98.3	97.4	95.7	100	82.3	89.3
DriveSuprim (Yao et al., 2026)	C&L	97.8	97.3	93.6	100	86.7	89.9
Vision-language-action methods							
AdaThinkDrive (Luo et al., 2025)	C	98.5	94.4	94.9	100	79.9	86.2
AutoVLA (Zhou et al., 2025)	C	98.4	95.6	**98.0**	99.9	85.9	89.1
ReCogDrive (Li et al., 2026b)	C	98.2	97.8	95.2	99.8	83.5	89.6
DriveVLA-W0 (Li et al., 2026a)	C	98.7	**99.1**	95.3	99.3	83.3	90.2
World-model methods							
DrivingGPT (Chen et al., 2025)	C	98.9	90.7	94.9	95.6	79.7	82.4
LAW (Li et al., 2025a)	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
WoTE (Li et al., 2025b)	C&L	98.5	96.8	94.9	99.9	81.9	88.3
PWM (Zhao et al., 2025)	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
Geometric-reconstruction methods							
DVGT-2 (Zuo et al., 2026)	C	97.8	97.2	93.9	100	83.4	88.6
DVGT-2-NAVSIM (Zuo et al., 2026)	C	98.7	97.9	95.8	100	84.3	90.3
LiAuto-GeoX (Lian et al., 2026)	C	98.6	97.5	95.2	100	85.8	90.6
Our method							
GeoAct (ours)	C	98.95	98.10	95.92	99.99	**87.36**	**91.78**

Evaluation. All quantitative main-text results are scored on the NAVSIM/OpenScene `test` split (136 logs). *Geometry* uses all 1961 non-overlapping scenes, camera `CAM_F0`, projected-LiDAR ground truth, Z-depth in $[1, 80]$ m and scene-macro means. No dynamic-object mask is applied, so we call this *current-frame* geometry. Query-head motion metrics use the same 1961 scenes. *Planning* uses all 12146 official `navtest` tokens with zero failures. Paired comparisons use a bootstrap over the 136 logs (20000 resamples). All arms use one training seed, so intervals reflect evaluation variability only; further protocol details are in Appendix A.7.

4.2 NAVSIM v1 planning benchmark

GeoAct reaches PDMS 91.78, above the strongest listed row (LiAuto-GeoX, 90.6), with most of the margin in ego progress. These are third-party numbers with different inputs, data and planners, so we treat them as context; our conclusions rest on the same-planner comparison of Table 4.

4.3 Metric geometry on the NAVSIM/OpenScene test split

Metric depth without test-time alignment. Without any test-time fit, GeoAct (0.1093 AbsRel, 1.257 m CD) is more accurate than GeoX even after GeoX receives a per-scene oracle scale (0.1462, 1.729 m), and than DVGT-2 under leave-one-out scaling (0.1779); DVGT-2 also fails on $8/1961$ scenes by emitting negative depth. VGGT-Omega, an RGB-only model about ten times larger, is not metric raw (0.9496) but reaches the best AbsRel (0.1010) once given an oracle scale; GeoAct keeps

Table 2: **Depth and point-cloud reconstruction, NAVSIM/OpenScene `test`.** Protocol: 1961 scenes / 136 logs, CAM_F0, projected LiDAR, Z-depth $[1, 80]$ m, scene-macro means. N is the number of scored scenes. RMSE, Acc, Comp and Chamfer (CD = Acc + Comp) are in meters. "None" means no test-GT alignment; LOO and oracle scaling consume test GT and are diagnostics, not deployment results. All rows are our evaluations, not original-paper scores, and inputs differ across models.

Method	Scale	N	Depth			Point cloud			
			AbsRel↓	$\delta_{1.25}$ ↑	RMSE↓	Acc↓	Comp↓	CD↓	AUC-F↑
External reference models (evaluated locally)									
VGGT-Omega	none	1961	0.9496	0.0000	25.866	—	—	—	—
VGGT-Omega	oracle	1961	**0.1010**	**0.8945**	7.115	—	—	—	—
GeoX	none	1961	0.9572	0.0000	—	—	—	—	—
GeoX	LOO	1961	0.2808	0.5335	10.491	1.5371	1.2736	2.8107	0.4361
GeoX	oracle	1961	0.1462	0.8595	8.589	1.0019	0.7268	1.7287	0.6756
DVGT-2	none	1953	0.3536	0.4322	10.489	1.7802	1.5201	3.3003	—
DVGT-2	LOO	1953	0.1779	0.7793	7.472	0.9209	1.1768	2.0977	0.6058
DVGT-2	oracle	1953	0.1383	0.8435	6.975	0.7798	0.7835	1.5633	0.6820
GeoAct dense read-out: main model and matched continuations									
GeoAct N1	none	1961	0.1093	0.8836	6.249	0.6859	0.5713	**1.2573**	**0.7163**
GeoAct D1	none	1961	0.1091	0.8835	6.283	0.6924	**0.5693**	1.2617	0.7158
GeoAct S0	none	1961	0.1091	0.8838	**6.243**	**0.6840**	0.5741	1.2581	0.7157

References: VGGT-Omega (Wang et al., 2026), GeoX (Lian et al., 2026), DVGT-2 (Zuo et al., 2026). GeoAct and GeoX receive calibration; DVGT-2 estimates poses; VGGT-Omega-1B-512 receives RGB only, on our 4×8-view 288×512 input. D1 drops the query branch; S0 drops the structural losses (Appendix A.4). Bold is best per column over all rows, so bold oracle values need test GT. DVGT-2 fails on 8 scenes; "—" is unavailable. Point clouds are unprojected from each model's depth.

