TRAJFORWARD: FEED-FORWARD DYNAMIC 3D GAUSSIAN RECONSTRUCTION FOR URBAN DRIVING SCENES WITH EXPLICIT MOTION TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstructing dynamic driving scenes with feed-forward 3D Gaussian Splatting remains fundamentally challenged by temporal inaccuracies and under-constrained entity-level correspondence, leading to limited dynamic expressiveness, unstable aggregation, and novel-view artifacts. These issues stem from the local or discrete low-order approximation and the lack of explicit entity-level constraint under self-supervision. To address this, we propose to derive target-time Gaussian centers in an *entity-aware* manner that not only captures coherent nonlinear motions but also preserves persistent temporal correspondence and intra-entity local variations. We materialize this concept through **TrajForward**, a feed-forward dynamic 3DGS framework that instantiates Gaussian centers along *entity-conditioned trajectories*. Specifically, to resolve **entity-association ambiguity**, we introduce learnable *motion-bearing slots* as latent proxies. These slots bind dynamically related Gaussians, providing explicit structural coupling and stable cross-time correspondence without object annotations. Building upon these proxies, we tackle **heterogeneous nonlinear dynamics** through a *hierarchical trajectory representation*. By decomposing motion into slot-level macroscopic trajectories and Gaussian-specific local residuals, this elegantly preserves both entity coherence and local flexibility. Finally, to prevent **self-supervised trajectory instability**, we formulate a suite of *trajectory-aware objectives*. By explicitly regularizing slot assignments and trajectory smoothness, these objectives mitigate static-dynamic entanglement and promote temporally coherent, artifact-reduced reconstructions. Experiments on challenging urban driving benchmarks validate *TrajForward* and demonstrate improved rendering fidelity. Code will be released.

1 INTRODUCTION

Dynamic scene reconstruction is a fundamental component of closed-loop autonomous driving simulation (Yang et al., 2023), requiring accurate modeling of traffic participant motions for realistic novel-view synthesis. Recently, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has emerged as a promising solution due to its real-time rendering capability. Existing methods for urban driving scenes mainly rely on per-scene optimization, using object-level bounding boxes or self-supervised temporal deformation fields (Chen et al., 2026b; Zhou et al., 2024; Yan et al., 2024; Chen et al., 2025; Huang et al., 2024b). However, such scene-specific approaches are computationally expensive and lack cross-scene generalization, limiting their scalability to large-scale driving simulations.

To overcome this scalability bottleneck, recent studies have shifted towards feed-forward dynamic 3DGS, enabling scene reconstruction in a single forward pass (Lu et al., 2024a; Tian et al., 2025; Yang et al., 2025; Chen et al., 2026a; Yu et al., 2026b). Nevertheless, accurate dynamic modeling still remains a central challenge, as traffic participants (*e.g.*, vehicles and pedestrians) simultaneously undergo *coherent yet nonlinear entity-level motions* and *complex local variations*. To understand the shortcomings of current approaches in capturing such dynamics, it is crucial to revisit their core mechanism: *how are Gaussian centers derived at target timestamps?* Existing methods generally fall into two paradigms, where *increment-based propagation* derives target centers by adding locally predicted velocities or displacements to source centers, and *state-based interpolation* estimates

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.

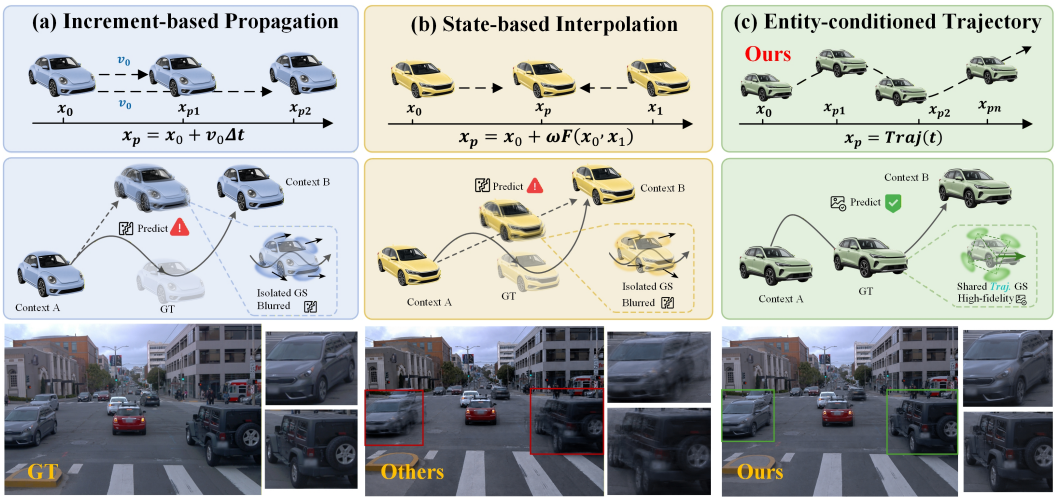

Figure 1: **Motivation.** Existing methods use either (a) increment-based propagation or (b) state-based interpolation to obtain target-time Gaussian. Their low-order and isolated Gaussian-level modeling often causes inaccurate trajectories and blurred dynamic objects, while (c) **TrajForward** uses shared continuous entity-conditioned trajectories to improve temporal correspondence and rendering fidelity.

Gaussian centers at discrete keyframes and interpolates between them. Despite their different mechanisms, both paradigms suffer from two critical limitations. **Temporally**, they attempt to model coherent nonlinear motions using merely local or discrete low-order operations, which inevitably leads to temporal inaccuracies. **Structurally**, driven purely by self-supervision, their motion modeling operates at the isolated Gaussian level without explicit entity-level constraints, making it difficult to maintain persistent cross-time correspondence for the entire object.

In practice, these limitations manifest as two primary failure modes. On one hand, the models exhibit limited dynamic expressiveness due to locally over-simplified motion reconstruction, which fails to faithfully capture nonlinear patterns like acceleration, braking, and turning (Fig. 1(a)(b)). On the other hand, they suffer from under-constrained entity-level correspondence. Consequently, the same physical region is often inconsistently represented across timestamps, leading to unstable aggregation and novel-view artifacts under complex dynamics (Fig. 1(a)(b)).

To overcome these bottlenecks, our goal is to derive target-time Gaussian centers in an *entity-aware* manner that not only captures coherent nonlinear motions but also preserves persistent temporal correspondence and intra-entity local variations. We materialize this concept through **TrajForward**, a feed-forward dynamic 3DGS framework that instantiates Gaussian centers along *entity-conditioned trajectories*. Rather than assembling isolated modules, *TrajForward* is designed as a cohesive system where three interconnected mechanisms naturally emerge from this trajectory-based formulation to resolve the core challenges of dynamic modeling(Fig. 1(c)). Specifically, to resolve **entity-association ambiguity**, we introduce learnable *motion-bearing slots* as latent proxies. These slots bind dynamically related Gaussians, providing explicit structural coupling and stable cross-time correspondence without relying on object annotations. Building upon these proxies, we tackle **heterogeneous nonlinear dynamics** through a *hierarchical trajectory representation*. By decomposing motion into slot-level macroscopic trajectories and Gaussian-specific local residuals, this design elegantly preserves both entity coherence and local flexibility. Finally, to prevent **self-supervised trajectory instability**, we formulate a suite of *trajectory-aware objectives*. By explicitly regularizing slot assignments and trajectory smoothness, these objectives mitigate static-dynamic entanglement and ensure temporally coherent, artifact-reduced reconstructions.

Experiments on Waymo (Sun et al., 2020), nuScenes (Caesar et al., 2020) and Argoverse2 (Wilson et al., 2023) demonstrate the strong performance of *TrajForward*, which consistently outperforms DGGT (Chen et al., 2026a) on dynamic rendering metrics, with a 1.72 dB PSNR gain in NVS.

Our main contributions are as follows: **(1)** We propose **TrajForward**, a feed-forward dynamic 3DGS framework for urban driving scenes that models dynamic Gaussians with explicit continuous

trajectories, improving nonlinear motion modeling and temporal consistency. (2) Underpinning this framework is a cohesive trajectory-based formulation that inherently binds Gaussians through latent slots, decomposes their heterogeneous motions via a coarse-to-fine hierarchy, and enables self-supervised learning with trajectory-aware objectives. (3) Extensive experiments on three challenging driving datasets demonstrate the effectiveness of *TrajForward* for dynamic scene reconstruction.

