LiGRe: Grounding Feed-Forward Gaussian Reconstruction in LiDAR Geometry for Joint RGB and LiDAR Rendering

Anonymous authors
Paper under double-blind review

Abstract

Autonomous driving simulation requires camera and LiDAR rendering beyond recorded trajectories. However, sparse images leave surface geometry and object motion ambiguous, so reconstructions that match the inputs may produce incorrect occlusions and LiDAR ranges at unseen viewpoints and times. LiDAR constrains observed surfaces, but hidden geometry and motion between frames remain uncertain. We introduce LiGRe (LiDAR-Grounded Reconstruction), a feed-forward framework that models these remaining possibilities as a conditional distribution over dynamic Gaussian scenes from sparse images and LiDAR scans. LiGRe gathers features near surface positions predicted by preliminary motion estimates, using LiDAR rays and measured distances to weight observations across views and timestamps. The combined features condition a measurement-budgeted prior over shared geometry codes, with LiDAR measurements guiding their mean and variance. Each return's weight is divided across the codes it supports, so reuse by multiple Gaussians does not multiply its total weight in the prior. A coupled decoder then uses the same sampled codes to predict Gaussian geometry and motion trajectories, yielding a dynamic scene for RGB and LiDAR rendering and actor editing without per-scene optimization. Experiments on the Waymo Open Dataset, Argoverse 2, PandaSet, and Para-Lane demonstrate improved RGB reconstruction and off-trajectory synthesis over the evaluated feed-forward baselines, together with competitive RGB and LiDAR rendering against per-scene optimization methods.

1 Introduction

Reconstructing real driving logs into controllable 3D scenes provides a practical basis for autonomous driving simulation. In closed-loop evaluation, the ego vehicle may deviate from the recorded path, requiring observations from previously unseen sensor poses. Rendering camera and LiDAR observations from shared geometry allows both modalities to describe the same environment, while actor editing enables alternative traffic scenarios. These applications motivate efficient scene reconstruction that supports joint sensor rendering and editing beyond the recorded trajectory.

Neural Radiance Fields (NeRFs) and 3D Gaussian Splatting (3DGS) have enabled high-quality scene reconstruction (Mildenhall et al., 2021; Kerbl et al., 2023), with subsequent work extending these representations to dynamic driving scenes and sensor simulation (Yang et al., 2023; Tonderski et al., 2024; Yan et al., 2024; Chen et al., 2025; Hess et al., 2025). Many methods optimize a separate representation for each sequence, making reconstruction over large collections of driving logs expensive. Recent advances in feed-forward geometry and Gaussian prediction (Wang et al., 2025; Jiang et al., 2025; Yang et al., 2025; Chen et al., 2026a; Yu et al., 2026a) reduce this cost by learning to infer scene representations directly from observations. Their progress has largely focused on visual reconstruction and novel-view synthesis, while joint RGB and LiDAR rendering from feed-forward dynamic scenes remains less explored.

A further challenge is that sparse driving images leave scene geometry and motion weakly constrained. Images captured along a narrow trajectory provide limited views of surfaces, so a reconstruction can match the input images without accurately recovering the underlying 3D geometry. Sparse temporal observations also make it difficult to infer object motion between observed frames.

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

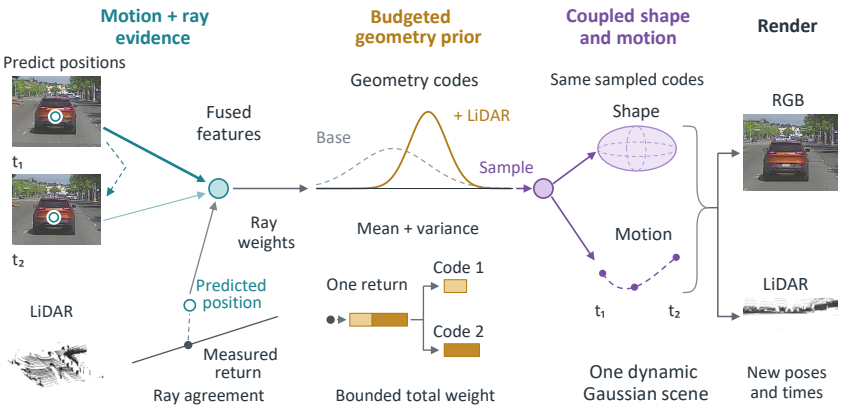

Figure 1: **Overview of LiGRe.** LiGRe uses preliminary motion estimates to gather features near predicted surface positions across views and times, weighting observations by their agreement with LiDAR rays and measured ranges. The fused features condition a distribution over geometry codes, with LiDAR measurements guiding its mean and variance. Each return's weight is divided among the codes it supports, so reusing a measurement does not multiply its total weight in the prior. A coupled decoder uses the same sampled geometry codes to predict Gaussian geometry and motion, producing one dynamic scene for RGB and LiDAR rendering at new poses and times without per-scene optimization.

When rendering at off-trajectory viewpoints or unobserved times, these geometry and motion errors can produce incorrect occlusions in rendered images and inaccurate distances in synthesized LiDAR scans. These limitations motivate geometric measurements that complement image appearance.

LiDAR measurements help reduce this ambiguity by directly constraining the locations of observed surfaces. However, sparse RGB and LiDAR observations still leave hidden surfaces and motion between observed frames only partially determined. Multiple shapes and object trajectories may therefore remain compatible with the available observations. To represent these alternatives, we learn a conditional distribution over dynamic Gaussian scenes, using measured geometry to guide reconstruction while jointly modeling the remaining variation in shape and motion. Each sampled scene is then used for both RGB and LiDAR rendering across query viewpoints and timestamps.

