CAMOSPLAT: TEMPORAL APPEARANCE MODELING FOR 3D GAUSSIAN SPLATTING IN AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Driving scenes contain local appearance changes over time, from gradual shading variations to abrupt traffic-light transitions. Capturing these changes with 3D Gaussian Splatting (3DGS) calls for temporal appearance control at the primitive level, yet current methods either keep per-Gaussian appearance fixed or tie its change to discrete per-frame codes. We introduce CamoSplat, which models a temporal appearance pattern for each Gaussian and controls how strongly it is expressed over time. These patterns come from a Mixture-of-Temporal-Bases (MoTB) that combines shared temporal functions with per-Gaussian weights, avoiding the overhead of separate per-Gaussian networks. To control when and how strongly each pattern acts, Morlet wavelet temporal gating modulates its amplitude across multiple temporal scales. The same wavelet control extends to visibility, where sensor-specific opacity wavelets add a temporal offset to a shared base opacity. CamoSplat achieves state-of-the-art photometric and LiDAR reconstruction on our temporal-variation splits from Waymo, Argoverse 2, and nuScenes, and the best overall results on standard benchmark scenes. Beyond these metrics, it precisely captures temporally localized appearance changes that prior methods collapse into ghosting or blur. We will release our code publicly.

1 INTRODUCTION

Novel view synthesis (NVS) enables autonomous driving simulation and scene replay by rendering recorded road environments from novel viewpoints (Wu et al., 2023; Tonderski et al., 2024; Hess et al., 2025). Beyond viewpoint, the scene itself changes as time passes, and capturing this variation is important for downstream tasks such as data generation and closed-loop simulation (Yang et al., 2023; Zhou et al., 2026): the ego vehicle, for instance, must see a traffic light turn red in time to stop. However, these variations are challenging to reconstruct because they are spatially localized and span multiple timescales, while neighboring regions stay visually stable. They range from gradual drifts, as in a change in illumination or a road entering a building's shadow, to abrupt flips, as in a traffic light or brake light switching. Faithful reconstruction therefore demands appearance control that is local in space and resolved in time, per region rather than per frame.

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has become the standard for reconstructing driving scenes. It represents a scene with explicit primitives and renders it in real time. Recent methods have advanced geometry, motion, and sensor realism through scene decomposition (Zhou et al., 2024b; Yan et al., 2024; Zhou et al., 2024a; Chen et al., 2025; Peng et al., 2025), actor dynamics (Chen et al., 2026; Xu et al., 2025; Song et al., 2025), and joint camera and LiDAR rendering (Hess et al., 2025; Kung et al., 2025). Temporal appearance, however, has received far less attention. Most methods keep each primitive's appearance fixed over time and at most correct the rendered images (Wang et al., 2025). The few exceptions either tie each Gaussian's change to per-frame codes (Kulhanek et al., 2024; Wu et al., 2026) or expand it in sinusoids that span the entire sequence (Yan et al., 2024; Yang et al., 2024b), so none makes that change both continuous and local in time. Without such a mechanism, optimization absorbs temporal photometric variation into geometry and opacity, producing ghosting, blur, and degraded reconstruction fidelity.

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.

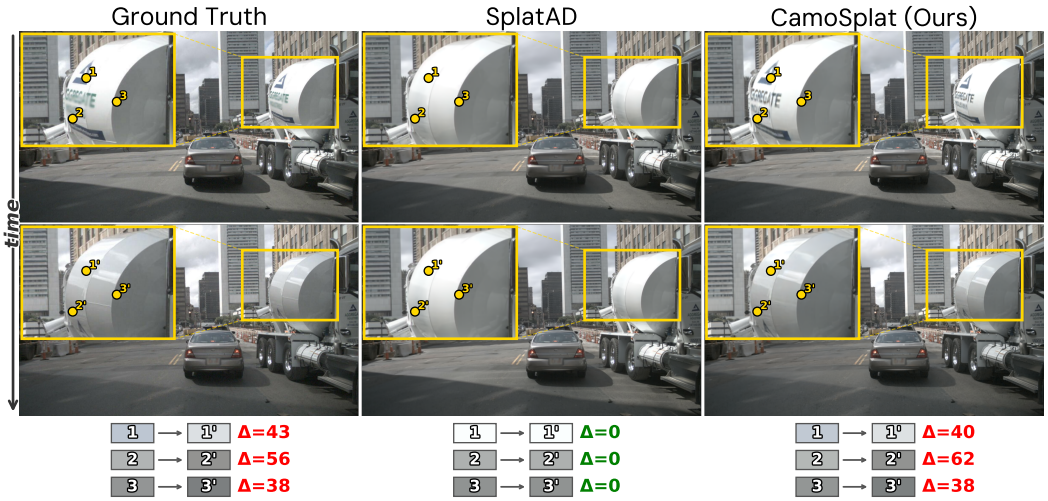

Figure 1: **A cement mixer truck with a rotating drum at two timesteps in nuScenes.** Numbers give the mean RGB difference between the timesteps at tracked pixels (numbered circles). SplatAD (Hess et al., 2025) keeps each Gaussian's appearance fixed and renders both timesteps identically, whereas CamoSplat reproduces the change.