2 Related Work

Dynamic Reconstruction via Per-scene Optimization. Neural scene representation has recently evolved from implicit continuous volumes (*e.g.*, NeRF (Mildenhall et al., 2020; Pumarola et al., 2021; Park et al., 2021; Tretschk et al., 2021; Yu et al., 2021; Wang et al., 2021; Johari et al., 2022; Suhail et al., 2022)) to explicit point-based primitives. As a representative, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) achieves real-time rendering with anisotropic Gaussians and differentiable rasterization. Beyond static reconstruction methods (Cheng et al., 2024; Huang et al., 2024a; Yu et al., 2024; Lu et al., 2024b), dynamic reconstruction seeks to recover time-varying geometry and appearance from image sequences. Based on their motion modeling strategies, existing methods can be broadly divided into two categories. The first uses temporally conditioned representations (Fridovich-Keil et al., 2023; Cao & Johnson, 2023; Li et al., 2023b; Liu et al., 2023), where time is explicitly incorporated into the scene representation. For instance, PVG (Chen et al., 2026b) introduces periodic temporal signals for dynamic urban scenes, while other methods model temporal deformation or persistent dynamic primitives (Luiten et al., 2024; Wu et al., 2024; Yang et al., 2024; Gao et al., 2024; Lei et al., 2025; Kratimenos et al., 2024; Wang et al., 2025b). The second adopts compositional representations, explicitly modeling dynamic entities as separate components (Ost et al., 2021; Fischer et al., 2024). DrivingGaussian (Zhou et al., 2024), Street Gaussians (Yan et al., 2024), and OmniRe (Chen et al., 2025) follow this paradigm to improve object-level controllability and editability in urban scenes. Despite the strong reconstruction performance, their reliance on per-scene optimization and auxiliary priors limits scalability and the use of transferable geometry and motion knowledge from large-scale data.

Feed-forward Reconstruction. Feed-forward methods learn generalizable scene priors for instant sparse-view reconstruction by predicting 3D scene representations in a single forward pass. Early methods mainly target static sparse-view reconstruction, either directly regressing Gaussian parameters from image features or enhancing geometric consistency with cost volumes and monocular depth priors, such as pixelSplat (Charatan et al., 2024), Splatter Image (Szymanowicz et al., 2024), MVSplat (Chen et al., 2024), NoPoSplat (Ye et al., 2025), and DepthSplat (Xu et al., 2025). While effective, driving scenes introduce sparse surround-view overlap and dynamic scene evolution, making multi-view correspondence and temporal aggregation more challenging. DrivingForward (Tian et al., 2025) and DrivingRecon (Lu et al., 2024a) extend feed-forward Gaussian prediction to surround-view driving scenarios, enabling scalable 4D scene modeling. More recent methods further extend this line of work. STORM (Yang et al., 2025) performs self-supervised reconstruction from sparse multi-timestep posed images using Gaussian velocities decoded from shared motion bases. DGGT (Chen et al., 2026a) predicts dynamic-region motion and reconstructs intermediate states through motion-guided interpolation, while UniSplat (Shi et al., 2026) enhances spatio-temporal fusion and scene completion with a 3D latent scaffold. Despite these advances, existing feed-forward formulations still mainly rely on instantaneous velocity fields or sparse inter-frame interpolation, which impose low-order motion assumptions and limit temporal coherence under complex nonlinear dynamics. Moreover, they typically lack an explicit entity-aligned representation for maintaining coherent motion correspondence across observations. To address this, we jointly learn slot-structured motion decomposition and continuous Gaussian trajectories.

3 Method

We introduce **TrajForward**, a feed-forward framework that reconstructs urban driving scenes as a continuous 4D Gaussian representation. We first define the dynamic reconstruction problem and motivate the need for motion-bearing slots in Sec. 3.1. Based on this analysis, we present the overall pipeline in Sec. 3.2, followed by motion-bearing slot association in Sec. 3.3 and hierarchical trajectory representation in Sec. 3.4. Finally, we describe the trajectory-aware objective in Sec. 3.5.

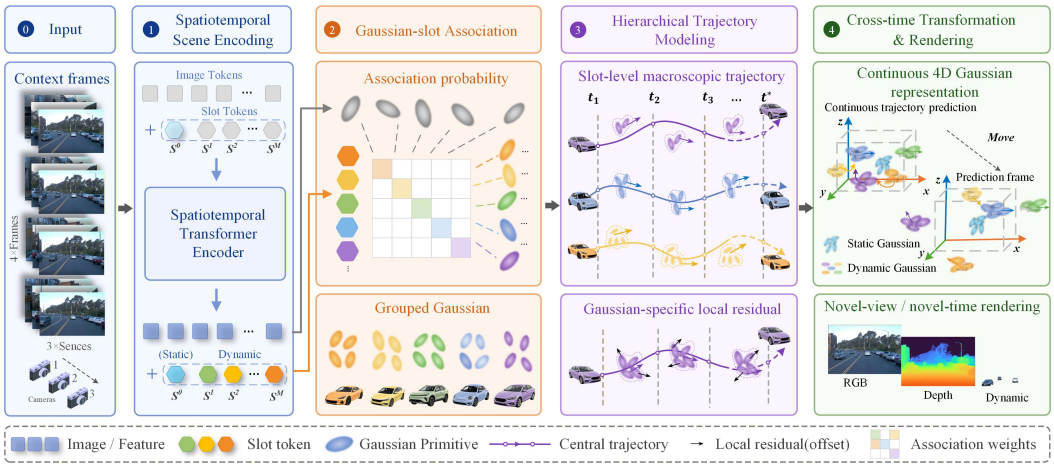

Figure 2: **Overview of TrajForward.** *TrajForward* encodes spatiotemporal scene features with learnable slot tokens, associates Gaussians with slots, and predicts hierarchical trajectories consisting of slot level macroscopic trajectories and Gaussian specific local residuals. These trajectories transform Gaussian anchors across time into a continuous 4D Gaussian representation.

3.1 Problem Formulation and Motivation

Problem Definition. Given sparse multi-view observations from a driving clip with M timestamps over $[\tau_s, \tau_e]$ and V camera views, we define the context set as

$$\mathcal{C} = \left\{ \left(\mathbf{I}^v_{\tau_m}, \mathbf{K}^v, \mathbf{E}^v_{\tau_m}, \tau_m \right) \mid m = 1, \ldots, M,\ v = 1, \ldots, V \right\}, \tag{1}$$

where $\mathbf{I}^v_{\tau_m}$ is the image captured by camera v at timestamp τ_m, with intrinsics $\mathbf{K}^v$ and extrinsics $\mathbf{E}^v_{\tau_m}$. Our goal is to learn a feed-forward model F_θ that maps $\mathcal{C}$ to a continuous 4D Gaussian representation. At normalized time $t = (\tau - \tau_s)/(\tau_e - \tau_s) \in [0, 1]$, the scene is represented as

$$\mathcal{G}_\theta(t) = \{g_i(t)\}_{i=1}^N, \quad g_i(t) = (\boldsymbol{\mu}_i(t), \mathbf{a}_i), \tag{2}$$

where N represents the number of Gaussians, $\boldsymbol{\mu}_i(t)$ is the time-dependent Gaussian center, and $\mathbf{a}_i = (\mathbf{R}_i, \mathbf{s}_i, \alpha_i, \mathbf{c}_i)$ denotes its rotation, scale, opacity, and appearance attributes. The key challenge is to estimate $\boldsymbol{\mu}_i(t)$ *with coherent nonlinear entity-level motion while preserving fine-grained local variations.* Given a target timestamp τ_q and camera view v_q, the corresponding image is rendered as

$$\hat{\mathbf{I}}^{v_q}_{\tau_q} = \mathcal{R}\left(\mathcal{G}_\theta(t_q), \mathbf{K}^{v_q}, \mathbf{E}^{v_q}_{\tau_q} \right), \quad t_q = \frac{\tau_q - \tau_s}{\tau_e - \tau_s}, \tag{3}$$

where $\mathcal{R}(\cdot)$ denotes the differentiable Gaussian splatting renderer. The model is trained by minimizing the photometric reconstruction loss over target timestamp-view pairs:

$$\min_\theta \sum_{(\tau_q, v_q) \in \mathcal{Q}} \mathcal{L}_{photo}\left(\hat{\mathbf{I}}^{v_q}_{\tau_q}, \mathbf{I}^{v_q}_{\tau_q} \right), \tag{4}$$

where $\mathcal{Q}$ denotes the set of supervised target timestamp-view pairs.