Table 3: Controlled continuation from one shared checkpoint, all on the 1961 `test` scenes. Left block: query-head point metrics. Right column: query-head AbsRel. $I_{\text{zero}} = 1 - \overline{\text{EPE}}_{\text{model}}/\overline{\text{EPE}}_{\text{zero}}$ (means over all moving queries), where the zero-motion baseline keeps the *ground-truth* source position; it is a diagnostic, not a deployable comparison.

	CurrentXYZ EPE (m)↓	moving EPE (m)↓	I_{zero} ↑	Test AbsRel↓
N1 (dense + query)	**3.208**	5.565	0.025	**0.1125**
D0 (query only)	3.276	**5.236**	**0.082**	0.1156

the best RMSE and point-cloud metrics, unaligned. Our claim is therefore competitive geometry *in metres without any test-time fit*, not the best shape. The query read-out of the same checkpoints reaches 0.1125 (N1) and 0.1156 (D0) on a different depth support, so it is only compared within the query head.

Lower AbsRel does not imply sharper edges. We split LiDAR-supported pixels into *edge* (local depth ratio > 1.25) and *smooth* (≤ 1.05) cells (Appendix A.4). Unscaled GeoAct has the lowest AbsRel on both, even against oracle-scaled references: smooth 0.0795 versus 0.1100 (GeoX) and 0.1096 (DVGT-2), edge 0.2518 versus 0.2991 and 0.2772. Yet all three models *attenuate* depth discontinuities alike: contrast fidelity, the median ratio of predicted to true log-depth contrast on edge cells (1.0 is exact), is 0.099, 0.094 and 0.105. Predicted jumps are about a tenth of their true magnitude, whichever model has the lowest depth error.

4.4 The dense/sparse supervision trade-off

N1 and D0 differ only in the dense-branch switch, and both are scored on the same 402k current and 367k moving-point queries. Removing the dense branch *improves* motion: moving-point EPE falls from 5.565 to 5.236 m (-5.9%) and I_{zero} rises from 0.025 to 0.082 (Figure 3); the gain grows with displacement, from -5.1% at 1–3 m to -6.8% above 10 m. The same removal *degrades*

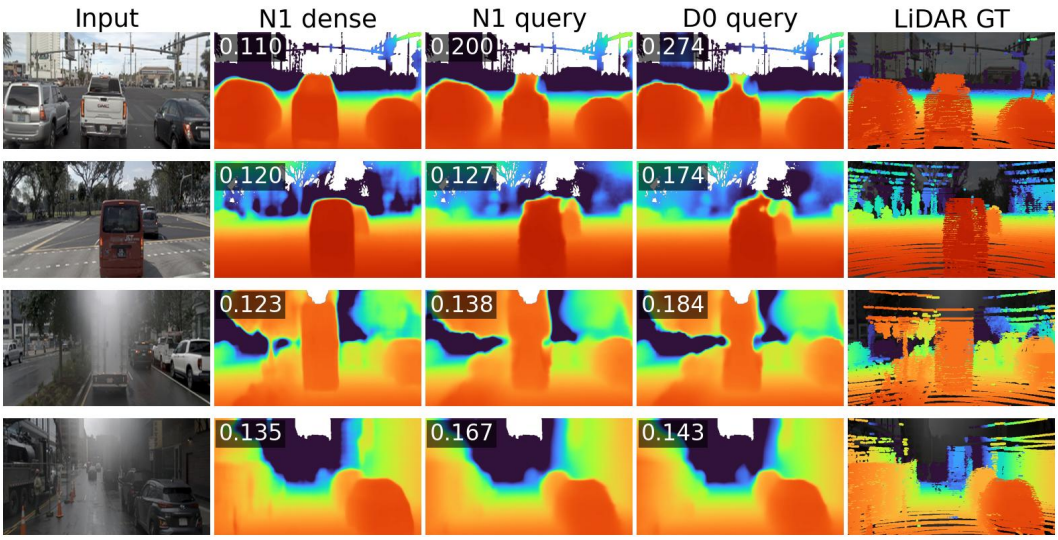


Figure 2: **Dense and query read-outs on `test` scenes (CAM_F0).** Columns: input, the N1 dense (DPT) head, the query heads of N1 and D0, and projected LiDAR on the same colour scale. Query heads are read by querying every fourth pixel at the current time and converting the answer to camera Z-depth. Pixels outside MoGe-2's native-validity mask (mostly sky) are blanked for every model alike. Inset numbers are per-scene AbsRel against projected LiDAR. The dense head gives the sharpest object boundaries and the lowest error in all four scenes (0.110–0.135). Part of the sharpness gap comes from the 4-pixel query grid, so it is not a pure property of the read-out. Scenes are selected; aggregate numbers are in Tables 2 and 3.

current-frame geometry: CurrentXYZ EPE rises from 3.208 to 3.276 m and query-head AbsRel from 0.1125 to 0.1156 (paired CI $[+0.0025, +0.0036]$; Figure 2). This ablates a supervision *package* — dense regression and the query–dense coupling — and D0 inherits dense-supervised history from the warm-up checkpoint, so any mechanism remains a hypothesis. The mirror ablation D1 leaves dense geometry essentially unchanged (0.10910 versus 0.10933 for N1, a 0.21% relative effect). We keep N1 as the main model because it is the only arm that is strong on both axes at once: it matches D1 on dense geometry, beats D0 on query-head geometry, and is statistically indistinguishable from the best arm on PDMS (Section 4.5). It is a balanced choice across geometry and planning, not the per-metric winner on either.

