GEOFORWARD: GEOMETRY-GUIDED CROSS-VIEW ALIGNMENT FOR FEEDFORWARD 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review



Figure 1: **From diverse observations to a unified 3D scene.** Driving scenes are observed from multiple cameras and timestamps, spanning different scenarios and viewpoints. GeoForward transforms these heterogeneous observations into a unified Gaussian scene representation, enabling geometrically consistent reconstruction across views and time.

ABSTRACT

Feedforward 3D reconstruction has largely followed an appearance-driven paradigm: multi-view images are encoded into visual representations, cross-view correspondence is implicitly inferred from learned features, and 3D primitives are subsequently predicted from these representations. While effective, this paradigm leaves a fundamental source of information underexploited: calibrated cameras and sparse metric measurements already provide explicit constraints on where the same physical point should appear across views. We propose a novel paradigm, **Geometry-Conditioned Feedforward Reconstruction**, in which metric geometry establishes cross-view correspondence before the 3D representation is formed, while learned visual features refine rather than discover this correspondence. Based on this paradigm, we introduce **GeoForward**. Sparse LiDAR points are unprojected into 3D and analytically projected across calibrated cameras to provide metric correspondence anchors. A bounded continuous local search then refines these anchors using visual features, and the resulting geometry-conditioned representation is fused with global multi-view context from VGGT. The shared representation drives separate metric-depth and Gaussian-attribute prediction, establishing a direct path from explicit geometry to both Gaussian centers and spatial attributes. A sparse 3D refinement module subsequently improves local attribute compatibility while preserving the geometrically determined centers. For temporal reconstruction, endpoint Gaussians are composed using vehicle masks and motion priors. On nuScenes, GeoForward achieves the lowest cross-camera depth-consistency error and Chamfer distance among the compared methods while maintaining competitive reconstruction quality. These results demonstrate that explicitly conditioning correspondence formation on metric geometry provides an effective alternative to purely appearance-driven feedforward 3D reconstruction.

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

Feed-forward 3D reconstruction has recently emerged as a promising alternative to optimization-based scene reconstruction, enabling 3D representations to be predicted directly from multi-view observations without per-scene fitting (Szymanowicz et al., 2024; Charatan et al., 2024; Chen et al., 2024b; Tian et al., 2025). A central challenge in this setting is establishing reliable correspondence across views: observations from different cameras should describe the same physical surface and therefore produce geometrically compatible representations. Existing approaches typically address this challenge by learning cross-view relationships from visual representations, where image features implicitly encode correspondence cues before 3D geometry is decoded. This appearance-driven formulation has achieved impressive performance, but it leaves an important source of information underexploited. In calibrated autonomous-driving platforms equipped with sparse LiDAR, metric measurements already provide physical constraints on where a scene point observed in one camera should appear in another.

This motivates a fundamental question: *can metric geometry directly participate in cross-view correspondence formation before 3D representation prediction?* Existing feed-forward reconstruction methods incorporate geometric information through different mechanisms, including learned depth prediction, explicit depth-hypothesis reasoning, or geometric foundation models that infer multi-view structure from images (Charatan et al., 2024; Chen et al., 2024b; Wang et al., 2024; 2025). These approaches demonstrate the power of learned visual representations for recovering geometry, but the correspondence underlying reconstruction remains primarily obtained through visual inference. In driving scenes, sparse LiDAR and calibrated cameras provide a complementary opportunity: instead of requiring a network to discover all possible cross-view associations from appearance, metric observations can provide a physically grounded initialization that narrows the correspondence search space.

We introduce **GeoForward**, a geometry-conditioned paradigm for feed-forward 3D reconstruction, as illustrated in Fig. 1. Different from appearance-driven correspondence learning, GeoForward first establishes metric cross-view correspondence using sparse LiDAR measurements and calibrated camera geometry. A LiDAR-supported point in a reference view is unprojected into 3D and projected into context views, providing an explicit correspondence anchor. Since sparse sensing, calibration errors, and discretization introduce residual uncertainty, a bounded continuous refinement module further adjusts the correspondence using local visual features. In this way, geometry constrains *where* cross-view evidence should be associated, while learning determines *how* the local correspondence should be refined.

The geometry-conditioned representation is then used for Gaussian scene reconstruction. GeoForward fuses the aligned features with global multi-view context from a frozen VGGT encoder (Wang et al., 2025) and jointly predicts metric depth and Gaussian attributes. The depth branch determines Gaussian centers through camera unprojection, while the attribute branch predicts spatial extent, opacity, and appearance conditioned on the geometry-aware representation. Beyond initial prediction, a sparse 3D neighborhood refinement module improves local attribute compatibility while preserving the predicted Gaussian centers. For dynamic driving scenes, endpoint Gaussian representations are further composed at arbitrary query times using vehicle masks and motion priors (Yu et al., 2026a; He et al., 2026; Song et al., 2026).

We evaluate GeoForward on two large-scale driving-scene benchmarks, nuScenes Caesar et al. (2020) and Waymo Sun et al. (2020). Experiments demonstrate that explicitly conditioning correspondence formation on metric geometry improves cross-view geometric consistency while maintaining competitive rendering quality. These results suggest a complementary direction for feed-forward reconstruction: rather than relying solely on learned appearance-based association, available metric information can directly participate in forming the 3D representation. Our contributions are summarized as follows:

- We propose a **geometry-conditioned paradigm for feed-forward 3D reconstruction**, where sparse metric geometry constrains cross-view correspondence before 3D representation formation, reducing ambiguous visual matches and anchoring depth and Gaussian prediction in metric space.
- We introduce **GeoForward**, which combines LiDAR-based correspondence initialization with bounded visual refinement and propagates geometry-conditioned features into Gaussian geometry and attribute prediction.

- We introduce **sparse 3D attribute refinement** to improve local Gaussian consistency without modifying geometrically determined centers, and demonstrate improved cross-view reconstruction quality on **nuScenes and Waymo**.

2 Related Work

2.1 Feed-Forward 3D Scene Reconstruction

Recent advances in feed-forward reconstruction aim to predict 3D scene representations directly from multi-view observations, avoiding expensive per-scene optimization. DUSt3R learns pairwise point maps from image pairs, while VGGT extends this idea to jointly estimate depth, point maps, camera parameters, and point tracks from multiple views (Wang et al., 2024; 2025). InfiniDepth further explores continuous coordinate-query decoding for high-resolution geometric prediction (Yu et al., 2026b).

Building on these geometric representations, recent works investigate feed-forward Gaussian reconstruction. MVSplat, AnySplat, and IDESplat predict Gaussian representations directly from images, improving reconstruction efficiency and scalability (Chen et al., 2024b; Jiang et al., 2025; Long et al., 2026). For autonomous driving scenarios, DrivingForward, DGGT, and StreetForward incorporate camera poses, temporal information, and dynamic scene modeling into feed-forward Gaussian reconstruction (Tian et al., 2025; Chen et al., 2026a; Yu et al., 2026c). DynamicVGGT and Envision4D further extend feed-forward reconstruction toward temporal prediction and future scene generation (He et al., 2026; Song et al., 2026). Other approaches improve multi-view representation through 3D latent structures, uncertainty-aware canonical representations, or sparse 3D queries, including UniSplat, CanonicalGS, and PointForward (Shi et al., 2026; Xu et al., 2026; Chi et al., 2026). ReconDrive further combines VGGT representations with hybrid Gaussian prediction and velocity-based temporal composition for driving scenes (Yu et al., 2026a).

Although these methods achieve impressive reconstruction quality, cross-view consistency is mainly obtained through learned visual representations and implicit feature interaction. GeoForward complements this direction by introducing explicit metric constraints into the cross-view association process before Gaussian prediction.

2.2 Metric-Guided Multi-View Geometric Reasoning

Metric measurements provide an important source of geometric constraints for multi-view reconstruction and depth estimation. Previous works incorporate sparse geometric priors into multi-view stereo through sparse cost-volume construction or depth hypothesis refinement (Qi et al., 2022; Poggi et al., 2022). In autonomous driving, LiDAR measurements are particularly valuable due to their accurate metric information, and several approaches explore LiDAR-guided geometric prediction. DriveMVS uses sparse LiDAR observations to anchor depth hypotheses and incorporate geometric cues into feature fusion, while DrivingDepth leverages sparse LiDAR measurements to calibrate monocular depth estimation through metric correction (Sun et al., 2026; Huang et al., 2026). Related LiDAR-camera fusion methods further combine sparse depth measurements with image features for dense depth estimation (Fu et al., 2019; Wei et al., 2023).

These approaches demonstrate the effectiveness of metric information for reducing geometric ambiguity. However, existing methods primarily introduce geometry as a constraint on depth estimation or feature enhancement, while the cross-view association underlying representation prediction is still largely determined by learned visual similarity. GeoForward explores a different use of metric information by treating sparse LiDAR measurements as geometric anchors for cross-view feature alignment. Rather than predicting correspondence purely from appearance or depth hypotheses, GeoForward establishes geometry-aware associations through calibrated projection and refines them using local visual evidence.