We introduce LiGRe (LiDAR-Grounded Reconstruction), a feed-forward framework that constructs this scene distribution from sparse images and LiDAR scans. Our motion- and ray-conditioned evidence aggregation uses preliminary motion estimates to gather features near each surface's predicted positions across views and timestamps. It compares these positions with LiDAR rays and measured distances to determine how strongly each observation contributes. Our measurement-budgeted geometry prior then uses the aggregated features to predict a distribution over geometry codes that describe scene shape. LiDAR measurements guide the codes' mean and variance. When several Gaussians share one LiDAR return, we divide its weight among their geometry codes, so reusing that measurement does not multiply its total weight in the prior. Finally, our shape–motion coupled decoder uses the same sampled geometry codes to predict both Gaussian geometry and motion trajectories, allowing the model to learn how shape and motion vary together.

Experiments on the Waymo Open Dataset (Sun et al., 2020), Argoverse 2 (Wilson et al., 2023), PandaSet (Xiao et al., 2021), and Para-Lane (Ni et al., 2025) demonstrate improved RGB reconstruction and off-trajectory synthesis over the evaluated feed-forward baselines. LiGRe also achieves competitive RGB and LiDAR rendering against per-scene optimization methods and supports actor editing, without requiring per-scene optimization for either reconstruction or editing.

2 Related Work

Optimization-Based Driving Scene Reconstruction. Optimization-based methods reconstruct driving sequences using neural fields (Ost et al., 2021; Wu et al., 2023; Yang et al., 2023; Ton-

derski et al., 2024; Yang et al., 2024; Sun et al., 2024) or Gaussian representations (Zhou et al., 2024b; Yan et al., 2024; Zhou et al., 2024a; Chen et al., 2026b; 2025; Peng et al., 2025; Zhou et al., 2025; Hess et al., 2025). Several also render camera and LiDAR observations from shared geometry (Yang et al., 2023; Tonderski et al., 2024; Hess et al., 2025). However, fitting each sequence separately limits reconstruction scalability. LiGRe predicts an editable dynamic Gaussian scene in one forward pass, supporting joint RGB–LiDAR rendering without per-scene optimization.

Feed-Forward Driving Scene Reconstruction. Feed-forward methods predict visual geometry (Wang et al., 2024; 2025; 2026a;b) or renderable Gaussian representations (Charatan et al., 2024; Chen et al., 2024; Wewer et al., 2024; Ye et al., 2025; Jiang et al., 2025) directly from images. Extensions to driving scenes exploit multi-view and temporal observations for efficient reconstruction (Tian et al., 2025; Yang et al., 2025; Fei et al., 2024; Lu et al., 2026; Miao et al., 2025; Chen et al., 2026a; He et al., 2026; Chi et al., 2026; Yu et al., 2026b;a). However, sparse observations leave hidden geometry and motion between frames ambiguous. LiGRe learns a LiDAR-conditioned distribution over dynamic Gaussian scenes, jointly decoding shape and motion from shared latent codes for RGB–LiDAR rendering.

3 METHOD

Overview. LiGRe reconstructs a dynamic Gaussian scene from context observations $\mathcal{O}$: sparse RGB images, local-frame LiDAR scans, known camera intrinsics, fixed inter-sensor extrinsics, and timestamps. We gather observations along estimated motion and sample geometry codes to jointly decode shape and motion, representing ambiguity in unobserved surfaces and motion between frames. Each sampled scene supports RGB and LiDAR rendering at new poses and times. Figure 1 summarizes the framework.

Encoding and Metric Initialization. A shared PointNet++ encoder (Qi et al., 2017) extracts LiDAR features, which calibrated projection pools into same-time RGB patches. Learned residual gates control their addition to RGB tokens, combining image appearance with measured geometry. VGGT-Ω (Wang et al., 2026a) then predicts features, depths, and poses. Using the first reference-camera frame as scene coordinates, we recover one metric scale from the median measured-to-predicted camera-axis depth ratio at valid context LiDAR projections into that camera. We scale predicted depths and reference-camera translations before composing poses with fixed metric extrinsics; LiDAR coordinates and sensor baselines remain unscaled.

Back-projected depths and aligned LiDAR initialize a fixed number of queries, each anchored at position r_i and time τ_i and decoded into one Gaussian. A softmax distributes each return j across nearby same-time queries and a null option for leaving it unassigned, giving weights a_{ji} with $\sum_i a_{ji} \leq 1$. A multilayer perceptron combines image features, normalized a_{ji}-weighted LiDAR features encoding point-to-query offsets, and total assigned weight to describe appearance, geometry, and measurement support. Missing modality features are zero.

Motion- and Ray-Conditioned Evidence Aggregation. To follow moving surfaces, a preliminary geometry–motion head uses initial query features to predict positions at source times. We gather image features at their projections under estimated poses and nearby LiDAR features. For each available return, we encode the predicted position's signed along-ray distance from the return and perpendicular offset vector from its ray to assess geometric agreement. These residuals, gathered features, ray measurements, time offsets from τ_i, and modality-validity indicators drive one residual attention update, producing h_i. Softmax attention includes a zero-valued null token to ignore unhelpful observations; image evidence remains usable without LiDAR.