4.5 Representation comparisons under a shared planner

All rows in Table 4 use the same planner architecture and recipe on a frozen encoder; the encoder and its input budget differ.

GeoAct versus GeoX. Both GeoAct memories beat the frozen GeoX memory under the same planner budget (+0.95 and +1.04 PDMS; log-clustered paired intervals $[+0.35, +1.59]$ and $[+0.40, +1.74]$ after Bonferroni correction over seven metrics), with the largest increase in ego progress. GeoX is also geometrically pre-trained but reads 4 cameras $\times$ 1 frame with 64 registers, so the gap measures representation and input budget jointly. N1 gains EP without losing NC or TTC. Within block B the mean EP order is D0 $>$ N1 $>$ D1, but none of these differences is significant. The GeoX arm is trained with the same budget and final-epoch rule as ours (Appendix A.4).

Better geometry does not give a higher PDMS here. N1 is reliably better than D0 on current-frame geometry, yet scores 91.78 versus 91.87 ($\Delta = -0.091$, 95% CI $[-0.453, +0.253]$). D1 reaches 91.32, and no difference within block B survives correction. This is a failure to detect a difference, not a proof of equivalence, and N1 and D0 differ in two abilities in opposite directions

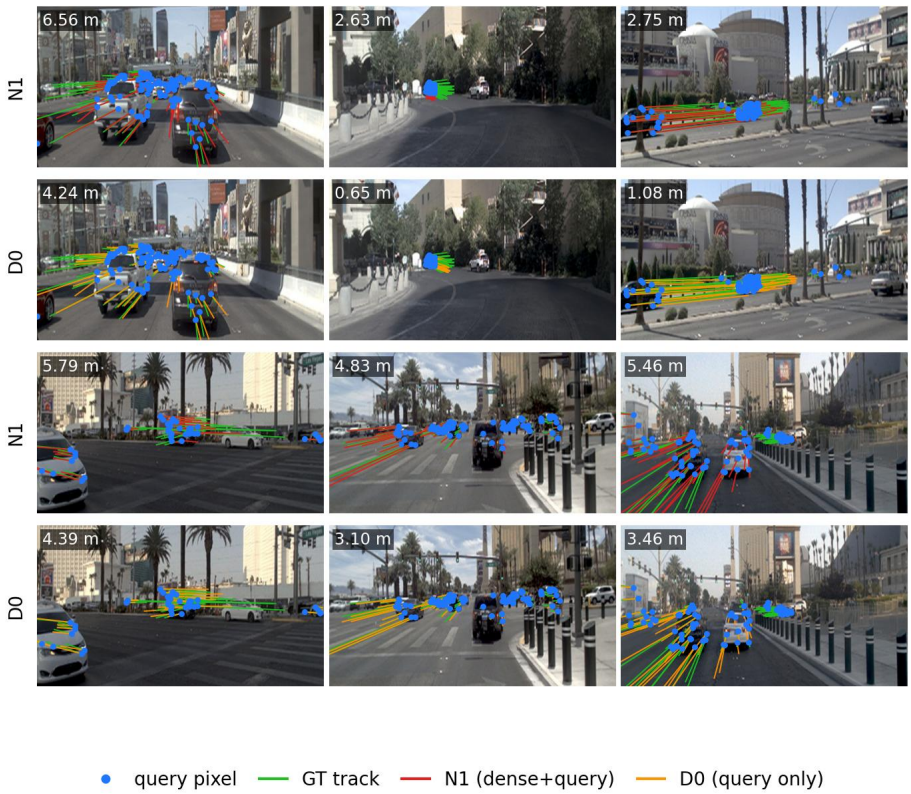


Figure 3: **Observed-motion tracking of moving points.** Six scenes in two blocks of three columns; in each block the upper row is N1 and the lower row D0, on the same queries. Each query pixel (blue) asks where its surface point was in an earlier observed frame; green is the ground-truth track, obtained by transporting the LiDAR point with its annotated box (Eq. 6); red and orange are the N1 and D0 predictions, with median 3D EPE inset. Scenes come from training-pool logs excluded from training by a log-level split, and are six of the twelve with the largest N1–D0 gap among 97 such scenes; aggregates on `test` are in Table 3.

rather than in geometry alone. The two advantages may offset inside the planner, the planner may be insensitive to geometry at this accuracy, or the 128-token adapter may bottleneck both; separating these needs an arm that improves geometry without hurting motion and an adapter-capacity control.

5 Discussion and Limitations

Information budget and labels. GeoAct consumes intrinsics, extrinsics and history ego poses. This is realistic for a vehicle and is why we report metric numbers without test-time alignment, but it makes every comparison with unposed models a *protocol* comparison; we never rank a GeoAct metric row against an oracle-scaled row without saying so. Dynamic query targets come from rigid transport of LiDAR points by annotated boxes: vehicles are well served, pedestrians and cyclists are approximations, and none of it is dense non-rigid 4D ground truth.

Statistics and scope. All results use one training seed, so bootstrap intervals capture evaluation variability only. The shared `test` data were inspected repeatedly during development, which makes borderline intervals weaker than nominal. Section 4.5 shows no detectable PDMS difference between two memories differing measurably in current-frame geometry, under one transfer protocol, planner, seed and non-reactive benchmark; it does not license the general claim that geometric accuracy is irrelevant for driving.