3 Methodology

Overview. GeoForward reconstructs a Gaussian scene from calibrated surround-view images $I_{t,v}$ and sparse LiDAR depth $D^{\mathrm{L}}_{t,v}$ observed at two endpoint timestamps. Camera intrinsics $K_{t,v}$ and

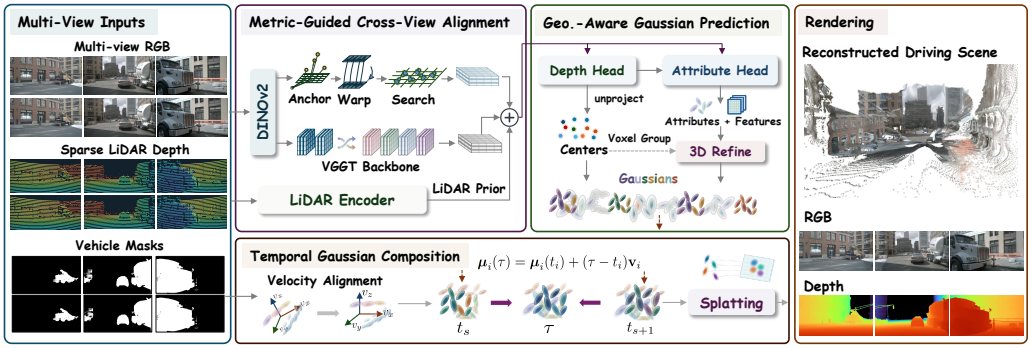


Figure 2: **Overview of GeoForward.** Given calibrated multi-view images and sparse LiDAR, GeoForward first establishes metric cross-camera correspondences through geometric projection and continuous local refinement. The aligned features are fused with global multi-view context and jointly condition metric depth and Gaussian-attribute prediction. A sparse 3D refinement module further improves local attribute consistency, while vehicle masks and motion annotations enable temporal Gaussian composition at arbitrary query times.

camera-to-reference poses $E_{t,v} = (R_{t,v}, \mathbf{t}_{t,v})$ transform all observations into a common reference-ego frame. The key idea of GeoForward is to convert sparse metric measurements into explicit cross-camera feature correspondence before Gaussian prediction. As illustrated in Fig. 2, a LiDAR-supported pixel in a reference camera is first projected into a context camera using calibrated geometry, providing a metric correspondence initialization. A bounded learned offset then refines this correspondence using local visual evidence. The resulting aligned features are combined with global multi-view features from a frozen VGGT encoder (Wang et al., 2025) and shared by the depth and Gaussian-attribute prediction branches. Finally, a sparse 3D refinement module improves local attribute compatibility while preserving the predicted Gaussian centers, and endpoint Gaussians are composed at arbitrary query times for dynamic-scene rendering. Sparse LiDAR is used at both training and inference.

3.1 Metric-Guided Cross-View Alignment

The central component of GeoForward is metric-guided cross-view alignment, as illustrated in Fig. 5. Rather than requiring a network to infer all cross-camera correspondence implicitly, we use sparse LiDAR measurements to provide an analytically determined correspondence and learn only a bounded local correction. This separates the problem into two complementary components: metric initialization from calibrated geometry and visual refinement from learned features.

We use a frozen VGGT encoder (Wang et al., 2025) to jointly process the endpoint images. Its global multi-view representation is retained to provide scene-level context, while intermediate DINOv2 features (Oquab et al., 2023) are used for local correspondence refinement. Freezing VGGT isolates the contribution of explicit metric correspondence and avoids relying on additional end-to-end adaptation of the geometric foundation model.

Metric correspondence initialization. For a reference view A, a sampled LiDAR depth at image location $\mathbf{p}$ defines a 3D anchor $\mathbf{X}_A = d_A^{\mathrm{L}}(\mathbf{p}) K_A^{-1} \widetilde{\mathbf{p}}$, where $\widetilde{\mathbf{p}}$ is the homogeneous image coordinate. Given the relative pose $E_B^{-1} E_A = (R_{B\leftarrow A}, \mathbf{t}_{B\leftarrow A})$, the anchor is transformed into context view B and projected onto its image plane:

$$\mathbf{g}_{AB}(\mathbf{p}) = \pi\left(K_B\left(R_{B\leftarrow A}\mathbf{X}_A + \mathbf{t}_{B\leftarrow A}\right)\right), \tag{1}$$

where $\pi(\cdot)$ denotes perspective division. We construct a support mask m_{AB} according to LiDAR availability, positive projected depth, image validity, and context-view visibility.

For cross-time alignment, the same projection is applied using the calibrated poses under a static-scene assumption. This approximation is used only to establish the initial visual correspondence; object motion is handled separately during temporal Gaussian composition in Sec. 3.4.

Learned local refinement. The metric projection provides a strong initialization but does not guarantee pixel-level feature correspondence because of sparse sampling, calibration errors, discretization, and view-dependent appearance. We therefore refine the projected location using local feature evidence. Let $\mathcal{S}$ denote the feature-query operator, we obtain $\mathbf{f}_A = \mathcal{S}(F_A, \mathbf{p})$,

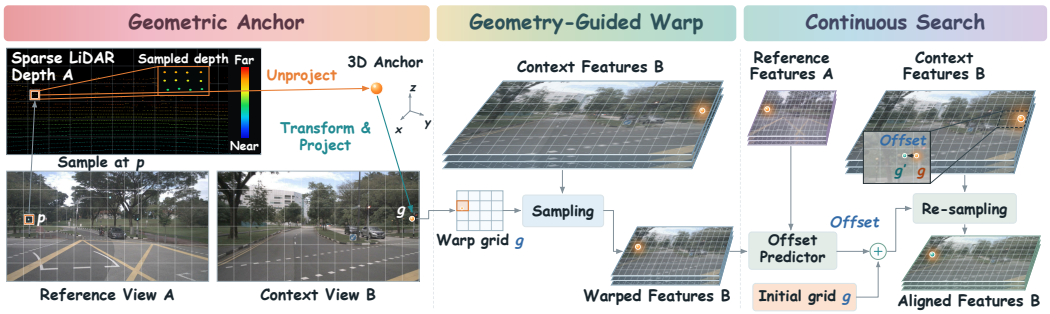


Figure 3: **Metric-guided cross-view alignment.** Sparse LiDAR provides a metric correspondence anchor by projecting a 3D point from the reference view into a context view using calibrated geometry. A bounded continuous offset then locally refines the projected location using visual features, after which the context feature is re-sampled at the refined coordinate.

$\mathbf{f}^0_{AB} = \mathcal{S}\left(\mathrm{sg}(F_B), \mathbf{g}_{AB}\right)$, where F_A and F_B are intermediate feature maps and $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator.

An offset predictor takes the local feature pair as input and predicts a bounded correction: $\Delta\mathbf{g}_{AB} = r\tanh\left(\Phi\left([\mathbf{f}_A, \mathbf{f}^0_{AB}]\right)\right)$, where r specifies the maximum search radius in pixels. The refined context feature is then obtained by re-sampling the original context feature map:

$$\widehat{\mathbf{f}}_{AB} = \mathcal{S}\left(\mathrm{sg}(F_B), \mathbf{g}_{AB} + \Delta\mathbf{g}_{AB}\right). \tag{2}$$

The bounded formulation restricts the learned refinement to a local neighborhood around the metric initialization rather than learning correspondence from scratch. Gradients propagate through the sampling coordinates while the sampled context feature values are detached.

For a reference view, refined features from supported context views are aggregated as $\overline{\mathbf{f}}_A = \frac{\sum_B m_{AB}\widehat{\mathbf{f}}_{AB}}{\sum_B m_{AB}+\epsilon}$. When no valid context view is available, we set $\overline{\mathbf{f}}_A = \mathbf{0}$. A residual fusion layer incorporates the aligned evidence: $\widetilde{\mathbf{f}}_A = \mathbf{f}_A + \mathcal{Z}\left([\mathbf{f}_A, \overline{\mathbf{f}}_A]\right)$.

We perform the alignment at multiple feature depths and concatenate the resulting aligned features with the corresponding VGGT representations to obtain the shared representation $\mathbf{z}_i$. This representation combines explicit metric correspondence with global multi-view context without constructing a depth-hypothesis cost volume.

3.2 Geometry-Aware Gaussian Prediction

The aligned representation $\mathbf{z}_i$ is used to predict a Gaussian for each valid source pixel. Following the coordinate-query formulation and recent continuous depth prediction methods (Yu et al., 2026b), we use separate depth and attribute branches. The separation allows the metric geometry of a Gaussian center to be predicted independently from its appearance and spatial attributes, while both branches receive the same geometry-guided cross-view evidence.

Metric depth and Gaussian centers. For pixel i, the depth branch combines the shared aligned representation with local RGB features to obtain a visual descriptor $\mathbf{v}_i$. Sparse LiDAR is additionally encoded as a metric prior $\boldsymbol{\ell}_i$ with support s_i. The visual and LiDAR-conditioned predictions are combined as

$$d_i = d_{\min} + (d_{\max} - d_{\min})\sigma\left(\Psi_{\mathrm{vis}}(\mathbf{v}_i) + \Psi_{\mathrm{prior}}(\mathbf{v}_i, \boldsymbol{\ell}_i, s_i)\right), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function and $[d_{\min}, d_{\max}]$ is the prescribed metric depth range. The LiDAR prior provides sparse metric information complementary to the correspondence mechanism rather than directly replacing the predicted depth.

The Gaussian center is obtained by camera unprojection: $\boldsymbol{\mu}_i = R_{t,v}\left(d_i K^{-1}_{t,v}\widetilde{\mathbf{p}}_i\right) + \mathbf{t}_{t,v}$.

We further derive camera-space surface normals $\mathbf{n}_i$ from the predicted depth and estimate a confidence value c_i from detached visual features. These quantities provide geometric and reliability cues to the attribute branch.