Measurement-Budgeted Geometry Prior. We represent possible completions with M latent slots, each holding codes shared across queries so multiple Gaussians can vary together. Query-to-slot softmax weights s_{ik}, predicted from h_i, satisfy $\sum_k s_{ik} = 1$. Normalized pooling of h_i with these weights and a learned slot embedding gives context features c_k. A return can support several slots; we divide its weight as $r_{jk} = \sum_i a_{ji} s_{ik}$, ensuring $\sum_k r_{jk} \leq 1$.

Each return feature predicts a code proposal v_j and nonnegative precision increment λ_j, computed once per physical return and shared across its camera projections. Precision is inverse latent vari-

ance. With context mean m_k^0 from c_k and learned, context-independent positive base precision τ_k^0, the geometry prior is

$$\begin{aligned} \tau_k^g &= \tau_k^0 + \sum_j r_{jk}\lambda_j, \\ m_k^g &= \frac{\tau_k^0 \odot m_k^0 + \sum_j r_{jk}\lambda_j \odot v_j}{\tau_k^g}, \\ z_k^g &\sim \mathcal{N}\big(m_k^g, \operatorname{diag}\big((\tau_k^g)^{-1}\big)\big). \end{aligned} \tag{1}$$

Operations are elementwise. The mean blends context and measurement proposals, while routing bounds each return's total direct precision increment by λ_j per channel. Separate appearance codes z_k^a have Gaussian means and variances predicted from c_k, since geometry does not determine unseen texture. Together, $Z = \{z_k^g, z_k^a\}_{k=1}^M$ follows $p(Z \mid \mathcal{O})$.

Shape–Motion Coupled Gaussian Decoding. Each query receives geometry and appearance codes pooled with s_{ik}. Predictions from h_i reconstruct visible detail, while residuals conditioned on h_i and sampled codes allow alternative completions. Geometry codes affect shape and RGB/LiDAR attributes; appearance codes affect only attributes. Outputs specify a free center offset from r_i, valid rotation, positive scale, and opacity. Rotation and scale define spatial covariance.

A temporal head maps c_k, z_k^g, and t to trajectory $b_k(t)$, coupling motion to the same geometry sample:

$$\mu_i(t) = \mu_i + \rho_i \left[d_i^{\text{base}}(h_i, t) + \sum_k s_{ik}\big(b_k(t) - b_k(\tau_i)\big) \right]. \tag{2}$$

Here, μ_i is the reference center and feature-based motion d_i^{base} is zero at τ_i, anchoring $\mu_i(\tau_i) = \mu_i$. A semantic head on h_i predicts movable-class probability $\rho_i \in [0, 1]$ to suppress static-content motion; movable objects can still have zero displacement. Trajectories support interpolation and extrapolation; covariance, opacity, and appearance parameters stay fixed over time. We reuse one scene sample across poses and times. Actor editing removes or transforms instance-associated Gaussians for both renderers; slots need not correspond to objects.

Rendering and Training. RGB rendering alpha-composites view-dependent Gaussian colors with a sky background. LiDAR rendering projects the same scene onto a spherical range image, aggregates range and features by visibility, and decodes intensity and ray-drop (no-return) probability from features and ray directions. The observation loss $\mathcal{L}_{\text{obs}}$ combines RGB ℓ_1/perceptual, compositing-weighted range, intensity/drop, and available semantic/motion losses. A coverage term penalizes insufficient accumulated weight on measured-return rays; predicted drop cannot mask their geometry losses.

A conditional variational autoencoder learns completions from additional frames $\mathcal{Y}$ on the recorded trajectory that can reveal hidden surfaces. Its training-only diagonal Gaussian posterior $q(Z \mid \mathcal{O}, \mathcal{Y})$ uses context-defined slots and separately encoded targets with shared encoder weights. Estimated poses, scale, query features, assignments, and prior parameters depend only on $\mathcal{O}$. For decoded scene $\mathcal{G}_Z$, we minimize

$$\begin{aligned} \mathcal{L} = \mathbb{E}_{Z\sim q}[\mathcal{L}_{\text{obs}}(\mathcal{G}_Z; \mathcal{O} \cup \mathcal{Y})] + \beta D_{\text{KL}}(q\|p) \\ + \lambda_{\text{ctx}}\mathbb{E}_{Z\sim p}[\mathcal{L}_{\text{obs}}(\mathcal{G}_Z; \mathcal{O})] + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}}. \end{aligned} \tag{3}$$

Posterior reconstruction uses context and target frames; KL divergence aligns the context prior with this posterior. Prior-context reconstruction encourages samples to preserve input measurements. Ground-truth relative poses supervise translation and rotation through $\mathcal{L}_{\text{pose}}$ and define training rendering poses in the first reference-camera frame; scene construction uses estimated poses. Inference samples the context prior and decodes one scene without per-scene optimization.

4 Experiments

4.1 Experimental Setup

Datasets and protocols. We evaluate LiGRe on the Waymo Open Dataset (WOD) (Sun et al., 2020), Argoverse 2 (Wilson et al., 2023), PandaSet (Xiao et al., 2021), and ParaLane (Ni et al., 2025). On

WOD, Argoverse 2, and PandaSet, both evaluation tracks use three-camera inputs (front, front-left, and front-right). For Para-Lane, we evaluate off-trajectory RGB rendering using front-camera images, as the dataset release used in our experiments does not include other camera views. For feed-forward comparisons, following STORM (Yang et al., 2025), each 20-frame window at 10 Hz uses context indices $\{0, 5, 10, 15\}$, with the remaining 16 frames reserved for temporal novel-view synthesis (NVS). For comparisons with per-scene optimization methods, we adopt the alternating-frame NVS split used by SplatAD (Hess et al., 2025). Per-scene baselines are optimized on even-indexed frames and evaluated on held-out odd-indexed frames. We train a separate LiGRe model for this track using the same even/odd context–target partition on training sequences only. At evaluation, LiGRe reconstructs each scene from the even-indexed observations without parameter updates. Within each track, methods use the same temporal extent, context timestamps, and evaluation timestamps. At evaluation, RGB and LiDAR observations used for reconstruction are restricted to context timestamps.