Table 4: **Representation comparisons and supervision ablations under a shared planner.** All rows use the same 12146 official `navtest` metric-cache tokens with zero failures, on the 0–100 scale of Table 1, and share planner architecture, loss recipe, training split and checkpoint-selection rule. **Bold marks the best value per column**; comfort is identical to two decimals and is not bolded. N1 is our main model but does not attain the best PDMS — D0 does, by 0.09 points. *None of the differences within block B survives our multiple-comparison correction*; the N1- and D0-versus-GeoX PDMS gaps do.

Encoder / supervision	NC↑	DAC↑	EP↑	TTC↑	Comf.↑	DDC↑	**PDMS↑**
A. Reference encoder							
GeoX	98.76	97.93	85.41	**95.92**	99.99	96.95	90.83
B. GeoAct supervision ablations							
Dense only (D1)	98.81	98.02	86.87	95.51	99.99	96.95	91.32
Query only (D0)	98.76	**98.29**	**87.53**	95.69	99.99	96.91	**91.87**
Dense + Query (N1)	**98.95**	98.10	87.36	**95.92**	99.99	**97.01**	91.78

All rows use the planner of Section 3.3 with independently trained weights. GeoX uses 4 cameras $\times$ 1 frame and 64 scene registers, whereas GeoAct uses 8×4 frames and 128 registers: the readout design is shared, the input budgets are not. In block B, Dense and Query name the supervision branch, and all three variants share a dual-head initialisation. DDC is driving-direction compliance and has zero weight in the v1 scorer. For reference, the original DrivoR reports 93.1 PDMS with its visual encoder trained end-to-end for 25 epochs; every row here keeps the encoder frozen and trains the planner for 20 epochs, so the two numbers are not comparable.

Missing controls. The comparison with GeoX is not input-matched, all planning results keep the encoder frozen, and we have no adapter-capacity study, so we cannot say whether a jointly trained planner or a wider adapter would expose geometry differences that the current interface hides. The future-prediction variant (Appendix A.5) lacks an information-matched constant-velocity baseline.

6 Conclusion

We built one calibration-aware spatio-temporal memory answering both dense and sparse metric geometry queries, measured how the two supervisions trade off under a controlled continuation, and tested how far the resulting geometry travels into a planner behind a frozen interface. The geometry is strong under its protocol, the trade-off is measured on the official test split, and the transfer result is mixed in an instructive way: frozen GeoAct memories score above a frozen GeoX memory under the same planner budget, though that reference still differs in input budget, while between our own matched arms — which differ in current-frame geometry and in motion prediction, in opposite directions — no planning difference is detectable. We also find that lower depth error does not buy higher edge-contrast fidelity, which suggests the field's default metric leaves a relevant axis unexamined. Input-matched baselines, adapter-capacity controls and repeated seeds are needed to turn these observations into causal attributions.

AI use statement

Generative AI tools were used in three ways. First, for code assistance during implementation of training, evaluation and analysis scripts; all such code was reviewed, executed and tested by the authors, and no result in this paper comes from code that was not run. Second, for literature-search assistance and for editing the manuscript's English; the authors are responsible for all claims and citations, and checked each reference against its source or, where a field could not be confirmed, flagged it internally rather than asserting it. Third, for drafting and revising prose in this manuscript under author direction. Generative AI was not used to generate, impute or adjust any experimental number. Every reported value was produced by a logged experiment run and independently re-read from the experiment ledger and its stored metric artifacts.

Ethics statement

This work uses publicly available driving datasets under their respective licences and involves no human subjects, no personally identifying annotation and no new data collection. The driving scenes contain recordings of public roads made by the dataset providers, who are responsible for the privacy processing applied to them. Our planning evaluation is non-reactive: the simulated environment does not respond to the ego trajectory, so the reported PDMS values measure agreement with a scoring function on logged scenarios and are *not* a safety certification. Nothing in this paper should be read as evidence that the described system is safe to deploy. The geometric read-outs are trained on and evaluated against sparse LiDAR and pseudo-labels, both of which carry the biases of the environments in which the data were collected.

Reproducibility statement

Section 3 specifies the architecture and objectives, and Appendix A.1 gives the exact configuration of every reported variant, including loss weights, query mixture, optimiser settings and the shared initialisation checkpoint from which the three supervision continuations branch. Appendix A.3 states the standard block definitions omitted from the main text. Evaluation protocols — scene lists, depth support, scale-alignment conventions, bootstrap procedure and failure accounting — are in Section 4.1 and Appendices A.4–A.7. The anonymised source drop accompanying this submission contains one training config per reported variant, named as in the paper, together with the training and evaluation entry points. Every training run is identified by a source-archive SHA256 and an initialisation-checkpoint SHA256, and every evaluation by its frozen token list, so that a reported number can be traced to the exact code and weights that produced it. Results in this paper use a single training seed; we note where that limits the strength of a comparison.

References

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *CVPR Workshop on Autonomous Driving*, 2021.

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. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2402.13243.

Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. DrivingGPT: Unifying Driving World Modeling and Planning with Multi-modal Autoregressive Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26890–26900, 2025. URL https://arxiv.org/abs/2412.18607.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

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. URL https://arxiv.org/abs/2504.19580.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *European Conference on Computer Vision (ECCV)*, 2024.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, Anh-Quan Cao, Nermin Samet, Tuan-Hung Vu, and Matthieu Cord. Driving on registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32058–32069, 2026. URL https://arxiv.org/abs/2601.05083.

Skanda Koppula, Ignacio Rocco, Yi Yang, Joe Heyward, João Carreira, Andrew Zisserman, Gabriel Brostow, and Carl Doersch. TAPVid-3D: A benchmark for tracking any point in 3D. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2407.05921.