Limitation in Dynamic Driving Scenes. However, existing feed-forward dynamic 3DGS methods commonly model Gaussian centers through increment-based propagation or state-based interpolation. The former propagates a Gaussian from a context time t_m to a target time t_q:

$$\boldsymbol{\mu}_i(t_q) = \boldsymbol{\mu}_i(t_m) + \mathbf{m}_i(t_m, t_q), \tag{5}$$

where $\mathbf{m}_i(t_m, t_q)$ can be induced by velocity, $\mathbf{v}_i(t_m)(t_q - t_m)$, or directly predicted as a displacement. The latter estimates motion between two observed times t^1_m and t^2_m, and interpolates the target state:

$$\boldsymbol{\mu}_i(t_q) = \boldsymbol{\mu}_i(t^1_m) + \omega \mathbf{F}_i(t^1_m, t^2_m), \quad \omega = \frac{t_q - t^1_m}{t^2_m - t^1_m}. \tag{6}$$

where $\mathbf{F}_i(t_m, t_n)$ denotes the estimated displacement of Gaussian i, with $t_q \in [t_m^1, t_m^2]$.

However, such local or discrete low-order formulations are limited in capturing coherent nonlinear motion over time. Moreover, Gaussian-level motion modeling lacks explicit entity-level constraints, making it difficult to maintain consistent cross-time correspondence.

Introducing Entity-aware Modeling. We argue that Gaussian motion should not be modeled solely by local or discrete low-order operations, nor should individual Gaussians be treated as independent motion units. Instead, Gaussians with coherent entity-level or motion patterns should be explicitly grouped, with their complex dynamics modeled through continuous trajectories. Accordingly, we derive target-time Gaussian centers in an *entity-aware* manner to capture coherent nonlinear motion while preserving temporal correspondence and intra-entity variations.

3.2 Pipeline of Entity-aware Reconstruction Model

Inspired by feed-forward reconstruction models (Liang et al., 2025; Zhang et al., 2024), **TrajForward** adopts a ViT-style backbone (Dosovitskiy et al., 2020). Input images are tokenized and augmented with calibrated Plücker rays (Xvii, 1865) and normalized timestamps to form spatio-temporal tokens $\mathbf{X}$. A compact set of learnable motion-bearing slots $\mathbf{Z}$ is prepended to $\mathbf{X}$ and jointly updated through Transformer self-attention to organize shared motion structures. The resulting features are decoded into Gaussian anchor positions and time-independent attributes, along with Gaussian-to-slot association weights and clip-wise trajectory parameters. Together, these predictions instantiate the continuous 4D Gaussian representation $\mathcal{G}_\theta(t)$ at arbitrary target timestamps, which is finally rendered using differentiable Gaussian splatting (Kerbl et al., 2023). Fig. 2 summarizes the overall pipeline.

3.3 Latent Entity Motion Binding

Motion-bearing Slot Instantiation. To resolve **entity-association ambiguity**, we introduce $K+1$ learnable motion-bearing slots as compact latent priors:

$$\mathbf{Z} = \{\mathbf{z}_k\}_{k=0}^{K}, \quad \mathbf{z}_k \in \mathbb{R}^C, \tag{7}$$

where $k = 0$ is reserved for static regions and $k \geq 1$ for dynamic components. Given the image-geometry-time token sequence in Sec. 3.2, we prepend the slots to it and update them jointly:

$$[\mathbf{H}, \mathbf{Y}] = \mathrm{Transformer}([\mathbf{Z}, \mathbf{X}]), \quad \mathbf{H} = \{\mathbf{h}_k\}_{k=0}^{K}. \tag{8}$$

The updated slots through self-attention serve as latent motion proxies that group Gaussians with coherent static or dynamic behavior without instance and motion masks or correspondence labels.

Gaussian-slot Association. We map slot and pixel-aligned embeddings to a shared association space:

$$\mathbf{q}_k = \phi_q(\mathbf{h}_k), \quad \boldsymbol{\kappa}_i = \phi_k(\mathbf{e}_i), \tag{9}$$

where $\phi_q(\cdot)$ and $\phi_k(\cdot)$ are learnable projection MLPs, $\mathbf{e}_i$ is pixel embedding, $\mathbf{q}_k, \boldsymbol{\kappa}_i \in \mathbb{R}^{C_a}$ denote the query of slot k and the key associated with Gaussian i, and C_a is the embedding dimension.

The association probability between Gaussian i and slot k is computed by query-key similarity:

$$\pi_{ik} = \frac{\exp\left(\mathbf{q}_k^\top \boldsymbol{\kappa}_i / \eta\right)}{\sum_{\ell=0}^{K} \exp\left(\mathbf{q}_\ell^\top \boldsymbol{\kappa}_i / \eta\right)}, \quad \sum_{k=0}^{K} \pi_{ik} = 1, \tag{10}$$

where η is the temperature controlling the sharpness of the association distribution. The vector $\boldsymbol{\pi}_i = (\pi_{i0}, \pi_{i1}, \ldots, \pi_{iK})$ defines the slot membership of Gaussian i.

To avoid blending incompatible motions, we further adopt a straight-through hard association:

$$\tilde{\boldsymbol{\pi}}_i = \mathrm{onehot}\left(\arg\max_k \pi_{ik}\right) - \mathrm{sg}(\boldsymbol{\pi}_i) + \boldsymbol{\pi}_i, \tag{11}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. Thus, each Gaussian follows a single slot trajectory in the forward pass, while gradients are propagated through the soft probabilities.

Since slot $k = 0$ is static, we define the dynamic mass of Gaussian i as

$$\rho_i = \sum_{k=1}^{K} \pi_{ik} = 1 - \pi_{i0}. \tag{12}$$

This is used to suppress unnecessary residual motion in static-dominated regions.

3.4 HIERARCHICAL TRAJECTORY FACTORIZATION

To model **heterogeneous nonlinear dynamics**, we formulate Gaussian motion in a coarse-to-fine manner. The coarse level captures dominant entity-level motion, while the fine level models local Gaussian-level deviations. This ensures temporal coupling without sacrificing fine-grained flexibility.

For slot k, we denote its center trajectory parameters by $\boldsymbol{\xi}_k^{\mathrm{ctr}}$. For Gaussian i, we denote its residual trajectory parameters by $\boldsymbol{\xi}_i^{\mathrm{res}}$. Let $\mathcal{T}_{\mathrm{ctr}}$ and $\mathcal{T}_{\mathrm{res}}$ be differentiable trajectory mappings. The slot-level center path and Gaussian-level residual path are defined as

$$\mathbf{p}_k^{\mathrm{ctr}}(t) = \mathcal{T}_{\mathrm{ctr}}\left(\boldsymbol{\xi}_k^{\mathrm{ctr}}, t\right), \quad \mathbf{p}_i^{\mathrm{res}}(t) = \mathcal{T}_{\mathrm{res}}\left(\boldsymbol{\xi}_i^{\mathrm{res}}, t\right), \tag{13}$$

where the center trajectory parameters are predicted from slot tokens, while the residual trajectory parameters are predicted from pixel-aligned features conditioned on the associated slot:

$$\boldsymbol{\xi}_k^{\mathrm{ctr}} = \psi_{\mathrm{ctr}}(\mathbf{h}_k), \quad \mathbf{s}_i = \sum_{k=0}^{K} \tilde{\pi}_{ik}\mathbf{h}_k, \quad \boldsymbol{\xi}_i^{\mathrm{res}} = \psi_{\mathrm{res}}\left([\mathbf{e}_i, \mathbf{s}_i]\right). \tag{14}$$

Here, ψ_{ctr} and ψ_{res} are learnable trajectory prediction heads, and $\mathbf{s}_i$ denotes the associated slot feature of Gaussian i. In this work, we instantiate these generic trajectory mappings with different curves, which are discussed in the **Appendix**.D and evaluated in the ablation study.

