X-SPLATTER: FEED-FORWARD METRIC 4D GAUSSIAN SPLATTING FOR SURROUND-VIEW DRIVING SCENES

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward scene reconstruction supports scalable simulation and world modeling for autonomous driving. We present X-Splatter, a surround-view, feed-forward 4D Gaussian Splatting (4DGS) framework that processes all surround cameras jointly on a VGGT-Ω backbone, with cross-camera attention throughout the network. It introduces two key components: Depth Anchoring Fine-tuning (DAF), which shifts geometry from relative to metric space, and Static–Dynamic Decomposition Learning (SDDL), which combines dynamic gating, static-promoting regularization, and intermediate-frame supervision to alleviate static–dynamic conflation and enhance motion learning. Following LiDAR-supervised depth adaptation, X-Splatter uses self-supervised motion learning, without bounding boxes, tracklets, or optical-flow labels. Trained and evaluated on all surrounding views, including the side views that share the least content with the front camera, X-Splatter achieves the best full-image and dynamic-region novel-view synthesis and the lowest metric-depth RMSE among the evaluated baselines on the Waymo Open Dataset. It also achieves the lowest cross-view 3D error, indicating better geometric agreement across adjacent cameras. The same model generalizes zero-shot to nuScenes and Argoverse 2, outperforming the evaluated baselines in rendering and metric-depth accuracy on both.

Project page: `https://x-splatter.github.io/page/`

1 INTRODUCTION

Replaying driving scenes along new ego trajectories is increasingly important for simulation, world modeling, and closed-loop reinforcement learning in autonomous driving (Hu et al., 2023; Gao et al., 2024; Feng et al., 2025). Gaussian Splatting offers an explicit representation with efficient, high-quality novel-view rendering (Kerbl et al., 2023). Dynamic extensions of 3DGS capture both the environment and moving agents but optimize each sequence individually, which is costly across large driving archives (Yan et al., 2024; Zhou et al., 2024; Chen et al., 2025). Feed-forward methods remove this cost by predicting Gaussians directly from images or video, and recent work extends them to metric surround-view and dynamic reconstruction (Tian et al., 2025; Lin et al., 2025; Yang et al., 2025; Chen et al., 2026a; Yu et al., 2026b). Replay along new trajectories needs these capabilities together: new maneuvers reveal regions beside and behind the vehicle, where static structure and moving traffic must both be rendered faithfully from sparsely overlapping cameras.

Three difficulties stand in the way: depth that is consistent across cameras, a correct separation of static from moving content, and velocity learning that the reconstruction loss can actually drive. Per-image depth can look plausible yet misalign across views; it must match the metric scale of the calibrated poses, even where neighboring cameras barely overlap. Depth errors also create ambiguity between geometry and motion, since the model can explain motion through changes in geometry or predict motion to compensate for geometric errors. Even with reliable depth, the model can take a shortcut by assigning static content to the dynamic branch, fitting each observation without learning shared structure. Velocity learning is further hampered because each Gaussian affects only a small image region: when its projection lands far from the object in the target frame, the reconstruction loss gives little gradient toward the correct displacement.

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.

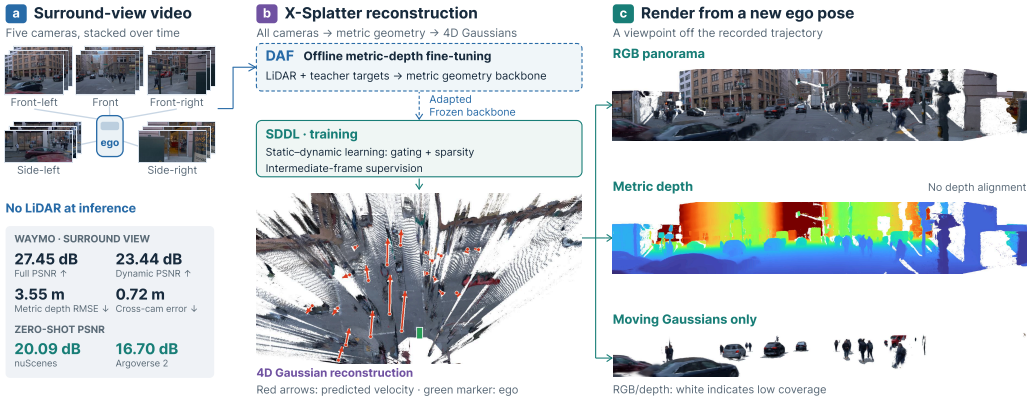

Figure 1: **X-Splatter: from surround-view images to novel views.** **(a)** Input images capture surrounding views across time. **(b)** X-Splatter reconstructs a shared 4D Gaussian scene in a single forward pass. DAF provides metric geometry, all cameras are processed jointly, and SDDL learns the static–dynamic decomposition. **(c)** The reconstructed scene supports novel-view rendering of appearance, metric depth, and moving content, without per-scene optimization. Bottom left: results on Waymo and zero-shot transfer (Tables 2, 3 and 4).

We introduce **X-Splatter**, a feed-forward 4D Gaussian Splatting framework for reconstructing static structure and moving objects from surround-view video (Figure 1). As the setting requires, it processes all surround cameras jointly, including side views that share little content with the front camera, with cross-camera attention throughout the network. **Depth Anchoring Fine-tuning (DAF)** adapts a pretrained geometry backbone to metric depth, which we then freeze to give motion learning a stable basis. Static Gaussians from all cameras and frames are pooled into one shared representation. **Static–Dynamic Decomposition Learning (SDDL)** learns the decomposition through dynamic gating, an opacity gate between static and velocity-warped dynamic copies; static-promoting regularization, which discourages assigning the scene to the dynamic branch; and intermediate-frame supervision, which constrains motion at shorter temporal offsets where full-interval gradients are weak. Neither per-scene optimization nor motion annotations are required.

On all five Waymo cameras, X-Splatter achieves 27.45 dB full-image PSNR, 23.44 dB dynamic-region PSNR, and 3.55 m metric-depth RMSE, outperforming the evaluated baselines (Table 2). The same weights transfer to nuScenes and Argoverse 2, with six and seven cameras, achieving higher PSNR and lower metric depth error than the baselines (Table 4). Adjacent cameras place shared static structure several-fold closer together than in the baselines (Table 3); ablations support the adapted depth head, joint camera processing, and temporal supervision (Sec. 4.5).

Our contributions are threefold:

- **A unified feed-forward framework for state-of-the-art dynamic surround-view reconstruction.** X-Splatter reconstructs static structure and moving traffic from calibrated multi-camera video without per-scene optimization. It achieves the best surround-view rendering quality, metric-depth accuracy, and cross-view geometric consistency among the evaluated baselines on Waymo, and transfers zero-shot to nuScenes and Argoverse 2.
- **Metric geometry through depth anchoring.** DAF anchors the pretrained geometry backbone to metric scale, so inference needs no depth alignment.
- **Static–dynamic decomposition learning.** SDDL combines gated static and dynamic Gaussian copies with static-promoting regularization and intermediate-frame supervision to learn the decomposition and velocities without motion annotations.

2 Related Work

Per-scene driving reconstruction. Per-scene methods fit a representation to each driving sequence: EmerNeRF (Yang et al., 2024a) models static and dynamic radiance with scene flow, and Street Gaussians (Yan et al., 2024), DrivingGaussian (Zhou et al., 2024), and OmniRe (Chen et al.,

2025) use dynamic Gaussians, sometimes with object annotations. Each new clip requires optimization, whereas X-Splatter makes no per-scene parameter update.

Feed-forward Gaussian and geometry reconstruction. Generalizable methods predict representations directly from images: pixelSplat and MVSplat predict Gaussians from sparse views (Charatan et al., 2024; Chen et al., 2024), and DUSt3R and VGGT predict geometry and cameras (Wang et al., 2024; 2025). Our backbone builds on VGGT-Ω and its DINOv2 features (Oquab et al., 2024; Wang et al., 2026a). For driving video, STORM predicts Gaussians and scene flow from calibrated views, with LiDAR-supervised metric depth (Yang et al., 2025); DGGT predicts dynamic Gaussians and camera parameters with instance and lifespan heads (Chen et al., 2026a); and StreetForward builds on Pi3 with causal attention and motion regularizers (Yu et al., 2026b; Wang et al., 2026b), which our motion formulation follows; Appendix A compares the three with X-Splatter. Static surround-view reconstructors also address sparse camera overlap (Tian et al., 2025; Lin et al., 2025), and concurrent methods such as ReconDrive address dynamic driving reconstruction (Yu et al., 2026a). Metric-depth adaptation is established in monocular depth estimation (Yang et al., 2024b); DAF applies it to a pretrained multi-view backbone that is then frozen for 4D training. DGGT and StreetForward align relative depth to ground truth for evaluation, so we report metric and affine-aligned depth separately (Sec. 4.1). Appendix E discusses further concurrent work.

Dynamic modeling and motion priors. Dynamic reconstruction either models motion through a scene-flow or velocity field, as in STORM and StreetForward, or separates movers with object boxes and tracklets (Yan et al., 2024; Chen et al., 2025). The latter requires motion annotations, while unconstrained motion fields can move static background. We regularize per-pixel velocities with established 3D rigidity, forward–backward consistency, and smoothness priors, and SDDL adds dynamic gating, static-promoting regularization, and intermediate-frame supervision on the metric geometry supplied by DAF (Sec. 3.2).

3 METHOD

Given T source frames from each of N calibrated cameras with ego poses, X-Splatter reconstructs a shared dynamic Gaussian scene in a single forward pass, enabling rendering at novel viewpoints and timestamps within the observed clip (Figure 2). DAF adapts a pretrained VGGT-Ω backbone (Wang et al., 2026a) to metric depth, which is then kept fixed to reduce ambiguity between depth and motion. All cameras are processed jointly: the backbone attends globally over all $N \times T$ views, and the trainable motion head lets each camera attend to all cameras at the neighboring source frames (Appendix C). SDDL then learns the static–dynamic decomposition and velocities.