Gaussian attributes. The attribute branch predicts the remaining Gaussian parameters using the aligned representation, local RGB features, sparse-depth features, and source color. Its geometric conditioning descriptor is $\boldsymbol{\gamma}_i = [\log d_i, \mathbf{n}_i, c_i, \nu_i]$, where ν_i denotes camera validity. The resulting feature is $\mathbf{h}_i = \mathcal{D}_{\mathrm{attr}}\left(\mathbf{z}_i, \boldsymbol{\ell}_i, \boldsymbol{\gamma}_i, I_i\right)$, and the initial Gaussian attributes are predicted as $\mathbf{b}_i = \mathcal{A}(\mathbf{h}_i)$,

including rotation, scale, opacity, and appearance parameters. The shared representation $\mathbf{z}_i$ therefore allows the metric correspondence established in Sec. 3.1 to influence both Gaussian center locations through depth prediction and the attributes describing the local surface.

3.3 Sparse 3D Attribute Refinement

Although geometry-guided alignment improves the consistency of independently decoded Gaussians, neighboring Gaussians may still exhibit incompatible attributes after prediction. We therefore perform a lightweight refinement directly in 3D space while keeping the predicted centers fixed.

Specifically, Gaussian centers are grouped into sparse voxels according to their reference-frame coordinates. For the Gaussians $\mathcal{I}_j$ in voxel j, we first pool their latent features: $\overline{\mathbf{h}}_j = \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} \mathbf{h}_i$. A sparse convolution $\mathcal{C}$ aggregates information from neighboring occupied voxels, and a readout $\mathcal{R}$ predicts an additive correction:

$$\mathbf{b}'_i = \mathbf{b}_i + \mathcal{R}\left(\mathcal{C}(\overline{\mathbf{h}})\right)_{j(i)}, \tag{4}$$

where $j(i)$ denotes the voxel containing Gaussian i. Gaussians within the same voxel share the predicted correction while retaining their individual decoded attributes.

Importantly, the refinement does not modify $\boldsymbol{\mu}_i$. It therefore regularizes local Gaussian attributes without overriding the metric geometry established by the depth branch. The corrected parameters determine the covariance $\boldsymbol{\Sigma}_i$, opacity o_i, and appearance coefficients $\mathbf{a}_i$ after the corresponding activation functions. We express the base color as a bounded residual around the source pixel color and transform the covariance and appearance parameters into the common reference frame. Additional activation and initialization details are provided in the appendix.

3.4 Temporal Gaussian Composition

GeoForward predicts Gaussian representations independently at two endpoint timestamps. To render an intermediate time τ, vehicle-associated Gaussians are translated according to their estimated motion and jointly composited.

Following (Yu et al., 2026a), a frozen SAM2 model (Ravi et al., 2025) is prompted with projected vehicle boxes to obtain object masks. Source-pixel Gaussians within accepted masks are associated with the corresponding annotated vehicle velocity $\mathbf{v}_i^{\text{motion}}$, while Gaussians outside the accepted masks are assigned zero velocity. A Gaussian originating at timestamp t_i is translated as

$$\boldsymbol{\mu}_i(\tau) = \boldsymbol{\mu}_i + (\tau - t_i)\mathbf{v}_i^{\text{motion}}. \tag{5}$$

This temporal composition is applied after Gaussian reconstruction and does not modify the proposed cross-camera alignment mechanism.

3.5 Learning Objectives

We train GeoForward in two stages. The first stage learns endpoint geometry and appearance from source-view reconstruction, while the second stage introduces temporal composition and intermediate-view supervision. The VGGT encoder remains frozen throughout both stages.

Endpoint reconstruction. For each endpoint, the predicted Gaussians are rendered into their source views and supervised using image and metric-depth observations. Offline dense metric-depth maps provide training targets only and are not used as model inputs. The endpoint objective is

$$\mathcal{L}_{\text{end}} = \lambda_{\text{img}}\mathcal{L}_{\text{img}} + \lambda_{\text{d}}\mathcal{L}_{\text{d}} + \lambda_{\text{conf}}\mathcal{L}_{\text{conf}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}} + \mathcal{L}_{\text{reg}}, \tag{6}$$

where $\mathcal{L}_{\text{img}}$ measures rendered-image reconstruction, $\mathcal{L}_{\text{d}}$ supervises metric depth, and $\mathcal{L}_{\text{conf}}$ supervises the predicted confidence. $\mathcal{L}_{\text{geo}}$ contains depth-boundary and multi-scale image constraints, while $\mathcal{L}_{\text{reg}}$ regularizes Gaussian attributes. Detailed formulations are provided in the appendix.

Confidence targets are constructed from the discrepancy between predicted depth and the offline dense metric-depth targets, with the corresponding validity masks. Thus, confidence represents the reliability of the predicted geometry rather than an additional source of metric supervision.

Temporal supervision. In the second stage, endpoint and intermediate renderings are jointly supervised. In addition to direct image reconstruction, we use a pose-based reprojection constraint to encourage temporal consistency. Within each camera, the first endpoint depth and known camera poses warp an intermediate observation into the endpoint coordinate system. On static and projection-valid regions, brightness-normalized L_1 and SSIM residuals define $\mathcal{L}_{\text{proj}} = \mathcal{L}_{\text{photo}} + \lambda_{\text{ssim}}\mathcal{L}_{\text{ssim}}$. The temporal objective is $\mathcal{L}_{\text{temporal}} = \lambda_{\text{img}}\mathcal{L}_{\text{img}} + \lambda_{\text{proj}}\mathcal{L}_{\text{proj}} + \mathcal{L}_{\text{reg}}$. The overall training procedure optimizes Gaussian reconstruction while using sparse LiDAR in two complementary ways: it establishes explicit cross-camera correspondence through the alignment module and provides sparse

Table 1: Comparison of geometric accuracy and consistency on nuScenes.

Method	Acc. ↓	Comple. ↓	CD ↓	AbsRel ↓	SqRel ↓	RMSE ↓	$\delta_{1.25}$ ↑	DepthCons ↓
Street Gaussians Yan et al. (2024)	2.505	7.136	9.641	0.719	10.081	15.450	0.100	5.992
PVG Chen et al. (2026b)	9.033	3.946	12.979	0.780	14.951	8.210	0.589	4.538
Deformable 3DGS Yang et al. (2024)	2.927	6.420	9.347	0.818	24.110	14.407	0.250	8.196
OmniRe Chen et al. (2025)	1.137	**0.489**	1.625	0.152	1.898	5.229	0.833	4.629
DrivingForward Tian et al. (2025)	1.228	1.195	2.423	0.255	3.365	7.900	0.666	3.756
ReconDrive Yu et al. (2026a)	3.735	1.057	4.793	0.607	23.662	10.466	0.680	5.733
DGGT Chen et al. (2026a)	1.690	5.459	7.149	0.421	6.041	10.477	0.461	4.985
GeoForward (Ours)	**0.674**	0.761	**1.435**	**0.148**	**1.548**	**5.016**	**0.849**	**2.446**

Table 2: Comparison of geometric accuracy and consistency on Waymo.

Method	Acc. ↓	Comple. ↓	CD ↓	AbsRel ↓	SqRel ↓	RMSE ↓	$\delta_{1.25}$ ↑	DepthCons ↓
Street Gaussians Yan et al. (2024)	3.189	1.520	4.710	0.356	8.818	9.549	0.586	2.799
PVG Chen et al. (2026b)	7.969	5.089	13.058	0.562	11.490	8.519	0.609	2.026
Deformable 3DGS Yang et al. (2024)	2.389	4.624	7.012	0.512	8.208	13.837	0.294	1.503
STORM Yang et al. (2025)	1.018	0.815	1.833	0.185	2.526	6.564	0.806	**1.272**
StreetForward Yu et al. (2026c)	4.882	4.270	9.152	0.457	9.152	12.069	0.498	16.826
Envision4D Song et al. (2026)	3.134	4.512	7.646	0.319	5.915	9.356	0.664	7.627
GeoForward	**0.569**	**0.688**	**1.257**	**0.104**	**0.875**	**3.993**	**0.917**	1.477

metric information to the depth prediction branch. Dense metric-depth targets are used only during training and do not replace the sparse geometric input available at inference time.

4 EXPERIMENTS

Datasets. We evaluate GeoForward on nuScenes Caesar et al. (2020) and Waymo Sun et al. (2020), which provide calibrated multi-camera images and LiDAR measurements of driving scenes across varied weather and traffic conditions. For nuScenes, we train GeoForward on the official training split. For Waymo, we start from a nuScenes-trained checkpoint, fine-tune it on temporally sampled Waymo training windows, and then conduct the Waymo comparison. To assess transfer beyond these two benchmarks, we evaluate geometric reconstruction on KITTI Geiger et al. (2012) and Argoverse 2 Wilson et al. (2023) using known calibrations without target-dataset fine-tuning.

Baselines. We compare against per-scene optimization methods, including Street Gaussians Yan et al. (2024), PVG Chen et al. (2026b), and Deformable 3DGS Yang et al. (2024). The feed-forward baselines include DrivingForward Tian et al. (2025), ReconDrive Yu et al. (2026a), and DGGT Chen et al. (2026a). For Waymo, we also report STORM Yang et al. (2025), StreetForward Yu et al. (2026c), and Envision4D Song et al. (2026), which span distinct scene representation strategies.