Then, the slot-conditioned center displacement of Gaussian i and the Gaussian-level residual displacement from its context time t_m to a target time t_q are

$$\Delta\boldsymbol{\mu}_i^{\mathrm{ctr}}(t_q) = \sum_{k=0}^{K} \tilde{\pi}_{ik}\left[\mathbf{p}_k^{\mathrm{ctr}}(t_q) - \mathbf{p}_k^{\mathrm{ctr}}(t_m)\right], \quad \Delta\boldsymbol{\mu}_i^{\mathrm{res}}(t_q) \quad = \rho_i\left[\mathbf{p}_i^{\mathrm{res}}(t_q) - \mathbf{p}_i^{\mathrm{res}}(t_m)\right], \tag{15}$$

where ρ_i is the dynamic mass in Eq. 12. This gating suppresses residual motion in static-dominated regions and reduces background jitter. The target-time Gaussian center is then obtained by

$$\boldsymbol{\mu}_i(t_q) = \bar{\boldsymbol{\mu}}_i + \Delta\boldsymbol{\mu}_i^{\mathrm{ctr}}(t_q) + \Delta\boldsymbol{\mu}_i^{\mathrm{res}}(t_q), \tag{16}$$

where $\bar{\boldsymbol{\mu}}_i$ denotes the decoded Gaussian anchor position at context time t_m. Both motion terms are defined relative to t_m and therefore vanish at the observation time, yielding $\boldsymbol{\mu}_i(t_m) = \bar{\boldsymbol{\mu}}_i$.

3.5 TRAJECTORY-STRUCTURED SELF-SUPERVISION

To prevent **self-supervised trajectory instability**, we formulate a compact set of *trajectory-aware objectives*. The primary supervision comes from rendering target timestamp-view pairs after transforming Gaussian anchors according to Eq. 16:

$$\mathcal{L}_{\mathrm{ren}} = \sum_{(\tau_q, v)\in\mathcal{Q}} \ell_{\mathrm{img}}\left(\hat{\mathbf{I}}_{\tau_q}^{v}, \mathbf{I}_{\tau_q}^{v}\right), \tag{17}$$

where ℓ_{img} combines photometric and perceptual reconstruction terms.

To regularize nonlinear trajectories, we penalize excessive acceleration for both centers and residuals:

$$\mathcal{L}_{\mathrm{traj}} = \sum_{k=1}^{K}\int_0^1 \left\| \frac{\partial^2}{\partial t^2}\mathbf{p}_k^{\mathrm{ctr}}(t)\right\|_2^2 dt + \frac{1}{N}\sum_{i=1}^{N}\rho_i\int_0^1 \left\| \frac{\partial^2}{\partial t^2}\mathbf{p}_i^{\mathrm{res}}(t)\right\|_2^2 dt, \tag{18}$$

where $\rho_i = \sum_{k=1}^{K}\pi_{ik}$ denotes the dynamic confidence of Gaussian i.

To encourage consistent slot assignments among neighboring Gaussians, we impose

$$\mathcal{L}_{\mathrm{assoc}} = \sum_{(i,j)\in\mathcal{N}} \omega_{ij}\left\|\boldsymbol{\pi}_i - \boldsymbol{\pi}_j\right\|_1, \tag{19}$$

where $\mathcal{N}$ denotes neighboring Gaussian pairs and ω_{ij} is an appearance-geometry affinity.

We regularize the magnitude of curve control points to avoid degenerate large-motion solutions:

$$\mathcal{L}_{\mathrm{ctrl}} = \sum_{k=1}^{K}\left\|\mathbf{C}_k^{\mathrm{ctr}}\right\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\rho_i\left\|\mathbf{C}_i^{\mathrm{res}}\right\|_F^2. \tag{20}$$

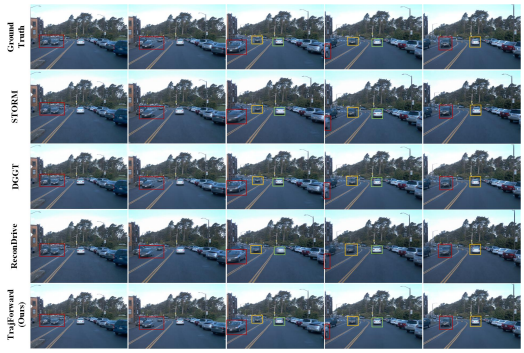

Figure 3: **Qualitative comparisons on Waymo.** RGB novel-view synthesis results.

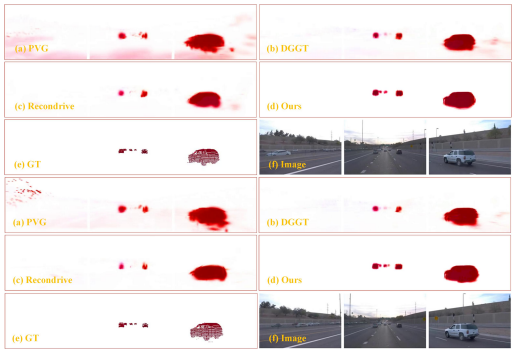

Figure 4: **Qualitative comparison on Waymo.** Rendered motion flow results.

To reduce ambiguous slot assignments, we minimize the normalized assignment entropy:

$$\mathcal{L}_{\text{ent}} = \frac{1}{N}\sum_{i=1}^{N}\frac{-\sum_{k=0}^{K}\pi_{ik}\log\pi_{ik}}{\log(K+1)}. \tag{21}$$

Since slot $k=0$ is reserved for static regions, we suppress its motion by

$$\mathcal{L}_{\text{static}} = \int_0^1 \left\|\mathbf{p}_0^{\text{ctr}}(t) - \mathbf{p}_0^{\text{ctr}}(0)\right\|_2^2 dt + \frac{1}{N}\sum_{i=1}^{N}\pi_{i0}\int_0^1 \left\|\mathbf{p}_i^{\text{res}}(t) - \mathbf{p}_i^{\text{res}}(t_i^{\text{obs}})\right\|_2^2 dt. \tag{22}$$

The final objective is

$$\mathcal{L} = \mathcal{L}_{\text{ren}} + \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} + \lambda_{\text{assoc}}\mathcal{L}_{\text{assoc}} + \lambda_{\text{ctrl}}\mathcal{L}_{\text{ctrl}} + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}} + \lambda_{\text{static}}\mathcal{L}_{\text{static}}. \tag{23}$$

Under this objective, motion-bearing slots, Gaussian-slot associations, and hierarchical trajectories are jointly learned from multi-view and multi-time rendering supervision.

4 Experiment

4.1 Experimental Setup

Datasets. We evaluate **TrajForward** on the Waymo Open Dataset (Sun et al., 2020), nuScenes (Caesar et al., 2020), and Argoverse2 (Wilson et al., 2023). To stress-test reconstruction under complex temporal dynamics, we construct the *Waymo-NUM (Non-Uniform Motion)* subset, which contains Waymo clips with surrounding vehicles exhibiting non-uniform motion patterns. Details of the datasets and the *Waymo-NUM* construction are provided in the **Appendix**.G.

Appearance Rendering. Following STORM (Yang et al., 2025) and DGGT (Chen et al., 2026a), each sequence is split into ten non-overlapping 2.0-second clips, each containing 20 frames from three camera views. We use the 1st, 5th, 10th, and 15th frames as context and render the remaining frames. We report PSNR and SSIM on full images and dynamic regions, dynamic PSNR for the next four target frames, and feed-forward inference time.

Flow Estimation. Following STORM (Yang et al., 2025) and Flux4d (Wang et al., 2025a), we evaluate motion estimation on Waymo without ground-truth supervision. Motion between consecutive timestamps is derived from the predicted dynamic Gaussian trajectories and compared with ground truth. We report rendered flow metrics (velocity RMSE, flow End-Point Error (EPE), and angular error) and 3D scene flow metrics (EPE3D, Acc5, Acc10, and angular error).

Entity-Level Slot Correspondence. We directly evaluate whether motion-bearing slots align with physical entities by associating dynamic Gaussians with reference objects using ground-truth 3D boxes and track IDs, with SAM masks (Kirillov et al., 2023; Ren et al., 2024) as auxiliary boundary cues. We report Slot Purity, Slot Switching Rate, Fragmentation Rate, and Object-Level Consistency. Slot indices remain fixed within each clip without frame-wise rematching, and annotations are used only for evaluation. Metric definitions are provided in the **Appendix**.F.

Table 1: **Comparison with state-of-the-art methods on the Waymo Open Dataset.** We evaluate novel view synthesis quality on full sequences, future-frame prediction quality, and inference speed. Dynamic-only metrics focus on moving regions. ***1st***, *2nd* and *3rd* are highlighted with different colors.