Gaussian representation. Each non-sky source pixel defines a Gaussian $g = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \alpha, \mathbf{c})$ with center, covariance, opacity, and color (Kerbl et al., 2023). Centers are obtained by back-projecting the predicted metric depth into a shared world frame using calibration and ego poses. We additionally predict forward and backward velocities $\mathbf{v}^{\pm}$ and a mixing weight $\sigma \in [0, 1]$ that splits each Gaussian into a static contribution, shared across cameras and source frames, and a dynamic one displaced to the target timestamp (Sec. 3.2). The scene is rendered with differentiable Gaussian splatting (Ye et al., 2025). Sky pixels, marked by the pretrained segmenter in VGGT-Ω (Wang et al., 2026a), produce no Gaussians and are excluded from 4D reconstruction losses and image-quality metrics.

3.1 DEPTH ANCHORING FINE-TUNING

Reconstructing a shared scene requires depth at the metric scale of the camera poses, which the relative depth of a pretrained backbone does not provide. DAF adapts this prior to metric depth before motion learning, combining sparse LiDAR measurements with dense supervision from a frozen teacher.

Target construction. For each camera frame, we align the relative depth of a frozen VGGT-Ω teacher to the LiDAR returns with an L_1-optimal scale, keep the measured depth at LiDAR pixels, and use the down-weighted aligned teacher depth at the remaining non-sky pixels. This alignment

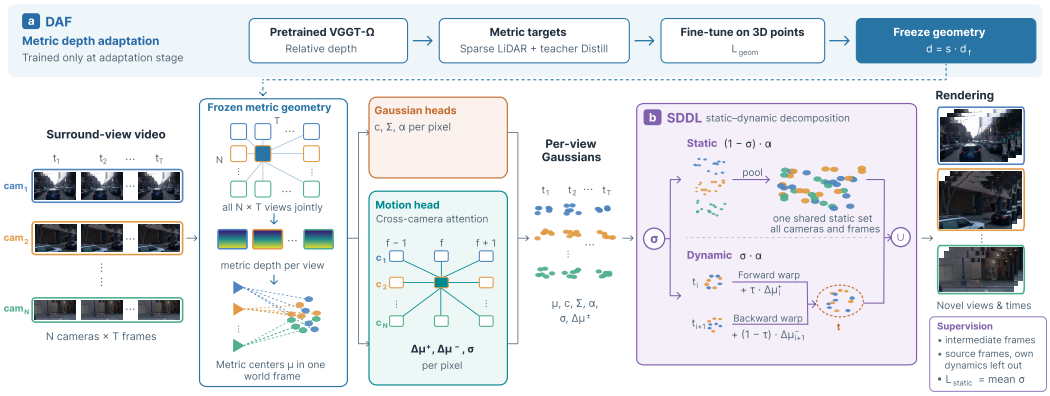

Figure 2: **X-Splatter: framework overview. (a)** DAF adapts pretrained geometry to metric depth with sparse LiDAR supervision. The adapted backbone is then frozen and processes all cameras and frames jointly, while the motion head adds trainable cross-camera attention. Metric depth and camera poses place per-view Gaussians in a shared world frame, and the heads predict their appearance, shape, opacity, velocities, and static–dynamic mixing weights. **(b)** SDDL splits each Gaussian by its mixing weight: static copies are pooled into one shared set, dynamic copies are warped to the target time, and their union is splatted to render the target view.

is used only to construct training targets. We then fine-tune the backbone, conditioned on known camera parameters, by comparing predicted and target camera-frame points without fitting a scale to the prediction:

$$\mathcal{L}_{\text{geom}} = \frac{1}{3|\mathcal{S}|} \sum_{p \in \mathcal{S}} w_p \left\| \hat{\mathbf{P}}_p - \mathbf{P}_p \right\|_1, \tag{1}$$

where $\mathcal{S}$ is the supervised pixel set, $\hat{\mathbf{P}}_p$ and $\mathbf{P}_p$ are the predicted and target 3D points, and w_p combines target reliability with inverse-depth weighting. Depth targets and camera translations share one fixed normalization, preserving metric scale across training examples (Appendix B).

Frozen metric geometry. After adaptation, we freeze the backbone and depth head and convert the normalized prediction d_{f} to meters by

$$d = s\, d_{\text{f}}, \tag{2}$$

where s=19.8 m is a preset depth scaling factor (Appendix B) that replaces the scene-dependent normalization of VGGT-Ω and is fixed across clips and datasets. Inference therefore requires no LiDAR or per-clip depth alignment. Keeping the resulting Gaussian centers fixed during 4D training prevents the model from adjusting depth to compensate for motion errors.

3.2 Static–Dynamic Decomposition Learning

Fixed geometry does not determine which content should move: a model can still assign static content to the dynamic representation or reconstruct individual frames without learning useful motion. SDDL addresses this with three objectives and needs no bounding boxes, tracklets, or optical-flow labels.

Dynamic gating and temporal reconstruction. For each source Gaussian, the mixing weight σ divides its opacity into a static contribution $\alpha_{\text{s}} = (1-\sigma)\alpha$ and a dynamic contribution $\alpha_{\text{d}} = \sigma\alpha$. Static contributions retain their centers and are pooled across cameras and source frames in the shared world frame, so each target view can render static content observed by any camera; dynamic contributions are displaced using the predicted forward and backward motion. When reconstructing a source frame during training, we exclude that frame's own dynamic contributions and warp them in from neighboring source frames, so the dynamic branch cannot reproduce the target from its own Gaussians and must explain it through motion.

Static-promoting regularization. Temporal reconstruction alone can still assign persistent background to the dynamic branch, so we penalize the dynamic mixing weights, which we refer to as

dynamic sparsity in the ablations:

$$\mathcal{L}_{\text{static}} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \sigma_g, \tag{3}$$

where $\mathcal{G}$ denotes the source Gaussians. A separate opacity regularizer encourages opacities toward one, and local 3D rigidity, velocity smoothness, and forward–backward consistency regularize how dynamic content moves.

Intermediate-frame supervision. Reconstruction at source timestamps weakly constrains motion between widely spaced frames, so we also supervise intermediate images withheld from the source inputs. For a target time t between source times t_i and t_{i+1}, let $\tau = (t - t_i)/(t_{i+1} - t_i)$. Dynamic Gaussians from the two endpoints are placed at

$$\boldsymbol{\mu}_i(t) = \boldsymbol{\mu}_i + \tau\, \Delta\boldsymbol{\mu}_i^{+}, \qquad \boldsymbol{\mu}_{i+1}(t) = \boldsymbol{\mu}_{i+1} + (1-\tau)\, \Delta\boldsymbol{\mu}_{i+1}^{-}, \tag{4}$$

where $\Delta\boldsymbol{\mu}_i^{+}$ and $\Delta\boldsymbol{\mu}_{i+1}^{-}$ are the displacements induced by $\mathbf{v}_i^{+}$ and $\mathbf{v}_{i+1}^{-}$ over one source-frame interval. These are combined with the shared static set and rendered with the photometric objective of Sec. 3.3. The shorter displacements help when full-interval warps are poorly aligned with the target content.

Optional weak motion supervision. As an optional extension that SDDL does not require, we test whether available object annotations improve moving-object reconstruction. We project annotated 3D boxes of moving vehicles, pedestrians, and cyclists into each camera; with $m_p \in \{0, 1\}$ marking pixels inside the projected rectangles, the RGB reconstruction term becomes

$$\mathcal{L}_{\text{rgb}}^{\text{weak}} = \frac{\sum_{p \in \mathcal{P}} w_p \, \|\hat{\mathbf{I}}_p - \mathbf{I}_p\|_1}{\sum_{p \in \mathcal{P}} w_p}, \qquad w_p = 1 + (\gamma - 1) m_p, \tag{5}$$

where $\mathcal{P}$ contains valid non-sky pixels of source and held-out frames, including intermediate frames, and $\gamma > 1$ sets the emphasis on moving regions. Velocities and mixing weights are still learned through rendering rather than regressed. We report this variant separately as *+weak supervision* in Sec. 4.

3.3 Training Objective

After DAF, we train the appearance, motion, and Gaussian render heads, including the motion-head cross-camera attention, with the backbone and depth head frozen, using

$$\begin{aligned} \mathcal{L}_{\text{4D}} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{p}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{d}}\mathcal{L}_{\text{depth}} + \lambda_{\text{s}}\mathcal{L}_{\text{static}} \\ + \lambda_{\alpha}\mathcal{L}_{\text{opacity}} + \lambda_{\text{sm}}\mathcal{L}_{\text{smooth}} + \lambda_{\text{fb}}\mathcal{L}_{\text{fb}} + \lambda_{\text{r}}\mathcal{L}_{\text{rigid}}. \end{aligned} \tag{6}$$

The RGB L_1 and LPIPS (Zhang et al., 2018) terms supervise rendered source and held-out frames, including the intermediate targets; the depth term compares rendered depth with the frozen metric prediction at source frames, keeping the composed scene consistent with the DAF geometry. LiDAR is used only in DAF, and the weakly supervised variant replaces only $\mathcal{L}_{\text{rgb}}$ with $\mathcal{L}_{\text{rgb}}^{\text{weak}}$. Every term is defined in Appendix C; loss weights and optimization settings are given in Appendix C.

4 Experiments

4.1 Experimental Setup

Datasets and evaluation protocol. We train X-Splatter on the Waymo Open Dataset (Sun et al., 2020) and evaluate on the 202 validation scenes used by STORM (Yang et al., 2025), using all five cameras; each 20-frame window provides source frames $\{0, 5, 10, 15\}$, and the remaining frames are held out. For zero-shot evaluation, we select 100 scenes from the nuScenes validation split (Caesar et al., 2020) and 100 logs from the Argoverse 2 Sensor validation split (Wilson et al., 2021), with six and seven cameras. We exclude scenes with less than 3 m of ego travel and select 40 high-, 40 medium-, and 20 low-motion scenes per dataset, emphasizing dynamic scenes (Appendix D). We report front-camera and all-camera results.

Table 1: Waymo rendering results as reported in the original papers.

