# SPECTRALSPLAT: APPEARANCE-DISENTANGLED FEED-FORWARD GAUSSIAN SPLATTING FOR DRIVING SCENES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Feed-forward 3D Gaussian Splatting methods reconstruct driving scenes efficiently, but their predicted colors are tied to the lighting and weather of the input images. We present SpectralSplat, which equips a feed-forward reconstruction with a learned *appearance library* for scene-level weather and time-of-day control without changing its geometry. Each color head is evaluated with a zero code to produce a canonical *base* color and with a global appearance code to produce an *adapted* color. A hybrid relighting pipeline combines physics-based intrinsic priors with diffusion refinement to create paired training images; invariance, reconstruction, appearance-swap, and base-color losses encourage the separation of content and appearance. The library averages codes extracted from relit exemplars, and an appearance-adaptable temporal history caches features rather than fixed colors so accumulated Gaussians can be recolored under a selected code. Reconstruction results on Waymo and nuScenes remain comparable to the UniSplat backbone, while Waymo experiments demonstrate library-conditioned rendering, cross-appearance reconstruction, and code interpolation. The method controls learned scene-level appearance rather than physically complete weather or arbitrary real-to-real illumination transfer.

## 1 INTRODUCTION

Reconstructing 3D driving scenes from multi-camera video underpins autonomous-vehicle simulation, planning validation, and closed-loop testing (Yang et al., 2023). Feed-forward 3D Gaussian Splatting (3DGS) methods (Chen et al., 2024a; Charatan et al., 2024; Shi et al., 2026) produce a full set of Gaussian primitives in a single network pass, enabling real-time novel view synthesis without per-scene optimization; UniSplat (Shi et al., 2026) is a strong backbone for large-scale driving reconstruction.

Current feed-forward methods, however, *entangle* appearance with geometry: predicted Gaussian colors are baked in and tied to the lighting, weather, and exposure of the input images. Changing these conditions requires additional appearance control rather than replaying the reconstruction as produced. Entanglement also undermines temporal accumulation: a history buffer that aggregates Gaussians observed under different illumination yields inconsistent coloring.

We introduce SpectralSplat, which separates appearance from geometry inside a feed-forward 3DGS framework and equips it with a learned *appearance library* (Fig. 1). The code controls color without changing reconstructed geometry, enabling scene-level appearance edits such as night, rain, snow, or golden hour. Concretely, we augment the UniSplat backbone with a global appearance code pooled from DINOv2 (Oquab et al., 2024) tokens and shared by all Gaussians, a *factored color prediction* that evaluates each color head with a zeroed code (appearance-agnostic *base color*) and with the actual code (*adapted color*), and an *appearance-adaptable temporal history* that caches features so accumulated Gaussians can be recolored under a selected code (Sec. 3). Training requires paired views of the same scene under different conditions, which forward-only driving logs do not provide. We build such pairs, and thereby the library itself, with a hybrid relighting pipeline that fuses physics-based intrinsic priors (Wu et al., 2025) with diffusion-based refinement (Zhang et al., 2025), and train with four complementary losses that encourage this factorization.



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.