Evaluation metrics. We evaluate scene reconstruction and novel-view synthesis using PSNR, SSIM, and LPIPS. For geometric quality, we report point-cloud accuracy, completeness, and Chamfer distance, as well as depth AbsRel, SqRel, RMSE, and $\delta_{1.25}$. DepthCons measures depth agreement across overlapping cameras after accounting for their relative calibration. Higher values are better for PSNR, SSIM, and $\delta_{1.25}$; lower values are better for the remaining metrics.

4.1 MAIN RESULTS

Geometry. We compare geometric accuracy and consistency in Tabs. 1 and 2. On nuScenes, GeoForward ranks first on seven of eight metrics and second on completeness. It reduces CD from 1.625 to 1.435 and DepthCons from 3.756 to 2.446, improving over the next-best methods by 11.69% and 34.88%. On Waymo, GeoForward again leads on seven of eight metrics. Its CD of 1.257 and RMSE of 3.993 improve over STORM by 31.42% and 39.17%. Acc. decreases from 1.018 to 0.569, completeness from 0.815 to 0.688, and $\delta_{1.25}$ rises from 0.806 to 0.917. DepthCons ranks second at 1.477, behind STORM. These results demonstrate improved surface placement and depth accuracy across both benchmarks, covering complementary aspects of reconstructed scene geometry.

Appearance. We evaluate scene reconstruction and novel-view synthesis in Tab. 3. On nuScenes, GeoForward achieves the best reconstruction PSNR of $36.615\,\text{dB}$, SSIM of 0.987, and LPIPS of 0.032. Compared with ReconDrive, it gains $3.320\,\text{dB}$ in reconstruction PSNR and reduces LPIPS

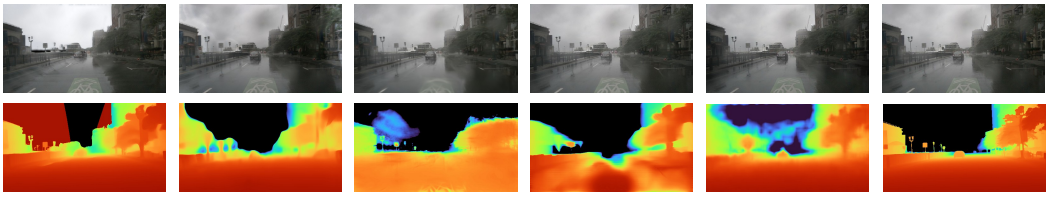
Table 3: Comparison of rendering quality on nuScenes and Waymo.

nuScenes

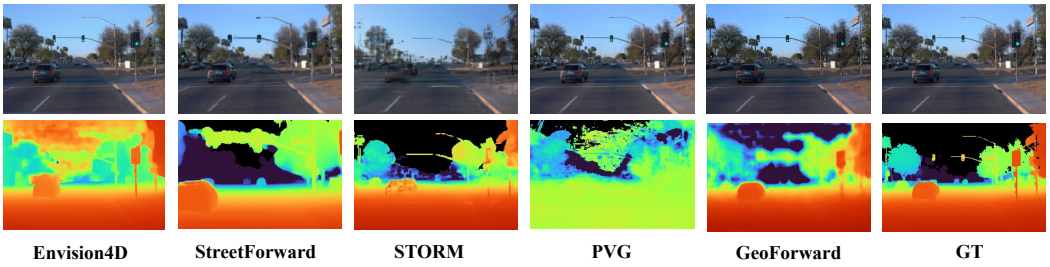
Method	Visual Scene Reconstruction			Novel-View Synthesis		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Street Gaussians Yan et al. (2024)	26.090	0.774	0.430	18.942	0.621	0.515
PVG Chen et al. (2026b)	33.040	0.923	0.213	**27.416**	**0.810**	0.302
DeformableGS Yang et al. (2024)	25.185	0.760	0.412	22.188	0.692	0.451
OmniRe Chen et al. (2025)	30.597	0.921	0.193	23.474	0.688	0.332
DrivingForward Tian et al. (2025)	23.791	0.878	0.131	18.240	0.665	0.303
ReconDrive Yu et al. (2026a)	33.295	0.961	0.059	23.986	0.730	**0.259**
DGGT Chen et al. (2026a)	22.503	0.803	0.305	22.498	0.808	0.307
GeoForward (Ours)	**36.615**	**0.987**	**0.032**	23.191	0.730	0.279

Waymo

Method	Visual Scene Reconstruction			Novel-View Synthesis		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Street Gaussians Yan et al. (2024)	34.5317	**0.9558**	0.0985	25.6590	0.7932	0.2333
PVG Chen et al. (2026b)	**35.5709**	0.9505	0.1212	**26.9835**	0.7998	0.2516
DeformableGS Yang et al. (2024)	27.3924	0.8327	0.3274	19.8208	0.6105	0.4921
OmniRe Chen et al. (2025)	31.8291	0.9357	0.1187	25.0610	0.7648	0.2316
STORM Yang et al. (2025)	21.6821	0.7362	0.3025	21.0298	0.7032	0.3251
DGGT Chen et al. (2026a)	25.0395	0.8148	0.2041	23.5523	0.7719	0.2353
StreetForward Yu et al. (2026c)	25.9594	0.7872	0.1734	24.5113	0.7470	0.2165
Envision4D Song et al. (2026)	32.5785	0.9518	**0.0694**	25.0997	0.7490	0.2337
GeoForward (Ours)	32.2823	0.9329	0.0797	25.5167	**0.8026**	**0.1965**



a. Qualitative comparison in a nuScenes driving scene.



b. Qualitative comparison in a dynamic traffic scene from the Waymo dataset.

Figure 4: **Qualitative reconstruction results.** For each method, the upper row shows visual-view RGB renderings, and the lower row shows the corresponding depth maps. GT denotes ground truth.

from 0.059 to 0.032. Its novel-view LPIPS of 0.279 ranks second behind ReconDrive, while PVG leads novel-view PSNR and SSIM. On Waymo, GeoForward achieves the best novel-view SSIM of 0.8026 and LPIPS of 0.1965, reducing LPIPS from StreetForward's 0.2165. Its novel-view PSNR of 25.5167 dB ranks third, and reconstruction LPIPS of 0.0797 ranks second behind Envision4D. These results highlight its structural and perceptual quality at unseen viewpoints.

Generalization. We evaluate geometric transfer to KITTI and Argoverse 2 without target-dataset fine-tuning. In Tab. 5, GeoForward achieves the lowest accuracy error, AbsRel, and RMSE among the compared methods on both datasets. Its RMSE reaches 7.3550 on KITTI and 10.0448 on Argoverse 2, improving over STORM by 16.63% and 31.36%. On Argoverse 2, Acc. decreases from STORM's 2.5556 to 1.6104, while AbsRel improves from the next-best DGGT result of 0.5605 to 0.2982. These gains support effective geometric transfer in both surface placement and metric depth across the evaluated sensor configurations, including different camera layouts and viewpoints.

4.2 Ablation Studies

Tab. 4 separates the effects of geometric alignment, Gaussian prediction, and temporal composition.

LiDAR-guided alignment. Removing LiDAR-guided alignment lowers novel-view PSNR from 23.1912 to 22.1891 dB and SSIM from 0.7302 to 0.6789, while depth RMSE rises from 5.0166 to 7.0338 and $\delta_{1.25}$ falls from 0.8486 to 0.2765. Thus, metric-guided correspondence improves both rendering quality and depth accuracy by reducing ambiguity in cross-view feature matching.

Table 4: Component ablations on nuScenes: novel-view rendering and depth quality.

Setting	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE ↓	$\delta_{1.25}$ ↑
w/o LiDAR Align.	22.1891	0.6789	0.3188	7.0338	0.2765
w/o Gaussian Head	20.0539	0.5754	0.4451	5.5165	0.8317
w/o Motion Compensation	23.1392	0.7292	0.2796	5.0284	0.8466
Full GeoForward	**23.1912**	**0.7302**	**0.2789**	**5.0166**	**0.8486**

Table 5: Zero-shot geometric comparison on KITTI and Argoverse 2.

	KITTI			Argoverse 2		
Method	Acc. ↓	AbsRel ↓	RMSE ↓	Acc. ↓	AbsRel ↓	RMSE ↓
DGGT	64.3272	1.0991	25.5043	5.3868	0.5605	20.4651
STORM	1.3977	0.2856	8.8221	2.5556	0.6836	14.6333
GeoForward	**1.0749**	**0.2596**	**7.3550**	**1.6104**	**0.2982**	**10.0448**

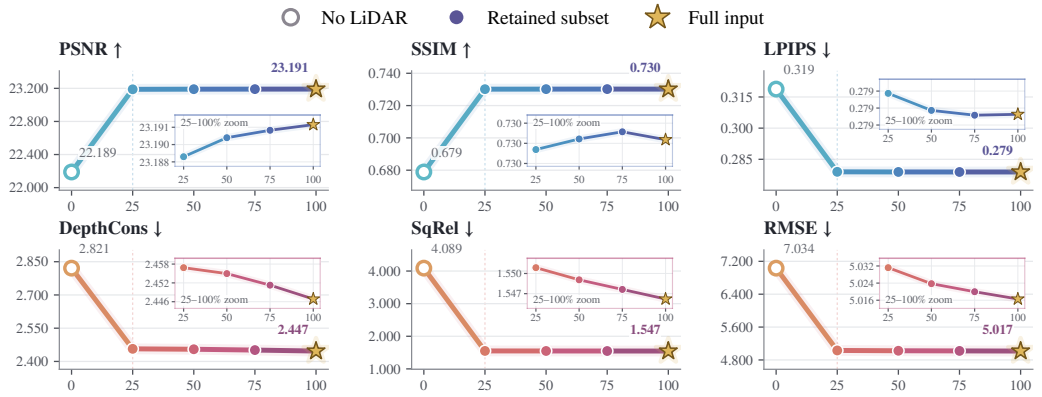


Figure 5: **LiDAR retention ablation on nuScenes.** Performance largely saturates after retaining 25% of the LiDAR points, with only minor changes at higher sampling densities.