Method	Scored cams	Full		Dynamic	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑
STORM (Yang et al., 2025)	3	26.38	0.794	22.10	0.624
DGGT (Chen et al., 2026a)	3	27.41	0.846	20.99	0.821
DeformableGS (Yang et al., 2024c)	3	25.29	0.761	17.10	0.266
GS-LRM (Zhang et al., 2024)	3	25.18	0.753	20.02	0.520
EmerNeRF (Yang et al., 2024a)	3	24.51	0.738	17.79	0.255
PVG (Chen et al., 2026b)	3	22.38	0.661	15.51	0.128
Xiaomi Auto WM (Zhou et al., 2026)	1	28.48	0.861	–	–
StreetForward (Yu et al., 2026b)	1	27.01	0.818	24.30	**0.827**
AnySplat (Jiang et al., 2025)	1	23.32	0.687	15.99	0.418
DA3 w/ GS head (Lin et al., 2026)	1	23.27	0.686	15.96	0.429
X-Splatter (self-supervised)	1	29.66	0.883	24.49	0.744
X-Splatter (+ weak sup.)	1	**29.73**	**0.884**	**25.29**	0.775

Cameras. Scored cams: number of cameras each method is evaluated on, as reported. Our rows use the five-camera-input model scored on the front camera.
Protocols. Camera sets and protocols differ between papers; Table 2 compares methods on a common protocol.
Metrics. Bold marks the best result per column.

Evaluation metrics. We report PSNR and SSIM (Wang et al., 2004), on Waymo also within dynamic regions defined by ground-truth boxes, and depth RMSE in meters. Metric depth is evaluated directly against ground truth; for comparison with relative-depth methods, we also report results after per-clip scale-and-shift alignment for every method (Appendix D).

Baselines. We evaluate the released STORM and DGGT checkpoints on the same clips, target views, resolutions, and masks. X-Splatter and STORM use known camera poses and intrinsics; DGGT predicts camera parameters and uses a per-clip, per-camera Sim(3) alignment to ground-truth poses, which also supplies scale. Single-camera models are applied independently to each camera. Both checkpoints were trained on the three front-facing Waymo cameras and not retrained for five-camera input, so the all-camera comparison also measures robustness outside their training configuration; their published results and StreetForward's are listed in Table 1.

Model variants and training. Our default model learns motion without object annotations. The optional *+weak supervision* variant (Sec. 3.2) uses box-derived masks that also define the dynamic-region evaluation mask, so its dynamic-region gains are measured on the region its supervision targets. Implementation details are given in Appendix C.

4.2 4D Novel-View Synthesis on Waymo

Table 2 reports results on all five cameras (a) and on the front camera (b).

Surround-view reconstruction. With five-camera input, the default model achieves 27.45 dB full-image and 23.44 dB dynamic-region PSNR, exceeding STORM by 4.8 and 2.6 dB, and SSIM rises from 0.680 to 0.814 over the full image and from 0.524 to 0.686 within dynamic regions, without weak supervision.

Front-camera reconstruction. With front-camera input only, our single-camera model reaches 29.73 dB full-image and 24.46 dB dynamic-region PSNR, against 25.16 and 21.07 dB for DGGT. The five-camera model retains comparable front-view quality (29.66 and 24.49 dB). The two models are trained separately, so their comparison characterizes the two settings rather than isolating the effect of camera count.

4.3 Metric Depth and Cross-Camera Consistency

On all five cameras, X-Splatter achieves 3.55 m metric depth RMSE, 47% below STORM's 6.67 m (Table 2); DGGT obtains 17.62 m under its pose-alignment protocol. Within dynamic regions, our RMSE is 5.25 m, against 9.85 m for STORM and 21.56 m for DGGT, and weak supervision lowers it to 5.18 m. After affine alignment, our error changes only from 3.55 to 3.33 m, against 17.62

Table 2: 4D novel-view synthesis on Waymo, evaluated across all five cameras (a) and the front camera (b).

	Cameras		Full		Dynamic		D-RMSE (m)↓	
Method	in	scored	PSNR↑	SSIM↑	PSNR↑	SSIM↑	metric	affine-aligned
(a) Scored on all five cameras — one protocol, evaluated by us								
STORM (Yang et al., 2025)	5	5	22.66	0.680	20.82	0.524	6.67	5.46
STORM (Yang et al., 2025)	1	5	21.65	0.632	19.90	0.466	8.24	6.29
DGGT (Chen et al., 2026a)	1	5	20.28	0.576	18.95	0.447	17.62†	5.03
X-Splatter (front-camera input)	1	5	24.60	0.720	21.44	0.577	5.40	4.27
X-Splatter	5	5	27.45	**0.814**	23.44	0.686	3.55	3.33
X-Splatter + weak sup.	5	5	**27.48**	**0.814**	**23.96**	**0.709**	**3.53**	**3.32**
(b) Scored on the front camera — one protocol, evaluated by us								
STORM (Yang et al., 2025)	1	1	21.60	0.726	20.72	0.519	6.99	6.07
DGGT (Chen et al., 2026a)	1	1	25.16	0.756	21.07	0.552	13.57†	4.69
X-Splatter (front-camera input)	1	1	**29.73**	0.880	24.46	0.743	5.18	3.62
X-Splatter	5	1	29.66	0.883	24.49	0.744	3.55	3.36
X-Splatter + weak sup.	5	1	**29.73**	**0.884**	**25.29**	**0.775**	**3.54**	**3.34**

Cameras. In/scored: input/evaluated cameras. Single-camera models are run independently per evaluated camera.
Metrics. Full/Dynamic: full-image/dynamic-region image quality. Full image-quality metrics exclude sky pixels. D-RMSE: full-image depth RMSE of the rendered depth. Bold marks the best result per column within each block, including ties.
Depth evaluation. Metric: predictions evaluated directly against ground-truth depth. Affine-aligned: predictions evaluated after fitting a scale and shift to ground-truth depth per clip for every method. †DGGT predicts relative geometry and aligns its camera trajectory to ground-truth poses for rendering.

Table 3: Cross-camera geometric consistency on Waymo.

	Cross-view 3D error (m)↓		Depth RMSE (m)↓	Coverage
Method	Mean	Median	Overlap	
STORM	2.51	1.06	7.39	91.2%
DGGT	9.23	2.60	30.84	90.1%
DGGT†	2.02	0.73	4.37	90.1%
X-Splatter	**0.72**	**0.21**	**2.33**	**100.0%**

Metrics. Cross-view 3D error: mean (scene-averaged) and pooled median distance between two adjacent cameras' estimates of the same static LiDAR point. Depth RMSE: against LiDAR on the same points. Coverage: share of correspondences with a valid estimate. Bold marks the best result per column.
Methods. STORM has no depth head, so its per-camera estimate is the render of that camera's Gaussians alone. DGGT uses its pose-alignment protocol (Sec. 4.1); DGGT† additionally re-scales each camera's depth to LiDAR, shown for reference.

to 5.03 m for DGGT, so our predictions need little correction and remain the most accurate after alignment.

Cross-camera consistency. Per-camera metrics cannot tell whether two cameras place a shared surface at the same 3D position. For every source frame, we lift each static LiDAR return visible in camera A with A's calibrated pose, project it into the adjacent camera B, and keep it if it passes a depth-edge and an occlusion check, giving 18.7M correspondences over 202 validation scenes. Each camera's own depth estimate is unprojected with its calibrated pose, and we report the mean 3D distance between the two estimates of the same point, with the depth RMSE against LiDAR on the same points so that agreement on a wrong surface cannot pass as consistency; no cross-camera alignment is fitted. X-Splatter places the same surface 0.72 m apart on average (median 0.21 m), against 2.51 m for STORM, with the lowest depth error on these points (Table 3). DGGT obtains 9.23 m with its per-clip, per-camera Sim(3) pose alignment; additional LiDAR-based scaling (DGGT†) reduces this to 2.02 m, indicating a substantial scale-related error. In Figure 4, the two cameras' points fall on one surface for X-Splatter and on two offset surfaces for the baselines (protocol in Appendix D).

4.4 Zero-Shot Cross-Dataset Generalization

We evaluate the Waymo-trained model on nuScenes and Argoverse 2 without further training, moving from five cameras to six and seven. The DAF teacher setting was selected on Argoverse 2 validation logs (Appendix B), the split our evaluation logs come from, so Argoverse 2 is not fully held out for that choice. X-Splatter exceeds STORM and DGGT in PSNR in every setting (Table 4); across all cameras it reaches 20.09 dB on nuScenes and 16.70 dB on Argoverse 2, 1.83 and 0.49 dB

Table 4: Zero-shot generalization from Waymo to nuScenes and Argoverse 2, evaluated on 100 scenes per dataset without further training. We report front-camera and all-camera image quality and depth accuracy.

	→ nuScenes			→ Argoverse 2		
		D-RMSE (m)↓			D-RMSE (m)↓	
Method	PSNR↑	metric	affine-aligned	PSNR↑	metric	affine-aligned
Front camera						
STORM	19.08	9.62	8.50	17.67	19.93	16.54
DGGT	22.27	10.33	**6.35**	16.27	86.08	13.30
X-Splatter (ours)	**23.00**	8.61	6.60	**17.91**	16.63	13.54
X-Splatter + weak sup.	22.80	**8.58**	6.60	17.82	**16.43**	**13.01**
All cameras (6 on nuScenes, 7 on Argoverse 2)						
STORM	17.98	10.92	8.51	16.21	18.65	13.28
DGGT	18.26	9.79	**6.36**	16.03	36.49	**10.65**
X-Splatter (ours)	**20.09**	**8.38**	6.76	16.70	13.53	11.84
X-Splatter + weak sup.	19.91	**8.38**	6.80	**16.75**	**13.37**	11.64

Metrics. Bold marks the best result per column within each camera block, including ties.
Depth evaluation. We follow the conventions in Table 2, including DGGT's pose alignment.

above the strongest baseline. Weak supervision gives only small, mixed changes. Our metric depth error is also the lowest in every setting, with all-camera RMSE of 8.38 and 13.53 m. After affine alignment, however, DGGT is lower across all cameras (6.36 and 10.65 m against our 6.76 and 11.84 m), so relative geometry under domain shift remains an area for improvement.

4.5 Component Ablations

Table 5 ablates the adapted depth head, joint processing, and the SDDL objectives on Waymo, using the same architecture, a global batch size of 16, and matched training steps. Each variant is a single training run; adjacent checkpoints of one run differ by at most about 0.07 dB. Depth evaluation follows Table 2, and Appendix F gives the protocol of each variant.

w/o DAF. Replacing the DAF-adapted depth head with the released VGGT-Ω head, without retraining, raises metric depth RMSE from 3.62 to 5.18 m and affine-aligned RMSE from 3.40 to 3.96 m, and lowers full-image PSNR by 1.61 dB. The adapted backbone is retained; the replacement protocol is detailed in Appendix F.