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. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://arxiv.org/abs/2406.08481.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27137–27146, 2025b. URL https://arxiv.org/abs/2504.01941.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. DriveVLA-W0: World Models Amplify Data Scaling Law in Autonomous Driving. In *International Conference on Learning Representations (ICLR)*, 2026a. URL https://arxiv.org/abs/2510.12796.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. ReCogDrive: A Reinforced Cognitive Framework for End-to-End Autonomous Driving. In *International Conference on Learning Representations (ICLR)*, 2026b. URL https://arxiv.org/abs/2506.08052.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, Yu-Gang Jiang, and Jose M. Alvarez. Hydra-MDP: End-to-end Multimodal Planning with Multi-target Hydra-Distillation. *arXiv preprint arXiv:2406.06978*, 2024. URL https://arxiv.org/abs/2406.06978.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision (ECCV)*, 2022.

Jiawei Lian, Haoyi Sun, Yang Wu, Lifu Mu, Siyuan Wang, Tao Wei, Le Hui, Ning Mao, Pan Zhou, Kun Zhan, and Jian Yang. LiAuto-GeoX: Efficient grounded driving transformer. *arXiv:2606.05774*, 2026.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. DiffusionDrive: Truncated Diffusion Model for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047, 2025. URL https://arxiv.org/abs/2411.15139.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, Long Chen, Bing Wang, and Zhi-Xin Yang. AdaThinkDrive: Adaptive Thinking via Reinforcement Learning for Autonomous Driving. *arXiv preprint arXiv:2509.13769*, 2025. URL https://arxiv.org/abs/2509.13769.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-Modal Fusion Transformer for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7077–7087, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Prakash_Multi-Modal_Fusion_Transformer_for_End-to-End_Autonomous_Driving_CVPR_2021_paper.html.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. *arXiv preprint arXiv:2605.15195*, 2026. URL https://arxiv.org/abs/2605.15195.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5261–5271, 2025b. URL https://arxiv.org/abs/2410.19115.

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. In *Advances in Neural Information Processing Systems*, volume 38, 2025c. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/336572db3e99930814d6b328d4220cb6-Abstract-Conference.html.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3d vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. DriveLaW: Unifying Planning and Video Generation in a Latent Driving World. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39701–39712,

2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xia_DriveLaW_Unifying_Planning_and_Video_Generation_in_a_Latent_Driving_CVPR_2026_paper.html.

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. GoalFlow: Goal-Driven Flow Matching for Multimodal Trajectories Generation in End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1602–1611, 2025. URL https://arxiv.org/abs/2503.05689.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11910–11918, 2026. URL https://arxiv.org/abs/2506.06659.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one D4RT at a time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7382–7392, 2026. URL https://arxiv.org/abs/2512.08924.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, Xun Cao, and Wei Yin. Epona: Autoregressive Diffusion World Model for Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27220–27230, 2025. URL https://arxiv.org/abs/2506.24113.

Zhida Zhao, Talas Fu, Yifan Wang, Lijun Wang, and Huchuan Lu. From Forecasting to Planning: Policy World Model for Collaborative State-Action Prediction. In *Advances in Neural Information Processing Systems*, volume 38, pp. 134585–134611, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/c3b3c5f9297ad0012cfe7188c34cea0e-Abstract-Conference.html.

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. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2506.13757.

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.

Table 5: Main model variants (S0 differs from N1 only in the structural loss weights; Appendix A.4). The three NAVSIM continuations start from the same shared initialisation checkpoint, reset optimizer and scheduler, and use the same frozen 17004-scene NAVSIM token list, 12000 additional steps and global batch 64.

ID	read-outs	init	steps	corpus	queries/scene
N1	dense + query, +structural	shared init	12000	NAVSIM 17004	512
D0	query only	shared init	12000	NAVSIM 17004	512
D1	dense only	shared init	12000	NAVSIM 17004	—
W0	query only	scratch	40000	0.32 Waymo / 0.68 NAVSIM	1024
F1	query only, causal, $t_{\mathrm{tgt}} > 0$	W0	60000	0.32 / 0.68	1024

A Appendix

A.1 Model Variants and Implementation Details

We collect experiment-specific configurations here rather than making variants part of the Methodology. The three NAVSIM supervision continuations share the encoder architecture, corpus and schedule, and start from one common dual-head warm-up checkpoint trained from DINOv3 on a small NAVSIM subset. The mixed-corpus and future-prediction configurations are separate recipes, not single-factor ablations of the NAVSIM main model.

D0 removes the 70 dense-head tensors, dense losses and query–dense coupling. D1 removes the 134 query-head tensors and the associated query losses. Retained weights are loaded unchanged from the shared checkpoint at initialization; the encoder is subsequently optimized under each arm's supervision. The comparison therefore concerns continuation objectives, not models with no prior exposure to the removed supervision.

Geometry optimization and targets. The NAVSIM dual-head continuation freezes DINOv3 for the first 1000 optimization steps, then optimizes it at learning rate 10^{-5}; it is not frozen throughout geometry training. The requested current/static/moving query mixture is $0.4/0.2/0.4$. A track is classified as moving when its displacement across the observed window exceeds 1 m. Returns within 0.30 m of a moving box's floor are excluded from rigid transport, not removed from the static query pool.