Method	Venue	Feed-forward?	Full sequence				Future frame				Inference speed
			Dynamic-only		Full image		Dynamic-only PSNR↑				
			PSNR↑	SSIM↑	PSNR↑	SSIM↑	Next 1st	Next 2nd	Next 3rd	Next 4th	
Methods with object annotations											
DGGT (Chen et al., 2026a)	CVPR'26	✔	21.39	0.613	**27.41**	**0.846**	20.98	20.92	20.83	20.44	0.39s
ReconDrive (Yu et al., 2026a)	arXiv'26	✔	22.23	0.628	26.75	0.806	21.80	21.49	21.50	21.38	2.94s
Unsupervised											
DeSiRe-GS (Peng et al., 2025)	CVPR'25	✗	16.74	0.485	24.59	0.714	15.38	15.02	14.79	14.53	38min
PVG (Chen et al., 2026b)	IJCV'26	✗	15.51	0.128	22.38	0.661	14.32	14.24	14.25	13.73	27min
AnySplat (Jiang et al., 2025)	TOG'25	✔	15.99	0.418	23.32	0.687	14.01	13.77	13.23	12.90	-
DA3 (Lin et al., 2025)	arXiv'25	✔	15.96	0.429	23.27	0.686	13.95	13.64	13.19	12.81	-
DrivingForward (Tian et al., 2025)	AAAI'25	✔	21.97	0.622	26.32	0.774	19.78	19.54	19.33	18.85	0.84s
STORM (Yang et al., 2025)	ICLR'25	✔	22.10	0.624	26.38	0.794	21.79	21.54	21.52	21.13	0.18s
DepthSplat (Xu et al., 2025)	CVPR'25	✔	16.98	0.476	24.15	0.703	15.11	14.89	14.20	13.60	0.15s
TrajForward	–	✔	**23.11**	**0.641**	27.13	0.829	**23.10**	**23.08**	**22.98**	**22.56**	**0.12s**

Table 3: **Comparison of rendered flow.**

Method	Feed-forward?	Velocity RMSE↓	EPE↓	Flow Angular Error↓
Methods with object annotations				
DGGT (Chen et al., 2026a)	✔	5.72	4.86	32.7
ReconDrive (Yu et al., 2026a)	✔	3.50	2.98	19.3
Unsupervised				
DeSiRe-GS (Peng et al., 2025)	✗	4.49	3.82	25.1
PVG (Chen et al., 2026b)	✗	6.76	5.75	39.8
STORM (Yang et al., 2025)	✔	3.68	3.13	20.4
TrajForward	✔	**3.01**	**2.56**	**16.5**

Table 4: **Comparison of scene flow estimation.**

Method	EPE_{3D} (m)↓	Acc_5 (%)↑	Acc_{10} (%)↑	θ (rad)↓
NSFP (Li et al., 2021)	0.698	42.17	54.26	0.919
NSFP++ (Najibi et al., 2022)	0.711	53.10	63.02	0.989
FastNSF (Li et al., 2023a)	0.732	48.79	65.89	0.876
STORM (Yang et al., 2025)	0.276	81.12	85.61	0.658
TrajForward	**0.179**	**88.38**	**92.40**	**0.317**

4.2 Comparison with the State-of-the-art

Appearance Rendering. Table 1 compares *TrajForward* with state-of-the-art driving reconstruction methods on Waymo. *TrajForward* consistently improves dynamic-region rendering and future-frame prediction, indicating more accurate temporal modeling of moving objects. It also remains competitive on full-image metrics and achieves the fastest inference. Fig. 3 further shows sharper moving objects and cleaner boundaries. These results demonstrate that entity-conditioned trajectories improve dynamic rendering fidelity while retaining high inference efficiency. For generalization, we evaluate two settings: in-domain training and evaluation on nuScenes, and zero-shot transfer to Argoverse2 using the Waymo-trained model. As shown in Table 2, *TrajForward* maintains reconstruction advantages in both settings, suggesting that the learned trajectory representation captures transferable motion priors rather than dataset-specific overfitting.

Table 2: **Comparison to state-of-the-art methods on more datasets.** We evaluate in-domain performance on nuScenes and zero-shot transfer to Argoverse2 using the Waymo-trained model.

Method	nuScenes				Argoverse2			
	Dynamic		Full		Dynamic		Full	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
L4GM	21.13	0.604	23.56	0.750	18.94	0.460	21.02	0.601
STORM	22.05	0.611	24.54	0.784	19.03	0.505	21.83	0.632
NoPoSplat	20.03	0.595	23.01	0.725	18.45	0.485	22.00	**0.646**
DepthSplat	19.98	0.557	23.19	0.733	18.53	0.497	22.05	0.636
TrajForward	**23.19**	**0.639**	**25.77**	**0.812**	**20.08**	**0.538**	**22.35**	0.640

Flow Estimation. Beyond appearance rendering, we evaluate the flow induced by the predicted trajectories. Table 3 evaluates rendered flow on Waymo, measuring motion accuracy after projecting the predicted 3D dynamics into camera views. Table 4 further removes the rendering step and evaluates the predicted motion directly in 3D space. Together, these two evaluations indicate that the learned trajectories induce accurate motion fields both after camera view rendering and in 3D space, validating the geometric consistency of our trajectory-based formulation. The rendered flow visualizations in Fig. 4 qualitatively confirm these results.

Entity-Level Slot Correspondence. Furthermore, we directly evaluate slot-entity correspondence. As shown in Table 5, *TrajForward* achieves 92.8% Slot Purity and 94.2% Object-Level Consistency, with low Slot Switching and Fragmentation Rates of 5.6% and 3.4%. These results indicate that slots form relatively stable entity-aligned motion groups across observations. Fig. 5 further shows that the same reference objects retain consistent dominant slot assignments across context frames, with slot identities kept fixed over time throughout each clip.

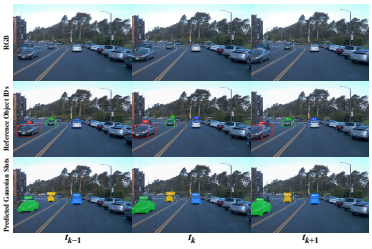

Figure 5: **Visualizations of entity-level slot correspondence.**

Table 5: **Entity-level correspondence of motion-bearing slots.**

Metric	Result
Slot Purity ↑	92.8%
Slot Switching Rate ↓	5.6%
Fragmentation Rate ↓	3.4%
Object-Level Consistency ↑	94.2%

Table 6: **Comparison on challenging *Waymo-NUM*.**

Method	PSNR↑	SSIM↑
DrivingForward	17.01	0.588
STORM	19.15	0.621
DGGT	17.74	0.607
ReconDrive	19.34	0.632
TrajForward	**21.62**	**0.676**

Table 7: **Comparison on long-horizon 4-second sequences.**

Method	PSNR↑	SSIM↑
DrivingForward	13.87	0.399
STORM	20.04	0.477
DGGT	18.78	0.473
ReconDrive	19.86	0.467
TrajForward	**21.74**	**0.492**

Table 8: **Comparison of different trajectory parameterizations.**

Method	PSNR↑	SSIM↑
Linear	21.03	0.598
Quadratic	22.01	0.617
Cubic	22.38	0.622
B-Spline	23.44	0.631
Bézier	**23.78**	**0.649**

Table 9: **Ablation study on the proposed objective components.**

Removed loss	Dynamic		Full	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
$\mathcal{L}_{\text{traj}}$	22.66	0.596	27.08	0.809
$\mathcal{L}_{\text{assoc}}$	22.91	0.611	27.15	0.812
$\mathcal{L}_{\text{ctrl}}$	22.42	0.583	26.98	0.805
$\mathcal{L}_{\text{ent}}$	23.12	0.620	27.21	0.814
$\mathcal{L}_{\text{static}}$	22.98	0.607	27.05	0.808
Full *obj.*	**23.74**	**0.648**	**27.33**	**0.817**

Table 10: **Ablation study on the number of entity-level slots.**

Method	Dynamic		Full	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
$K = 0$	22.35	0.619	26.33	0.748
$K = 1$	23.28	0.631	27.10	0.807
$K = 2$	23.52	0.640	27.23	0.812
$K = 3$	23.66	0.645	27.29	0.815
$K = 4$	**23.74**	**0.648**	**27.33**	**0.817**
$K = 8$	23.34	0.634	27.17	0.810
$K = 16$	23.20	0.628	27.05	0.806

Challenging Temporal Dynamics. We evaluate *TrajForward* under non-constant-velocity motion and longer temporal horizons. Table 6 reports results on Waymo clips (*Waymo-NUM*) with acceleration, turning, and other non-uniform motions, while Table 7 evaluates 4s long-horizon reconstruction. *TrajForward* achieves the best performance in both settings, suggesting that stable slot-conditioned trajectories help model nonlinear dynamics and mitigate error accumulation over time.