Gaussian head. Removing the Gaussian head causes the largest appearance loss: PSNR drops to 20.0539 dB, SSIM to 0.5754, and LPIPS increases from 0.2789 to 0.4451. Its smaller effect on depth RMSE, from 5.0166 to 5.5165, suggests that this component is particularly important for converting the aligned representation into renderable Gaussian attributes with consistent appearance across camera views and improved visual fidelity in unseen camera viewpoints.

Motion compensation. Removing motion compensation changes novel-view PSNR by 0.0520 dB and RMSE by 0.0118, although the vehicle boundaries in Fig. 6 show more noticeable local differences that are diluted when metrics are averaged over large, predominantly static image regions.

LiDAR sparsity. Fig. 5 shows sharp, consistent improvements in all six reported metrics between 0% and 25% LiDAR retention. Additional points yield only small, occasionally non-monotonic changes, indicating that relatively sparse measurements already provide useful geometric anchors.

5 Conclusion

Limitations. GeoForward requires calibrated multi-camera images and sparse LiDAR at inference. The ablation results show a substantial loss in depth accuracy when LiDAR-guided alignment is removed, which may limit its use when range measurements are unavailable. Cross-time feature projection also assumes static correspondences, while temporal rendering depends on tracked vehicle masks and velocity estimates. Untracked objects and non-rigid motion therefore remain challenging. Future work could improve robustness to missing geometric observations and extend temporal reconstruction beyond objects with reliable motion priors.

Conclusion. We introduced GeoForward, which uses sparse LiDAR to align camera features for depth and Gaussian prediction from calibrated multi-camera observations of driving scenes. Bounded local refinement reduces residual cross-camera projection errors, while global multi-view context complements the aligned local features. This design incorporates explicit metric constraints into correspondence formation before predicting the Gaussian scene representation. GeoForward leads seven of eight geometric metrics on both nuScenes and Waymo, leads Waymo novel-view SSIM and LPIPS, and transfers to KITTI and Argoverse 2 without fine-tuning. These results show how metric-guided correspondence supports accurate and transferable driving-scene reconstruction.

AI USE STATEMENT

We used ChatGPT and Codex to assist with manuscript drafting and editing, LaTeX formatting, literature searches, and interpretation and consistency checking of experimental results. The authors take responsibility for the final content, including all AI-assisted text, claims, and artifacts.

REFERENCES

Samer Abualhanud, Christian Grannemann, and Max Mehltretter. Cylinderdepth: Cylindrical spatial attention for multi-view consistent self-supervised surround depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 169–179, 2026.

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 *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11618–11628. IEEE, 2020.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19457–19467, 2024.

Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 31 (9):6100–6111, 2024a.

Xiaoxue Chen, Ziyi Xiong, Yuantao Chen, Gen Li, Nan Wang, Hongcheng Luo, Long Chen, Haiyang Sun, Bing Wang, Guang Chen, et al. Dggt: Feedforward 4d reconstruction of dynamic driving scenes using unposed images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1265–1276, 2026a.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European conference on computer vision*, pp. 370–386. Springer, 2024b.

Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *International Journal of Computer Vision*, 134(3):83, 2026b.

Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo De Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. In *International Conference on Learning Representations*, volume 2025, pp. 85508–85527, 2025.

Cheng Chi, Xianqi Wang, Hongcheng Luo, Mingfei Tu, Gangwei Xu, Zehan Zhang, Bing Wang, Guang Chen, Hangjun Ye, Sida Peng, et al. Pointforward: Feedforward driving reconstruction through point-aligned representations. *arXiv preprint arXiv:2605.11594*, 2026.

Chen Fu, Christoph Mertz, and John M Dolan. Lidar and monocular camera fusion: On-road depth completion for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 273–278. IEEE, 2019.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361. IEEE, 2012.

Zhuolin He, Jing Li, Guanghao Li, Xiaolei Chen, Jiacheng Tang, Siyang Zhang, Zhounan Jin, Feipeng Cai, Bin Li, Jian Pu, et al. Dynamicvggt: Learning dynamic point maps for 4d scene reconstruction in autonomous driving. *arXiv preprint arXiv:2603.08254*, 2026.

Chi Huang, Wenhao Zhang, Hang Yin, YuAn Wang, Hao Li, Bosheng Wang, Xun Sun, and Liang Wang. Drivingdepth: Sparse-prompted pixel-wise scale correction for driving depth estimation. *arXiv preprint arXiv:2606.31488*, 2026.

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

Wei Long, Haifeng Wu, Shiyin Jiang, Jinhua Zhang, Xinchun Ji, and Shuhang Gu. Idesplat: Iterative depth probability estimation for generalizable 3d gaussian splatting. *arXiv preprint arXiv:2601.03824*, 2026.

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

Matteo Poggi, Andrea Conti, and Stefano Mattoccia. Multi-view guided multi-view stereo. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8391–8398. IEEE, 2022.

Yuhang Qi, Wanjuan Su, Qingshan Xu, and Wenbing Tao. Sparse prior guided deep multi-view stereo. *Computers & Graphics*, 107:1–9, 2022.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Chen Shi, Shaoshuai Shi, Xiaoyang Lyu, Chunyang Liu, Kehua Sheng, Bo Zhang, and Li Jiang. Unisplat: Unified spatio-temporal fusion via 3d latent scaffolds for dynamic driving scene reconstruction. In *International Conference on Learning Representations*, volume 2026, pp. 125218–125235, 2026.

Qi Song, Yifei He, Chi Zhang, Zheng Fu, Xuhe Zhao, Mengmeng Yang, Kun Jiang, Rui Huang, and Diange Yang. Envision4d: Envisioning visual futures via feed-forward 4d gaussian splatting for autonomous driving. *arXiv preprint arXiv:2606.10656*, 2026.

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2451. IEEE, 2020.

Qihao Sun, Jiarun Liu, Ziqian Ni, Jianyun Xu, Sheng Yang, Tao Xie, Lijun Zhao, and Ruifeng Li. Lidar prompted spatio-temporal multi-view stereo for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14567–14577, 2026.

Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Qijian Tian, Xin Tan, Yuan Xie, and Lizhuang Ma. Drivingforward: Feed-forward 3d gaussian splatting for driving scene reconstruction from flexible surround-view input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7374–7382, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

Ming Wei, Ming Zhu, Yaoyuan Zhang, Jiarong Wang, and Jiaqi Sun. Real-time depth completion based on lidar-stereo for autonomous driving. *Frontiers in Neurorobotics*, 17:1124676, 2023.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023.

Xiaoyu Xu, Jian Zou, Sheyang Tang, Zhihua Wang, Jing Liao, and Kede Ma. Learning stable canonical worlds for novel view synthesis and beyond. *arXiv preprint arXiv:2606.23027*, 2026.

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

Jiawei Yang, Jiahui Huang, Boris Ivanovic, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Apoorva Sharma, Maximilian Igl, Peter Karkus, et al. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. In *International Conference on Learning Representations*, volume 2025, pp. 50446–50465, 2025.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20331–20341. IEEE, 2024.

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

Hao Yu, Haotong Lin, Jiawei Wang, Jiaxin Li, Yida Wang, Xueyang Zhang, Yue Wang, Xiaowei Zhou, Ruizhen Hu, and Sida Peng. Infinidepth: Arbitrary-resolution and fine-grained depth estimation with neural implicit fields. *arXiv preprint arXiv:2601.03252*, 2026b.

Zhongrui Yu, Zhao Wang, Yijia Xie, Yida Wang, Xueyang Zhang, Yifei Zhan, and Kun Zhan. Streetforward: Perceiving dynamic street with feedforward causal attention. *arXiv preprint arXiv:2603.19552*, 2026c.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14658–14668, 2026.

A Implementation Details

A.1 Input representation and pretrained features

GeoForward takes two endpoint observations separated by approximately half a second. Each observation contains calibrated surround-view RGB images and LiDAR measurements. We use six cameras and endpoint offsets $(0, 6)$ for the 12 Hz nuScenes sequences, and five cameras and offsets $(0, 5)$ for the 10 Hz Waymo sequences. Images are resized to 280×518 and the intrinsics are rescaled accordingly. For a projected LiDAR pixel with multiple points, we retain its nearest valid depth. We also retain the valid raw LiDAR points and their measurement times for the point-token conditioner described below. Offline dense metric-depth targets are used for training and, where available, evaluation; they are never inputs to the reconstruction network.

We initialize the visual trunk from VGGT with its DINOv2 ViT-L/14 encoder and multi-view aggregator. The pretrained base weights remain fixed. Low-rank adapters provide trainable capacity during the reconstruction phases in which the trunk is unfrozen; the final color-only adaptation freezes the entire reconstruction network. The adapters use rank 8, scaling 32, and dropout 0.05. The four DINO feature levels have zero-based indices $\{8, 12, 16, 20\}$. Each level is compressed to 256 channels before alignment; their fused output has 1024 channels. The VGGT aggregator contributes a separate 2048-channel feature, and a local RGB encoder contributes 128 channels. Thus, the visual depth descriptor has 3200 channels before the additional LiDAR-token condition is concatenated.