In this paper, we introduce CamoSplat, which makes appearance a *per-primitive function of time*: every Gaussian across the scene, not only foreground objects, evolves its own appearance, decomposed into *what* changes and *when and how strongly* it changes. It realizes this through three components. i) A *Mixture-of-Temporal-Bases (MoTB)* mixes a few shared temporal networks with per-Gaussian weights to form each primitive's appearance pattern. ii) *Morlet wavelet temporal gating* modulates the amplitude of that pattern through per-Gaussian coefficients over a dyadic dictionary, where coarse scales express slow drift and fine scales sharp events. iii) *Sensor-specific opacity wavelets* add a per-sensor temporal offset to a shared base opacity, letting camera and LiDAR differ in what they see over time while sharing one geometry.

We evaluate CamoSplat on temporal-variation splits curated from Waymo, Argoverse 2, and nuScenes. Each split gathers sequences that contain such temporal appearance changes and leaves wide temporal gaps between training frames, so interpolation cannot recover them. CamoSplat sets a new state of the art in both photometric and LiDAR reconstruction. It improves PSNR by up to 2.5 dB on nuScenes, and cuts LiDAR Chamfer distance and depth error by 57% and 44% on Waymo over the strongest baseline. Notably, it reconstructs the localized temporal changes that prior methods blur into ghosting: brake lights and traffic signals switch at the right moment, and surfaces track illumination as it drifts.

Contributions. In summary, our work makes the following contributions:

- We propose **CamoSplat**, a temporal appearance modeling framework that modulates per-Gaussian appearance via a Mixture-of-Temporal-Bases (MoTB) with shared networks and per-Gaussian mixture weights, regulated by Morlet wavelet temporal gating across temporal scales.
- We introduce sensor-specific Morlet wavelet coefficients for temporal opacity, decoupling visibility changes across camera and LiDAR while sharing geometry.
- On the standard benchmark and newly constructed temporal-variation splits, CamoSplat achieves consistent state-of-the-art results on both photometric rendering and LiDAR reconstruction metrics, while faithfully reproducing per-scene appearance dynamics in qualitative comparisons.

2 Related Work

2.1 Gaussian Splatting for Driving Scenes

3D Gaussian Splatting (Kerbl et al., 2023) has become prevalent for driving-scene reconstruction due to its explicit representation, high visual fidelity, and real-time rendering. Gaussian-based methods have advanced dynamic-scene reconstruction through scene decomposition (Zhou et al., 2024b; Yan

et al., 2024; Zhou et al., 2024a; Chen et al., 2025; Peng et al., 2025) and modeling of actor motion and deformation (Chen et al., 2026; Xu et al., 2025; Song et al., 2025). Recent work has also moved toward sensor realism, enabling joint camera–LiDAR rendering from shared Gaussians and LiDAR-specific synthesis (Hess et al., 2025; Jiang et al., 2025; Kung et al., 2025). Most of these methods, however, keep each Gaussian's appearance fixed across the sequence; CamoSplat complements them by letting that appearance evolve over time.

2.2 Modeling Appearance Variation

Neural rendering methods have mainly handled appearance variation by conditioning on individual observations. Per-image embeddings absorb photometric differences between captures (Martin-Brualla et al., 2021; Tancik et al., 2022; Chen et al., 2022; Kulhanek et al., 2024; Zhang et al., 2024; Dahmani et al., 2024), bilateral grids correct pixel-level color (Wang et al., 2024) with multi-scale extensions for driving scenes (Wang et al., 2025), and per-frame codes drive per-Gaussian affine color transforms (Wu et al., 2026). Because these parameters are tied to observed frames, a held-out frame has none of its own. Street GS (Yan et al., 2024) and 4DGS (Yang et al., 2024b) make appearance continuous in time through Fourier series, yet every sinusoid spans the whole sequence, so a brief change such as a lamp switching on tends to be smeared across it (Section 3.3). Temporal opacity (Chen et al., 2026; Xu et al., 2025; Yang et al., 2024b; Li et al., 2024) lets Gaussians appear and vanish, but it scales their entire contribution rather than their color and does not distinguish between sensors. CamoSplat instead composes each Gaussian's pattern from shared temporal networks, decides when and how strongly it acts with time-localized Morlet wavelets, and separates camera and LiDAR visibility through sensor-specific opacity wavelets.

3 Method

We represent a driving scene as a set of N 3D Gaussians, each carrying a latent appearance feature $\mathbf{z}_i \in \mathbb{R}^D$ that is rasterized onto the image plane and decoded into sensor-specific outputs by per-sensor neural decoders: RGBs for camera and intensity with ray-drop probability for LiDAR. Static Gaussians model the background, while dynamic Gaussians are rigidly transformed along tracked bounding-box trajectories (Ost et al., 2021; Yan et al., 2024). This scene representation and rendering pipeline follow SplatAD (Hess et al., 2025).