4.3 Ablation Study

Trajectory Parameterization. Table 8 compares different trajectory parameterizations. Linear motion underperforms, indicating that the constant-velocity assumption is insufficient for driving scenes. Higher-order polynomial models improve performance, while Bézier and B-spline trajectories provide greater flexibility. On the full test set in Table 8, the Bézier parameterization achieves the best overall performance, whereas B-splines perform better for highly complex or high-frequency motions (see the **Appendix**.E for detailed results). These results suggest that compact and smooth nonlinear trajectories are effective for modeling diverse scene dynamics.

Objective Components. Table 9 ablates the proposed *trajectory-aware objectives* by removing each loss term. The consistent degradation across ablations shows that image-level supervision alone is insufficient for learning coherent trajectories in dynamic regions. Together, these constraints turn ambiguous supervision into a stable learning signal for trajectory learning.

Motion-bearing Slots. Table 10 analyzes the number of motion-bearing slots. Without slots ($K = 0$), the model lacks explicit entity-level motion coupling and suffers a clear drop, especially on full image metrics. Increasing K improves performance up to $K = 4$, indicating that a compact set of latent motion proxies is sufficient for dominant motion patterns. Larger values of K degrade the results, likely due to over-fragmented assignments and weakened motion coherence.

5 Conclusion

Targeting scalable and temporally coherent reconstruction, this paper presents **TrajForward**, a feed-forward dynamic 3DGS framework for urban driving scene reconstruction. By introducing entity-conditioned trajectories with motion-bearing slots, hierarchical motion modeling, and trajectory-aware optimization, *TrajForward* explicitly models coherent nonlinear motions while preserving persistent cross-time correspondence under self-supervision. As a result, it improves dynamic motion expressiveness and temporal consistency, achieving high-quality novel-view synthesis in complex driving scenes. We hope *TrajForward* will facilitate scalable closed-loop autonomous driving simulation and inspire future research on trajectory-centric dynamic scene representations.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for English language editing and polishing, literature search, LaTeX formatting, and minor debugging of general experimental scripts. We have not used generative AI tools for formulating research hypotheses, developing the theoretical framework, designing the core methodology, implementing the core algorithms, interpreting experimental results, or generating synthetic datasets, and other required disclosure tasks are not applicable to this work. Additionally, generative AI tools were used only in an auxiliary capacity and did not contribute to the substantive scientific content or core technical contributions of this work. We carefully reviewed and verified all AI-assisted content, including manually checking revised text, validating retrieved references against original sources, and testing any AI-assisted script modifications for correctness. We take full responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

TrajForward is intended for research on dynamic scene reconstruction and novel-view rendering in autonomous driving scenarios. Our experiments use established public driving benchmarks and do not involve direct interaction with human subjects. As with other learning-based methods, performance may be affected by biases and limitations in the training data. The method is not intended for direct deployment in safety-critical driving systems without further validation and safety assessment.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

To facilitate reproducibility, we provide the model formulation and training objectives in Sec. 3 and the experimental protocols in Sec. 4.1. Further details are provided in Appendix. A (datasets), Appendix B (baselines), Appendix D (trajectory parameterizations), Appendix F (slot correspondence metrics), and Appendix G (motion miner). Code will be released to reproduce the reported experiments.

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 (CVPR)*, pp. 11621–11631, 2020.

Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 130–141, 2023.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 (CVPR)*, pp. 1265–1276, 2026a.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision (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 (IJCV)*, 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.

Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first International Conference on Machine Learning (ICML)*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Tobias Fischer, Lorenzo Porzi, Samuel Rota Bulo, Marc Pollefeys, and Peter Kontschieder. Multi-level neural scene graphs for dynamic urban environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21125–21135, 2024.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–11, 2024a.

Nan Huang, Xiaobao Wei, Wenzhao Zheng, Pengju An, Ming Lu, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. s^3gaussian: Self-supervised street gaussians for autonomous driving. *arXiv preprint arXiv:2405.20323*, 2024b.

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

Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18365–18375, 2022.

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

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. In *European Conference on Computer Vision (ECCV)*, pp. 252–269. Springer, 2024.

Jiahui Lei, Yijia Weng, Adam W Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6165–6177, 2025.

Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:7838–7851, 2021.

Xueqian Li, Jianqiao Zheng, Francesco Ferroni, Jhony Kaesemodel Pontes, and Simon Lucey. Fast neural scene flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9878–9890, 2023a.

Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4273–4284, 2023b.

Hanxue Liang, Jiawei Ren, Ashkan Mirzaei, Antonio Torralba, Ziwei Liu, Igor Gilitschenski, Sanja Fidler, Cengiz Oztireli, Huan Ling, Zan Gojcic, et al. Feed-forward bullet-time reconstruction of dynamic scenes from monocular videos. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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. *arXiv preprint arXiv:2511.10647*, 2025.

Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13–23, 2023.

Hao Lu, Tianshuo Xu, Wenzhao Zheng, Yunpeng Zhang, Wei Zhan, Dalong Du, Masayoshi Tomizuka, Kurt Keutzer, and Yingcong Chen. Drivingrecon: Large 4d gaussian reconstruction model for autonomous driving. *arXiv preprint arXiv:2412.09043*, 2024a.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20654–20664. IEEE, 2024b.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *International Conference on 3D Vision (3DV)*, 2024.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xinchen Yan, Scott Ettinger, and Dragomir Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving. In *European Conference on Computer Vision (ECCV)*, pp. 424–443. Springer, 2022.

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

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)*, 40(6), 2021.

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

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

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 (ICLR)*, 2026.

Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision (ECCV)*, pp. 156–174. Springer, 2022.

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 (CVPR)*, pp. 2446–2454, 2020.

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

Qijian Tian, Xin Tan, Yuan Xie, and Lizhuang Ma. Drivingforward: Feed-forward 3d gaussian splatting for driving scene reconstruction from flexible surround-view input. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12959–12970, 2021.

Jingkang Wang, Henry Che, Yun Chen, Ze Yang, Lily Goli, Sivabalan Manivasagam, and Raquel Urtasun. Flux4d: Flow-based unsupervised 4d reconstruction. *arXiv preprint arXiv:2512.03210*, 2025a.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4690–4699, 2021.

Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

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

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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 (CVPR)*, 2025.

Julius Plucker. Xvii. on a new geometry of space. *Philosophical Transactions ofthe Royal Society of London*, 1865.

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

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

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 (CVPR)*, 2024.

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 (ICLR)*, 2025.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4578–4587, 2021.

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.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19447–19456. IEEE, 2024.

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.

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 (CVPR)*, 2024.

APPENDIX

A DATASET DETAILS

Waymo Open Dataset. We primarily evaluate **TrajForward** on the Waymo Open Dataset (Sun et al., 2020), which contains 1,000 real world driving sequences, with 798 sequences for training and 202 sequences for validation. Following recent dynamic driving scene reconstruction methods (Chen et al., 2026a; Yang et al., 2025), we use the frontal three cameras. This setting provides sparse multi-view observations while preserving sufficient spatial detail for evaluating dynamic object reconstruction.

For appearance rendering on Waymo, each validation sequence is divided into non-overlapping 2.0-second clips. Each clip contains 20 frames and three camera views per frame. We use the 1st, 5th, 10th, and 15th frames as context observations and render the remaining frames as target views. We report novel-view synthesis quality on both full images and dynamic regions, as well as future-frame prediction performance on the next 1st–4th target frames. The per clip inference time is also reported to evaluate feed-forward reconstruction efficiency.

Waymo-NUM (Non-Uniform Motion) subset. To specifically evaluate complex temporal dynamics, we construct **Waymo-NUM**, an additional Waymo subset that includes all clips identified by *Motion Miner* as containing surrounding vehicles that deviate from constant-velocity straight-line motion (see Fig. 6). These clips cover challenging motion patterns such as acceleration, deceleration, turning, and other non-uniform trajectories. The retrieval procedure and motion criteria of *Motion Miner* are detailed in Sec. G.

Cross-dataset evaluation. We further evaluate cross-dataset generalization on nuScenes (Caesar et al., 2020) and Argoverse 2 (Wilson et al., 2023). In this setting, models are trained and evaluated on nuScenes, and then directly evaluated on Argoverse 2 without additional fine-tuning. This protocol is designed to test whether the learned trajectory representation captures transferable motion priors rather than overfitting to a single dataset.

B BASELINE DETAILS

We compare **TrajForward** with representative baselines that are closely related to dynamic driving-scene reconstruction, feed-forward 3D Gaussian reconstruction, and motion estimation. The selected methods cover the main evaluation dimensions in our experiments, including novel-view synthesis, future-frame prediction, inference efficiency, cross-dataset generalization, rendered flow, and 3D scene flow.