A.2 METRIC-GUIDED FEATURE ALIGNMENT

We use the same projected metric anchor for the depth and Gaussian-attribute heads. For a target pixel $\mathbf{p}$ with valid LiDAR prior depth d_a, intrinsic matrix K_a, and an eligible context view b, its initial correspondence is

$$\mathbf{u}^0_{a\to b} = \pi\big(K_b T_{b\leftarrow a}[d_a K_a^{-1}\bar{\mathbf{p}}; 1]\big), \qquad T_{b\leftarrow a} = T_b^{-1}T_a. \tag{7}$$

The transforms are expressed in a common reference-ego frame. The code flattens both endpoints into $2V$ source views and makes the other $2V-1$ views eligible contexts for each target view: 11 contexts for nuScenes and nine for Waymo. Cross-time projection is a static-scene approximation at this feature-association step; moving objects are treated separately during temporal Gaussian composition. Invalid LiDAR support, negative projected depth, absent cameras, and out-of-frame correspondences are masked.

Let $\mathbf{f}_a$ and $\mathbf{f}^0_b$ be target and initially sampled context features. A shared offset network predicts

$$\Delta\mathbf{u}_{a\to b} = 2\tanh\big(f_\Delta([\mathbf{f}_a, \mathbf{f}^0_b])\big), \qquad \widehat{\mathbf{f}}_b = \mathcal{S}(F_b, \mathbf{u}^0_{a\to b} + \Delta\mathbf{u}_{a\to b}), \tag{8}$$

where the bound is ± 2 image pixels per axis and $\mathcal{S}$ denotes bilinear sampling. The offset output is zero-initialized. For a valid context, the fusion weight is the clipped input-depth confidence multiplied by a nonnegative, learned-threshold cosine match between the target and resampled features. We normalize the weighted context sum and fuse it residually into the target feature. A support-dependent gate keeps the original target feature when no valid context exists. Context feature values are detached in this alignment operation, while gradients can still update the sampling coordinates. The four aligned levels are mixed into the 1024-channel feature used by both prediction branches.

A.3 SPATIOTEMPORAL LIDAR-TOKEN CONDITIONING

In addition to providing projected anchors, raw LiDAR points condition metric depth through a two-step attention module. We express each valid point in the reference-ego coordinate system and pair its (x, y, z) coordinate with the measurement-time offset from the first endpoint. A point MLP maps these four values to a 1024-dimensional token. Fixed Fourier features encode position and time for the attention queries and keys; position and time are normalized by 50 m and 1 s, respectively.

First, point tokens attend to all valid, frozen raw VGGT aggregator patches from both endpoints. Second, image-grid queries attend to the enhanced point tokens, producing a 1024-channel condition for each endpoint view. Both steps use eight attention heads. The same LiDAR-token pool is shared across the cameras, rather than constructing a separate point pool for every view. When no valid LiDAR point is available, the conditioner returns a zero condition. The depth head uses this learned condition together with the aligned visual descriptor; sparse LiDAR therefore enters both the explicit correspondence calculation and the learned depth features.

A.4 DEPTH PREDICTION, GAUSSIAN ATTRIBUTES, AND 3D REFINEMENT

An implicit depth head predicts a bounded metric depth d_i at each valid source pixel. The corresponding Gaussian center is obtained by calibrated unprojection into the reference-ego frame. A separate confidence head takes detached visual features. For the attribute branch we form a six-dimensional geometric descriptor consisting of $\log d_i$, the depth-derived 3D normal, the confidence, and camera validity. This descriptor is fused with the shared alignment/aggregator representation and LiDAR-token condition before predicting a six-dimensional rotation representation, anisotropic scale, opacity, and degree-four spherical harmonic appearance. Initial appearance includes a bounded residual around the source RGB color.

The attribute refiner uses metric Gaussian centers only to assign voxels; it does not modify these centers. We quantize the centers into occupied 0.5 m voxels in the common ego frame, mean-pool the 256-dimensional per-Gaussian features within each voxel, and apply one full-width $3\times 3\times 3$ submanifold convolution over occupied neighbors. A zero-initialized linear readout maps the resulting voxel features to additive residuals for the raw Gaussian attributes. Each Gaussian receives the residual of its own voxel before the attribute activations are applied. The formal checkpoints use this single-layer attribute aggregation module.

During full-window adaptation, a bounded depth adapter can further correct the predicted center depth. Its zero-initialized head predicts a residual in log depth, so that

$$d_i' = d_i \exp(0.7 \tanh h(\mathbf{q}_i)) . \tag{9}$$

The implemented `wide_v3` input $\mathbf{q}_i$ combines compact Gaussian features, a fixed 64-dimensional sketch of depth features, the base depth, confidence, and local log-depth statistics. The correction is bounded and starts as an identity map. The corrected depth supplies the center used by the renderer; in this adaptation route its depth-supervision gradient updates the adapter, while the center value passed to the Gaussian decoder is detached. Accordingly, the depth and attribute paths should not be described as a single jointly trained U-Net.

A.5 Temporal composition and endpoint color calibration

Gaussians are decoded at both endpoints. A frozen SAM 2 model receives projected tracked-object boxes as prompts and produces source-image masks. For accepted, tracked moving objects, we compute velocity from matched endpoint annotations using the actual time difference and express it in the common reference frame. For a Gaussian originating at time t_i,

$$\boldsymbol{\mu}_i(\tau) = \boldsymbol{\mu}_i(t_i) + (\tau - t_i)\mathbf{v}_i. \tag{10}$$

Gaussians not assigned to a tracked moving object receive zero velocity. After translation, the Gaussians from the two endpoints and all available cameras are composited for intermediate views. This implementation uses annotation-based constant-velocity translation; it does not use STORM's learned motion tokens or OmniRe's articulated scene-graph nodes.

The final checkpoint additionally contains an endpoint color head. For each source Gaussian, a small FiLM predictor maps its 256-dimensional feature to three scale and three bias values. It changes only the degree-zero color coefficient,

$$\mathbf{c}'_{i,0} = (1 + \boldsymbol{\gamma}_i) \odot \mathbf{c}_{i,0} + \boldsymbol{\beta}_i. \tag{11}$$

The final layer is zero-initialized, and the formal configuration conditions this head on Gaussian features without the optional raw-RGB encoder. The correction is applied to endpoint source-view rendering only. It is not used for intermediate-time compositing and does not change predicted geometry.

A.6 Optimization and supervision

The checkpoint was produced through a sequence of endpoint reconstruction, full-window adaptation, and a final endpoint color-only calibration. These are successive optimization configurations of the same v6.11 model, not three independently designed prediction networks. For nuScenes, the documented adaptation starts from a Waymo-trained checkpoint. Its endpoint phase updates the reconstruction path while keeping the optional depth adapter and FiLM head frozen; the full-window phase freezes the visual/depth trunk and trains the depth adapter and Gaussian head; the final phase freezes the reconstruction path and trains only the FiLM color head. The documented Waymo final pass likewise trains only FiLM from its full-window predecessor. The base VGGT weights remain frozen throughout, while LoRA trainability depends on the active phase.

For a rendered target $\widehat{I}$ and RGB observation I, the implemented reconstruction term is

$$\mathcal{L}_{\mathrm{rgb}} = \mathrm{MSE}(\widehat{I}, I) + 0.05\ \mathrm{LPIPS}_{\mathrm{VGG}}(\widehat{I}, I). \tag{12}$$

Its configured weight is 2.0. Endpoint depth is supervised by Smooth-L1 against valid cached metric-depth targets; the configured endpoint depth weight in the final full-window adaptation is 1.0. The endpoint image objective also includes a multiscale Laplacian reconstruction term and Gaussian scale/opacity regularization. The formal launch configurations use a scale penalty weight of 0.02 and an opacity penalty weight of 0.01. They turn off the earlier contour, bend, multiscale depth, and second-order depth terms. They also turn off direct sparse-LiDAR depth and confidence losses in the final temporal route. At inference, the sparse LiDAR input is still used for alignment and point-token conditioning.

During the full-window phase, training renders selected endpoint and intermediate frames and supervises their RGB images; the set of intermediate frames sampled per step is controlled by the

training launcher. The code also computes a pose-based reprojection residual using 0.85 brightness-normalized ℓ_1 and 0.15 SSIM on valid static regions. In the supplied training implementation, predicted depth is detached before this warp, and both compared images are observations. Consequently, this residual has no gradient with respect to the model parameters; it is a diagnostic term rather than an effective learning constraint in the reported implementation. The final FiLM-only phase receives gradients from endpoint RGB reconstruction and leaves intermediate images and all geometry unchanged.

B DATASETS

nuScenes. The nuScenes dataset Caesar et al. (2020) contains 1,000 driving scenes of approximately 20 seconds each, with 700, 150, and 150 scenes in the official training, validation, and test splits. We use all six surround-view cameras and sparse LiDAR. Training uses the official training scenes, including 12 Hz interpolated sequences. Each reconstruction window takes two endpoint observations six frames apart, spanning approximately 0.5 s. Images are resized to 280×518 (height $\times$ width), with camera intrinsics scaled accordingly. The main comparison uses 14 validation scenes; the separate full-validation audit covers all 150 validation scenes.