In Eqs. (7)–(8), the main configuration uses $\lambda_{\Delta} = 0.5$, $\lambda_c = 0.1$, $\lambda_d = 4.0$, $\lambda_{qd} = 0.1$ and $\lambda_{rp} = 0.05$. Inside the dense objective, $\beta_{\mathrm{lidar}} = 2.0$, $\beta_{\mathrm{grad}} = 0.5$ and $\beta_{\mathrm{normal}} = 0.1$; the XYZ regression weight within the teacher and LiDAR terms is 1.0. Thus the outer dense weight also multiplies the LiDAR and structural terms. Dense teacher reliability is $w(d) = 1 - (1 - 0.2)\,\mathrm{clamp}((d - 2)/(80 - 2), 0, 1)$ on valid pixels and zero elsewhere. Query regression does not use this dense-teacher weight. Structural losses use reliable teacher support, and point-normal comparisons across depth discontinuities are excluded. Learned log-scales are clipped to $[-5, 5]$; the signed inverse-log XYZ activation is numerically bounded at 1000 m. Optimization uses AdamW (Loshchilov & Hutter, 2019).

Offline teacher alignment. The cached teacher depth is produced once, before training. For each image we take the MoGe-2 prediction $\hat{d}$ and the projected LiDAR depth d^*, keep pixels where both are finite and positive, subsample to at most 25000 such pixels, and fit a single scale and shift by the robust affine alignment of MoGe (Wang et al., 2025b;c), weighted by inverse target depth so that near-field LiDAR dominates the fit. MoGe-2 is metric in its own right; the fit re-anchors it to this sensor rig and absorbs residual domain bias, so that both supervision sources share one scale. The cached label is $\mathrm{clamp}(\alpha\hat{d} + \beta,\, 0,\, \infty)$, where the clamp removes the negative depths that a fitted shift can otherwise produce. Images with fewer than two jointly valid pixels fall back to the projected LiDAR itself. The fit is per image, uses only training-split data, and is never recomputed at evaluation time. Because one scale and shift are shared by the whole image, the procedure transfers LiDAR's metric scale to the teacher but does not correct local teacher errors, which is one reason the

LiDAR term is retained as a separate, unweighted supervision signal rather than being folded into the teacher label.

The two coupling terms. Eq. (8) contains two terms that link the heads and that are only described in words in the main text. Let q be a query with source camera c, source pixel (u, v) and source time τ_s, and let $\widehat{P}_{\tau_s,c}$ be the dense point map of that same view. Write $T_{\tau_s,c}$ for the known camera-to-`ego_0` transform. The query–dense term pulls the two read-outs of the same surface together,

$$\mathcal{L}_{\text{query-dense}} = \big\langle m_{qd}\, \rho\big(\widehat{x}_{q,\tau_s} - T_{\tau_s,c}\ \text{bilinear}(\widehat{P}_{\tau_s,c}; u, v)\big)\big\rangle, \tag{10}$$

where ρ is smooth-L1 and the transform is applied to the dense prediction rather than the query prediction, so that both sides are compared in `ego_0`. The mask m_{qd} requires a valid query, an in-bounds source pixel, and a finite dense prediction at that pixel. This term is *not* stop-gradient on either side: gradients flow into both heads and hence into the shared encoder, which is precisely why removing the dense branch in D0 also removes this coupling and why Section 4.4 describes D0 as an ablation of a supervision package rather than of one loss.

The reprojection term supervises a predicted point through a second, observed view. For a query whose target position $\widehat{x}_q$ is visible in view (τ_q, c') with projection $\Pi_{\tau_q,c'}$, and whose ground-truth pixel in that view is (u^*, v^*),

$$\mathcal{L}_{\text{reproj}} = \big\langle m_{rp}\, \rho\big(\Pi_{\tau_q,c'}(\widehat{x}_q) - (u^*, v^*)\big)\big\rangle. \tag{11}$$

The mask m_{rp} requires the target view to be an *observed* frame, the projected point to fall in front of that camera and inside its canvas, and the supporting target pixel to exist; queries failing any condition contribute nothing rather than a default value. We do not perform explicit occlusion reasoning, so a point that is geometrically in frustum but occluded in the target view may still be supervised; this is a known approximation and is the reason λ_{rp} is kept small (0.05). Both terms use the same valid-query, per-group reduction as Eq. (7), and neither is uncertainty-discounted.

Planning implementation. The adapter uses eight attention heads, an FFN expansion ratio of four and dropout 0.1. The planner predicts 64 candidates, each with eight future (x, y, θ) poses. The main NAVSIM frozen-transfer recipe uses 32×A100-80G, batch 8/GPU (global 256), seed 0, bf16, gradient clip 1.0, AdamW with base learning rate 2×10^{-4} scaled to 4×10^{-4}, 10% warmup plus cosine decay, 20 epochs and the final epoch's weights, never a test-selected checkpoint. About 18.0M of the 103.9M parameters are trainable. The encoder references and GeoAct arms share the planner architecture but train its weights independently; input-budget differences remain explicit in Table 4.

A.2 Detailed parameterisation

Inputs. Let $I_{t,c}$ be the image from camera c at observed time τ_t, with $\tau_T = 0$ at the latest frame. With intrinsics $K_{t,c}$ and camera-to-`ego_0` rotation and origin $(R_{t,c}, o_{t,c})$, the ray of patch-centre pixel π_p is

$$d_{t,c,p} = \frac{R_{t,c} K_{t,c}^{-1} \pi_p}{\|K_{t,c}^{-1} \pi_p\|_2}, \qquad \ell_{t,c,p} = \big[d_{t,c,p};\ o_{t,c} \times d_{t,c,p}\big]. \tag{12}$$

DINOv3 class and register tokens are discarded. Position and time use Fourier features with learned projections, camera identity an embedding table. Padding masks exclude invalid frames, views and tokens. Each encoder stage uses six heads and FFN ratio four; temporal slots are image-grid locations, not assumed point correspondences. $M = \text{Flatten}(\text{LN}(Z^{(12)}))$.