Joint surround-view processing. Removing cross-camera attention, so that every camera is processed independently throughout the network while camera coverage and temporal context are retained, raises metric depth RMSE from 3.62 to 4.31 m and lowers dynamic-region PSNR from 22.61 to 22.32 dB. It also raises the cross-view 3D error from 0.72 to 2.10 m (Appendix D).

SDDL: decomposition objectives. Removing held-out-time supervision and dynamic sparsity together, with dynamic gating and the remaining regularizers retained, lowers dynamic-region PSNR from 22.61 to 21.88 dB and raises metric depth RMSE from 3.62 to 3.88 m. Removing held-out-time supervision alone lowers full-image PSNR to 26.18 dB and dynamic-region PSNR to 22.00 dB, so this objective accounts for the gain. Dynamic sparsity instead trades full-image for dynamic-region quality: relative to removing both, it raises dynamic-region PSNR by 0.12 dB and lowers full-image PSNR by 0.21 dB, as in the cumulative study in Appendix G.

4.6 Qualitative Results

Figure 3 compares three Waymo cases from the front camera at held-out interpolation times. In the highlighted regions, X-Splatter preserves sharper vehicle contours and static details than STORM, and its depth rendering delineates vehicles, the road, and buildings. DGGT also reconstructs detailed RGB appearance here; its metric depth is assessed in Table 2. Zero-shot examples on nuScenes and Argoverse 2 are in Appendix H.

Table 5: Component ablations on Waymo using all five cameras.

DAF	Joint	SDDL		PSNR↑		D-RMSE (m)↓	
		Time	Sparsity	Full	Dynamic	metric	affine-aligned
✓	✓	✓	✓	**26.97**	**22.61**	**3.62**	**3.40**
✗	✓	✓	✓	25.36*	21.92*	5.18*	3.96*
✓	✗	✓	✓	26.63	22.32	4.31	3.69
✓	✓	✗	✓	26.18	22.00	3.85	3.61
✓	✓	✗	✗	26.39	21.88	3.88	3.64

Components. Time: supervision at held-out times (interpolation and extrapolation). Sparsity: static-promoting regularization ($\mathcal{L}_{\text{static}}$).
*w/o DAF: released depth head substituted without retraining.
Metrics. Full/Dynamic: full-image/dynamic-region PSNR, excluding sky pixels. D-RMSE: full-image depth RMSE of the rendered depth. Bold marks the best result per column.
Depth evaluation. Metric and affine-aligned depth follow the conventions in Table 2.

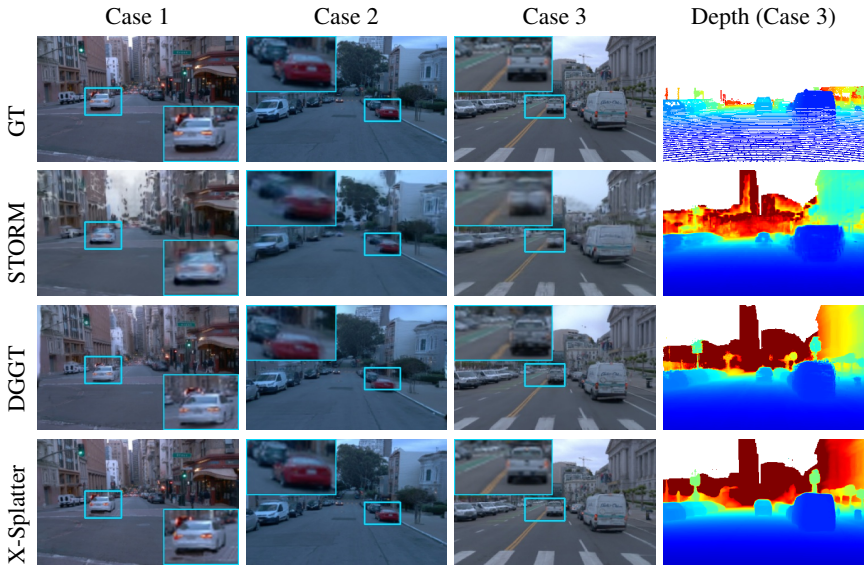

Figure 3: **Waymo qualitative comparison.** Three front-camera cases at held-out interpolation times. Matched boxes and insets highlight the same regions across methods. The final column shows projected LiDAR and rendered depth for Case 3 using a shared 0–80 m colormap. Values above 80 m are clipped; white indicates masked or missing values.

5 Conclusion

We presented X-Splatter, a feed-forward framework for metric 4D Gaussian reconstruction from surround-view video, in which DAF provides a frozen metric geometry prior, all surround cameras are processed jointly, and SDDL learns shared static structure and moving content without motion labels. On Waymo, it improves full-image and dynamic-region rendering, metric-depth accuracy, and cross-view geometric consistency over the evaluated baselines. The same weights transfer to nuScenes and Argoverse 2 across different camera rigs. Ablations support the adapted depth head, joint surround-view processing, and held-out-time supervision.

Limitations

X-Splatter requires calibrated cameras and ego poses, with LiDAR used only during DAF. Frozen-depth errors affect Gaussian placement and motion estimation; linear warping and prolonged occlusion limit motion reconstruction. Transfer is tested on two driving datasets, without broader domain shifts or calibration errors. The model reconstructs observed clips without predicting future agent behavior. Ablations use single runs at a reduced budget; baselines are not retrained for five-camera input. Motion and decomposition are assessed through rendering, without direct velocity or segmentation metrics. Novel views are evaluated at held-out times on recorded trajectories.

Ethics Statement

This work uses the publicly released Waymo Open Dataset, nuScenes and Argoverse 2 under their respective licenses; these datasets apply their own anonymization of faces and license plates. We collect no new data involving people. Reconstructed driving scenes are intended for simulation and evaluation of autonomous-driving systems.

Reproducibility Statement

Architecture, training schedule, loss weights and optional weak-supervision settings are given in Appendix C. The evaluation protocols, depth-alignment conventions, cross-camera consistency procedure and the deterministic zero-shot scene selection are given in Appendix D, and the ablation protocol in Appendix F.

Use of Large Language Models

Large language models assisted with manuscript editing, figure preparation, experimental-design feedback, result interpretation, and analysis and plotting code. The authors conducted the experiments, verified the AI-assisted work and results, and take full responsibility for the content.

References

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Xiaoxue Chen, Ziyi Xiong, Yuantao Chen, Gen Li, Nan Wang, Hongcheng Luo, Long Chen, Haiyang Sun, Bing Wang, Guang Chen, Hangjun Ye, Hongyang Li, Ya-Qin Zhang, and Hao Zhao. DGGT: Feedforward 4D reconstruction of dynamic driving scenes using unposed images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a. arXiv:2512.03004.

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 (ECCV)*, 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*, 2026b.

Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. OmniRe: Omni urban scene reconstruction. In *International Conference on Learning Representations (ICLR)*, 2025.

Hoonhee Cho, Jae-Young Kang, Giwon Lee, Hyemin Yang, Heejun Park, Seokwoo Jung, and Kuk-Jin Yoon. VR-Drive: Viewpoint-robust end-to-end driving with feed-forward 3D Gaussian splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2510.23205.

Tuo Feng, Wenguan Wang, and Yi Yang. A survey of world models for autonomous driving. *arXiv preprint arXiv:2501.11260*, 2025.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: Feed-forward 3D Gaussian splatting from unconstrained views. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4), 2023.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2511.10647.

Junhong Lin, Kangli Wang, Shunzhou Wang, Songlin Fan, Ge Li, and Wei Gao. VGD: Visual geometry Gaussian splatting for feed-forward surround-view driving reconstruction. *arXiv preprint arXiv:2510.19578*, 2025.

Carl Lindström, Mahan Rafidashti, Maryam Fatemi, Lars Hammarstrand, Martin R. Oswald, and Lennart Svensson. IDSplat: Instance-decomposed 3D Gaussian splatting for driving scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2511.19235.

Hongyuan Liu, Haochen Yu, Bochao Zou, Jianfei Jiang, Qiankun Liu, Jiansheng Chen, and Huimin Ma. InstDrive: Instance-aware 3D Gaussian splatting for driving scenes. *arXiv preprint arXiv:2508.12015*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. arXiv:2304.07193.

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.

Chenpeng Su, Wenhua Wu, Chensheng Peng, Tianchen Deng, Zhe Liu, and Hesheng Wang. DIAL-GS: Dynamic instance aware reconstruction for label-free street scenes with 4D Gaussian splatting. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. arXiv:2511.06632.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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 *AAAI Conference on Artificial Intelligence*, 2025. arXiv:2409.12753.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2503.11651.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a. arXiv:2605.15195.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 (ICLR)*, 2026b. arXiv:2507.13347.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks*, 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 (ECCV)*, 2024.

Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. EmerNeRF: Emergent spatial-temporal scene decomposition via self-supervision. In *International Conference on Learning Representations (ICLR)*, 2024a.

Jiawei Yang, Jiahui Huang, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Maximilian Igl, Apoorva Sharma, Peter Karkus, Danfei Xu, Boris Ivanovic, Yue Wang, and Marco Pavone. STORM: Spatio-temporal reconstruction model for large-scale outdoor scenes. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2501.00602.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

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

Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *Journal of Machine Learning Research*, 26(34), 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.

Haochen Yu, Qiankun Liu, Hongyuan Liu, Jianfei Jiang, Juntao Lyu, Jiansheng Chen, and Huimin Ma. XYZCylinder: Towards compatible feed-forward 3D Gaussian splatting for driving scenes via unified cylinder lifting method. *arXiv preprint arXiv:2510.07856*, 2025.