Waymo. The Waymo Open Dataset Sun et al. (2020) provides calibrated camera images, LiDAR measurements, and vehicle poses. We use its five cameras at 10 Hz and resize their images to 280×518, updating the intrinsics accordingly. The two endpoint observations are five frames apart, again spanning approximately 0.5 s. The separate full-validation audit covers all 202 official validation contexts.

Zero-shot datasets. We evaluate geometric generalization on Argoverse 2 and KITTI Tracking without fine-tuning on either dataset. For Argoverse 2, all compared methods use the same ten validation logs:

```
05fa5048-f355-3274-b565-c0ddc547b315  0b86f508-5df9-4a46-bc59-5b9536dbde9f  185d3943-dd15-397a-8b2e-69cd86628fb7
25e5c600-36fe-3245-9cc0-40ef91620c22  27be7d34-ecb4-377b-8477-ccfd7cf4d0bc  280269f9-6111-311d-b351-ce9f63f88c81
2f2321d2-7912-3567-a789-25e46a145bda  3bffdcff-c3a7-38b6-a0f2-64196d130958  44adf4c4-6064-362f-94d3-323ed42cfda9
                                      5589de60-1727-3e3f-9423-33437fc5da4b
```

For KITTI Tracking, we use the stereo cameras `image_02` and `image_03`. All compared methods use the same three sequences and frame ranges: `0001` (380–430), `0002` (140–220), and `0006` (65–120).

C BASELINES

We briefly describe each method included in the nuScenes or Waymo comparison, focusing on its scene representation and treatment of dynamics.

• **Street Gaussians** Yan et al. (2024) reconstructs the background and foreground vehicles with separate Gaussian sets. Tracked vehicle poses place the foreground Gaussians in the scene, and the method refines these poses during scene-specific optimization. It also models time-dependent appearance with four-dimensional spherical harmonics. This object-centric baseline lets us compare feed-forward prediction with an optimized scene whose moving vehicles are represented explicitly.

• **PVG** Chen et al. (2026b) represents a dynamic urban scene using Gaussians with periodic motion and time-varying visibility. Its parameters describe vibration, lifespan, and the time at which opacity peaks. PVG optimizes one temporal Gaussian representation per scene without requiring annotated object boxes to assign motion to individual instances. We include it to compare our reconstructed geometry and appearance with a scene-optimized model of continuous temporal change.

• **DeformableGS** Yang et al. (2024) places Gaussians in a canonical space and predicts timestamp-dependent changes to their properties with a deformation network. These changes move the canonical representation to the observed state of the scene. The method was introduced for monocular dynamic reconstruction; its deformation formulation provides a comparison for driving scenes with substantial camera motion and multiple viewpoints.

• **OmniRe** Chen et al. (2025) organizes an urban scene into a Gaussian scene graph rather than assigning every actor the same motion model. It reconstructs a static background, rigid actors,

articulated pedestrians, and other deformable objects with their corresponding representations. The nodes are optimized jointly for each scene. OmniRe therefore provides a strong comparison for visual fidelity and geometry when instance structure and per-scene optimization are available.

• **DrivingForward** Tian et al. (2025) predicts a driving scene's Gaussian representation from flexible surround-view input. It couples depth and Gaussian prediction with image-based supervision from neighboring spatial and temporal observations, allowing predictions from separate cameras to be placed in a common scene. We compare with it to examine the contribution of explicit metric-guided correspondence and LiDAR-conditioned depth in a feed-forward setting.

• **ReconDrive** Yu et al. (2026a) builds on VGGT features and uses distinct heads for Gaussian-center depth and the remaining Gaussian attributes. It aligns endpoint Gaussians in time using object velocities and combines them for dynamic rendering. ReconDrive is a closely related feed-forward baseline for assessing projected cross-view alignment and LiDAR-token conditioning in a depth-and-attribute prediction framework.

• **DGGT** Chen et al. (2026a) reconstructs dynamic driving scenes from sparse images without requiring input camera poses. It jointly estimates camera parameters and per-frame Gaussian maps, while separate dynamic predictions describe Gaussian motion and temporal visibility. This baseline tests rendering and geometric quality when the model must also infer camera geometry rather than relying on calibrated input poses.

• **STORM** Yang et al. (2025) predicts dynamic 3D Gaussians and their motion from sparse posed observations in a single forward pass. Learned motion tokens define a set of velocity bases, and image-dependent weights compose these bases into per-Gaussian motion. Transforming and aggregating Gaussians across observed frames allows the method to supervise motion through reconstruction. We include STORM on Waymo as a feed-forward comparison with learned scene motion.

• **StreetForward** Yu et al. (2026c) reconstructs a dynamic street scene without input poses or object tracks. Its temporal masked attention extends VGGT-style feature aggregation to extract motion-aware features from image sequences. The method predicts a unified Gaussian representation for static and moving content together with pixelwise velocities. We include it to compare with a feed-forward system that learns motion without annotated tracks.

• **Envision4D** Song et al. (2026) addresses future-view extrapolation with a feed-forward 4D Gaussian representation. It predicts future camera poses through iterative denoising and models nonlinear scene evolution using temporal attention and conditioned motion lifting. Progressive training stabilizes its future-frame predictions. Its Waymo results provide a comparison to a model designed for future appearance and motion prediction.

D EVALUATION

We provide detailed definitions of the image reconstruction and geometric metrics used in our experiments as follows.

Image Reconstruction Quality. Following ReconDrive Yu et al. (2026a), we report PSNR, SSIM, and LPIPS for both visual scene reconstruction and novel-view synthesis. Let $\hat{I}$ and I denote the rendered and reference images, respectively, and let M be the total number of evaluated pixel values. These metrics are defined as

$$\mathrm{MSE} = \frac{1}{M}\sum_{j=1}^{M}(\hat{I}_j - I_j)^2, \qquad \mathrm{PSNR} = 10\log_{10}\left(\frac{L^2}{\mathrm{MSE}}\right),$$

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)},$$

$$\mathrm{LPIPS}(I,\hat{I}) = \sum_l \frac{1}{H_l W_l}\sum_{h,w}\left\|\mathbf{w}_l \odot \left(\tilde{\phi}_l(I)_{hw} - \tilde{\phi}_l(\hat{I})_{hw}\right)\right\|_2^2.$$

Here, L is the maximum image intensity. For SSIM, x and y denote corresponding local image windows; μ, σ^2, and σ_{xy} are their local means, variances, and covariance, while C_1 and C_2 are

stabilizing constants. For LPIPS, $\tilde{\phi}_l(\cdot)$ denotes normalized deep features at layer l, and $\mathbf{w}_l$ denotes the corresponding channel weights. PSNR measures pixel-level fidelity, SSIM evaluates luminance, contrast, and structural similarity, and LPIPS measures perceptual discrepancy in deep feature space. Higher PSNR and SSIM and lower LPIPS indicate better image quality.

Point-Cloud Reconstruction. Following DVGT Zuo et al. (2026), let $\mathcal{P}$ and $\mathcal{G}$ denote the predicted and reference point clouds, respectively. Accuracy and Completeness are defined as

$$\text{Accuracy} = \frac{1}{|\mathcal{P}|}\sum_{\mathbf{p}\in\mathcal{P}}\min_{\mathbf{g}\in\mathcal{G}}\|\mathbf{p}-\mathbf{g}\|_2, \qquad \text{Completeness} = \frac{1}{|\mathcal{G}|}\sum_{\mathbf{g}\in\mathcal{G}}\min_{\mathbf{p}\in\mathcal{P}}\|\mathbf{g}-\mathbf{p}\|_2.$$

Accuracy measures the fidelity of the predicted geometry, whereas Completeness measures its coverage of the reference geometry. Following PGSR Chen et al. (2024a), the symmetric Chamfer Distance is

$$\text{CD} = \text{Accuracy} + \text{Completeness},$$

which jointly accounts for reconstruction fidelity and coverage.

Depth Accuracy. Let $\hat{d}_i$ and d_i denote the predicted and reference metric depths at valid pixel i, respectively, and let N be the number of valid pixels. Following DVGT Zuo et al. (2026), we use ray depth, defined as the Euclidean distance from a reconstructed 3D point to the ego-vehicle center of its corresponding frame. The depth metrics are

$$\text{AbsRel} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{d}_i - d_i|}{d_i}, \qquad \text{SqRel} = \frac{1}{N}\sum_{i=1}^{N}\frac{(\hat{d}_i - d_i)^2}{d_i},$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{d}_i - d_i)^2}, \qquad \delta_{1.25} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left[\max\left(\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i}\right) < 1.25\right].$$

AbsRel measures average relative depth error. SqRel and RMSE assign greater penalties to large depth deviations, with RMSE directly measuring their magnitude in metric units. The threshold accuracy $\delta_{1.25}$ reports the fraction of predictions lying within a multiplicative factor of 1.25 of the reference depth.

Cross-Camera Consistency. Following CylinderDepth Abualhanud et al. (2026), DepthCons measures whether overlapping cameras predict consistent metric distances for the same physical points. Correspondences are established using reference depth and known metric relative poses, after which the corresponding predictions are expressed in a common ego coordinate system. Let r_i^a and r_i^b denote the predicted distances to the common origin from cameras a and b. DepthCons is defined as

$$\text{DepthCons} = \sqrt{\frac{1}{N_c}\sum_{i=1}^{N_c}(r_i^a - r_i^b)^2},$$

where N_c denotes the number of valid cross-camera correspondences. DepthCons measures agreement between views rather than their individual accuracy with respect to the reference geometry.