For both tracks, we report PSNR, SSIM, and LPIPS for RGB reconstruction at context timestamps and temporal NVS at held-out timestamps. We use the same metrics for off-trajectory RGB rendering on Para-Lane, with held-out cross-lane images as ground truth. For off-trajectory RGB rendering without paired ground truth and actor editing, we report pooled FID. We evaluate LiDAR rendering on WOD, Argoverse 2, and PandaSet under the alternating-frame protocol. Following SplatAD (Hess et al., 2025), we report median squared range error (MedSE), Chamfer distance (CD), and ray-drop accuracy at recorded viewpoints. For laterally shifted viewpoints, we report MedSE and CD against pseudo-references.

Baselines. We include only baselines with publicly available training and inference code and conduct all baseline experiments ourselves using their official implementations. Feed-forward baselines include STORM (Yang et al., 2025), ReconDrive (Yu et al., 2026a), DGGT (Chen et al., 2026a), AnySplat (Jiang et al., 2025), and NoPoSplat (Ye et al., 2025). Per-scene comparisons include SplatAD (Hess et al., 2025), OmniRe (Chen et al., 2025), PVG (Chen et al., 2026b), Street Gaussians (Yan et al., 2024), Difix3D+ (Wu et al., 2025), and StreetCrafter (Yan et al., 2025). We evaluate feed-forward and per-scene optimization baselines in separate tracks using their official implementations, with matched camera settings and metric computation within each track. LiDAR comparisons include SplatAD and depth-based point-cloud rendering adaptations of OmniRe, PVG, and Street Gaussians following SplatAD.

Implementation details. LiGRe uses a pretrained VGGT-Ω backbone (Wang et al., 2026a) with LoRA (rank 16, $\alpha = 32$) and a shared PointNet++ LiDAR encoder (Qi et al., 2017). We train separate models for the two protocols using identical architecture and optimization settings. Training uses AdamW for 30 epochs on four NVIDIA RTX PRO 6000 GPUs with a global batch size of four scene windows. Learning rates are 10^{-5} for LoRA and the final camera-pose prediction MLP, and 10^{-4} for other trainable modules, with weight decay 10^{-2} and cosine decay following 5% linear warmup. Inference requires no per-scene optimization.

4.2 RGB Reconstruction and Temporal Novel-View Synthesis

Table 1 compares three-camera RGB reconstruction and temporal NVS on Waymo, Argoverse 2, and PandaSet. Recon is evaluated at observed timestamps and NVS at held-out timestamps. The feed-forward track uses four context timestamps per 20-frame window, whereas the per-scene track uses half of the timestamps as context under the alternating-frame split.

LiGRe outperforms all feed-forward baselines across the three metrics and datasets for both reconstruction and NVS. Using a separately trained model for the alternating-frame track, LiGRe also achieves competitive reconstruction and NVS performance against per-scene optimization methods, despite requiring no test-time optimization on the evaluation scenes. Figure 2(a,b) compares temporal NVS on Waymo and Argoverse 2, showing LiGRe's reconstruction of vehicle contours and road markings.

Table 1: Three-camera RGB reconstruction (Recon) and temporal novel-view synthesis (NVS). The two tracks use their respective context–target protocols and are ranked separately. Best and second-best available values within each track are marked in **bold** and underline. LiGRe requires no per-scene optimization in either track.