Our contribution lies in the temporal appearance domain: we decompose temporal variation into **what** changes and **when and how strongly** it changes, and augment the base representation with three modules without altering the geometric pipeline (fig. 2).

3.1 Preliminary

Morlet wavelet basis. We construct a fixed temporal basis dictionary from the Morlet wavelet (Grossmann & Morlet, 1984):

$$\psi(t) = \exp\Big(-\frac{t^2}{2}\Big)\cos(\omega_0\, t), \tag{1}$$

where ω_0 is a hyperparameter controlling the center frequency. A family of basis functions is generated by dyadic scaling and translation (Daubechies, 1992):

$$\psi_{j,k}(\bar{t}) = \psi\left(\frac{\bar{t} - c_{j,k}}{s_j}\right), \quad s_j = 2^{-j}, \;\; c_{j,k} = \frac{k + 0.5}{2^j}, \tag{2}$$

for level $j = 0, \ldots, J-1$ and position index $k = 0, \ldots, 2^j - 1$. Together with a constant DC basis $\psi_0(\bar{t}) = 1$, this yields $B = 2^J$ fixed basis functions that tile the normalized time axis $\bar{t} \in [0, 1]$ at geometrically increasing resolution. Its Gaussian envelope concentrates each function around its center $c_{j,k}$, a locality that section 3.3 shows to be essential.

3.2 Mixture-of-Temporal-Bases (MoTB)

To model time-varying appearance, each Gaussian requires a temporal modulation vector $\mathbf{m}_i(\bar{t}) \in \mathbb{R}^D$ to adjust its base feature at every timestamp. A straightforward approach is to learn a function

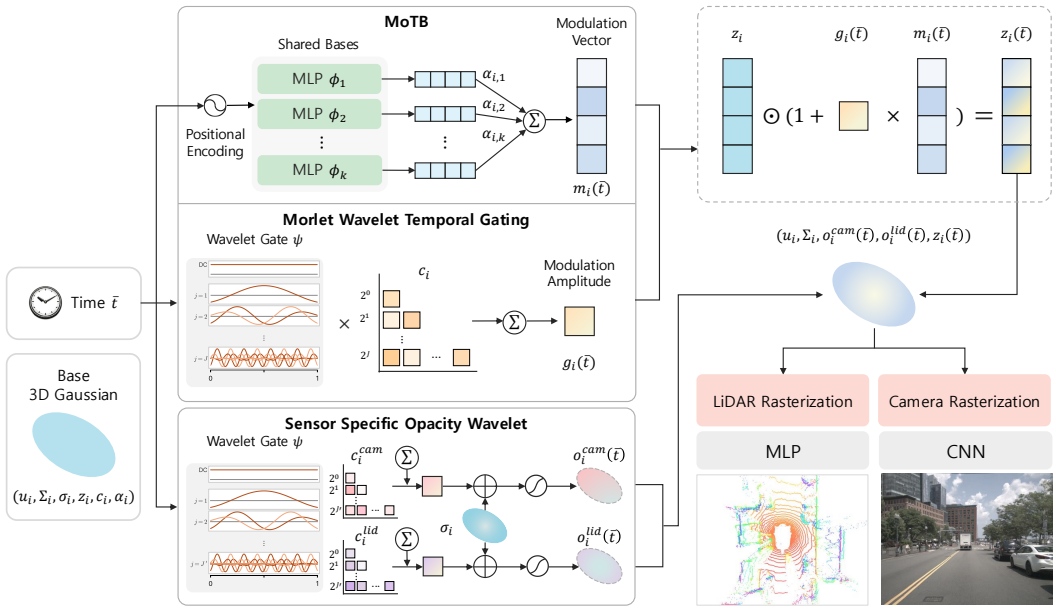

Figure 2: **Overview of CamoSplat.** We augment each Gaussian with three temporal components: (1) MoTB that mixes shared temporal networks with per-Gaussian weights into a modulation vector $\mathbf{m}_i(\bar{t})$, (2) a Morlet wavelet gate $g_i(\bar{t})$ that controls when and how strongly this modulation acts, and (3) sensor-specific opacity wavelets that produce time-varying camera and LiDAR opacities. The gated modulation is applied multiplicatively to the base feature $\mathbf{z}_i$, and the resulting features and opacities are rendered through a separate branch for each sensor.

$\phi : \bar{t} \mapsto \mathbb{R}^D$ that generates this vector, but a single function shared across all Gaussians usually has limited representational capacity to capture diverse temporal variations: a traffic light switching state requires a different temporal pattern than an adjacent building under gradual illumination drift. A per-Gaussian function would offer full flexibility but requires N independent networks, which is impractical for scenes with millions of primitives.