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.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: Large reconstruction model for 3D Gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Lijun Zhou, Hongcheng Luo, Zhenxin Zhu, Cheng Chi, Mingfei Tu, Kaixin Xiong, Lei Gong, Zhanqian Wu, Zehan Zhang, Fangzhen Li, Hao Li, Yingying Shen, Jiale He, Haohui Zhu, Shan Zhao, Kai Wang, Zhiwei Zhan, Yuechuan Pu, Kaiyuan Tan, Ruiling Yang, Xianqi Wang, Tianyi Yan, Jiawei Zhou, Lei Zhang, Jingyang Zhao, Xi Zhou, Chitian Sun, Chenming Wu, Jiong Deng, Hongwei Xie, Ming Lu, Kun Ma, Long Chen, Guang Chen, Hangjun Ye, Bing Wang, and Haiyang

Sun. Xiaomi Auto World Model: A joint world model integrating reconstruction and generation for autonomous driving. *arXiv preprint arXiv:2605.18137*, 2026.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

A Comparison of Feed-Forward Methods

Table 6 compares X-Splatter with the feed-forward methods for dynamic driving scenes discussed in Sec. 2: STORM, DGGT, and StreetForward.

Table 6: Comparison of feed-forward methods for dynamic driving-scene reconstruction.

	STORM (Yang et al., 2025)	DGGT (Chen et al., 2026a)	StreetForward (Yu et al., 2026b)	**X-Splatter** (ours)
Metric depth[†]	Yes	No	No	Yes
Geometry during motion learning	Jointly learned	Jointly learned	Jointly learned	Metric-adapted, then fixed
Motion labels	None	Masks	None	None
Camera coverage	3 front-facing	1 or 3 front-facing	Front only	All 5 (surround)

Notes. [†]Metric depth is predicted without ground-truth scale alignment. DGGT and StreetForward use ground truth to align relative depth for evaluation and predict camera parameters; STORM and X-Splatter use known camera parameters. Camera coverage denotes views used for training and reconstruction on Waymo, not architectural limits. Motion labels exclude depth and camera supervision.

B Depth Anchoring Fine-tuning: Details

This section specifies the DAF stage that precedes 4D training. The geometry it produces is frozen before appearance and motion learning begins, so every 4D result inherits the depth model described here.

Metric scale. Depth is predicted in normalized units of s: a LiDAR return at d meters supervises a prediction of d/s, and the geometry loss is evaluated in *scale-aware* mode, with no free per-sample depth scale. A prediction therefore stands in a fixed, clip-independent relation to meters, recovered at inference by multiplying by s. We set $s{=}19.8\,\mathrm{m}$, the per-scene normalization factor of VGGT-Ω computed from LiDAR and averaged over 100 randomly sampled Waymo scenes.

Depth completion from the teacher. A frozen VGGT-Ω teacher, initialized from the released 1B backbone, predicts relative depth $\tilde{d}$ at every pixel. For each frame f we fit a single positive scalar a_f to the LiDAR pixels $\mathcal{L}_f$, with measured depth D_p^{L}, by minimizing an L_1 residual,

$$a_f = \arg\min_{a>0} \sum_{p\in\mathcal{L}_f} \left| D_p^{\mathrm{L}} - a\,\tilde{d}_p \right|, \tag{7}$$

whose solution is the weighted median of $D_p^{\mathrm{L}}/\tilde{d}_p$ with weights $\tilde{d}_p$, taken over pixels with positive predicted depth. The aligned prediction $a_f\tilde{d}$ then supplies a dense pseudo-metric target wherever LiDAR does not reach. The target depth in meters and its reliability weight are

$$D_p = \begin{cases} D_p^{\mathrm{L}}, & p \in \mathcal{L}_f \\ a_f\,\tilde{d}_p, & p \text{ filled} \end{cases} \qquad r_p = \begin{cases} 1, & \text{LiDAR} \\ w_{\mathrm{t}}, & \text{teacher fill} \\ 0, & \text{unsupervised,} \end{cases} \qquad w_{\mathrm{t}} = 0.01, \tag{8}$$

so that a teacher target carries $1/100$ of the weight of a LiDAR return. Sky pixels receive no teacher target; real LiDAR measurements are retained even where the sky mask incorrectly marks them as sky. Frames with fewer than 32 valid LiDAR pixels receive no teacher fill.

LiDAR fixes the metric scale on the $\approx 5\%$ of supervised-resolution pixels it reaches, and the teacher propagates shape over much of the remainder, with its scale verified only at those returns; the small w_{t} reflects that indirect verification. Sky pixels admit no return at all, so the teacher's scale cannot be verified there, and we do not substitute a nominal far-depth sky target.

Table 7: Sensitivity to the depth scaling factor s on Waymo.

	Depth		Pose		
s (m)	RMSE (m)↓	AbsRel↓	RTA@5↑	AUC@30↑	ATE (m)↓
19.8 (adopted)	**3.381**	**0.0522**	**0.845**	**0.913**	**0.0125**
30.0	3.393	0.0525	0.812	0.899	0.0143
40.0	3.393	0.0523	0.795	0.890	0.0159
50.0	3.391	0.0526	0.768	0.877	0.0171

Metrics. Depth: absolute metric depth pooled over all LiDAR returns, with no test-time alignment; AbsRel is the mean absolute relative error. Pose: RTA@5 is the fraction of frame pairs whose relative translation direction errs by less than 5°, AUC@30 is the area under the pose-accuracy curve up to 30°, and ATE is the absolute trajectory error. Bold marks the best result per column.

Scale-aware geometry objective. For each training sample, let $\mathcal{S} = \{p : r_p > 0\}$ collect supervised pixels across cameras and frames, and let $z_p = D_p/s$ be the normalized target depth. We back-project the predicted and target depths into their camera frames,

$$\hat{\mathbf{P}}_p = d_{\mathrm{f},p}(\mathbf{K}_p^{\mathrm{pred}})^{-1}\bar{\mathbf{u}}_p, \qquad \mathbf{P}_p = z_p(\mathbf{K}_p^{\mathrm{gt}})^{-1}\bar{\mathbf{u}}_p, \tag{9}$$

where $\bar{\mathbf{u}}_p$ is the homogeneous pixel coordinate. The two intrinsic matrices are decoded at the depth-map resolution from the predicted and the ground-truth camera encoding, respectively, so the objective supervises the predicted field of view alongside depth. The point-map objective is

$$\mathcal{L}_{\mathrm{geom}} = \frac{1}{3|\mathcal{S}|}\sum_{p\in\mathcal{S}} w_p \left\|\hat{\mathbf{P}}_p - \mathbf{P}_p\right\|_1, \qquad w_p = \frac{r_p}{\max(z_p, 0.1\bar{z}) + \epsilon}, \tag{10}$$

where $\bar{z}$ is the mean target depth over $\mathcal{S}$ and $\epsilon = 10^{-6}$. The floor $0.1\bar{z}$ bounds the weight that very near points can attract. The loss is averaged over supervised pixels and the three coordinate components, and then over training samples. No scale is fitted to the student point maps, so errors in absolute scale remain penalized; the teacher alignment of Eq. (7) serves only to construct the targets.

Objective and training. We train the geometry and pose heads with a π^3-style objective (Wang et al., 2026b) combining a geodesic rotation loss, a Huber loss on scale-aligned relative translations, the weighted point-map term of Eq. (10), and a confidence BCE on thresholded point error. The point-map term supervises metric geometry without scale alignment, and the confidence targets are likewise derived from scale-aware point errors. The DINO patch embedding remains frozen. Training uses Muon+AdamW, a 20k-step one-cycle schedule with peak learning rate 5×10^{-5}, bf16 precision, and gradient clipping at 1.0. Each five-camera window contains 4 past and 1 future key frames at stride 5; the input resolutions are 416×624 for the front camera and 352×768 for the side cameras. At evaluation, predicted normalized depth is multiplied by s and no test-time depth alignment is applied.

Sensitivity to s. We ablate s by retraining DAF with different $s \in \{30, 40, 50\}$ m, and evaluate on the five-camera Waymo evaluation set (Table 7). Each entry is the mean over the last seven evaluations (17k–20k steps). Depth is insensitive to the choice: RMSE varies by at most 0.012 m (0.4%) and does not change monotonically with s. Pose, in contrast, degrades monotonically as s grows: from s=19.8 to 50 m, RTA@5 falls from 0.845 to 0.768 and ATE rises by 37%.

Range-binned depth ablation. We perform a series of changes to the DAF recipe and evaluate them as a cumulative ladder on the Waymo Open Dataset (Table 8). The ladder begins at the released backbone with no depth training, adds teacher distillation at full weight, then down-weights the teacher to w_{t}=0.01 so that the sparse LiDAR returns are not outweighed by the dense teacher fill, then raises the input resolution to 416×624 front and 352×768 side, and finally excludes sky pixels from the teacher fill, which is the adopted recipe. We score each rung in three distance bands rather than pooled, because a pooled metric is dominated by the near field: 83% of Waymo's LiDAR returns lie within 30 m, so a recipe that improved only the near field would move it nearly as much as one that improved every range. Bands are aggregated from a uniform 5 m grid recorded at evaluation time, combining squared residuals and weighting by pixel count, so that each band is the exact RMSE over its own pixels rather than an average of the cells it spans. We exclude 0–5 m, where Waymo's roof LiDAR does not return and the reference comes almost entirely from the low-mounted short-range lasers. The three bands hold $82.7/12.4/4.8\%$ of the returns, and the Average column is their pixel-weighted aggregate over 5–75 m. Base-0 predicts scale-invariant depth and is therefore

Table 8: Range-binned depth RMSE of the cumulative DAF ladder on Waymo.

	Depth RMSE (m)↓ per range band (m)			
Recipe	5–30	30–50	50–75	Average
Base-0: released backbone, no depth training†	3.57	7.61	12.66	5.05
+ teacher distillation, w_t=1.0	3.31	7.64	13.03	4.95
+ teacher down-weighted to w_t=0.01	2.03	5.98	12.17	3.87
+ high resolution	1.79	**5.45**	**11.45**	3.56
+ sky excluded from the teacher fill (adopted)	**1.74**	**5.45**	11.53	**3.55**

Metrics. Average: pixel-weighted over the 5–75 m bands. †Scored after a global scale fit. Bold marks the best result per column.
Protocol. Every trained recipe is evaluated at its 10k-step checkpoint; Base-0 is untrained. Table 7 instead averages the last seven in-training evaluations over all LiDAR returns, so its RMSE is not directly comparable.

Table 9: Range-binned depth RMSE with and without the teacher, on Waymo and zero-shot on Argoverse 2.