	Waymo						Argoverse 2						PandaSet					
	Recon			NVS			Recon			NVS			Recon			NVS		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Feed-forward comparison																		
STORM	26.738	0.8468	0.1814	26.181	0.8205	0.2505	27.187	0.8508	0.2013	25.176	0.7879	0.2462	26.328	0.8120	0.2658	23.218	0.7283	0.2935
ReconDrive	25.236	0.7954	0.2017	23.937	0.7193	0.2743	25.860	0.8109	0.1732	23.987	0.7469	0.2381	25.560	0.7698	0.2428	22.707	0.6769	0.2872
DGGT	30.697	0.8797	0.0780	27.659	0.8409	0.1023	29.613	0.8981	0.0810	26.813	0.8343	0.1098	28.052	0.8535	0.1573	25.574	0.7825	0.1873
AnySplat	24.659	0.7508	0.2628	23.156	0.6890	0.3239	24.560	0.7464	0.2197	22.592	0.6976	0.2739	24.083	0.7175	0.3281	21.708	0.6378	0.3691
NoPoSplat	23.821	0.7198	0.2879	22.503	0.6410	0.3247	23.549	0.7150	0.2076	21.857	0.6492	0.2408	23.125	0.6592	0.3401	20.539	0.5754	0.4035
LiGRe (Ours)	**31.219**	**0.9109**	**0.0731**	**28.286**	**0.8543**	**0.0864**	**30.691**	**0.9241**	**0.0647**	**27.874**	**0.8391**	**0.0975**	**28.901**	**0.8769**	**0.1202**	**26.247**	**0.7984**	**0.1670**
Per-scene optimization comparison																		
SplatAD	32.763	**0.9405**	0.1061	30.010	**0.8921**	0.1427	**30.928**	0.8904	0.2137	28.183	0.8217	0.2767	29.363	0.8903	0.1836	26.743	**0.8185**	0.1952
OmniRe	31.643	0.9259	0.1560	28.945	0.8905	0.1943	28.101	0.8206	0.3072	25.321	0.7450	0.3843	26.786	0.8021	0.2987	24.592	0.7371	0.3298
PVG	30.156	0.8901	0.2431	27.230	0.8148	0.2965	26.872	0.7657	0.4501	24.175	0.7010	0.5064	25.235	0.7457	0.4376	23.948	0.7037	0.4694
Street Gaussians	29.121	0.8798	0.1832	28.198	0.8404	0.2174	27.670	0.8178	0.3257	25.439	0.7479	0.3751	26.450	0.7989	0.3074	24.528	0.7392	0.3297
Difix3D+	30.628	0.8902	0.2198	26.690	0.7975	0.2472	27.803	0.8145	0.2659	25.736	0.7583	0.3154	27.087	0.8202	0.2040	24.693	0.7451	0.2301
StreetCrafter	31.230	0.9195	**0.0621**	29.675	0.8906	**0.0824**	29.670	0.8537	**0.0732**	27.565	0.8075	**0.1096**	29.198	0.8786	**0.0630**	26.418	0.7929	**0.0784**
LiGRe (Ours)	**32.901**	0.9367	0.0738	**30.268**	0.8908	0.1021	30.879	**0.8907**	0.0879	**28.432**	**0.8313**	0.1201	**29.763**	**0.9020**	0.0753	**26.957**	0.8161	0.1076

4.3 Off-Trajectory RGB Synthesis

We evaluate rendering beyond the logged trajectory. On Waymo, Argoverse 2, and PandaSet, we shift the three-camera rig laterally by 1, 2, and 3 m, perpendicular to the local driving direction. We keep the shift on the same side within each scene and preserve camera orientations and intrinsics. At each track's held-out timestamps, we compute FID separately for each dataset and displacement, pooling images across scenes, timestamps, and cameras. Real logged images with the same camera configuration form a reference set fixed across methods and displacements within each track.

On Para-Lane, we use front-camera observations from a single source traversal to render recorded viewpoints in held-out adjacent and second-adjacent lanes. We report PSNR, SSIM, and LPIPS against paired ground truth, excluding dynamic regions using the provided masks.

Table 2 shows that LiGRe achieves the lowest FID across all tested datasets and displacements and the best paired Para-Lane scores in both comparison tracks. These results support visual realism at shifted viewpoints and reconstruction accuracy in held-out lanes. Figure 2(c) shows an adjacent-lane view on Para-Lane, where LiGRe preserves lane markings and roadside structure.

Table 2: Off-trajectory RGB synthesis. Methods are ranked separately within each comparison track. LiGRe requires no per-scene optimization.

	Lateral shifts: FID↓									Para-Lane: cross-lane RGB					
	Waymo			Argoverse 2			PandaSet			Adjacent lane			Second-adjacent lane		
Method	1 m	2 m	3 m	1 m	2 m	3 m	1 m	2 m	3 m	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Feed-forward comparison															
STORM	35.29	45.84	60.05	39.52	52.30	69.72	53.79	73.52	94.74	19.45	0.5798	0.3298	17.98	0.5498	0.3621
ReconDrive	42.64	56.01	71.53	42.79	56.43	75.17	58.38	78.56	101.49	19.05	0.5670	0.3219	17.48	0.5379	0.3525
DGGT	37.03	45.38	56.08	36.03	47.19	61.08	46.12	63.72	80.10	20.34	0.6197	0.2713	18.90	0.5902	0.3076
AnySplat	51.81	69.01	91.16	51.52	70.30	93.05	69.23	91.80	120.01	18.14	0.5409	0.3750	16.56	0.5087	0.4109
NoPoSplat	56.01	74.32	101.01	56.01	76.59	102.72	74.54	98.98	127.29	17.58	0.5157	0.3879	16.06	0.4875	0.4321
LiGRe (Ours)	**16.21**	**21.26**	**27.38**	**16.79**	**21.98**	**28.37**	**21.89**	**30.05**	**37.26**	**23.01**	**0.6989**	**0.1781**	**21.48**	**0.6781**	**0.2061**
Per-scene optimization comparison															
SplatAD	46.73	64.97	82.04	54.98	76.98	103.56	62.86	93.19	121.80	20.04	0.6265	0.3297	18.59	0.5943	0.3639
OmniRe	34.93	46.36	58.56	45.12	61.18	81.65	48.15	70.40	90.76	19.83	0.6058	0.3401	18.45	0.5901	0.3657
PVG	64.61	83.23	104.60	65.68	91.75	120.01	71.47	101.04	130.26	18.93	0.5712	0.3906	17.67	0.5437	0.4218
Street Gaussians	39.74	52.39	69.68	47.32	65.09	90.57	50.47	72.82	95.35	19.62	0.5936	0.3545	18.39	0.5798	0.3761
Difix3D+	38.57	49.68	62.32	38.69	53.27	67.16	43.50	61.63	74.51	20.56	0.6124	0.2198	19.32	0.5787	0.2515
StreetCrafter	34.01	42.54	52.23	37.64	51.30	65.48	43.10	60.10	70.26	20.82	0.6230	0.2018	19.54	0.5903	0.2385
LiGRe (Ours)	**15.85**	**20.09**	**25.27**	**16.52**	**21.60**	**28.08**	**20.90**	**29.01**	**34.43**	**23.08**	**0.7217**	**0.1383**	**21.89**	**0.6892**	**0.1603**