MoTB resolves this by maintaining K shared temporal networks $\{\phi_k\}_{k=1}^K$, which serve as learned bases for all Gaussians. Each ϕ_k encodes $\bar{t}$ via a Fourier positional encoding (Mildenhall et al., 2020; Tancik et al., 2020) $\gamma(\bar{t}) = \left[\sin(2\pi \cdot 2^l \bar{t}), \cos(2\pi \cdot 2^l \bar{t})\right]_{l=0}^{L-1}$ and transforms it through a two-layer MLP to produce $\phi_k(\bar{t}) \in \mathbb{R}^D$. Each Gaussian mixes their outputs through per-Gaussian weights $\boldsymbol{\alpha}_i \in \mathbb{R}^K$:

$$\mathbf{m}_i(\bar{t}) = \sum_{k=1}^{K} \alpha_{i,k}\, \phi_k(\bar{t}). \tag{3}$$

The K networks are independent of scene size; only the weights $\boldsymbol{\alpha}_i$ (K scalars per Gaussian) grow with N, providing per-Gaussian heterogeneity without per-Gaussian networks. We apply this modulation to the base appearance feature $\mathbf{z}_i$ via element-wise multiplication:

$$\mathbf{z}_i(\bar{t}) = \mathbf{z}_i \odot \big(1 + \lambda\, \mathbf{m}_i(\bar{t})\big), \tag{4}$$

where λ is a fixed scalar that sets the modulation strength. Because the residual is scaled by the base feature itself, modulation represents only a relative change in each Gaussian's appearance rather than an independent additive signal. When $\mathbf{m}_i \approx \mathbf{0}$, the original appearance is preserved.

3.3 Morlet Wavelet Temporal Gating

MoTB determines **what** pattern each Gaussian follows over time, but the Gaussian cannot choose **when** or **how strongly** that pattern acts. Its mixture weights $\boldsymbol{\alpha}_i$ stay constant across the sequence, so the pattern changes only when the shared network outputs do. Events, however, occur at times specific to each Gaussian: brake lights on two cars switch on at different instants, and a few shared

networks cannot supply a switch for each of them. Each Gaussian therefore needs its own gate that varies with time.

A single learnable scalar per Gaussian would set the strength but not the timing. Instead, we build the gate on the wavelet dictionary defined in section 3.1, which provides time-localized basis functions at multiple scales. Each Gaussian stores a learnable coefficient vector $\mathbf{c}_i \in \mathbb{R}^B$, a scalar weight per wavelet basis, and the gate value at time $\bar{t}$ is:

$$g_i(\bar{t}) = \mathbf{c}_i^\top \boldsymbol{\psi}(\bar{t}), \tag{5}$$

where $\boldsymbol{\psi}(\bar{t}) = [\psi_0(\bar{t}), \ldots, \psi_{B-1}(\bar{t})]^\top$. The gate is a linear combination of the wavelet bases, where each coefficient $c_{i,b}$ determines how strongly the corresponding basis ψ_b contributes. Coarse-scale bases capture slow amplitude envelopes, such as gradual illumination drift, while fine-scale bases enable sharp on/off transitions at specific moments, such as a traffic signal changing state.

Inserting the gate into eq. (4) gives the full temporally-gated modulation:

$$\mathbf{z}_i(\bar{t}) = \mathbf{z}_i \odot \big(1 + \lambda\, g_i(\bar{t})\, \mathbf{m}_i(\bar{t})\big). \tag{6}$$

Compared to the ungated form, the scalar $g_i(\bar{t})$ now controls the modulation amplitude for each Gaussian and timestep. A Gaussian that requires no temporal change simply learns $\mathbf{c}_i \approx \mathbf{0}$, zeroing out the gate across all timestamps and preserving its base appearance. Conversely, a Gaussian that must respond to a brief event activates only the fine-scale coefficients corresponding to that moment, leaving the rest of the sequence unaffected.

Why Morlet rather than Fourier. Prior temporal appearance models instead use Fourier functions (Yan et al., 2024; Yang et al., 2024b). A sinusoid keeps the same amplitude over the entire sequence, whereas a Morlet wavelet decays around its center. Sinusoids represent a brief event such as a lamp switching on by reinforcing each other at that instant and canceling out elsewhere, yet the loss enforces this cancellation only at the training frames. Between those frames, leftover oscillations spread the event onto other times, so the change is smeared in time and diluted at the event itself, whereas a few fine-scale wavelets capture it without disturbing the rest of the sequence. Figure 3 tests this by swapping the Morlet dictionary for a Fourier dictionary of the same size and keeping everything else unchanged. Even with its frequencies optimized during training, the Fourier variant weakens the glow of a lit tail lamp and turns a warm orange lamp nearly white on held-out frames, whereas the Morlet gate reproduces both. MoTB can still build its shared patterns from Fourier features, since each Gaussian's timing comes from its gate.