	Depth RMSE (m)↓ per range band (m)								
	Waymo (in-domain)				Argoverse 2 (zero-shot)				
Recipe	5–30	30–50	50–75	Average	5–30	30–50	50–75	75–200	Average
Adopted (with teacher)	1.74	5.45	11.53	3.55	**5.40**	**11.41**	**19.08**	**47.63**	**14.10**
No teacher (w_t=0)	**1.69**	**5.26**	**10.80**	**3.38**	5.90	17.54	30.20	80.91	22.75

Metrics. Each Average aggregates its own dataset's bands. Bold marks the best result per column.
Protocol. The Argoverse 2 protocol is distinct from Table 4. Both recipes are evaluated at their 10k-step checkpoints.

scored after a single global scale fit; every other row is absolute metric depth with no test-time alignment.

The decomposition reveals what the pooled metric cannot: the benefit of the ladder decays sharply with range. From the zero-shot backbone to the adopted recipe, RMSE at 5–30 m falls by 51% ($3.57 \rightarrow 1.74$ m), whereas at 50–75 m it falls by only 9% ($12.66 \rightarrow 11.53$ m). The pooled reduction of 30% lies far closer to the near-field figure than to the far-field one solely because of how the returns are distributed. Two rungs are additionally separated by less than the bands resolve: at 50–75 m the high-resolution rung leads the adopted recipe by 0.08 m on the 4.8% of returns that reach that far.

In-domain and zero-shot evidence for the teacher. In domain, under the protocol of Table 8, removing the teacher is better in every band — by 3% near and 6% far (Table 9). This advantage does not transfer to Argoverse 2, where the teacher-free model has higher error in every band. We evaluate it with no Argoverse 2 training and no adaptation: `stereo_front_left` together with the four `ring` cameras closest to Waymo's front-left, front-right, and side orientations, center-cropped from 2048 to 1589 px so that the model observes Waymo's $50.4^\circ \times 34.8^\circ$ field of view rather than Argoverse 2's native 62.5°, and the same five-frame window, stride, and input resolutions. We use 140 of the 150 validation logs; the remaining ten omit the stereo extrinsics and cannot be calibrated. This protocol is separate from the zero-shot evaluation of Table 4, and its numbers are not comparable with that table. All entries are absolute metric depth with no test-time alignment. Each Average column aggregates that dataset's own bands (5–75 m for Waymo, 5–200 m for Argoverse 2), so Averages are comparable within a dataset but not across the two; Argoverse 2's four bands hold $71.2/15.6/7.5/5.8\%$ of its returns. The Argoverse 2 evaluation includes a 75–200 m band beyond the evaluated Waymo range. The teacher-free model's RMSE is $1.09\times$ that of the adopted recipe at 5–30 m and $1.70\times$ at 75–200 m, showing a larger transfer gap at long range.

The teacher supplies supervision outside the sparse LiDAR support. These results support retaining it for cross-dataset transfer, despite its slightly higher in-domain RMSE.

C Implementation Details

Training setup. The VGGT-Ω backbone uses a patch size of 16 and a feature dimension of 1024. The DAF schedule and input resolutions are provided in Appendix B. For 4D training, we freeze the adapted VGGT-Ω backbone, camera head, and dense-geometry head. We optimize the appearance, motion, and Gaussian render heads, including the additional cross-camera attention in the motion

head. Training starts from a static-reconstruction checkpoint, which initializes scene appearance before motion learning.

Optimization. We train for 120k steps with a global batch size of 256 clips. We use AdamW with a peak learning rate of 2×10^{-4} and weight decay of 10^{-2}. A one-cycle schedule increases the learning rate from 8×10^{-6} to its peak over the first 3% of steps, then applies cosine decay back to the initial rate. Training uses bf16 precision and differentiable Gaussian splatting through gsplat (Ye et al., 2025).

Loss weights. For the objective in Sec. 3.3, we use $\lambda_{\mathrm{p}} = 0.05$, $\lambda_{\mathrm{d}} = 0.5$, $\lambda_{\mathrm{s}} = 0.1$, $\lambda_{\alpha} = 8.0$, and $\lambda_{\mathrm{sm}} = \lambda_{\mathrm{fb}} = \lambda_{\mathrm{r}} = 1.0$. The RGB reconstruction term has unit weight, and no LiDAR depth loss is used during 4D training. The optional weakly supervised variant uses the same objective and weights, replacing only the RGB term with the region-weighted loss in Sec. 3.2. Its mask construction settings and region weight are specified in the next paragraph.

Optional weak-supervision settings. We construct moving-object masks from projected 3D boxes of vehicles, pedestrians, and cyclists with annotated speeds above $1\,\mathrm{m/s}$. We set $\gamma = 10$ in the weighted RGB objective of Sec. 3.2, giving masked pixels ten times the weight of other valid non-sky pixels. The loss is normalized by the total pixel weight and applied to both source and held-out target frames, including intermediate frames. The LPIPS term and all other loss terms remain unchanged.

Motion convention and regularization. The motion head predicts forward and backward world-space displacements in meters over one source-frame interval. These correspond to velocities with time measured in source-frame intervals; intermediate rendering scales each displacement by the corresponding fractional interval, as in Sec. 3.2.

Loss definitions. Let $\mathcal{P}$ be the valid non-sky pixels of all rendered source and held-out frames, $\mathcal{P}_{\mathrm{s}} \subset \mathcal{P}$ those of the source frames, and $\mathcal{G}$ the source Gaussians, which exclude sky (Sec. 3). For pixel p, $\hat{\mathbf{I}}_p$ and $\hat{D}_p$ are the rendered color and depth, $\mathbf{I}_p$ the image, d_p the frozen metric depth of Eq. (2), and $\mathbf{v}_p^{\pm}$ the forward and backward velocities. The photometric and depth terms are

$$\mathcal{L}_{\mathrm{rgb}} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \|\hat{\mathbf{I}}_p - \mathbf{I}_p\|_1, \qquad \mathcal{L}_{\mathrm{depth}} = \frac{1}{|\mathcal{P}_{\mathrm{s}}|} \sum_{p \in \mathcal{P}_{\mathrm{s}}} \left|\hat{D}_p - \mathrm{sg}(d_p)\right| + \mathcal{L}_{\nabla}, \tag{11}$$

where sg stops the gradient, depths are in meters, and $\mathcal{L}_{\nabla}$ is the multi-scale depth-gradient matching term of VGGT (Wang et al., 2025) on the same pixels; $\mathcal{L}_{\mathrm{rgb}}$ is the $\gamma = 1$ case of $\mathcal{L}_{\mathrm{rgb}}^{\mathrm{weak}}$. $\mathcal{L}_{\mathrm{LPIPS}}$ is the VGG-based LPIPS distance (Zhang et al., 2018) between $\hat{\mathbf{I}}$ and $\mathbf{I}$, computed per pixel and averaged over $\mathcal{P}$. The opacity and forward–backward terms are

$$\mathcal{L}_{\mathrm{opacity}} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} (1 - \alpha_g)^2, \qquad \mathcal{L}_{\mathrm{fb}} = \frac{1}{|\mathcal{P}_{\mathrm{s}}|} \sum_{p \in \mathcal{P}_{\mathrm{s}}} \left\|\mathbf{v}_p^{+} + \mathbf{v}_p^{-}\right\|_1, \tag{12}$$

the latter without dynamic weighting. The two spatial motion terms weight each pair (i, j) by $\omega_{ij} = \min(\sigma_i, \sigma_j)$, so that they act inside dynamic regions:

$$\mathcal{L}_{\mathrm{smooth}} = \frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} \omega_{ij} \left\|\mathbf{v}_i^{+} - \mathbf{v}_j^{+}\right\|_2, \qquad \mathcal{L}_{\mathrm{rigid}} = \frac{\sum_i \sum_{j \in \mathcal{N}_8(i)} \omega_{ij} \left\|[\mathbf{v}_i^{+}; \mathbf{v}_i^{-}] - [\mathbf{v}_j^{+}; \mathbf{v}_j^{-}]\right\|_2}{\sum_i \sum_{j \in \mathcal{N}_8(i)} \omega_{ij}}, \tag{13}$$

where $\mathcal{E}$ holds horizontally and vertically adjacent pixel pairs, and for $\mathcal{L}_{\mathrm{rigid}}$ we sample up to 1024 points per camera frame and take each point's eight nearest neighbors $\mathcal{N}_8(i)$ in world coordinates within the sampled set. $\mathcal{L}_{\mathrm{static}}$ is given in Sec. 3.2.

Cross-camera attention layout. The backbone receives a camera-major list of all cameras and source frames. Each of its 24 blocks applies within-view attention followed by inter-view attention. Using zero-based indices, blocks $2, 6, 9, 14, 20$ restrict inter-view attention to the camera/register prefixes; patch tokens pass through this inter-view operation. The other 19 blocks use global patch

attention across all cameras and source frames. This backbone attention remains frozen during 4D training.

The motion head instead groups patch grids in frame-major order, concatenating all cameras at each source frame. Its cross-camera path accepts unequal patch counts across cameras. Successive attention blocks alternate between forward and backward temporal directions: queries at frame f attend to all cameras at frame $f+1$ in forward blocks and at frame $f-1$ in backward blocks. Let $\mathbf{F}_{c,f}$ denote the motion-head tokens for camera c at source frame f, augmented with camera and temporal embeddings. We concatenate the tokens from all cameras at each frame:

$$\mathbf{F}_f = \mathrm{Concat}_{c=1}^{N} \mathbf{F}_{c,f}. \tag{14}$$

Within each attention block, queries from camera c at frame f attend to this joint token set at frame $f+\delta$:

$$\mathbf{A}_{c,f}^{\delta} = \mathrm{Attention}\big(\mathbf{Q}_{c,f}, \mathbf{K}_{f+\delta}, \mathbf{V}_{f+\delta}\big), \qquad \delta \in \{-1, +1\}, \tag{15}$$

where queries, keys, and values are projected from the corresponding tokens. The fused features are decoded into per-pixel velocities and static–dynamic mixing weights. Camera identity is encoded sinusoidally using 16 dimensions allocated from the existing time-embedding dimension, preserving the embedding size for warm-start compatibility. This motion-head attention is trained during the 4D stage. When its cross-camera option is disabled, the head runs separately for each camera.

