S2D-SPLAT: STATIC-FIRST GAUSSIAN RECONSTRUCTION OF DYNAMIC DRIVING SCENES WITH LAYER-WISE FEATURE GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Moving objects introduce cross-frame inconsistencies that complicate camera estimation and static geometry recovery in driving scenes. Recent feed-forward Gaussian methods explicitly separate static and dynamic primitives, but often estimate camera parameters and geometry from shared features that jointly encode both components, leaving static reconstruction susceptible to dynamic interference. To address this challenge, we propose **S2D-Splat**, a feed-forward framework for **static-first Gaussian reconstruction** with **layer-wise feature guidance**. Our framework separates static and dynamic features before camera and geometry estimation, enabling the static branch to recover camera parameters and static 3D Gaussians with reduced dynamic interference. Intermediate static features are cached at each layer and reused as fixed conditioning at corresponding layers of the dynamic branch. This asymmetric information flow preserves the established static representations while providing stable geometric context for reconstructing dynamic 3D Gaussians. An instance-level motion decoder further queries the fused static and dynamic representations to estimate object trajectories, supporting motion-aligned Gaussian fusion and temporal interpolation. Experiments on Waymo and zero-shot evaluations on nuScenes and the MM-AU accident dataset demonstrate that S2D-Splat outperforms existing approaches on most evaluated metrics and generalizes to highly dynamic, safety-critical driving scenarios.

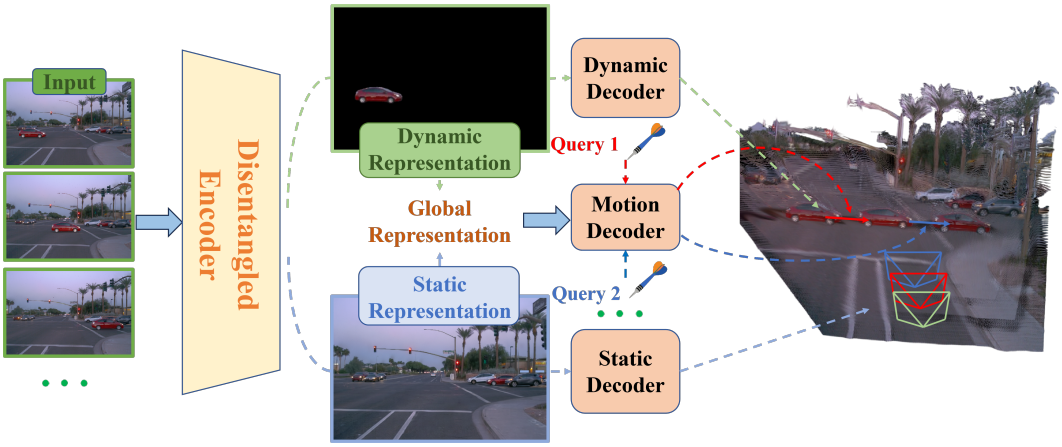


Figure 1: We propose S2D-Splat, a feedforward Gaussian reconstruction framework that achieves precise dynamic-static decoupling and instance-level motion prediction for complex driving scenes. It disentangles multi-frame input into static and dynamic representations, reconstructs each representation independently, and fuses them into a global representation that guides a motion decoder to recover accurate and temporally consistent per-instance 3D trajectories (e.g., Query 1, Query 2).

1 INTRODUCTION

Reconstructing dynamic driving scenes from image sequences provides a geometric basis for autonomous driving simulation, accident analysis, and near-miss replay. A useful reconstruction must

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.

recover camera motion, preserve the static environment, and represent moving agents coherently in a common 3D space. These requirements become particularly challenging in dense traffic and accident scenarios, where abrupt object motion and changing occlusions disrupt cross-frame correspondences. Errors in camera estimation can then distort static geometry and confound object motion, while fragmented dynamic reconstructions compromise the fidelity of scene replay. Reliable reconstruction therefore requires carefully managing the interaction between static scene recovery and dynamic object modeling.

Per-scene optimization has enabled high-quality reconstruction through methods such as 3DGS (Kerbl et al., 2023), EmerNeRF (Yang et al., 2024a), and DeformableGS (Yang et al., 2024b). However, substantial scene-specific computation and reliance on camera poses limit their scalability and complicate their application to uncalibrated driving videos. Feed-forward Gaussian methods address the computational burden by directly predicting renderable 3D representations from input images (Charatan et al., 2024; Chen et al., 2024). Approaches such as MVSplat (Chen et al., 2024), NoPoSplat (Ye et al., 2025), and DepthSplat (Xu et al., 2025) primarily target static scenes, leaving moving objects challenging to reconstruct consistently. STORM (Yang et al., 2025) extends feed-forward reconstruction to dynamic driving scenes through motion-based Gaussian aggregation, but requires known camera parameters and operates within short temporal windows.

More recent methods, including DGGT (Chen et al., 2026) and StreetForward (Yu et al., 2026b), support reconstruction from unposed images and explicitly distinguish static and dynamic Gaussian primitives. DGGT predicts dynamic masks to partition Gaussian maps, while StreetForward uses motion-aware features to predict velocities and dynamic probabilities for Gaussian separation and temporal aggregation. Despite their dedicated dynamics modules, camera parameters and Gaussian geometry are estimated from backbone features that jointly aggregate static and dynamic image content. Their Gaussian-level decomposition therefore does not explicitly isolate the static features used by these earlier estimation stages. Moving objects can still influence the representations from which camera parameters and static geometry are recovered, even when the resulting Gaussian primitives are subsequently assigned to separate components.

This observation points to an asymmetric dependency between static and dynamic reconstruction. Static scene recovery benefits from suppressing interference from moving objects, whereas dynamic reconstruction benefits from the geometric reference provided by the static environment. Early feature separation addresses the first requirement, but effective reconstruction also requires controlled interaction between the separated representations. In particular, static information should remain available to the dynamic branch without allowing dynamic updates to modify the established static representations. This motivates a static-first design that combines feature separation before camera and geometry estimation with directed, layer-wise guidance from static to dynamic features.