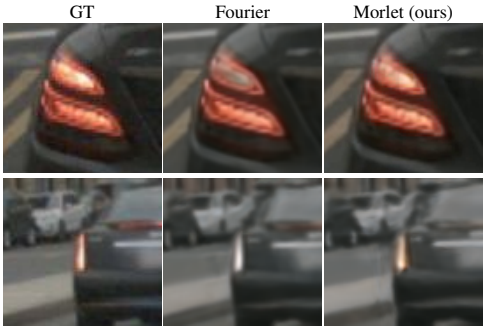

Figure 3: **Fourier vs. Morlet dictionary** on held-out nuScenes frames. The Fourier variant weakens the glow of a lit tail lamp (top) and turns a warm orange lamp nearly white (bottom).

3.4 Sensor-Specific Opacity Wavelet

The temporal gating in section 3.3 modulates appearance features, but visibility itself can also vary over time in a sensor-dependent manner. Camera and LiDAR differ not only in viewpoint and capture rate but in their fundamental sensing modalities: passive photometric imaging versus active range measurement. Consequently, a Gaussian that is clearly visible to one sensor may be transparent or irrelevant to the other, and this discrepancy can change over time. For example, the glow around a lit brake light appears in camera images only while the light is on and never returns a LiDAR point, whereas the lamp housing stays visible to both sensors. We model this by introducing a separate set of learnable wavelet coefficients dedicated to opacity, independent of the appearance gate coefficients $\mathbf{c}_i$ in eq. (5). Each Gaussian receives per-sensor opacity coefficients $\mathbf{c}_i^{\mathrm{cam}} \in \mathbb{R}^{B'}$ and $\mathbf{c}_i^{\mathrm{lid}} \in \mathbb{R}^{B'}$, while sharing the base opacity logit σ_i. These coefficients weight $\boldsymbol{\psi}'(\bar{t}) \in \mathbb{R}^{B'}$, the first $B'{=}2^{J'}$ functions of the same dictionary, which cover the DC term and the $J'{<}J$ coarsest

levels. The per-sensor temporal opacity is

$$o_i^{\text{cam}}(\bar{t}) = \text{sigmoid}\big(\sigma_i + \mathbf{c}_i^{\text{cam}\top} \boldsymbol{\psi}'(\bar{t})\big), \tag{7}$$

$$o_i^{\text{lid}}(\bar{t}) = \text{sigmoid}\big(\sigma_i + \mathbf{c}_i^{\text{lid}\top} \boldsymbol{\psi}'(\bar{t})\big). \tag{8}$$

The wavelet signal acts as a time-varying bias in logit space. Because these coefficients are initialized to zero, the model recovers baseline behavior $o_i^s(\bar{t}) = \text{sigmoid}(\sigma_i)$ at the start of training. Each sensor subsequently learns independent temporal visibility adjustments while the shared σ_i preserves geometric consistency. Because opacity now varies over time, MCMC densification (Kheradmand et al., 2024) uses each Gaussian's maximum opacity over time (**??**).

3.5 Training Objective

We adopt the loss terms of SplatAD (Hess et al., 2025): an L1+SSIM (Wang et al., 2004) camera loss $\mathcal{L}_{\text{cam}}$, and a LiDAR loss $\mathcal{L}_{\text{lid}}$ comprising depth, intensity, ray-drop, and line-of-sight (Rematas et al., 2022) terms. The MCMC densification strategy (Kheradmand et al., 2024) adds scale and opacity regularization $\mathcal{L}_{\text{mcmc}}$. The total objective is:

$$\mathcal{L} = \mathcal{L}_{\text{cam}} + \mathcal{L}_{\text{lid}} + \lambda_{\text{mcmc}}\, \mathcal{L}_{\text{mcmc}}. \tag{9}$$

All temporal modules are activated after τ_{start} warmup iterations to let the base geometry stabilize before temporal modulation begins.

4 Experiment

We evaluate how well CamoSplat captures temporal appearance variations in driving scenes, ranging from gradual changes to abrupt traffic-light transitions. We assess overall quality through quantitative comparisons and examine the fidelity of local appearance changes over time through qualitative analysis. Implementation details are provided in Appendix **??**.

4.1 Experimental Setup

Datasets. We evaluate CamoSplat on three autonomous driving datasets: Waymo (Sun et al., 2020), Argoverse 2 (Wilson et al., 2021), and nuScenes (Caesar et al., 2020). Although existing scene selections (Yang et al., 2024a; Hess et al., 2025) include temporal appearance variation, they provide limited coverage of challenging cases with pronounced appearance changes over time. To more directly evaluate reconstruction quality under such conditions, we curate 10 sequences from each dataset. Our selection focuses on clearly observable temporal changes.

For each selected scene, we adopt a linspace split with a 50% ratio. Our evaluation split widens the temporal gap between training frames, providing a more challenging setting of temporal modeling fidelity. Evaluation is performed at native resolution. Scene selections and preprocessing details are provided in Appendix **??**.