Dense head. Each tap $F^{(l)}$ has width $3D = 1152$. The DPT decoder projects the taps to four scales and fuses them coarse-to-fine; a shared convolutional trunk G feeds $\widehat{D} = \text{softplus}(g_d(G))$ and $\widehat{P} = \chi(g_p(G))$ with $\chi(r) = \text{sign}(r)(e^{|r|} - 1)$, plus a per-pixel log-uncertainty. $\widehat{D}$ is camera Z-depth and is predicted independently of $\widehat{P}$, which is in the OpenCV camera frame. Teacher and LiDAR terms use independent masks with no reconciliation between them; teacher pixels are masked by MoGe-2's validity output. Depth residuals are in log space and XYZ residuals in metres.

Query head. The full query is $(c, u, v, \tau_s, \tau_q, \ell_s, \mathbf{p}_s)$, with $u = (x + 0.5)/W$, $v = (y + 0.5)/H$ and $\mathbf{p}_s$ a bilinearly sampled 9×9 RGB patch. Its initial embedding is the sum of separate embeddings of each field plus a constant task embedding e_{xyz}, which does not encode any static/dynamic label; the RGB-patch MLP has a zero-initialised output. Each of the eight layers is pre-normalised cross-attention to M followed by an FFN (Appendix A.3). Position and displacement heads are independent MLPs, so $\widehat{x}_q$ is not hard-constrained to $\widehat{x}_s + \widehat{\Delta x}_q$; their learned log-scales are clamped and are not calibrated probabilities. In Eq. (7), $\ell(e, s) = e^{-s}\rho(e) + s$ with ρ smooth-L1, and every term is averaged separately over current, static and moving query groups. Near-ground returns are excluded from rigid transport but remain static targets.

Scene-token adapter. For each camera, the TP tokens of M are projected from $D = 384$ to 256 and read by 16 camera-specific queries through one residual cross-attention and one FFN (Eq. (16)); projection, attention and FFN weights are shared across cameras.

A.3 Standard block definitions

The encoder, query decoder and scene adapter all use the same pre-normalised residual pattern, stated here once. For an attention neighborhood $\mathcal{A}$ with its padding mask,

$$\begin{aligned} \bar{Z} &= Z + \mathrm{MHA}_{\mathcal{A}}\big(\mathrm{LN}_1(Z), \mathrm{LN}_1(Z), \mathrm{LN}_1(Z)\big), \\ \mathcal{B}_{\mathcal{A}}(Z) &= \bar{Z} + \mathrm{FFN}_{\mathcal{A}}\big(\mathrm{LN}_2(\bar{Z})\big). \end{aligned} \tag{13}$$

Encoder stage l composes three such blocks with independent parameters,

$$S^{(l)} = \mathcal{B}^{(l)}_{\mathrm{self}}(Z^{(l-1)}), \quad U^{(l)} = \mathcal{B}^{(l)}_{\mathrm{time}}(S^{(l)}), \quad Z^{(l)} = \mathcal{B}^{(l)}_{\mathrm{view}}(U^{(l)}), \tag{14}$$

and the dense head reads $F^{(l)} = [S^{(l)} \| U^{(l)} \| Z^{(l)}]$ in Eq. (3). The query decoder replaces self-attention with cross-attention to the shared memory; during geometric pre-training gradients propagate through M into the encoder, which is frozen only during planning adaptation. Layer j is

$$\begin{aligned} \bar{h}^{(j)}_q &= h^{(j-1)}_q + \mathrm{MHA}_j\big(\mathrm{LN}_j(h^{(j-1)}_q), M, M\big), \\ h^{(j)}_q &= \bar{h}^{(j)}_q + \mathrm{FFN}_j\big(\mathrm{LN}'_j(\bar{h}^{(j)}_q)\big), \end{aligned} \tag{15}$$

with no query-to-query attention at any layer. The scene adapter applies the same pattern once, with the camera-specific registers R_c as queries and a projected, doubly normalised memory as keys and values,

$$\begin{aligned} \bar{M}_c &= \mathrm{LN}_m\big(\mathrm{LN}_p(M_c)W_p + b_p\big), \\ H_c &= R_c + \mathrm{MHA}\big(\mathrm{LN}_q(R_c), \bar{M}_c, \bar{M}_c\big), \\ Z_c &= \mathrm{LN}_o\big(H_c + \mathrm{FFN}(\mathrm{LN}_f(H_c))\big). \end{aligned} \tag{16}$$

A.4 Supporting detail for Section 4.3 and Section 4.5

The sharpness diagnostic. For each of the 1961 scenes we take the projected-LiDAR depth on the canonical support (Z-depth in $[1, 80]$ m, at least 1000 valid pixels). For every valid pixel we collect the minimum and maximum valid ground-truth depth over its 24-neighbour window at radius 2; radius 1 is too small, because sparse LiDAR rarely has adjacent valid columns. A pixel with at least two valid neighbours forms a *pair*; pairs whose ground-truth high/low ratio exceeds 1.25 are *edge* cells and those at or below 1.05 are *smooth* cells, giving medians of about 1857 and 8135 cells per scene (3.76M and 16.2M in total). Contrast fidelity is $\mathrm{median}\big(\log r_{\mathrm{pred}} / \log r_{\mathrm{gt}}\big)$ over a scene's edge cells, where r denotes the high/low ratio in the same window; scenes with fewer than 20 usable cells are skipped, and scene medians are then averaged. Because the quantity is a ratio of log contrasts within one image, it is invariant to a global scale factor. Sobel log-depth gradient energy is the mean $|\nabla \log d|$ over the lower two-thirds of the image, excluding the border, and is likewise scale-invariant. Each reference is given its own per-case oracle median scale — the setting most favourable to it, matching the *oracle* rows of Table 2 — while GeoAct is evaluated unscaled.