To this end, we propose **S2D-Splat**, a feed-forward framework for **static-first Gaussian reconstruction** of dynamic driving scenes with **layer-wise feature guidance**. As illustrated in Figure 1, an initial encoder extracts image features, which are partitioned using instance masks obtained with SAM2 (Ravi et al., 2025). A feature separator uses masked attention to restrict interactions from dynamic regions and deformable attention to refine dynamic features and recover static cues from mixed boundary tokens. By separating features before camera estimation and Gaussian reconstruction, the framework reduces dynamic interference during the recovery of static scene structure.

The static branch processes the separated features through an alternating-attention backbone (Wang et al., 2026a) to estimate camera parameters and reconstruct static 3D Gaussians. Its intermediate static features are cached at each layer and subsequently reused to guide the dynamic branch. Specifically, at every layer of dynamic reconstruction, static token positions are replaced with the corresponding cached static tokens, while dynamic tokens are updated through attention to the combined representation. This mechanism preserves the static representations established at each layer while allowing dynamic features to draw on their geometric context. The recovered camera parameters further place static and dynamic Gaussians in a common coordinate system, connecting the two reconstruction branches through a consistent geometric reference.

To recover object motion, we introduce an instance-level decoder that queries the fused static and dynamic representations. Each query specifies an instance and timestep, enabling selective decoding of instance poses. The estimated poses provide shared transformations for aligning an instance's Gaussians across frames, allowing complementary observations to be fused in a common object

reference frame. This supports temporal interpolation and coherent reconstruction of an object's overall motion under partial observations, extending the static-first geometry framework into a dynamic scene representation.

We train S2D-Splat on the Waymo Open Dataset (Sun et al., 2020) and evaluate its zero-shot generalization on nuScenes (Caesar et al., 2020) and the MM-AU accident dataset (Fang et al., 2024). Experiments demonstrate improved rendering quality and depth accuracy on Waymo, together with strong reconstruction and temporal interpolation performance on nuScenes. Evaluation on MM-AU further examines generalization to accident scenarios beyond the training distribution. Ablation studies investigate the effects of static-first decoupling, asymmetric layer-wise feature guidance, and instance-level motion decoding.

Our main contributions are:

- We introduce a static-first feed-forward Gaussian reconstruction framework that separates static and dynamic features before camera and geometry estimation, reducing dynamic interference during static scene recovery.
- We propose asymmetric layer-wise feature guidance that reuses cached static features as fixed conditioning for the dynamic branch, preserving the established static representations while providing geometric context for dynamic reconstruction.
- We develop an instance-level motion decoder that queries the fused scene representation to estimate object poses, enabling motion-aligned Gaussian fusion across frames and temporal interpolation.
- We demonstrate the effectiveness of S2D-Splat on Waymo and its zero-shot generalization to nuScenes and MM-AU, covering both conventional driving scenes and safety-critical accident scenarios.

2 Related Work

Feed-forward Geometry and Gaussian Reconstruction. Neural 3D scene representations support diverse robotic tasks (Deng et al., 2025) and Gaussian-based driving world models (Deng et al., 2026a). NeRF-based methods (Deng et al., 2026b; Yang et al., 2024a) and Gaussian-based driving reconstruction approaches (Zhou et al., 2024; Yan et al., 2024) achieve high visual fidelity through per-scene optimization, but incur substantial per-scene computation. Feed-forward methods instead learn transferable reconstruction priors across scenes. DUSt3R and VGGT establish generalizable geometric representations (Wang et al., 2024; 2025a), while VGGT-Ω improves reconstruction across static and dynamic scenes through scaled training (Wang et al., 2026a). Driving-specific models further incorporate sensor configurations and scene priors (Jia et al., 2025; Zuo et al., 2026a;b). In parallel, generalizable Gaussian methods directly predict renderable primitives from one or a few images (Xu et al., 2025; Ye et al., 2025; Zhang et al., 2025; Jiang et al., 2025; Zhang et al., 2024; Szymanowicz et al., 2025). For driving scenes, limited camera overlap and incomplete visibility motivate joint depth–pose learning (Tian et al., 2025), complementary pixel–volume representations (Wei et al., 2025), and volumetric fusion (Miao et al., 2025). Sparse LiDAR observations (Song et al., 2025) and pretrained geometry models (Lin et al., 2025; 2026) provide additional geometric constraints.

Static–Dynamic Separation and Feature Guidance. Existing methods address static–dynamic interaction at different stages of reconstruction. DrivingRecon (Lu et al., 2026) separates static and dynamic rendering supervision to exploit temporal observations. DGGT (Chen et al., 2026) predicts dynamic masks to partition Gaussian maps, while StreetForward (Yu et al., 2026b) constructs motion-aware features through causal attention and predicts velocities and dynamic probabilities for Gaussian separation and temporal aggregation. In both methods, camera parameters and Gaussian geometry are recovered from backbone features that jointly aggregate static and dynamic image content without explicitly static features isolation. Other approaches intervene earlier. MUT3R (Shen et al., 2025) derives motion cues from attention maps and suppresses dynamic interference in early transformer layers without retraining. DrivingScene (Hou et al., 2026) adopts a staged learning strategy that first establishes a static scene prior and subsequently trains a residual flow network with the static backbone frozen. S2D-Splat combines feature separation before camera and geometry estimation with directed, layer-wise static-to-dynamic guidance. After recovering camera parameters and