Lower Accuracy, Completeness, CD, AbsRel, SqRel, RMSE, and DepthCons indicate better geometric quality, whereas a higher $\delta_{1.25}$ is preferred.

E Additional Results

Figure 6 shows an intersection with a foreground car and a tree-lined road with a truck. Removing LiDAR-guided alignment introduces visible horizontal artifacts on the nearby road and makes the curb and depth transitions less stable. Without the Gaussian head, the car, truck, and roadside details appear softer; the depth boundary around the truck also becomes broader. Without motion compensation, the car shows a displaced, ghosted outline, and its predicted depth extends beyond the compact vehicle silhouette in the full model. The truck boundary in the second scene likewise

becomes less distinct. With all three components, GeoForward retains clearer vehicle contours and more continuous road surfaces in both RGB and depth.

Figure 7 compares a strongly backlit street view and two scenes with prominent roadside trees. Envision4D preserves much of the road appearance, but its depth maps smooth over tree structure, especially in the second scene. StreetForward renders recognizable RGB images, while its depth maps contain broad, nearly uniform regions around vegetation and miss thin branches. STORM loses detail in the backlit RGB view and produces fragmented depth around tree canopies. PVG retains lane markings and the overall RGB layout, but individual tree silhouettes are less distinct in depth; in the second scene, broad horizontal depth bands replace much of the tree structure. Deformable3DGS produces noticeably blurred RGB images in the first and third scenes, together with nearly uniform depth over large regions. GeoForward retains road markings and roadside objects while producing more distinguishable tree trunks, vegetation boundaries, and road regions in depth. Fine branches remain less detailed than in the ground truth, and regions without displayed ground-truth depth should not be interpreted as pixelwise depth agreement.

Figure 8 presents GeoForward's outputs from all six cameras across three nuScenes scenes. The views cover different road directions, building facades, vegetation, and lighting conditions. The RGB renderings retain recognizable scene structure in both forward-facing and rear-facing views. In the predicted depth maps, road surfaces form largely continuous regions, while vehicles, buildings, and trees introduce distinct depth boundaries. These examples complement the single-view comparisons by showing the reconstructed appearance and geometry around the vehicle.

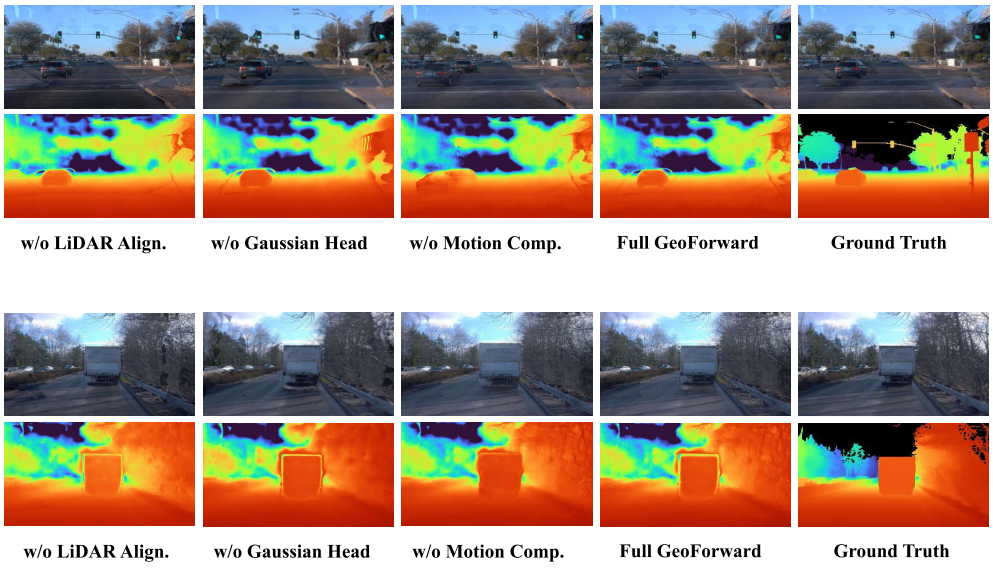


Figure 6: **Qualitative ablation results on nuScenes.** For each of two scenes, the upper row shows RGB renderings and the lower row shows depth maps. We compare variants without LiDAR-guided alignment, the Gaussian head, or motion compensation with the full GeoForward model and ground truth (GT).

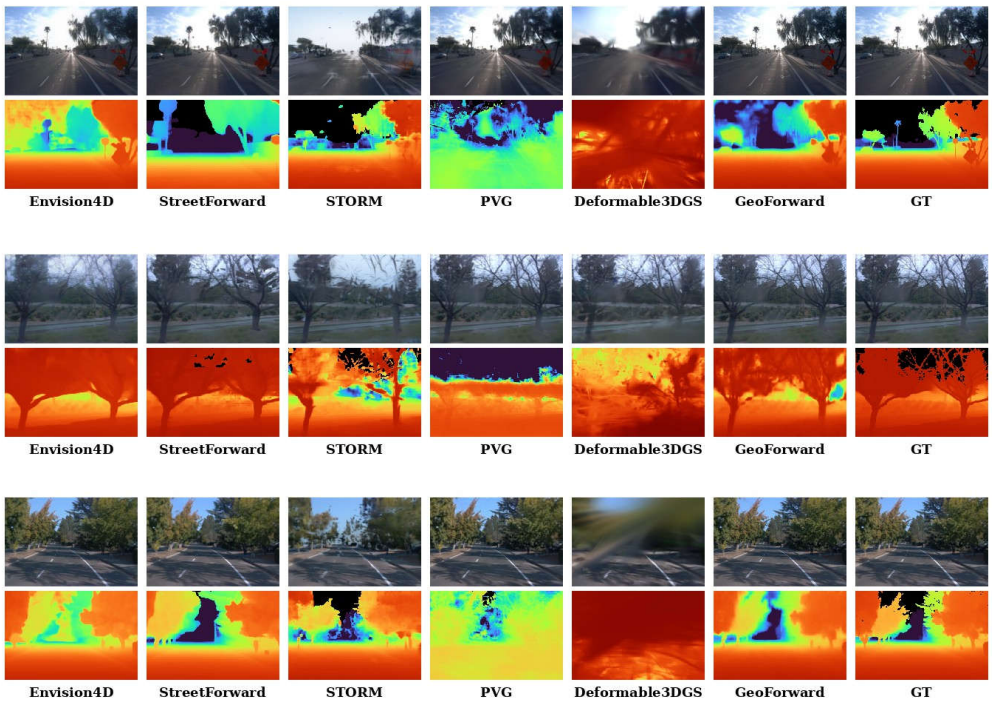


Figure 7: **Qualitative comparison with baselines.** Three driving scenes are shown, each with RGB renderings above the corresponding depth maps. Columns compare Envision4D, StreetForward, STORM, PVG, DeformableGS, GeoForward, and ground truth (GT).

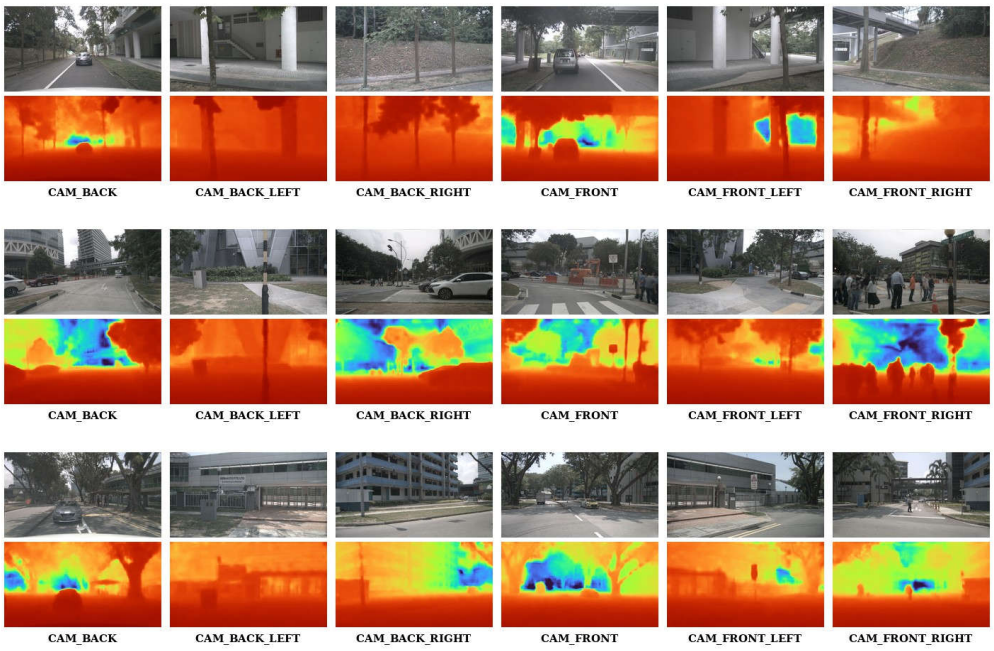


Figure 8: **Six-camera qualitative results on nuScenes.** Each row shows one scene. From left to right, the columns correspond to CAM_BACK, CAM_BACK_LEFT, CAM_BACK_RIGHT, CAM_FRONT, CAM_FRONT_LEFT, and CAM_FRONT_RIGHT. Each camera view shows the rendered RGB image above its colorized predicted depth map.