4.4 LiDAR Reconstruction and Novel-View Synthesis

We evaluate LiDAR rendering along and beyond the logged trajectory on Waymo, Argoverse 2, and PandaSet using a three-view setting. Under the alternating-frame protocol, per-scene baselines optimize on even-indexed observations, whereas LiGRe reconstructs from the same observations without test-time optimization. Reconstruction and temporal NVS are evaluated at context and held-out odd-indexed timestamps, respectively.

Table 3 reports median squared range error (MedSE) for range accuracy, Chamfer distance (CD) for point-cloud geometry, and ray-drop accuracy for return/no-return prediction against recorded scans. We adopt SplatAD's (Hess et al., 2025) depth-based point-cloud rendering adaptations for OmniRe, PVG, and Street Gaussians. MedSE uses common reference-return rays, and CD compares the corresponding point clouds before predicted ray-drop filtering. Ray-drop accuracy covers returning and non-returning rays for methods supporting this prediction.

For off-trajectory evaluation, we shift LiDAR poses at held-out timestamps laterally by 1, 2, and 3 m following the RGB shift convention, preserving sensor orientation and scan configuration. Adapting the fusion and reprojection procedure of LiDAR-EVS (Huang et al., 2026), we align the target-time scan with static points from nine neighboring scans using recorded sensor poses, retaining dynamic points only from the target timestamp. We project the fused points onto each shifted sensor's angular sampling grid and retain the nearest point in each occupied bin. MedSE and CD are computed on common reference-supported rays and their corresponding point clouds. Held-out scans and pseudo-references are used only for evaluation. LiGRe achieves the lowest temporal-NVS CD on all three datasets and remains competitive on range and ray-drop metrics. At shifted viewpoints, it achieves the lowest CD in all settings and the lowest MedSE in most settings. These comparisons assess agreement with fused measurements on reference-supported rays. Figure 2(e) shows two Waymo temporal-NVS examples, pairing recorded scans with LiGRe's rendered point clouds.

Table 3: LiDAR reconstruction and novel-view synthesis with three-camera RGB and context LiDAR inputs. Recon and NVS use recorded scans at context and held-out timestamps, respectively; lateral shifts use pseudo-references at held-out timestamps. Acc. denotes ray-drop accuracy (%). † denotes depth-based rendering adaptations; – indicates unsupported predictions. Best and second-best values within each dataset and column are marked in **bold** and underline. LiGRe requires no test-time optimization.

		On-trajectory: recorded reference						Off-trajectory: pseudo-reference					
		Recon			NVS			Lateral 1 m		Lateral 2 m		Lateral 3 m	
Dataset	Method	MedSE↓	CD↓	Acc.↑	MedSE↓	CD↓	Acc.↑	MedSE↓	CD↓	MedSE↓	CD↓	MedSE↓	CD↓
Waymo	SplatAD	0.0120	0.784	**96.94**	**0.0137**	0.867	96.20	0.0231	1.126	0.0386	1.573	0.0647	2.284
	OmniRe†	1.483	14.60	–	1.926	15.03	–	2.418	18.47	3.165	23.68	4.372	31.52
	PVG†	18.73	68.39	–	24.16	74.61	–	28.43	83.72	35.61	102.46	46.30	131.57
	Street Gaussians†	2.637	15.73	–	3.284	17.38	–	4.176	21.46	5.823	28.10	8.361	37.84
	LiGRe (Ours)	**0.0109**	**0.731**	96.82	0.0146	**0.806**	**96.37**	**0.0184**	**0.973**	**0.0260**	**1.218**	**0.0382**	**1.537**
Argoverse 2	SplatAD	0.0164	**2.53**	92.90	0.0216	2.87	**92.83**	**0.0286**	3.318	0.0491	4.173	0.0837	5.624
	OmniRe†	2.763	20.38	–	3.491	23.16	–	4.138	27.49	5.687	35.26	7.914	46.80
	PVG†	12.80	28.73	–	16.39	32.41	–	20.76	41.58	28.47	57.69	39.12	78.34
	Street Gaussians†	3.126	19.84	–	4.075	21.80	–	5.291	26.18	7.436	34.72	10.583	48.19
	LiGRe (Ours)	**0.0153**	2.60	**93.18**	**0.0191**	**2.73**	92.71	0.0291	**3.047**	**0.0374**	**3.386**	**0.0526**	**4.012**
PandaSet	SplatAD	**0.0091**	1.43	97.30	0.0108	1.72	**96.58**	**0.0141**	2.037	0.0197	2.584	0.0273	3.428
	OmniRe†	2.63	27.46	–	3.04	31.10	–	3.582	36.94	4.713	46.25	6.395	59.73
	PVG†	34.67	109.63	–	40.12	132.48	–	47.83	149.37	59.30	181.54	76.41	230.87
	Street Gaussians†	5.50	32.97	–	6.43	38.62	–	7.815	46.39	10.263	58.47	14.586	76.21
	LiGRe (Ours)	0.0094	**1.37**	**97.62**	**0.0107**	**1.64**	96.40	0.0147	**1.892**	**0.0169**	**2.137**	**0.0208**	**2.486**