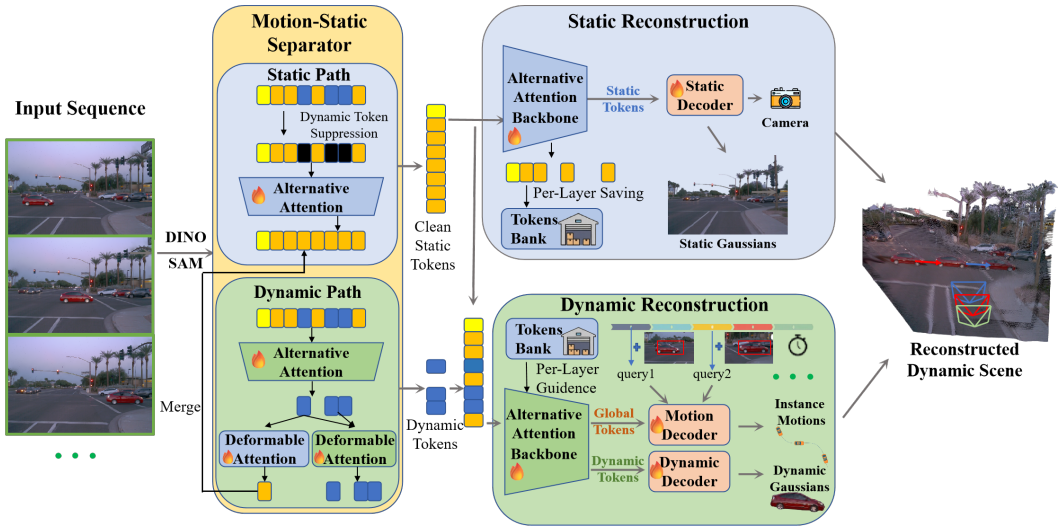


Figure 2: Overview of **S2D-Splat**. Given unposed images, we encode each frame into patchified features and use SAM2 to classify tokens as static or dynamic, which a Motion-Static Separator then refines into clean static and dynamic tokens (camera in ■, static in ■, dynamic in ■, suppressed dynamic in ■). The static tokens pass through alternating attention and are decoded by heads into camera parameters and a static Gaussian field, with each layer's camera and static tokens cached in a Tokens Bank. The dynamic reconstruction mirrors this but takes the fused static and dynamic tokens as input, re-fusing the extracted dynamic tokens with the Tokens Bank after each layer to yield global tokens and a dynamic Gaussian field. From the global tokens, instance-specific queries retrieve each instance's motion, and all components merge to form the full scene reconstruction.

static Gaussians, it reuses cached intermediate static features as fixed conditioning at corresponding layers of the dynamic branch. This preserves the established static representations while allowing dynamic tokens to use their geometric context throughout reconstruction.

Motion Modeling in Dynamic Scenes. Feed-forward dynamic reconstruction methods represent temporal changes through several complementary mechanisms. STORM, Flux4D, and UFO-4D predict motion fields to align dynamic geometry (Yang et al., 2025; Wang et al., 2025b; Hur et al., 2026). ReconDrive models Gaussian motion via segment-wise velocities (Yu et al., 2026a), while Instant NuRec predicts piecewise-linear trajectories (NVIDIA, 2026). UniSplat fuses observations within a 3D latent scaffold (Shi et al., 2026), and Ground4D performs localized temporal Gaussian aggregation for off-road scenes (Wang et al., 2026b). Object-centric approaches offer another form of temporal organization: UFO uses object poses to guide dynamic modeling (Tan et al., 2026), while PointForward uses 3D boxes and scene graphs for instance-level motion propagation (Chi et al., 2026). D4RT (Zhang et al., 2026b) introduces a query-based formulation that independently decodes point positions across time from a global spatio-temporal representation, supporting both sparse and dense queries. Following this formulation, our motion decoder operates at the instance level, by querying fused static and dynamic features to estimate a selected instance's pose at a requested timestep. The resulting transformations align its Gaussians across observations, supporting motion-aligned fusion and temporal interpolation.

3 S2D-SPLAT

We propose **S2D-Splat**, a pose-free feedforward reconstruction framework that achieves precise decoupling of static and dynamic reconstruction. To prevent dynamic noise from contaminating the static reconstruction, we first design a Motion-Static Separator that precisely decouples the dynamic and static features (§ 3.1). Building on these separated features, we adopt a two-stage reconstruction strategy that constructs the static and dynamic Gaussian fields separately: once static reconstruction is shielded from dynamic objects, the dynamic objects are in turn reconstructed with reference to

this clean static scene information (§ 3.2). The framework is **static-first**: separation precedes camera and geometry estimation, and the two stages are coupled by **asymmetric layer-wise feature guidance**. Finally, inspired by D4RT (Zhang et al., 2026b), we introduce a target-driven motion decoder that directly estimates each dynamic object's overall relative motion between frames, rather than aggregating point-wise motion (§ 3.3). The complete architecture is illustrated in Figure 2; we additionally build a separate sky model to handle the sky region independently (see Appendix A).

3.1 Motion-Static Separator

Given an unposed RGB image sequence $\{I^t \mid I^t \in \mathbb{R}^{H\times W\times 3}, t = 1, \ldots, N\}$, we encode each frame with a frozen DINOv3 (Siméoni et al., 2025) encoder to obtain patch token sequences $F^t \in \mathbb{R}^{L\times d}$, where L is the number of patches and d the feature dimension. In parallel, SAM2 (Ravi et al., 2025) produces a binary dynamic mask $M^t \in \{0,1\}^L$ for each frame by segmenting from the 2D box of each dynamic object, where $M_i^t = 1$ indicates a dynamic token at position i. Based on M^t, the token sequence is partitioned into a static subset $F_s^t = \{F_i^t \mid M_i^t = 0\}$ and a dynamic subset $F_d^t = \{F_i^t \mid M_i^t = 1\}$.

The separator processes these tokens through two parallel paths. In the **Static Path**, the token sequence is processed through three layers of alternating attention (Wang et al., 2026a), following the within-frame and cross-frame design of VGGT-Ω. To ensure the static tokens remain uncontaminated by dynamic content throughout all layers, we impose a masking constraint on each attention computation: the attention logits over dynamic key positions are forced to $-\infty$ before the softmax:

$$\hat{A}^{(t)}_{i,(t',j)} = \begin{cases} \mathbf{q}_i^{(t)\top}\mathbf{k}_j^{(t')}/\sqrt{d} & \text{if } M_j^{t'} = 0, \\ -\infty & \text{if } M_j^{t'} = 1, \end{cases} \qquad \tilde{F}_s^t = \mathrm{softmax}\big(\hat{A}^{(t)}\big)\, V, \tag{1}$$

where t and t' denote the frames of the query and of the key, respectively. This ensures the refined static tokens $\tilde{F}_s^t$ carry no dynamic information. In the **Dynamic Path**, all tokens are processed through the same three-layer alternating attention (Wang et al., 2026a) without any masking, allowing full cross-frame and within-frame context to flow freely across both static and dynamic positions. At the output stage, two improved deformable attention branches (Zhu et al., 2021) are applied. The first branch operates over all dynamic tokens F_d^t to further refine and extract pure dynamic representations. Each dynamic token attends to P neighboring tokens whose positions are predicted by the query via learned offsets:

$$\hat{\mathbf{f}}_i = W_{\mathrm{proj}} \sum_{k=1}^{P} \alpha_{ik}\, \mathbf{v}_{ik}, \quad \alpha_{ik} = \mathrm{softmax}_k\left(\frac{\mathbf{q}_i^\top \mathbf{k}_{ik}}{\sqrt{d}}\right), \tag{2}$$

where P is the number of sampled tokens per query, $\mathbf{q}_i = W_q\mathbf{f}_i$, $\mathbf{k}_{ik}$ and $\mathbf{v}_{ik}$ are key and value features interpolated from the patch token grid at query-predicted offset positions, and W_{proj} is the output projection. The second branch operates over the boundary subset $\mathcal{B} = \{i \mid M_i^t = 1, \exists j \in \mathcal{N}(i) : M_j^t = 0\}$, *i.e.,* tokens within the dynamic mask that also overlap with static regions and therefore carry mixed dynamic-static information. This branch extracts the static component from these mixed tokens, which is then weighted-fused into the corresponding positions in $\tilde{F}_s^t$ from the Static Path, ensuring that static background information at dynamic-object boundaries is fully recovered rather than discarded. Separation precedes camera and geometry estimation, so Stage 1 recovers camera parameters and static geometry from features no moving object can reach.

3.2 Two-stage reconstruction

Once the static and dynamic tokens have been separated, we initiate a two-stage reconstruction process: a static reconstruction stage, referred to as Stage 1 below, followed by a dynamic reconstruction stage, referred to as Stage 2 below.

Static Reconstruction Stage 1 takes the clean static tokens $\tilde{F}_s^t$ from the Motion-Static Separator and processes them through L layers of alternating attention (Wang et al., 2026a). Since all dynamic content has been suppressed in $\tilde{F}_s^t$, the backbone can focus entirely on static scene geometry. At every layer, the camera and static tokens are cached into the **Tokens Bank**, later reused by Stage 2

to condition dynamic reconstruction. The resulting multi-layer tokens are decoded by three dedicated heads. The **camera head** refines the camera and register tokens through a cross-frame self-attention trunk to decode each frame's camera-to-world extrinsics and intrinsics. The **depth head** and **Gaussian head** instead operate on the patch tokens: both fuse features from several intermediate backbone layers through a shared DPT-style multi-scale fusion pyramid, before a final convolutional layer predicts, at every pixel, a depth value with confidence for the depth head and a set of Gaussian attributes for the Gaussian head, namely opacity, spherical-harmonic color, scale, and rotation. The predicted depth is combined with the camera pose and intrinsics and back-projected to obtain each Gaussian's 3D center in world coordinates, which, with the remaining attributes, instantiates the static 3D Gaussian field.

Dynamic Reconstruction Stage 2 reconstructs the dynamic tokens from the Motion-Static Separator. To fully exploit the static features from Stage 1, the clean static tokens and the dynamic tokens are first fused into combined tokens, which serve as the initial input to L layers of alternating attention. After each layer's attention computation, only the dynamic tokens are extracted and fused with the cached static tokens retrieved from the Tokens Bank, to form the input to the next layer; the attention outputs at the camera and static positions are discarded rather than carried forward. We term this **asymmetric layer-wise feature guidance**: the cached static tokens act as fixed conditioning the dynamic branch attends to but never modifies, so the static representations from Stage 1 are preserved while dynamic tokens draw on their geometric context:

$$\hat{F}_i^{(l)} = \begin{cases} F_i^{(l),\,\text{stage1}} & \text{if } M_i^t = 0, \\ \mathrm{AltAttn}\big(\hat{F}^{(l-1)}\big)_i & \text{if } M_i^t = 1, \end{cases} \tag{3}$$

where t is the frame to which token i belongs and $F^{(l),\,\text{stage1}}$ is the static token cached at layer l in the **Tokens Bank**; only the dynamic token positions evolve through the network. We denote the final-layer tokens, fused across all frames, as $F^{(L)}$; besides serving as the global memory bank for the instance-level motion decoder (Section 3.3), the dynamic tokens extracted from $F^{(L)}$ at the dynamic-masked positions are decoded by a depth head and a Gaussian head into per-pixel depth and Gaussian attributes. Back-projecting this depth using the camera pose and intrinsics recovered in Stage 1 yields each dynamic Gaussian's 3D center in world coordinates for every frame, which, with the remaining attributes, instantiates the dynamic 3D Gaussian field.

3.3 Instance-level motion decoder

To model the 4D motion of dynamic Gaussians, we draw inspiration from D4RT (Zhang et al., 2026b) and introduce a motion decoder D, a compact cross-attention Transformer that constructs instance queries attending to spatial representations to estimate each instance's motion. After the dynamic Gaussians are reconstructed in Stage 2, for each instance we form a query by adding a learnable embedding of its 2D bounding-box prompt $b_m = (x_1, y_1, x_2, y_2)$ to a discrete timestep embedding of the frame t_m in which it is observed. We then adaptively size a sampling window based on the box dimensions, extract the corresponding RGB patch embedding, and add it to the query, giving the query appearance context beyond the box coordinates.

Each query is decoded through M stacked cross-attention blocks against the global memory bank $F^{(L)}$ (Section 3.2), formed by fusing the static representations from Stage 1 with the dynamic representations from Stage 2 across every frame, jointly encoding static and dynamic information; with no self-attention among queries, every instance is decoded independently of the others:

$$\mathbf{q}_m^{(l)} = \mathrm{CrossAttn}\big(\mathbf{q}_m^{(l-1)}, F^{(L)}\big), \quad l = 1, \ldots, M, \tag{4}$$

where $\mathbf{q}_m^{(0)}$ is the initial query described above and $\mathbf{q}_m^{(M)}$ is passed to the prediction head below. The refined query is then passed to an MLP head that predicts the 2D projection of the instance's 3D center, its depth, and its rotation angle. Back-projecting the predicted 2D projection and depth through standard pinhole geometry, using the source frame's camera intrinsics from the Stage-1 output, yields its 3D center in that frame's camera coordinates, which, together with the predicted rotation, gives its 3D pose in the queried frame's local camera space. We then transform this pose into a common world coordinate system using the corresponding frame's predicted camera extrinsics. Querying the same instance at a different frame yields a second pose, independently decoded

and likewise transformed into the same world coordinate system, and the rigid motion between the two frames is obtained directly as the difference between these two world-space poses, rather than being regressed as a separate output. This directly predicts object-level pose rather than per-point displacement, making the formulation more robust to occlusion and fast motion.