Compared Methods. We select eight baselines covering dynamic scene reconstruction and appearance modeling. PVG (Chen et al., 2026) and AD-GS (Xu et al., 2025) are included to compare with methods that explicitly model temporal visibility over time. DeSiRe-GS (Peng et al., 2025), a representative 4D Gaussian method, models scene dynamics as a function of time t. To assess performance against methods that explicitly address appearance variation, we include ArmGS (Wu et al., 2026) and OmniRe+BD (Wang et al., 2025), which employ multi-level appearance refinement and bilateral-grid-based photometric correction, respectively. We additionally include StreetGS (Yan et al., 2024), OmniRe (Chen et al., 2025), and SplatAD (Hess et al., 2025) as representative driving-scene reconstruction baselines.

4.2 Quantitative Results

We evaluate all methods on our temporal-variation splits; results on the standard benchmark scenes used by SplatAD (Hess et al., 2025) are reported in Appendix **??**.

Camera reconstruction. We evaluate camera reconstruction quality using PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). Table 1 reports camera reconstruction results on the

Table 1: **Quantitative results** of novel view synthesis across three driving datasets. First second third. †: Input images were downsampled to half resolution due to the extremely high computational cost. ‡: Results are obtained using our reimplementation of ArmGS based on the paper.

Method	Waymo			Argoverse 2			nuScenes		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
OmniRe (Chen et al., 2025)	25.18	0.786	0.262	–	–	–	25.17	0.752	0.321
OmniRe+BD (Wang et al., 2025)	–	–	–	–	–	–	26.24	0.757	0.306
SplatAD (Hess et al., 2025)	28.74	0.879	0.231	27.07	0.810	0.312	28.95	0.853	0.278
StreetGS (Yan et al., 2024)	26.97	0.884	0.241	24.09	0.725	0.457	26.49	0.829	0.252
PVG (Chen et al., 2026)	25.92	0.810	0.332	23.11	0.711	0.453	27.01	0.808	0.347
AD-GS (Xu et al., 2025)	26.91	0.851	0.232	–	–	–	27.29	0.837	0.246
DeSiRe-GS† (Peng et al., 2025)	26.98	0.835	0.279	–	–	–	–	–	–
ArmGS‡ (Wu et al., 2026)	29.62	0.889	0.243	26.13	0.780	0.392	28.52	0.849	0.322
CamoSplat (Ours)	29.73	0.886	0.204	28.47	0.833	0.287	31.46	0.878	0.233

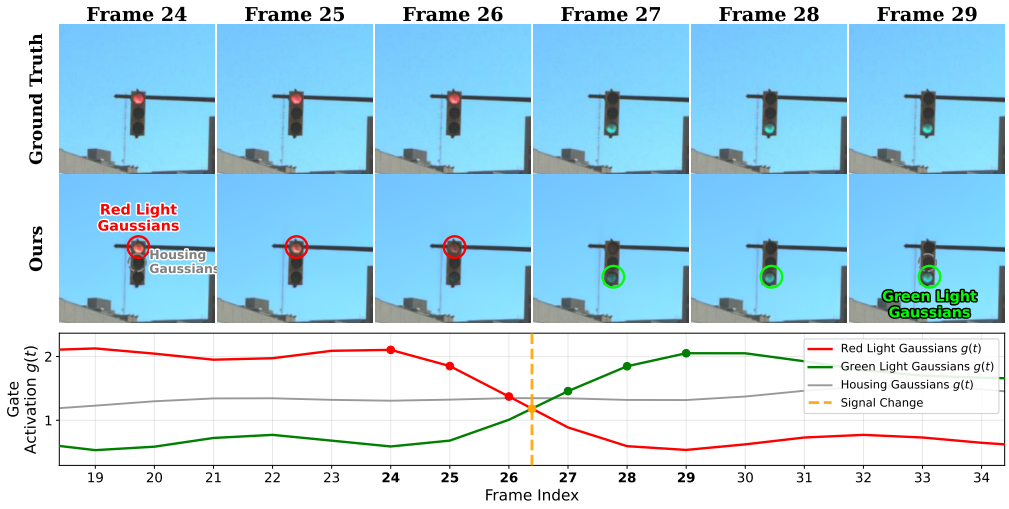

Figure 4: **Effect of temporal gating** on traffic light signal change on held-out Argoverse 2 frames. Gate activations $g(t)$ for Gaussians on the red and green lights show a precise crossing at the signal transition (orange dashed line), while the housing remains stable.

curated temporal-variation scenes from Waymo, Argoverse 2, and nuScenes. CamoSplat achieves the highest PSNR and lowest LPIPS (Zhang et al., 2018) among the evaluated methods on all three datasets, with PSNR gains of 0.11–2.51 dB over the best competing method on each dataset. CamoSplat also achieves the highest SSIM (Wang et al., 2004) on Argoverse 2 and nuScenes.

LiDAR reconstruction. We evaluate LiDAR reconstruction using depth error, intensity RMSE, ray-drop accuracy, and Chamfer distance (CD). Table 2 reports LiDAR reconstruction on the selected temporal-variation scenes. CamoSplat reduces depth error, intensity RMSE, and CD on all three datasets; ray-drop accuracy improves on Waymo and decreases slightly on Argoverse 2 and nuScenes.