Other failure modes. GeoAct and GeoX emit no non-positive depths; DVGT-2 does so on 1.6% of supported pixels on average. Near-collapse — predicting under 5 m where ground truth exceeds 15 m — affects 0.014%, 0.021% and 0.26% of pixels respectively.

Structural losses (S0). S0 matches N1 in initialisation, corpus, split, schedule and budget; its config differs from N1's only in `lambda_depth_gradient` and `lambda_normal` ($0.5 \to 0$ and $0.1 \to 0$), with the direct dense and LiDAR weights unchanged. S0 reaches AbsRel 0.109101 against 0.109330 for N1, a difference of -0.000230 with paired whole-log 95% CI $[-0.000350, -0.000107]$. The interval excludes zero, but the effect is 0.21% relative and the point-cloud metrics do not move consistently (Table 2), which is why the main text reports no consistent benefit rather than a benefit or a harm.

GeoX training budget. The GeoX arm uses the GeoAct planner budget (20 epochs, global batch 256, final-epoch weights). A shorter schedule (10 epochs, global batch 64, validation-selected checkpoint) gives 89.99 PDMS with validation still rising, so matching the budget matters: it removes about half of the apparent gap.

Training cost. The N1 continuation ran at 3.093 s per iteration on $32\times$A100-80G with global batch 64 and 18.64 GiB peak allocated memory (20.69 scenes/s, including periodic evaluation overhead); the query-only mixed-corpus run reached 1.969 s per iteration and 17.19 GiB. These are *training* figures, and the two runs differ in corpus and query count as well as in read-outs, so the difference cannot be attributed to the dense head alone.

Inference latency. On one A100-SXM4-80GB (bf16, batch 1, 20 warm-up and 50 timed iterations, median of CUDA-event timings, synthetic GPU-resident inputs, no I/O), the planning path of N1 — encoder, adapter and planner, with both geometric heads disabled — takes 191.6 ms at 3.88 GiB peak memory for 4 frames $\times$ 8 cameras; the encoder alone takes 182.0 ms. The planning-path model has 103.9 M parameters, of which 86.9 M belong to the encoder (frozen) and the adapter (trained). DVGT-2 on the same 32-image input takes 841.0 ms (760.6 ms without its point head).

Streaming variant with cached history. The architecture supports streaming inference. With causal temporal attention and time and ray embeddings injected after the spatial–temporal–view stages, past-frame states can be cached, so each step encodes only the newest frame. This cuts planning-path latency from 191.9 to 75.7 ms ($2.5\times$), with cached outputs identical to full causal recomputation. Fine-tuned from N1, the streaming model stays within a few percent of a matched bidirectional control on an internal held-out set (dense AbsRel 0.1451 versus 0.1417; query EPE $+4.1\%$ current, $+5.2\%$ moving). It is not the model evaluated in the main paper.

A.5 Short-horizon future prediction (exploratory)

A causal variant (F1, Table 5) reuses the same point query with $t_{\mathrm{tgt}} > 0$, under a strict no-leakage contract: future RGB, future ego poses and future boxes are used only to build supervision and never enter the model. On the internal held-out set after 60000 steps, absolute EPE is 4.4385 m over all future queries and 3.3428 m for same-time queries. Relative to a baseline that keeps each *predicted* source point stationary, the model improves moving points by 25.68% but slow-moving points by only 0.64%, and is 0.83% worse on static points; by horizon the improvement is 5.73% at $+0.5$ s and 12.03% at $+1.0$ s; by source distance the future EPE is 1.93, 3.91 and 7.41 m for 0–20, 20–40 and 40–80 m. Static retention under the same test geometry protocol degrades slightly relative to its initialisation (0.12702 versus 0.12418 AbsRel, paired log CI $[+0.00201, +0.00375]$), and $30000 \to 60000$ steps bought only 1.38%.

We deliberately keep this out of the main claims. The reported counts are query *visits* under a distributed sampler, not independent trajectories, so we do not compute significance from them; F1 differs from its initialisation in causal attention, label thresholds, extra steps and optimisation recipe simultaneously; and, most importantly, the comparison baseline keeps points stationary rather than extrapolating them.

A.6 Negative and null results retained

Adding a second corpus. A from-scratch query-only run mixing 0.32 Waymo and 0.68 NAVSIM is worse under the test geometry protocol than the NAVSIM-only query-only continuation (0.1242 versus 0.1156). Four factors changed together — corpus mixture, $12000 \to 40000$ steps, $512 \to$

1024 queries, and initialisation from scratch instead of from the shared checkpoint — and we consider the initialisation the most likely explanation, since the from-scratch arm never received dense supervision at all. No single-factor attribution is warranted.

A.7 Evaluation protocol details

Planning results are reported at full token coverage, because evaluation failures are not random and a partial subset can bias paired gaps. Shards are split by token hash rather than by position, since per-process dataset enumeration is not order-stable, and aggregation indexes by token. Macro means use a NaN-aware mean with explicit scored and failed counts; a model that fails on hard scenes is still scored on an easier surviving subset, so failure counts are printed beside every mean and the affected row carries its own N. PDMS aggregation excludes the per-file summary row, so every planning result counts exactly 12146 tokens.