For driving-scene reconstruction, we primarily compare with recent dynamic 3DGS methods that are designed for autonomous driving scenarios. DrivingForward (Tian et al., 2025) extends feed-forward Gaussian prediction to flexible surround-view driving inputs. STORM (Yang et al., 2025) performs spatio-temporal reconstruction from sparse multi-timestep posed images and predicts Gaussian motion using shared motion bases. DGGT (Chen et al., 2026a) reconstructs dynamic driving scenes in a feed-forward manner by estimating 3D motion for dynamic regions and interpolating intermediate states. ReconDrive (Yu et al., 2026a) provides a strong recent feed-forward 4D Gaussian reconstruction baseline for autonomous driving scenes. We also include PVG (Chen et al., 2026b) and DeSiRe-GS (Peng et al., 2025) as representative dynamic Gaussian reconstruction references, since they model time-varying urban scenes with strong scene-level reconstruction capacity.

In addition to driving-specific dynamic methods, we compare with several generalizable 3DGS baselines when their input assumptions are compatible with our evaluation protocol. AnySplat, DA3, DepthSplat (Xu et al., 2025), NoPoSplat (Ye et al., 2025), and L4GM are included to assess whether generic feed-forward Gaussian reconstruction models can handle sparse multi-view driving scenes with dynamic objects. These methods provide useful references for evaluating the importance of explicitly modeling temporal motion and dynamic-region correspondence.

For motion evaluation, we follow the Waymo flow evaluation protocol and compare the motion induced by our predicted trajectories with representative scene-flow methods, including NSFP (Li et al., 2021), NSFP++ (Najibi et al., 2022), FastNSF (Li et al., 2023a), and STORM (Yang et al., 2025). Ground-truth scene flow is used only for evaluation and is not used to supervise *TrajForward*.



Figure 6: **Examples of nonlinear motion clips extracted by *Motion Miner* from Waymo.** Top: turning. Bottom: acceleration and deceleration. These clips are used to evaluate reconstruction under challenging non-constant-velocity dynamics.

C Ablation of Association and Residual Offset.

Table 11 evaluates hard association and Gaussian-level residual offsets. Removing hard association allows a Gaussian to mix multiple slot motions, which reduces dynamic reconstruction quality. Removing the residual offset limits local flexibility and weakens intra-entity motion modeling. Combining both designs gives the best performance, confirming that *TrajForward* benefits from entity-level coherent trajectories and Gaussian-level local adaptation.

Table 11: **Ablation study on hard association and residual offset.** We evaluate different design choices on novel view synthesis metrics.

Hard assoc.	Residual offset	Dynamic-only		Full image	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑
✗	✔	23.28	0.626	27.05	0.807
✔	✗	23.02	0.621	26.98	0.804
✔	✔	**23.74**	**0.648**	**27.33**	**0.817**

D Trajectory Parameterizations

We provide additional details on the trajectory parameterizations used in our ablation study. In dynamic 3D Gaussian reconstruction, each dynamic Gaussian is associated with a time-varying 3D center. Therefore, the choice of trajectory parameterization directly determines the type of motion that the model can represent. Let $t \in [0, 1]$ denote the normalized time within a clip, and let $\mathbf{x}_i(t) \in \mathbb{R}^3$ be the 3D center of the i-th dynamic Gaussian at time t.

The simplest option is a linear trajectory:

$$\mathbf{x}_i(t) = \mathbf{a}_{i,0} + \mathbf{a}_{i,1} t. \tag{24}$$

This formulation corresponds to a constant-velocity assumption. It is compact and stable, but it is insufficient for modeling non-linear vehicle motion, such as braking, turning, lane changing, or interaction-induced acceleration.

We also consider higher-order polynomial trajectories:

$$\mathbf{x}_i(t) = \sum_{k=0}^{K} \mathbf{a}_{i,k} t^k, \quad K \in \{2, 3\}. \tag{25}$$

The quadratic form can represent constant acceleration, while the cubic form further increases the flexibility of temporal motion modeling. However, polynomial trajectories are expressed in the monomial basis and may become sensitive under sparse temporal observations, especially when extrapolating future states beyond the observed context frames.

The Bézier trajectory is:

$$\mathbf{x}_i(t) = \sum_{k=0}^{K} b_{k,K}(t) \mathbf{p}_{i,k}, \quad b_{k,K}(t) = \binom{K}{k} (1-t)^{K-k} t^k, \tag{26}$$

where $\{\mathbf{p}_{i,k}\}_{k=0}^{K}$ are learnable control points and $b_{k,K}(t)$ are Bernstein basis functions. Compared with standard polynomial trajectories, the Bézier formulation provides a more geometrically interpretable representation: the motion is controlled by a compact set of control points, while the



Figure 7: **Qualitative comparisions on the NuScenes dataset.** Best viewed with zoom-in.

resulting trajectory remains smooth over continuous time. This property is particularly suitable for autonomous driving scenes, where surrounding vehicles often exhibit non-constant-velocity motion.

We further compare with B-spline trajectories, which represent motion using piecewise polynomial basis functions:

$$\mathbf{x}_i(t) = \sum_{j=0}^{M} N_{j,p}(t)\mathbf{c}_{i,j}, \tag{27}$$

where $\{\mathbf{c}_{i,j}\}_{j=0}^{M}$ are control points and $N_{j,p}(t)$ denotes the B-spline basis function of degree p defined over a knot vector $\{\tau_j\}$. The basis functions are recursively defined as

$$N_{j,0}(t) = \begin{cases} 1, & \tau_j \le t < \tau_{j+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{28}$$

and

$$N_{j,p}(t) = \frac{t - \tau_j}{\tau_{j+p} - \tau_j} N_{j,p-1}(t) + \frac{\tau_{j+p+1} - t}{\tau_{j+p+1} - \tau_{j+1}} N_{j+1,p-1}(t). \tag{29}$$

B-splines provide strong local flexibility because each control point affects only a limited temporal interval, allowing them to model complex motion variations with piecewise polynomial segments. However, they require additional control points and knot specifications, increasing the complexity of trajectory prediction.

E Bézier vs. B-spline under Complex Motion

Although Bézier trajectories achieve the best overall performance on the full test set (Table 8 in the ablation study of the main paper), their globally supported parameterization, where each control point influences the trajectory over the entire temporal interval, can provide less local flexibility for highly articulated or high-frequency motion. To further examine this limitation, we compare Bézier and B-spline parameterizations on two challenging subsets, including *car-dense* clips and clips with a *high density of non-rigid actors* such as pedestrians and cyclists.

Specifically, we construct the *car-dense subset* by ranking test clips according to the maximum number of simultaneously visible and moving cars and selecting the top 20%. The *high-density non-rigid-actor subset* is constructed similarly, based on the maximum number of simultaneously visible and moving pedestrians and cyclists. For this ablation, we replace the Gaussian-specific Bézier residual trajectories with B-spline trajectories while keeping all other components unchanged.

Table 12: Comparison between Bézier and B-spline trajectory parameterizations on challenging motion subsets.

Evaluation Subset	Bézier	B-spline
Car-dense clips	**21.86**	21.54
High-density non-rigid-actor clips	20.35	**20.71**

As shown in Table 12, Bézier trajectories perform better on *car-dense* clips, suggesting that their compact and smooth parameterization is sufficient for typical multi-vehicle driving scenes. In contrast, B-splines achieve better performance on the *high-density non-rigid-actor subset*, where pedestrians and cyclists often exhibit more rapid and locally varying motion. This difference may be explained by the local support of B-splines, which allows individual control points to affect only local trajectory segments and thus provides greater flexibility for modeling high-frequency or locally varying motion.

These results help explain why Bézier trajectories perform better on the full benchmark, while B-splines can be advantageous for more highly articulated motion. We therefore adopt Bézier trajectories as the default parameterization for their compactness, smoothness, and overall effectiveness, while the B-spline results suggest a promising direction for extending the model toward more complex non-rigid dynamics.

F Associations and Metric Definitions

Reference object identities are obtained from ground-truth 3D boxes and track IDs when evaluating the learned slot assignments. SAM-generated masks provide auxiliary cues for visible object boundaries only and are not treated as ground-truth instance masks. Object identities are determined independently from the corresponding observations and are not propagated from the predicted Gaussian trajectories.