4.3 Analysis

To verify that the gates capture when each Gaussian's appearance changes, Figure 4 examines temporal modulation within a traffic signal in Argoverse 2. We plot the scalar gates $g_i(\bar{t}) = \mathbf{c}_i^\top \boldsymbol{\psi}(\bar{t})$ for selected Gaussians associated with its red light, green light, and housing. In the ground truth, the signal state changes from red to green between frames 26 and 27. Over this interval, the gate associated with the red-light region decreases, while that associated with the green-light region increases, with the two responses precisely crossing between the frames. The housing gate exhibits comparatively smaller variation. These responses illustrate distinct temporal modulation among neighboring Gaussians within the same traffic light.



Figure 5: **Qualitative comparisons on selected driving scenes with temporal appearance variations.** Each row shows two timestamps with enlarged crops highlighting the changing regions.

Table 2: **LiDAR reconstruction on our temporal-variation splits.** Metrics are median depth/CD (m), intensity RMSE, and ray-drop accuracy (%). Best values are bold.

Method	Waymo				Argoverse 2				nuScenes			
	Depth↓	Int.↓	Ray-drop↑	CD↓	Depth↓	Int.↓	Ray-drop↑	CD↓	Depth↓	Int.↓	Ray-drop↑	CD↓
SplatAD	0.009	0.010	90.40	0.7	0.014	0.056	**93.29**	3.3	0.018	0.036	**94.74**	2.0
CamoSplat (Ours)	**0.005**	**0.009**	**94.23**	**0.3**	**0.011**	**0.055**	92.86	**3.1**	**0.006**	**0.032**	94.56	**1.5**

Table 3: **Ablation study** of the proposed modules on the curated subset of nuScenes. First second third.

MoTB	Gating	Opacity Wavelet		Camera			LiDAR			
		Camera	LiDAR	PSNR↑	SSIM↑	LPIPS↓	Depth↓	Int.↓	Ray-drop (%)↑	CD↓
–	–	–	–	28.95	0.853	0.278	0.018	0.036	94.74	2.0
✓	–	–	–	30.06	0.865	0.282	0.016	0.036	94.93	1.9
✓	✓	–	–	31.12	0.875	0.256	0.014	0.034	94.66	2.3
✓	✓	✓	–	31.47	0.879	0.236	0.018	0.033	94.55	2.3
✓	✓	✓	✓	31.46	0.878	0.233	0.006	0.032	94.56	1.5

Figure 5 compares the reconstruction of local appearance changes at two timestamps in selected driving scenes. Appendix **??** provides additional results on nuScenes and Argoverse 2. **(a)** CamoSplat reproduces the changing pedestrian-signal indications and on/off brake-light states. AD-GS, which uses temporal opacity masks to regulate object visibility, captures this change but renders the brake-light region with noticeable blur. ArmGS captures the brake-light change, but under-reproduces part of the illuminated pedestrian-signal indication. **(b)** As the white vehicle moves between illuminated and shadowed regions, CamoSplat reproduces the resulting brightness change more faithfully than SplatAD and ArmGS. **(c)–(d)** CamoSplat reproduces the distinct signal states in both night-time and day-time examples, capturing changes between red and green in (c) and between yellow and red in (d). SplatAD and ArmGS exhibit residual activation of inactive signal lights, while AD-GS captures the color changes but renders the signal regions with noticeable blur. These artifacts weaken the distinction between signal states in the baseline renderings.

4.4 Ablation Study

Table 3 reports the contribution of each component on the curated nuScenes scenes. MoTB allows individual Gaussians to follow distinct temporal appearance patterns through learned combinations of shared temporal networks, improving PSNR and SSIM, although LPIPS slightly increases. Morlet wavelet gating allows their strength to vary locally in time. Its addition improves all three camera metrics, supporting the benefit of controlling temporal expression separately from pattern construction. Adding the camera opacity wavelets yields further image-quality gains but increases depth error. The LiDAR opacity wavelets introduce a separate temporal adjustment for LiDAR rendering, improving depth and point-cloud reconstruction while preserving camera quality.

5 Conclusion

We presented CamoSplat, which makes appearance a per-primitive function of time for driving-scene Gaussian Splatting and decomposes its temporal variation into *what* changes and *when and how strongly* it changes. A Mixture-of-Temporal-Bases compose each Gaussian's pattern from a few shared temporal networks, and Morlet wavelet gating modulates its amplitude through per-Gaussian coefficients, where coarse scales express slow drift and fine scales sharp events. Sensor-specific opacity wavelets extend this control to camera and LiDAR visibility while both sensors share one geometry. On temporal-variation splits from Waymo, Argoverse 2, and nuScenes, CamoSplat achieves state-of-the-art photometric and LiDAR reconstruction and reproduces brake lights and traffic signals that prior methods collapse into blur or ghosting. Our analysis shows why temporal locality matters: the gates of adjacent red and green lamps cross exactly at the transition, whereas a Fourier dictionary smears such brief changes in time and weakens them.