3.4 Training Objectives

We adopt a three-stage training strategy that mirrors the three-stage reconstruction pipeline (Section 3.2 and Section 3.3): Stage 1 supervises the static reconstruction together with camera estimation, Stage 2 supervises the dynamic reconstruction, and Stage 3 supervises the instance-level motion decoder once static and dynamic reconstruction are complete.

Static-region loss. In Stage 1, we supervise the static reconstruction with the loss function:

$$\mathcal{L}_{static} = \lambda_1 \, \ell_{rgb}^{static} + \lambda_2 \, \ell_{depth}^{static} + \lambda_3 \, \ell_{camera}. \tag{5}$$

Here, ℓ_{rgb}^{static} is the rendering loss used to supervise the static region, computed between the rendered and ground-truth frames restricted to static pixels; it mainly consists of an ℓ_1 term, an SSIM term, and an LPIPS perceptual term (Zhang et al., 2018). ℓ_{depth}^{static} supervises the static point cloud with a log-space ℓ_1 regression term combined with a multi-scale gradient-consistency term; the supervision is applied in a scale-normalized space, with the ground-truth depth normalized by the scene's median LiDAR depth. ℓ_{camera} supervises the camera by comparing the predicted pose encoding to the ground truth with an ℓ_1 loss on the translation and focal-length components and a quaternion-based cosine loss on the rotation component.

Dynamic-region and motion loss. In Stage 2, we train the model to reconstruct the dynamic region. In Stage 3, once static and dynamic reconstruction are complete, we train the instance-level motion decoder. The two objectives are

$$\mathcal{L}_{dynamic} = \lambda_1 \, \ell_{rgb}^{dynamic} + \lambda_2 \, \ell_{depth}^{dynamic}, \qquad \mathcal{L}_{motion} = \ell_{motion}. \tag{6}$$

Here, $\ell_{rgb}^{dynamic}$ and $\ell_{depth}^{dynamic}$ take the same form as ℓ_{rgb}^{static} and ℓ_{depth}^{static} (Eq. 5) in Stage 1, but are evaluated only over the dynamic region. ℓ_{motion} supervises the motion of each dynamic instance: we compute the ℓ_1 distance between the rigid motion predicted by the instance-level motion decoder (Section 3.3) and the ground-truth relative rotation and translation of that instance between consecutive frames, and average the result over all instances.

4 Experiments

4.1 Experimental settings

We train the feed-forward model on the original training split of the Waymo Open Dataset (Sun et al., 2020), a widely used autonomous driving benchmark comprising real-world driving logs collected across diverse urban scenarios (798 scenes, approximately 190–200 frames each), and evaluate on the test split (202 scenes). **To assess generalization**, we conduct zero-shot experiments on both the nuScenes dataset (Caesar et al., 2020) and the MM-AU traffic accident dataset (Fang et al., 2024). MM-AU is the largest and most fine-grained multimodal accident dataset, comprising 58 accident categories, from which we select one scene per category for evaluation. To measure visual quality, we used standard image quality metrics including PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018); depth accuracy is measured with depth RMSE (D-RMSE), computed after scale alignment between the predicted and ground-truth depth. Beyond reconstruction at the original observed viewpoints, we additionally evaluate novel view synthesis at viewpoints interpolated between input frames. Full implementation and training details are provided in Appendix B.

4.2 MAIN RESULTS

Waymo We evaluate on a single-camera setting, sampling context frames at a stride of 5 for a total of 8 input frames and interpolating the intermediate frames as target views for evaluation. We compare against feed-forward baselines spanning static and dynamic settings, including MVSplat (Chen et al., 2024), DepthSplat (Xu et al., 2025), and AnySplat (Jiang et al., 2025) for static scenes, as

Table 1: **Quantitative Comparison on Waymo.** All methods use 8 input frames, sampled at a stride of 5, as context under our single-camera setting, with intermediate frames as interpolated novel views for evaluation. Higher PSNR/SSIM and lower D-RMSE are better.

Method	Full Image			Dynamic Region		
	PSNR ↑	SSIM ↑	D-RMSE ↓	PSNR ↑	SSIM ↑	D-RMSE ↓
MVSplat	20.285	0.601	10.798	16.263	0.431	12.811
DepthSplat	21.603	0.586	9.923	17.828	0.316	11.995
AnySplat	24.146	0.761	7.395	19.613	0.477	9.816
DGGT	24.277	0.763	4.052	19.827	0.505	6.078
StreetForward	25.237	0.809	3.017	20.252	0.524	5.439
Ours	**25.953**	**0.814**	**2.152**	**22.043**	**0.719**	**4.065**

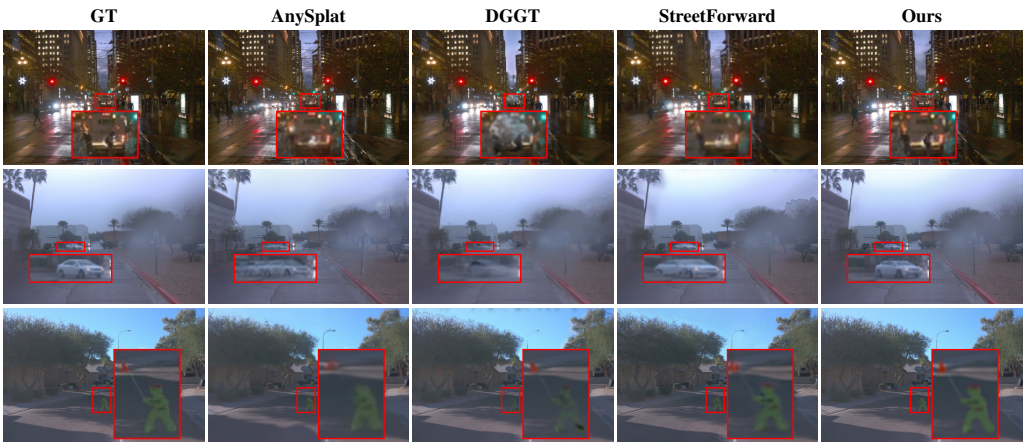


Figure 3: **Qualitative comparison on Waymo.** Compared to AnySplat, DGGT, and StreetForward, our method produces sharper structures and more accurate geometry around dynamic objects, consistent with its lower D-RMSE and higher PSNR/SSIM in Table 1.