D Evaluation Protocol Details

Cross-dataset scene selection. Candidates are restricted to the official validation splits: 150 nuScenes scenes and 150 Argoverse 2 Sensor logs. No training scenes are included. We retain scenes whose ego trajectory length is at least 3 m, leaving 136 nuScenes scenes and 149 Argoverse 2 logs.

We measure scene motion by the peak number of moving objects over its keyframes. An object is considered moving when its speed, estimated from adjacent annotations, exceeds 1 m/s. We include the `vehicle.*` and `human.pedestrian.*` categories for nuScenes, and vehicles, pedestrians, and cyclists for Argoverse 2. Within each filtered dataset, the lower and upper tertiles of the peak counts define low-, medium-, and high-motion groups. The thresholds are $(7, 13)$ for nuScenes and $(54, 85)$ for Argoverse 2: low includes counts at or below the lower threshold, medium includes counts above the lower and at or below the upper threshold, and high includes counts above the upper threshold.

We select 40 high-, 40 medium-, and 20 low-motion scenes from each dataset. Within each motion group, selection cycles through diversity groups defined by location, day/night, and rain/clear conditions for nuScenes, and by city for Argoverse 2. Candidates within each diversity group are ordered by decreasing peak moving-object count, with ties broken using seed 42. Both datasets satisfy the target allocation without redistributing selections across motion groups.

The selected nuScenes subset contains 55 Boston and 45 Singapore scenes, including 10 nighttime and 20 rainy scenes. The Argoverse 2 subset contains 29 Pittsburgh, 29 Miami, 18 Washington, D.C., 13 Detroit, 7 Palo Alto, and 4 Austin logs.

Camera evaluation. We distinguish the number of input cameras from the number of evaluated cameras. When a method receives one camera at a time but is evaluated across multiple cameras, we run it independently on each evaluated camera and aggregate the scores. This setting does not involve joint multi-camera processing.

Depth and pose alignment. For affine-aligned depth evaluation, we fit one scale and one shift to all valid ground-truth depth samples in a clip. The same transform is applied across its frames and regions. Using a single clip-level transform preserves sensitivity to temporal scale drift that separate per-frame fits could conceal. Metric depth evaluation uses no fit to ground-truth depth.

DGGT additionally requires a per-clip, per-camera Sim(3) alignment of its predicted camera trajectory to ground-truth poses for rendering. This pose alignment includes a scale factor and therefore affects its metric depth results, even though that column uses no direct depth alignment.

Table 10: Where the cross-camera agreement comes from.

	Cross-view 3D error (m)↓		Depth RMSE (m)↓
Variant	Mean	Median	Overlap
Full model	**0.72**	**0.21**	**2.33**
w/o DAF	1.01	0.31	4.91
Per-camera backbone	2.10	1.22	3.22

Protocol. Same model and protocol as Table 3; both variants change only how the frozen geometry backbone is run or which depth head it uses.
Metrics. As in Table 3. Bold marks the best result per column.

Cross-camera consistency evaluation. Table 3 uses the Waymo validation scenes of Table 2, comprising 397 clips from 202 scenes, and the four adjacent camera pairs (front–front-left, front–front-right, front-left–side-left, front-right–side-right); non-adjacent pairs share no field of view. A correspondence is a LiDAR return in camera A at a source frame that is outside the ground-truth moving-object boxes, whose 5×5 LiDAR neighborhood varies by less than 10% in depth (no depth edge), and whose projection into the adjacent camera B with the calibrated poses lies inside the image at $z_B > 0.5\,\text{m}$, outside the boxes, and is not occluded (no LiDAR return in B's 5×5 neighborhood closer than $0.9\,z_B - 0.5\,\text{m}$). Both directions of each pair are used; the fixed set of 18.7M correspondences is scored for every method. For each method the per-camera depth at the two projections is read at the method's own resolution (nearest pixel) and unprojected with the calibrated poses; we report the mean Euclidean distance between the two 3D estimates. With a small camera baseline, cross-view error can remain low despite a common depth-scale error. We therefore also report depth RMSE against LiDAR on the same points. Scores are averaged per camera pair within a scene (pairs with at least 50 correspondences), then over pairs and over scenes. The per-camera depth of X-Splatter is the depth-anchored backbone output that positions its Gaussians; DGGT's is its depth head divided by the scale of its per-clip, per-camera Sim(3) pose alignment (Sec. 4.1), which is fitted on four source-frame poses and occasionally ill-conditioned, so 9.9% of its estimates are invalid and DGGT† re-scales each camera by the median ratio to LiDAR for reference. STORM has no depth head; its per-camera estimate is the expected depth of rendering that camera's Gaussians alone into its own source views, invalid where the accumulated opacity is below 0.5 (8.8% of the correspondences).

The 4D training stage keeps the geometry backbone frozen, so the per-camera depth, and hence this metric, is unchanged across 4D checkpoints that share the same frozen geometry and camera-processing configuration. Table 10 compares joint processing with running the frozen backbone separately for each camera, and the full model with *w/o DAF*, which replaces the adapted depth head with the released relative-depth head. Both heads use the same fixed scale (Appendix F). Joint processing accounts for most of the agreement, and the adapted head adds a further reduction together with most of the metric accuracy (Figure 5).

E Concurrent Feed-Forward Driving Gaussian Methods

DrivingForward (Tian et al., 2025) and VGD (Lin et al., 2025) reconstruct static surround scenes. ReconDrive (Yu et al., 2026a) separates coordinate and appearance prediction and uses per-Gaussian motion in a VGGT-based 4D system. Other feed-forward driving methods include VR-Drive (Cho et al., 2025), Envision4D (Song et al., 2026), XYZCylinder (Yu et al., 2025), DIAL-GS (Su et al., 2026), InstDrive (Liu et al., 2025), and IDSplat (Lindström et al., 2026). Their representations, supervision, and evaluation protocols differ. Table 6 lists the methods for which we compare scale conventions and camera coverage explicitly.

F Component Ablation Details

Training and evaluation protocol. Table 5 uses the same architecture as the main model, with a global batch size of 16 clips (16 devices, one clip per device). The trained variants share the reference's initialization, training data, temporal sampling, and optimization schedule. We use AdamW with weight decay 10^{-2} and a one-cycle schedule configured for 120k steps, with a peak learning rate of 5×10^{-5}, 3% warm-up, and cosine decay. The initial and final learning rates are 2×10^{-6}. All reported scores are averaged over the 30k, 35k, and 40k-step checkpoints, evaluated with the

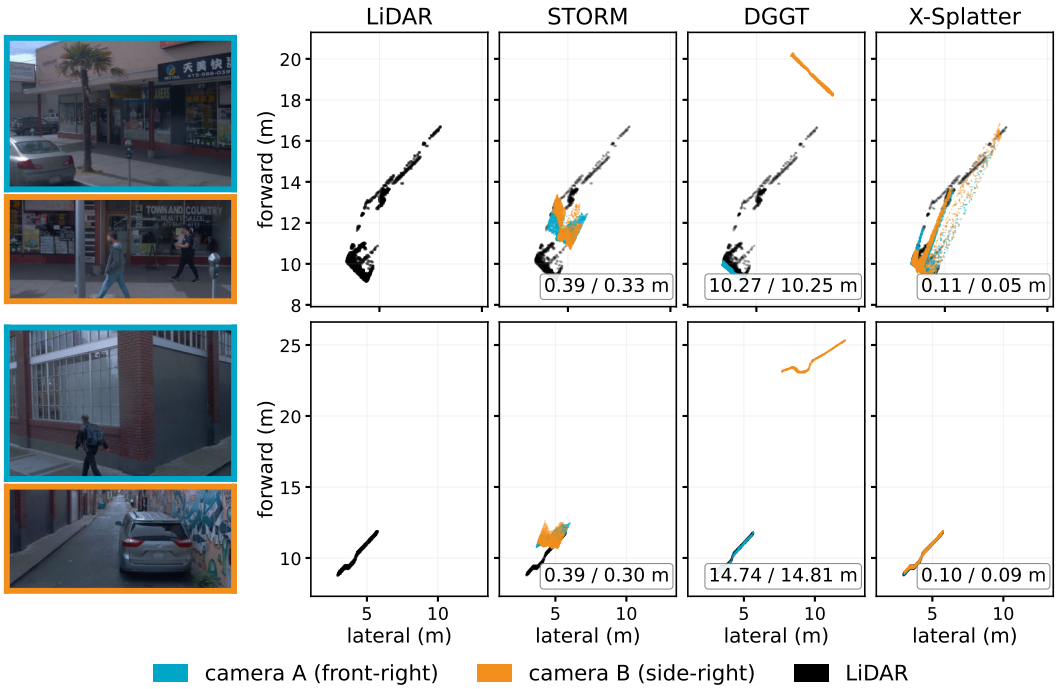

Figure 4: Cross-camera consistency in the overlap of the front-right (cyan) and side-right (orange) cameras, seen from above (lateral vs. forward distance in the front-right camera frame). Each panel unprojects every pixel of the two cameras' own depth estimates that falls into the shared field of view; LiDAR returns are shown in black. A consistent reconstruction puts both colors on the same surface, whereas a per-camera scale or depth error separates them into two offset surfaces. Boxes give the mean / median 3D distance between the two cameras' estimates of the same LiDAR points for that example.

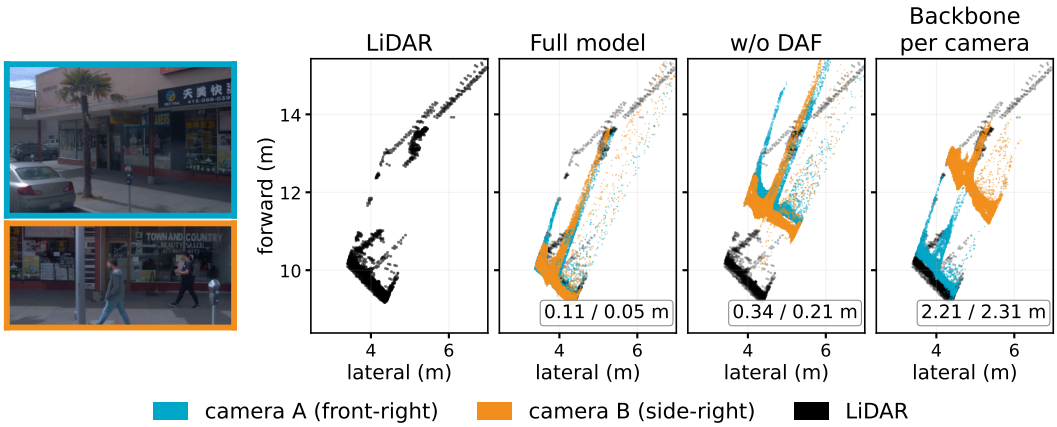

Figure 5: Cross-camera consistency of the model variants in Table 10 on the first example of Figure 4; same layout and legend.