At a context timestamp t, n_{okt} denotes the number of visible, reference-associated Gaussian anchors that belong to object o and are assigned to dynamic slot k. The anchors are taken from the corresponding context observation. We define

$$q_{okt} = \frac{n_{okt}}{\sum_{\ell} n_{o\ell t}}, \qquad d_o(t) = \arg\max_k q_{okt},$$

with q_{okt} giving the normalized slot distribution for object o and $d_o(t)$ its dominant slot. We only form this distribution when valid Gaussian support is available. Slot identities remain fixed over each clip, and temporal metrics are computed only for objects with at least two valid observations.

Slot Purity. Slot purity quantifies how strongly each slot is associated with a single reference object over a clip:

$$\mathrm{SP} = \frac{\sum_k \max_o \sum_t n_{okt}}{\sum_{o,k,t} n_{okt}}.$$

Slot Switching Rate. For each object o, $\mathcal{P}_o$ contains pairs of consecutive valid observations in which the object is visible. The switching rate is

$$\mathrm{SSR} = \frac{\sum_o \sum_{(t,t')\in\mathcal{P}_o} \mathbf{1}[d_o(t) \neq d_o(t')]}{\sum_o |\mathcal{P}_o|}.$$

Fragmentation Rate. At a valid observation, an object is regarded as fragmented if more than one slot accounts for a non-negligible fraction of its Gaussian support:

$$\mathrm{FR}_\tau = \frac{1}{|\mathcal{V}|} \sum_{(o,t)\in\mathcal{V}} \mathbf{1}\left[\sum_k \mathbf{1}[q_{okt} > \tau] > 1\right],$$

where $\mathcal{V}$ contains all valid object–observation pairs and τ controls the minimum slot fraction counted as effective support.

Object-Level Consistency. For each eligible object o, let $\mathcal{T}_o$ be its set of valid observation timestamps. Object-level consistency is the fraction of these observations assigned to the object's most frequently occurring dominant slot:

$$\mathrm{OC} = \frac{1}{|\mathcal{O}|} \sum_{o \in \mathcal{O}} \frac{\max_k \sum_{t \in \mathcal{T}_o} \mathbf{1}[d_o(t) = k]}{|\mathcal{T}_o|}.$$

All metrics are computed independently within each clip, and slot indices are not assumed to correspond across different clips. The reported values are given as percentages.

G Motion Miner for Non-Uniform Vehicle Motion Retrieval

Objective. We use *Motion Miner* to retrieve Waymo segments in which surrounding vehicles deviate from rectilinear constant-velocity motion. Rather than inferring motion categories from images, Motion Miner operates directly on annotated 3D vehicle trajectories and frame-wise ego poses. Its purpose is to provide an interpretable offline procedure for locating acceleration, deceleration, turning, and other non-uniform vehicle motion patterns.

Global trajectory representation. Motion Miner first converts each surrounding vehicle trajectory from the ego coordinate frame to a global coordinate frame. This step is essential because the raw Waymo 3D box centers are defined relative to the ego vehicle at each frame. As a result, the same static or uniformly moving object may appear to have complex motion in ego coordinates whenever the ego vehicle accelerates, brakes, or turns. Directly applying temporal differencing to ego-frame box centers would therefore introduce ego-motion artifacts into the estimated target motion.

For each frame, we use the ego pose provided by Waymo to map the 3D center of every annotated vehicle box from the ego frame into the global frame. Conceptually, this transformation first rotates the box center according to the ego vehicle's global orientation and then translates it by the ego vehicle's global position. After this transformation, the trajectory of a surrounding vehicle is represented in a stationary world coordinate system, so changes in its position primarily reflect the vehicle's own motion rather than the motion of the data-collection platform.

Since our goal is to detect driving maneuvers on the road plane, Motion Miner keeps only the horizontal coordinates of the transformed center:

$$\mathbf{u}_{o,i} = (x^g_{o,i}, y^g_{o,i})^\top, \tag{30}$$

where $\mathbf{u}_{o,i}$ denotes the global planar location of vehicle o at frame i. The time-ordered sequence $\{(t_i, \mathbf{u}_{o,i})\}$ is then used as the trajectory for subsequent velocity, acceleration, turning, and residual computation. This global trajectory representation is the key step that prevents ego acceleration or yaw from being falsely attributed to surrounding vehicles.

Windowed kinematic cues. For each object trajectory, Motion Miner estimates velocity, acceleration, and speed by temporal differentiation in the global frame:

$$\mathbf{v}_i = \nabla_t \mathbf{u}_i, \qquad \mathbf{a}_i = \nabla_t \mathbf{v}_i, \qquad s_i = \|\mathbf{v}_i\|_2. \tag{31}$$

To distinguish changes in speed from changes in direction, acceleration is decomposed relative to the current velocity direction. The longitudinal component measures acceleration along the direction of travel,

$$a_i^{\mathrm{lon}} = \frac{\mathbf{v}_i^\top \mathbf{a}_i}{\|\mathbf{v}_i\|_2 + \epsilon}, \tag{32}$$

while the lateral component measures the perpendicular component associated with turning,

$$a_i^{\mathrm{lat}} = \frac{v_{x,i} a_{y,i} - v_{y,i} a_{x,i}}{\|\mathbf{v}_i\|_2 + \epsilon}. \tag{33}$$

The heading direction is obtained from the velocity vector, and its temporal derivative gives the heading angular velocity. Instead of classifying motion from single-frame derivatives, Motion Miner aggregates these cues within a centered window $W_i = \{i-h, \ldots, i+h\}$, which covers approximately one second in Waymo Perception sequences. The resulting window-level statistics include mean longitudinal acceleration, mean lateral-acceleration magnitude, speed change, heading change, and mean heading-rate magnitude.

Residual from constant-velocity motion. Acceleration and turning cues cover common maneuvers, but some non-uniform patterns are better characterized by their deviation from straight-line constant-velocity motion. For each window W_i, Motion Miner fits a local constant-velocity trajectory,

$$\hat{\mathbf{u}}_j = \mathbf{p} + \mathbf{v}(t_j - \bar{t}_i), \qquad \bar{t}_i = \frac{1}{|W_i|} \sum_{j \in W_i} t_j, \tag{34}$$

where $\mathbf{p}$ and $\mathbf{v}$ are obtained by least-squares fitting over the window. The residual is the root-mean-square fitting error:

$$r_i = \sqrt{\frac{1}{|W_i|} \sum_{j \in W_i} \|\mathbf{u}_j - \hat{\mathbf{u}}_j\|_2^2}. \tag{35}$$

A large r_i indicates that the local trajectory cannot be well explained by uniform rectilinear motion, and is therefore evidence of generic non-uniform motion.

Non-uniform motion detection. Motion Miner applies a small set of complementary tests on the window-level cues. A frame is first required to satisfy basic validity conditions, including sufficient speed, sufficient 3D annotation support, and a complete temporal window. It is then labeled according to the following criteria: acceleration is detected when longitudinal acceleration is consistently positive and the window-level speed increases; deceleration is detected when longitudinal acceleration is consistently negative and the window-level speed decreases; turning is detected when heading change is accompanied by lateral acceleration or angular velocity; and generic non-uniform motion is detected when the constant-velocity residual in Eq. equation 35 exceeds a threshold. When multiple criteria are active, the primary label is assigned in the order of turning, deceleration, acceleration, and residual-based non-uniform motion, so that more specific maneuvers take precedence over generic residual detections.

Temporal aggregation. The above detection is performed on individual object trajectories, since kinematic features require temporally consistent tracks. To reduce derivative noise, isolated positives are removed and small gaps between nearby positives are bridged. The surviving positives from all surrounding vehicles are then projected onto the sequence timeline and merged into frame segments. A final segment is retained whenever at least one surrounding vehicle exhibits non-uniform motion within that temporal range, and the segment label is inherited from the highest-confidence positive sample inside the segment.

Discussion. Motion Miner is a rule-based geometric miner rather than a learned perception model. Its main correctness property is the use of global trajectories, which removes ego-motion contamination before any kinematic feature is computed. Its recall is limited by the quality and temporal continuity of the underlying 3D annotations: far-range vehicles, severe occlusion, short tracks, and fragmented object identities may lead to missed or shortened segments. In our pipeline, Motion Miner is used only for offline retrieval and quality control; downstream models are trained and evaluated independently on the mined data.

H LIMITATION

Although **TrajForward** improves temporal consistency and nonlinear motion modeling through entity-conditioned trajectories, severe occlusion can substantially reduce the visible evidence available for associating dynamic entities across observations. In such cases, incomplete or ambiguous entity cues may occasionally lead to less reliable slot assignments, which can further affect the recovery of fine-grained local dynamic details. Improving robustness to severe occlusion remains an important direction for future research.