4.5 Scene Editing

We evaluate actor editing on Waymo using three-camera inputs and the two main context–target protocols. Using the same annotated actor tracks across methods, we remove an actor, rotate it by ± 0.5 rad about its vertical center axis, or translate it laterally by ± 2 m, while keeping camera poses fixed. Each reconstructed scene is edited without further optimization; LiGRe also requires no

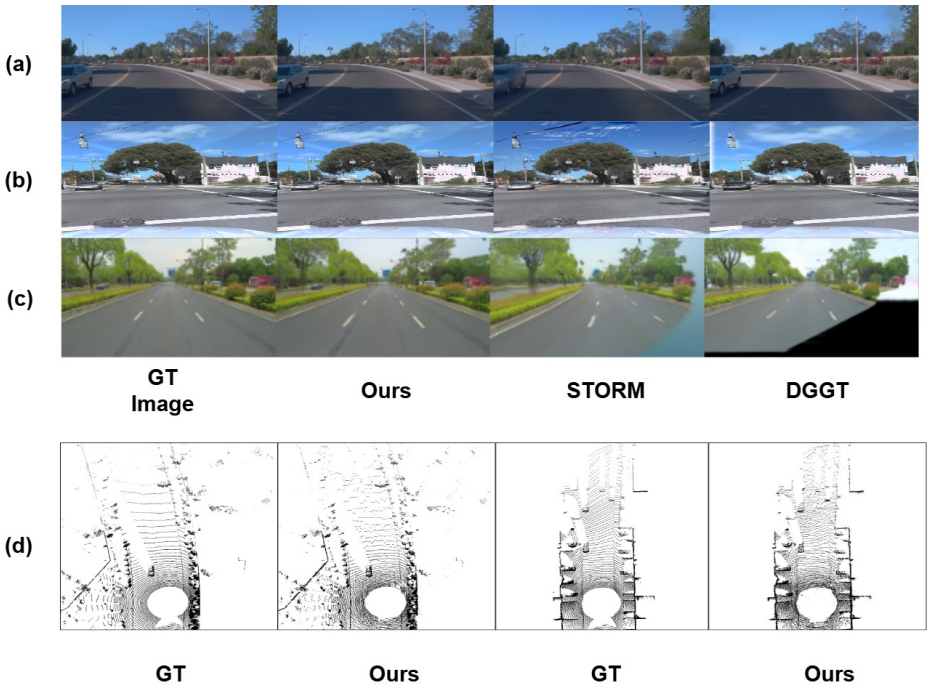

Figure 2: **Qualitative RGB and LiDAR rendering.** (a,b) Temporal novel-view synthesis on Waymo and Argoverse 2, respectively. (c) Rendering an adjacent-lane viewpoint on Para-Lane. RGB columns show ground truth, LiGRe, STORM, and DGGT. (e) Temporal LiDAR novel-view synthesis on Waymo, shown as two pairs of recorded scans (GT) and LiGRe renderings (Ours).

optimization for the initial reconstruction. At each track's held-out timestamps, we compute pooled FID across scenes and cameras against the same fixed real-image reference used in the off-trajectory evaluation. Actors, edit transforms, and evaluated images are matched across methods within each track. LiGRe reuses its initial scene sample for all edits.

Table 4(a) shows that LiGRe achieves the lowest reported FID across all five edit conditions in both tracks. These results indicate that its reconstructed scenes retain strong visual plausibility after actor manipulation.

4.6 Ablation Study

We progressively enable LiGRe's components on Waymo under the three-camera, alternating-frame protocol. The base retains multimodal encoding, metric calibration, query initialization, latent scene decoding, and both renderers. It uses the initial query features directly, omits LiDAR precision updates ($\lambda_j = 0$) while retaining a context-conditioned geometry prior, and predicts motion using only the feature-based head d_i^{base}. Context LiDAR remains available to the encoder and queries. All variants are trained separately with identical training objectives, training budgets, and evaluation inputs.

Table 4(b) reports temporal RGB PSNR and LiDAR CD, together with RGB FID and LiDAR CD at a 3 m lateral shift. Adding motion- and ray-conditioned evidence aggregation enriches queries with motion-aligned observations and yields the largest improvement in temporal RGB PSNR. The measurement-budgeted geometry prior then incorporates LiDAR latent proposals with precision distributed across slots, producing the largest reductions in LiDAR CD and off-trajectory FID. Finally, shape–motion coupling adds trajectories decoded from the shared geometry latents to the feature-based motion prediction, further improving temporal RGB–LiDAR synthesis while also benefiting off-trajectory rendering.

Table 4: Actor editing and cumulative ablation on Waymo with three-camera inputs. (a) Editing FID, ranked separately within each track. (b) Components are added cumulatively under the alternating-frame protocol; PSNR and FID evaluate RGB, and squared CD (m^2) evaluates LiDAR.