Limitations and future work. The three components come at a cost: training time and peak memory grow by about $1.8\times$ and $2.4\times$ over SplatAD, and rendering drops from 37 to 21 FPS (Appendix **??**). CamoSplat also inherits geometry and actor motion from SplatAD; we plan to extend its temporal modeling to both, so that wavelet-gated deformations can capture non-rigid actors such as walking pedestrians.

AI USE STATEMENT

We used large language model assistants (Claude by Anthropic and GPT by OpenAI) while preparing this paper. They helped draft and edit parts of the text for readability, identify and format references, check that the equations match our implementation, and give feedback on how the method is presented. The research ideas, method design, implementation, experiments, and results are the authors' own, and the authors reviewed all AI-assisted content and take full responsibility for the 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 *CVPR*, 2020.

Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *CVPR*, 2022.

Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *IJCV*, 2026.

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 *ICLR*, 2025.

Hiba Dahmani, Moussab Bennehar, Nathan Piasco, Luis Roldão, and Dzmitry Tsishkou. SWAG: Splatting in the wild images with appearance-conditioned Gaussians. In *ECCV*, 2024.

Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

A. Grossmann and J. Morlet. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.*, 15(4):723–736, 1984.

Georg Hess, Carl Lindström, Maryam Fatemi, Christoffer Petersson, and Lennart Svensson. SplatAD: Real-time lidar and camera rendering with 3D gaussian splatting for autonomous driving. In *CVPR*, 2025.

Junzhe Jiang, Chun Gu, Yurui Chen, and Li Zhang. GS-LiDAR: Generating realistic LiDAR point clouds with panoramic gaussian splatting. In *ICLR*, 2025.

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

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3D gaussian splatting as Markov chain Monte Carlo. In *NeurIPS*, 2024.

Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. WildGaussians: 3D Gaussian splatting in the wild. In *NeurIPS*, 2024.

Pou-Chun Kung, Xianling Zhang, Katherine A. Skinner, and Nikita Jaipuria. LiHi-GS: LiDAR-supervised Gaussian splatting for highway driving scene reconstruction. *IEEE Robotics and Automation Letters*, 2025.

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime Gaussian feature splatting for real-time dynamic view synthesis. In *CVPR*, 2024.

Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021.

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 *ECCV*, 2020.

Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *CVPR*, 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 *CVPR*, 2025.

Konstantinos Rematas, Andrew Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, 2022.

Rui Song, Chenwei Liang, Yan Xia, Walter Zimmer, Hu Cao, Holger Caesar, Andreas Festag, and Alois Knoll. CoDa-4DGS: Dynamic gaussian splatting with context and deformation awareness for autonomous driving. In *ICCV*, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, 2020.

Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In *CVPR*, 2022.

Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. NeuRAD: Neural rendering for autonomous driving. In *CVPR*, 2024.

Nan Wang, Lixing Xiao, Yuantao Chen, Weiqing Xiao, Pierre Merriaux, Lei Lei, Ziyang Yan, Saining Zhang, Shaocong Xu, Chongjie Ye, Bohan Li, Zhaoxi Chen, Tianfan Xue, and Hao Zhao. Unifying appearance codes and bilateral grids for driving scene Gaussian splatting. In *NeurIPS*, 2025.

Yuehao Wang, Chaoyi Wang, Bingchen Gong, and Tianfan Xue. Bilateral guided radiance field processing. *ACM TOG*, 43(4), 2024.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks*, 2021.

Guile Wu, Dongfeng Bai, and Bingbing Liu. ArmGS: Composite Gaussian appearance refinement for modeling dynamic urban environments. In *ICRA*, 2026.

Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, Yuxin Huang, Xiaoyu Ye, Zike Yan, Yongliang Shi, Yiyi Liao, and Hao Zhao. MARS: An instance-aware, modular and realistic simulator for autonomous driving. In *CICAI*, 2023.

Jiawei Xu, Kai Deng, Zexin Fan, Shenlong Wang, Jin Xie, and Jian Yang. AD-GS: Object-aware B-spline gaussian splatting for self-supervised autonomous driving. In *ICCV*, 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 *ECCV*, 2024.

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

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 *CVPR*, 2023.

Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4D Gaussian splatting. In *ICLR*, 2024b.

Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3D gaussian splatting for unconstrained image collections. In *ECCV*, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Hongyu Zhou, Jiahao Shao, Lu Xu, Dongfeng Bai, Weichao Qiu, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. HUGS: Holistic urban 3D scene understanding via Gaussian splatting. In *CVPR*, 2024a.

Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. HUGSIM: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *IEEE TPAMI*, 48(4):4673–4691, 2026.

Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. DrivingGaussian: Composite Gaussian splatting for surrounding dynamic autonomous driving scenes. In *CVPR*, 2024b.