well as DGGT (Chen et al., 2026) and StreetForward (Yu et al., 2026b), which model dynamic content but process static and dynamic reconstruction jointly without explicit decoupling. Quantitative results are summarized in Table 1, with all baselines evaluated by us under this same setting. Our method consistently outperforms all baselines across PSNR, SSIM, and D-RMSE while remaining pose-free. The qualitative comparisons in Figure 3 illustrate these gains: coupled baselines exhibit artifacts and blurred geometry around moving objects, whereas our decoupling strategy prevents dynamic contamination, yielding sharper and more geometrically accurate reconstructions.

nuScenes and MM-AU We evaluate generalization on nuScenes and MM-AU. For nuScenes, we take the 8 keyframes as input and evaluate on the intermediate non-keyframes; results are shown in Table 2. For MM-AU, all results in Table 3 are obtained zero-shot with the Waymo-trained model: the clips selected for evaluation are all drawn from before the collision, and each clip's 8 frames are fed to the model to evaluate reconstruction under extreme driving scenarios. Under these conditions, the quality of all baseline methods degrades substantially, whereas our method continues to generalize robustly; we attribute this to our dynamic-static decoupled design, which prevents sudden object motion from degrading reconstruction even in unseen traffic-accident scenarios. Qualitative comparisons are provided in Appendix C (Figures 4 and 5).

4.3 Ablation Study

Static-first decoupling. To evaluate static-first decoupling as a whole, we remove the entire decoupled design, namely the Motion-Static Separator, the two-stage reconstruction pipeline, and the dedicated depth and Gaussian heads, and replace it with a single coupled baseline: patch tokens from the Feature Extractor are directly processed by one VGGT-Ω backbone and decoded by a sin-

Table 2: **Quantitative Comparison under the Zero-Shot Setting on nuScenes.** Our Waymo-trained model shows strong generalization ability in zero-shot NVS on nuScenes.

Method	Scene Reconstruction			NVS		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
AnySplat	25.136	0.837	0.165	22.204	0.701	0.289
DGGT	25.352	0.862	**0.152**	21.861	0.707	0.227
StreetForward	25.090	0.937	0.182	24.610	0.787	0.295
Ours	**29.342**	**0.948**	0.157	**25.503**	**0.805**	**0.217**

Table 3: **Quantitative Reconstruction Comparison on MM-AU.** Zero-shot results show our method's strong generalization to extreme accident scenarios.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
AnySplat	20.150	0.754	0.316
DGGT	20.652	0.794	**0.189**
StreetForward	22.150	0.920	0.328
Ours	**25.371**	**0.932**	0.204

Table 4: **Ablation on static-first decoupling, asymmetric layer-wise feature guidance, and the instance-level motion decoder.**

Method	PSNR ↑	SSIM ↑	D-RMSE ↓
w/o Decoupling	24.719	0.772	3.129
w/o Guidance	25.521	0.795	2.539
w/o Decoder	25.841	0.808	2.269
Ours	**25.953**	**0.814**	**2.152**

gle shared Gaussian head. Since the design is removed as a unit rather than one component at a time, the difference this baseline reports is the aggregate benefit of decoupling; the component-level effects of the guidance mechanism and of the motion decoder are isolated in the ablations below. Dynamic-object motion then interferes with static scene recovery directly, dropping all metrics substantially (Table 4).

Asymmetric layer-wise feature guidance. Table 4 evaluates our asymmetric layer-wise feature guidance, which at every layer fuses the dynamic tokens produced by the current layer with the corresponding cached static tokens retrieved from the Tokens Bank into a complete token set for the next layer. Removing this Tokens Bank fusion causes the dynamic branch to lose access to clean, accurate static priors, so it can no longer exploit this useful information when rendering the dynamic region, visibly lowering the fidelity of that region.

Instance-level motion decoder. To evaluate the motion decoder's contribution, we replace it with a commonly used alternative: instance point clustering combined with TAPIP3D (Zhang et al., 2026a), an external point-tracking model, to estimate motion instead. Removing the decoder degrades performance across all evaluated metrics (Table 4). This indicates that the motion decoder's direct prediction of instance-level motion is more accurate than the alternative approach of geometric cluster centroids tracked by an external pure-translation tracker, since the latter not only fails to predict rotation but is also prone to being biased by noise points from dynamic objects.

5 Conclusion

We presented **S2D-Splat**, a pose-free, feed-forward framework for static-first Gaussian reconstruction of dynamic driving scenes. By separating static and dynamic features before camera and geometry estimation, our method reduces dynamic interference during static scene recovery. Cached static features then provide fixed guidance to corresponding layers of the dynamic branch, preserving the established static representations while supplying geometric context for dynamic reconstruction. An instance-level motion decoder further queries the fused scene features to estimate object poses, supporting motion-aligned Gaussian fusion and temporal interpolation. Experiments on Waymo and zero-shot evaluations on nuScenes and MM-AU demonstrate the effectiveness and cross-domain generalization of this design, including in highly dynamic, safety-critical scenarios.

AI USE STATEMENT

In this work, we used generative AI tools, including GPT-5.4 and Claude Sonnet 5, for writing assistance, text polishing, code debugging, and support during the evaluation process. All reported metrics and derived comparisons were computed directly from experimental evaluations, not estimated or fabricated by generative AI. We have reviewed all AI-assisted content and take responsibility for the final text, claims, and artifacts presented in this paper.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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*, 2026.

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*. Springer Nature Switzerland, 2024.