Table 11: Cumulative 4D-stage ablation on Waymo (batch 16), each row adding to the preceding variant.

Variant	Dyn↑	Full↑	D-RMSE (m)↓
Cumulative ablation, batch 16			
Baseline (no sparsity, no held-out-time sup.)	21.98	27.10	5.03
+ dynamic sparsity	22.07	26.54	4.98
+ held-out time supervision	**22.32**	**27.38**	**4.83**

Budget. These runs use a smaller training allocation than the main model.
Metrics. Dyn/Full: PSNR on the dynamic region and the full image. D-RMSE: absolute depth RMSE. Bold marks the best result per column.

same protocol across all five Waymo cameras. The 120k-step schedule specifies the learning-rate trajectory; the table compares checkpoints at matched earlier steps. These reduced-budget results are compared within the ablation study, separately from the main model in Table 2.

w/o DAF. This variant replaces only the DAF-adapted depth head with the released VGGT-Ω head at each evaluated checkpoint, without further training. The DAF-adapted backbone and all other heads remain unchanged. Both variants use the same fixed conversion factor $s{=}19.8$ m, without fitting an additional scale. Metric and affine-aligned depth follow Table 2. This comparison measures the adapted head's contribution within the trained pipeline, including its compatibility with the adapted features.

Per-camera processing. The frozen backbone processes each camera's source frames separately, and cross-camera attention is disabled in the motion head during 4D training and evaluation. The adapted geometry weights, temporal attention, camera coverage, Gaussian aggregation, and SDDL objectives are retained. This variant removes cross-camera attention at every location where it is applied.

Joint removal of SDDL objectives. We disable supervision at intermediate and extrapolated timestamps and set the dynamic-sparsity weight to zero. Source-frame reconstruction, dynamic gating, opacity regularization, and the other motion regularizers remain unchanged. The reported comparison measures the combined effect of the two objectives; it does not establish the effect of removing dynamic sparsity alone from the full model.

G Supplementary Cumulative Ablation

Table 11 records the existing cumulative 4D-stage variants trained with batch 16, separately from the component study in Sec. 4.5. These runs use a smaller training allocation than the main model. Comparisons are made between rows of this block, not against the main model's absolute scores. Adding dynamic sparsity to the preceding variant lowers depth RMSE from 5.03 to 4.98 m and raises dynamic PSNR from 21.98 to 22.07 dB, while full-image PSNR falls from 27.10 to 26.54 dB. Adding temporal interpolation/extrapolation supervision then gives 22.32 dB dynamic PSNR, 27.38 dB full-image PSNR, and 4.83 m depth RMSE. This is a cumulative design record under reduced training scale, not an isolation of the metric-depth prior. In particular, the sparsity comparison precedes the addition of intermediate-frame supervision; it therefore does not measure removal of sparsity from a model trained with both objectives. These scores are not reused as results for the reference model or variants in Table 5.

H Further Qualitative Results

Figures 6 and 7 show three selected cases from nuScenes and Argoverse 2, respectively, without further training, using the same front-camera, held-out-time protocol as Figure 3. Each figure presents three RGB comparisons and the depth corresponding to Case 3. Matched boxes and insets highlight the same regions across methods. Depth visualization follows Figure 3; LiDAR points are dilated for visibility, with isolated returns above the horizon omitted.

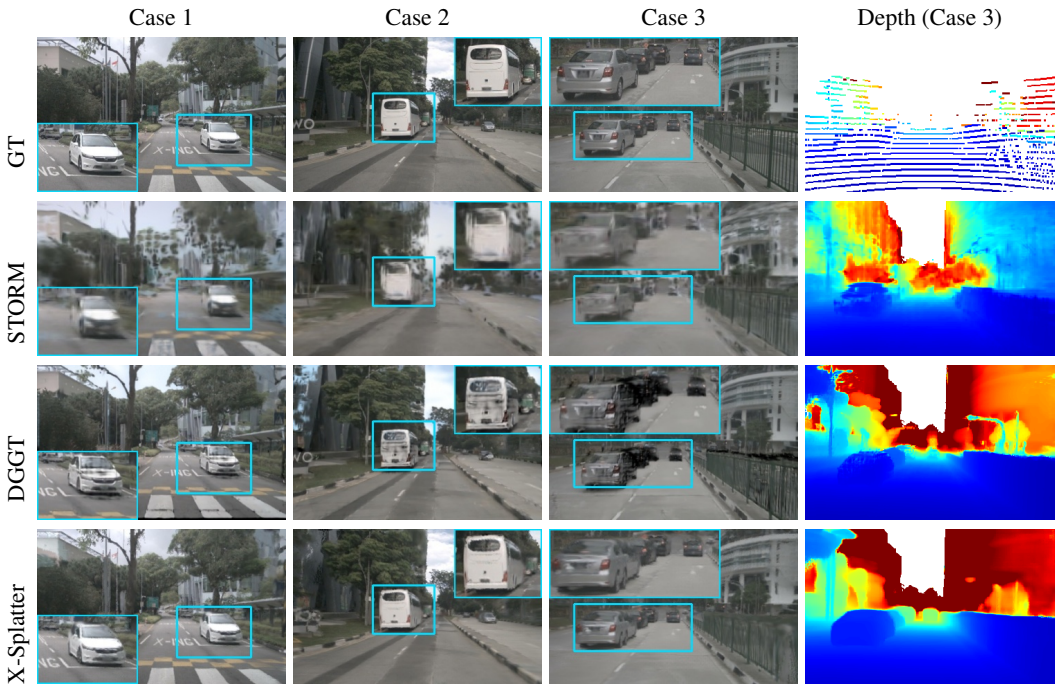

Figure 6: **Zero-shot qualitative comparison on nuScenes.** Three front-camera cases at held-out interpolation times. The final column shows depth for Case 3. Insets and depth visualization follow Figure 3.

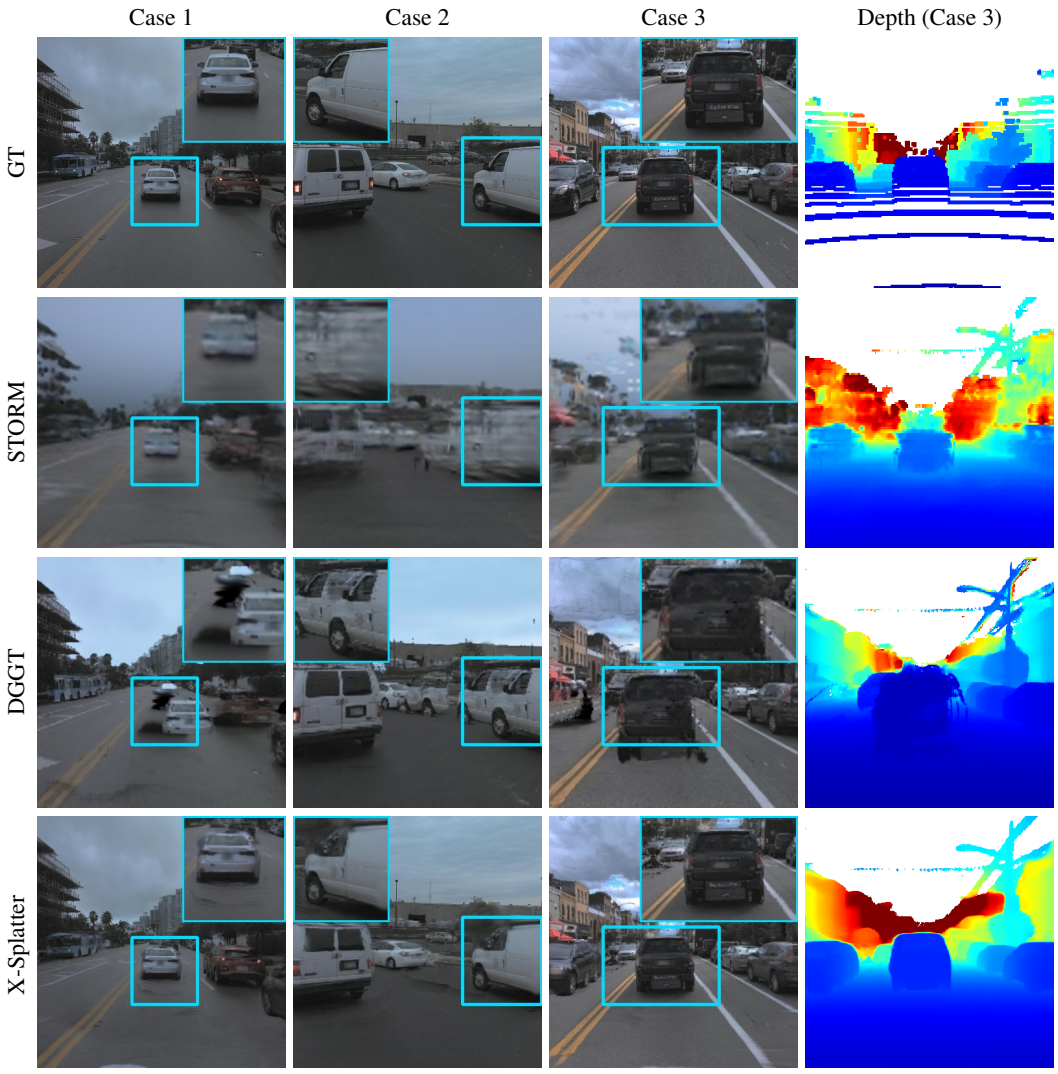

Figure 7: **Zero-shot qualitative comparison on Argoverse 2.** Three front-camera cases at held-out interpolation times. The final column shows depth for Case 3. Insets and depth visualization follow Figure 3.