(a) Actor editing: FID↓

	Remove	Yaw (rad)		Lateral (m)	
Method		$+0.5$	-0.5	$+2$	-2
Feed-forward comparison					
STORM	35.40	36.28	37.65	38.24	<u>36.54</u>
DGGT	<u>34.81</u>	<u>35.73</u>	<u>35.29</u>	<u>36.18</u>	37.06
LiGRe (Ours)	**21.36**	**20.84**	**21.57**	**22.93**	**22.18**
Per-scene optimization comparison					
SplatAD	29.18	29.06	<u>28.86</u>	29.23	<u>28.39</u>
OmniRe	<u>29.04</u>	<u>28.76</u>	29.34	<u>28.94</u>	28.69
Street Gaussians	31.46	32.28	32.40	31.89	31.75
LiGRe (Ours)	**20.47**	**19.83**	**20.16**	**21.64**	**21.08**

(b) Cumulative ablation

	Temporal NVS		Lateral 3 m	
Variant	PSNR↑	CD↓	FID↓	CD↓
Base	24.517	2.436	68.92	6.084
+ Evidence aggregation	28.173	1.752	49.83	4.213
+ LiDAR prior update	<u>29.014</u>	<u>0.969</u>	<u>28.46</u>	<u>1.916</u>
+ Shape–motion coupling (Full)	**30.268**	**0.806**	**25.27**	**1.537**

5 Conclusion

We presented LiGRe, a feed-forward framework for reconstructing dynamic Gaussian scenes from sparse RGB and LiDAR observations. Motion- and ray-conditioned evidence aggregation and a measurement-budgeted geometry prior ground reconstruction in observed evidence, while shared geometry latents couple Gaussian shape and motion. The resulting representation supports joint RGB–LiDAR rendering and actor editing without per-scene optimization. Experiments across Waymo, Argoverse 2, PandaSet, and Para-Lane demonstrate strong reconstruction and novel-view synthesis, particularly beyond the recorded trajectory. These results highlight LiGRe's potential for efficient, controllable multimodal driving simulation.

AI Use Statement

We used ChatGPT (OpenAI) for grammar correction and language polishing. The authors reviewed all suggested edits and take responsibility for the final manuscript.

AI USE STATEMENT

REFERENCES

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467. IEEE, 2024.

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, 2024.

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.

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

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.

Georg Hess, Carl Lindström, Maryam Fatemi, Christoffer Petersson, and Lennart Svensson. Splatad: Real-time lidar and camera rendering with 3d gaussian splatting for autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11982–11992. IEEE, 2025.

Yiming Huang, Xin Kang, Sipeng Zhang, Hongliang Ren, Weihua Zhang, and Junjie Lai. Lidar-evs: Enhance extrapolated view synthesis for 3d gaussian splatting with pseudo-lidar supervision. *arXiv preprint arXiv:2603.14763*, 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.

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

Hao Lu, Tianshuo Xu, Wenzhao Zheng, Yunpeng Zhang, Wei Zhan, Dalong Du, Masayoshi Tomizuka, Kurt Keutzer, and Yingcong Chen. Drivingrecon: Large 4d gaussian reconstruction model for autonomous driving. *Advances in Neural Information Processing Systems*, 38:167933–167952, 2026.

Sheng Miao, Jiaxin Huang, Dongfeng Bai, Xu Yan, Hongyu Zhou, Yue Wang, Bingbing Liu, Andreas Geiger, and Yiyi Liao. Evolsplat: Efficient volume-based gaussian splatting for urban view synthesis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11286–11296. IEEE, 2025.

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

Ziqian Ni, Sicong Du, Zhenghua Hou, Chenming Wu, and Sheng Yang. Para-lane: Multi-lane dataset registering parallel scans for benchmarking novel view synthesis. In *2025 International Conference on 3D Vision (3DV)*, pp. 316–325. IEEE, 2025.

Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2856–2865, 2021.

Chensheng Peng, Chengwei Zhang, Yixiao Wang, Chenfeng Xu, Yichen Xie, Wenzhao Zheng, Kurt Keutzer, Masayoshi Tomizuka, and Wei Zhan. Desire-gs: 4d street gaussians for static-dynamic decomposition and surface reconstruction for urban driving scenes. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6782–6791. IEEE, 2025.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

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.

Shanlin Sun, Bingbing Zhuang, Ziyu Jiang, Buyu Liu, Xiaohui Xie, and Manmohan Chandraker. Lidarf: Delving into lidar for neural radiance field on street scenes. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19563–19572. IEEE, 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.

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

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.

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*, 2026a.

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.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-Equivariant Visual Geometry Learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026b.

Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *European conference on computer vision*, pp. 456–473. Springer, 2024.

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.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26024–26035. IEEE, 2025.

Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pp. 3–15. Springer, 2023.

Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE international intelligent transportation systems conference (ITSC)*, pp. 3095–3101. IEEE, 2021.

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.

Yunzhi Yan, Zhen Xu, Haotong Lin, Haian Jin, Haoyu Guo, Yida Wang, Kun Zhan, Xianpeng Lang, Hujun Bao, Xiaowei Zhou, et al. Streetcrafter: Street view synthesis with controllable video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 822–832. IEEE, 2025.

Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. In *International Conference on Learning Representations*, volume 2024, pp. 16739–16766, 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.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1389–1399. IEEE, 2023.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *International Conference on Learning Representations*, volume 2025, pp. 54009–54033, 2025.

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.

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*, 2026b.

Chenxu Zhou, Lvchang Fu, Sida Peng, Yunzhi Yan, Zhanhua Zhang, Yong Chen, Jiazhi Xia, and Xiaowei Zhou. Lidar-rt: Gaussian-based ray tracing for dynamic lidar re-simulation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1538–1548. IEEE, 2025.

Hongyu Zhou, Jiahao Shao, Lu Xu, Dongfeng Bai, Weichao Qiu, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugs: Holistic urban 3d scene understanding via gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21336–21345. IEEE, 2024a.

Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21634–21643. IEEE, 2024b.