Cheng Chi, Xianqi Wang, Hongcheng Luo, Mingfei Tu, Gangwei Xu, Zehan Zhang, Bing Wang, Guang Chen, Hangjun Ye, Sida Peng, Xin Yang, and Haiyang Sun. PointForward: Feedforward Driving Reconstruction through Point-Aligned Representations. *arXiv preprint arXiv:2605.11594*, 2026.

Tianchen Deng, Yue Pan, Shenghai Yuan, Dong Li, Chen Wang, Mingrui Li, Long Chen, Lihua Xie, Danwei Wang, Jingchuan Wang, Javier Civera, Hesheng Wang, and Weidong Chen. What Is The Best 3D Scene Representation for Robotics? From Geometric to Foundation Models. *arXiv preprint arXiv:2512.03422*, 2025.

Tianchen Deng, Xuefeng Chen, Yi Chen, Qu Chen, Yuyao Xu, Lijin Yang, Le Xu, Yu Zhang, Bo Zhang, Wuxiong Huang, and Hesheng Wang. GaussianDWM: 3D Gaussian Driving World Model for Unified Scene Understanding and Multi-Modal Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2026a.

Tianchen Deng, Yanbo Wang, Yejia Liu, Chenpeng Su, Jingchuan Wang, Hesheng Wang, Danwei Wang, Shao-Yuan Lo, and Weidong Chen. ProSGNeRF: Progressive dynamic neural scene graph with frequency modulated foundation model in urban scenes. *International Journal of Computer Vision*, 134:365, 2026b.

Jianwu Fang, Lei-lei Li, Junfei Zhou, Junbin Xiao, Hongkai Yu, Chen Lv, Jianru Xue, and Tat-Seng Chua. Abductive Ego-View Accident Video Understanding for Safe Driving Perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Qirui Hou, Wenzhang Sun, Chang Zeng, Chunfeng Wang, Hao Li, and Jianxun Cui. DrivingScene: A Multi-Task Online Feed-Forward 3D Gaussian Splatting Method for Dynamic Driving Scenes. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2026.

Junhwa Hur, Charles Herrmann, Songyou Peng, Philipp Henzler, Zeyu Ma, Todd Zickler, and Deqing Sun. UFO-4D: Unposed Feedforward 4D Reconstruction from Two Images. *arXiv preprint arXiv:2602.24290*, 2026.

Xiaosong Jia, Yanhao Liu, Junqi You, Renqiu Xia, Yu Hong, and Junchi Yan. DriveVGGT: Visual Geometry Transformer for Autonomous Driving. *arXiv preprint arXiv:2511.22264*, 2025.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, et al. AnySplat: Feed-forward 3D Gaussian Splatting from Unconstrained Views. *ACM Transactions on Graphics*, 44:1–16, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42, July 2023.

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.

Junhong Lin, Jinlong Wang, Xianda Guo, Yanlun Peng, Wei Zheng, Guoqing Liu, Hanli Wang, Tiesong Zhao, and Wei Gao. Visual Geometry Foundation-Aware Gaussians for Single-Frame Surround-View Driving Reconstruction. *arXiv preprint arXiv:2608.10682*, 2026.

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. In *Advances in Neural Information Processing Systems*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2025.

NVIDIA. Instant NuRec: Feed-Forward 3D Gaussian Reconstruction for Driving Scene Simulation. Technical report, NVIDIA, 2026.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos. In *International Conference on Learning Representations*, 2025.

Guole Shen, Tianchen Deng, Xingrui Qin, Nailin Wang, Jianyu Wang, Yanbo Wang, Yongtao Chen, Hesheng Wang, and Jingchuan Wang. Mut3r: Motion-aware updating transformer for dynamic 3d reconstruction. *arXiv preprint arXiv:2512.03939*, 2025.

Chen Shi, Shaoshuai Shi, Xiaoyang Lyu, Chunyang Liu, Kehua Sheng, Bo Zhang, and Li Jiang. UniSplat: Unified Spatio-Temporal Fusion via 3D Latent Scaffolds for Dynamic Driving Scene Reconstruction. In *International Conference on Learning Representations*, 2026.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, and Francisco Massa. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Qi Song, Chenghong Li, Haotong Lin, Sida Peng, and Rui Huang. ADGaussian: Generalizable Gaussian Splatting for Autonomous Driving with Multi-modal Inputs. *arXiv preprint arXiv:2504.00437*, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F. Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3D: Feed-Forward Generalisable 3D Scene Reconstruction from a Single Image. In *International Conference on 3D Vision (3DV)*, 2025.

Kaiyuan Tan, Yingying Shen, Ziyue Zhu, Mingfei Tu, Haohui Zhu, Haiyang Sun, Bing Wang, Guang Chen, and Hangjun Ye. UFO: Unifying Feed-Forward and Optimization-based Methods for Large Driving Scene Modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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*, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2026a.

Jingkang Wang, Henry Che, Yun Chen, Ze Yang, Lily Goli, Sivabalan Manivasagam, and Raquel Urtasun. Flux4D: Flow-based Unsupervised 4D Reconstruction. In *Advances in Neural Information Processing Systems*, 2025b.

Shuo Wang, Jilin Mei, Fuyang Liu, Wenfei Guan, Fanjie Kong, Zhihua Zhao, Shuai Wang, Chen Min, and Yu Hu. Ground4D: Spatially-Grounded Feedforward 4D Reconstruction for Unstructured Off-Road Scenes. *arXiv preprint arXiv:2605.04435*, 2026b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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*, 13:600–612, 2004.

Dongxu Wei, Zhiqi Li, and Peidong Liu. Omni-Scene: Omni-Gaussian Representation for Ego-Centric Sparse-View Scene Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. DepthSplat: Connecting Gaussian Splatting and Depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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*, 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*, 2024a.

Jiawei Yang, Jiahui Huang, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Apoorva Sharma, Maximilian Igl, Peter Karkus, Danfei Xu, et al. STORM: Spatio-Temporal Reconstruction Model for Large-Scale Outdoor Scenes. In *International Conference on Learning Representations*, 2025.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

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*, 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.

Bowei Zhang, Lei Ke, Adam Harley, and Katerina Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry. In *Advances in Neural Information Processing Systems*, 2026a.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently Reconstructing Dynamic Scenes One D4RT at a Time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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*, Lecture Notes in Computer Science. Springer, Cham, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. FLARE: Feed-forward Geometry, Appearance and Camera Estimation from Uncalibrated Sparse Views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations*, 2021.

Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Shengyin Jiang, Long Chen, Zhi-Xin Yang, and Jiwen Lu. DVGT: Driving Visual Geometry Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Hanbing Li, Long Chen, Zhi-Xin Yang, and Jiwen Lu. DVGT-2: Vision-Geometry-Action Model for Autonomous Driving at Scale. *arXiv preprint arXiv:2604.00813*, 2026b.

APPENDIX

OVERVIEW

This appendix provides additional details to complement the main paper:

- Details on sky handling and sky modeling are provided in Section A.
- Implementation details of the two-stage framework, including training configurations and hyperparameters, are provided in Section B.
- Additional qualitative results on nuScenes and MM-AU are provided in Section C.

A SKY MODELING

The sky has no finite depth and no rigid 3D structure, so folding it into the static or dynamic Gaussian field would force those networks to explain infinite-depth content with a finite-depth representation. We instead exclude the sky entirely from the scene branch, masking the predicted opacity of scene Gaussians to zero at every sky pixel, and model it with a separate sky Gaussian model, following the sky representation of DGGT (Chen et al., 2026), shared across all frames in the input window.

Sky Gaussians lie on a sphere of fixed radius R centered at the world origin, with $N{=}90{,}000$ points $\{\mathbf{d}_i\}$ drawn from a Fibonacci sphere sampling scheme for an approximately uniform angular distribution, giving means $\boldsymbol{\mu}_i = R\,\mathbf{d}_i$. Each point's scale is fixed (not learned) from the average distance to its 3 nearest neighbors on the sphere, so neighboring sky Gaussians overlap smoothly; opacities and orientations are likewise fixed, so only color is estimated from the input images. Each $\boldsymbol{\mu}_i$ is projected into every view of the window, and a sample is valid only if it lands in front of the camera, inside the image bounds, and on a pixel labeled sky by that view's sky mask; the raw color $\bar{\mathbf{c}}_i$ is the average of valid samples across views, and points with no valid sample in any view are discarded. This raw projected color is then refined by a lightweight learned correction MLP g_θ (two linear layers with a ReLU and a final $\tanh$),

$$\mathbf{c}_i = \tfrac{1}{2}\tanh\big(g_\theta(\bar{\mathbf{c}}_i)\big)_{[:3]} + \tfrac{1}{2} \in [0,1]^3,$$

whose weights are the only trainable parameters of the sky model. Because the sphere is shared across the window, the same sky Gaussians (means, scales, colors) are reused when rendering every frame, differing only in the camera pose used for rasterization.

At render time, the sky Gaussians are rasterized with the camera pose *detached*, so sky gradients cannot affect camera pose estimation, and composited with the scene render $\mathbf{I}_{\text{scene}}$ (accumulated alpha α_{scene}) via a hard gate given by the sky mask m:

$$\mathbf{I} = m\,\mathbf{I}_{\text{sky}} + (1-m)\,\mathbf{I}_{\text{scene}}$$

This hard, per-pixel gate also decouples optimization: the render loss propagates into the sky correction MLP only through sky-masked pixels and into the static Gaussians only through non-sky pixels, so the two branches never influence the same pixel's loss. The sky model is trained only during Stage 1 of the static reconstruction, supervised solely by its render loss, with sky and scene parameters updated fully independently throughout training.

B IMPLEMENTATION DETAILS

Each frame is encoded by a frozen DINOv3 (Siméoni et al., 2025) image encoder. Training uses the same **single-camera** input configuration as evaluation: 8 input frames sampled at a stride of 5, with the intermediate frames held out and used as interpolated novel views for supervision. At inference, each dynamic instance only needs a single 2D box in one frame, and SAM2 is prompted with this box to obtain that instance's dynamic mask across all frames; during training, we instead project each instance's annotated 3D bounding box in the training set into a 2D box to speed up data processing. The Motion-Static Separator's alternating-attention backbone consists of 3 layers, while the backbone used in the static and dynamic reconstruction stages consists of 21 layers; both are initialized from frozen VGGT-Ω (Wang et al., 2026a) weights and fine-tuned with LoRA adapters.

The Gaussian head shares its architecture with the depth head, and the motion decoder consists of 6 cross-attention layers. Training follows a three-stage scheme: Stage 1 trains the static Gaussian reconstruction network while caching intermediate features in a Tokens Bank; Stage 2 trains the dynamic Gaussian reconstruction network conditioned on these static priors once static reconstruction reaches sufficient quality; and Stage 3 trains the motion decoder on the resulting static and dynamic global features to estimate each instance's overall motion. All stages use the AdamW optimizer with a base learning rate of 1×10^{-4} and weight decay of 1×10^{-4}, and are trained on 8 H20 GPUs with a batch size of 8. The backbone itself remains frozen throughout and is only adapted through its LoRA parameters. All three stages are trained for 50 epochs each. In Stage 1, the static loss weights (Eq. 5) are set to λ_1=1.0, λ_2=0.8, λ_3=2.0, with the rendering term ℓ_{rgb}^{static} itself combining an ℓ_1 term, an SSIM term, and an LPIPS term with weights 1, 0.5, and 0.3, respectively. In Stage 2, the dynamic loss weights (Eq. 6) are set to λ_1=1.0, λ_2=0.8, with $\ell_{rgb}^{dynamic}$ using the same ℓ_1/SSIM/LPIPS weighting as Stage 1.

C Additional Qualitative Results

More Qualitative Results. Figure 4 shows qualitative results on nuScenes. With the overall motion predicted by our motion decoder, we integrate an instance's Gaussians across different frames into a unified space before computing motion interpolation, avoiding the large numbers of independently moving points produced by StreetForward's per-point motion estimation; this is particularly evident in the occlusion regions shown in the figure, where the lack of a unified motion causes the interpolated intermediate frame to exhibit large missing regions. Figure 5 shows a qualitative visualization on MM-AU accident scenes.



Figure 4: Qualitative comparison of motion interpolation on nuScenes. Left to right: preceding, interpolated, and following frame.



Figure 5: Qualitative comparison on MM-AU accident scenes. Left to right: GT, StreetForward, and Ours.