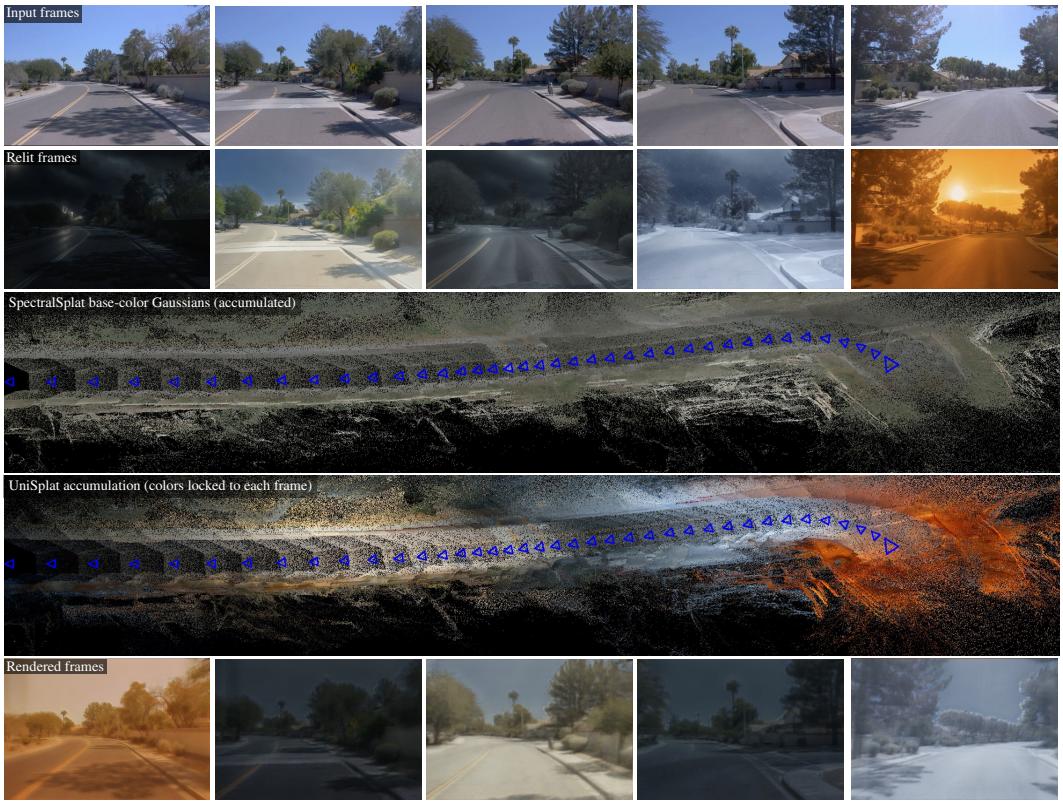



Figure 1: **Appearance-disentangled Gaussian reconstruction.** Given input frames (row 1) relit under progressively varying conditions (row 2, IC-Light), SpectralSplat accumulates Gaussians whose base colors stay consistent along the sequence (row 3, bird's-eye view), whereas the UniSplat baseline bakes each frame's appearance into its Gaussians and accumulates inconsistent colors (row 4). Row 5 re-renders the SpectralSplat Gaussians with the appearance codes extracted from the relit frames, changing only appearance.

Appearance-conditioned rendering exists in optimization-based Gaussian Splatting (Kulhanek et al., 2024; Dahmani et al., 2024; Lin et al., 2025), where a separate code is fitted per training image and per scene. Our contribution is not appearance conditioning itself but making it compatible with feed-forward reconstruction and streaming temporal history, with a library of conditions that is learned once and applied to every new scene. The library is deliberately limited to the scene-level conditions produced by the relighting pipeline: it does not model spatially varying illumination, and it is not a tool for copying the appearance of one arbitrary real scene onto another (App. E). Table 1 positions SpectralSplat among existing approaches. Our contributions are:

- A hybrid relighting pipeline (physics-based intrinsic priors + diffusion refinement with frequency-aware latent guidance) that uses shared lighting and structural priors to generate paired training data and library conditions.
- A global appearance embedding with factored color prediction that re-renders a reconstructed scene under any library condition without touching the geometry pathway.
- Paired supervision losses that encourage base-color invariance and appearance transfer.
- A temporal history that caches features and recomputes colors of accumulated Gaussians under a selected condition.

## 2 Related Work

**Optimization-based and feed-forward 3D reconstruction.** Classical pipelines estimate cameras with Structure-from-Motion (Schonberger & Frahm, 2016; Snavely et al., 2006) and stereo matching (Furukawa & Ponce, 2009; Schönberger et al., 2016), then optimize a neural scene representation per scene with NeRF (Mildenhall et al., 2020) or its variants (Barron et al., 2022; Kerbl et al., 2023;

Table 1: **Method comparison.** We compare along three axes: *Feed-forward* – inference without per-scene optimization; *Disentangled* – appearance is disentangled from geometry; *Cross-scene control* – a condition or style can be applied across scenes without fitting a new appearance code for each scene. Per-scene code switching alone does not meet this criterion. Among these methods, SpectralSplat combines all three in a driving-scene setting.

| Method | Feed-forward | Disentangled | Cross-scene control |
|---|---|---|---|
| NeRF (Mildenhall et al., 2020), 3DGS (Kerbl et al., 2023) | ✗ | ✗ | ✗ |
| NeRF-W (Martin-Brualla et al., 2021) | ✗ | ✓ | ✗ |
| WildGaussians (Kulhanek et al., 2024) | ✗ | ✓ | ✗ |
| SWAG (Dahmani et al., 2024) | ✗ | ✓ | ✗ |
| StyleRF (Liu et al., 2023) | ✗ | ✓ | ✓ |
| MVSplat (Chen et al., 2024a), DepthSplat (Xu et al., 2025) | ✓ | ✗ | ✗ |
| UniSplat (Shi et al., 2026) | ✓ | ✗ | ✗ |
| **SpectralSplat (ours)** | ✓ | ✓ | ✓ |

Barron et al., 2023; Wang et al., 2021; Huang et al., 2024; Herau et al., 2023; 2024b;a; 2026; Yang et al., 2024), which takes minutes to hours per scene. Feed-forward methods instead regress 3D representations directly: Splatter Image (Szymanowicz et al., 2024), PixelSplat (Charatan et al., 2024), MVSplat (Chen et al., 2024a), and DrivingRecon (Lu et al., 2025) predict Gaussians from posed images in one pass, with extensions to pose-free inputs (Ye et al., 2025; Smart et al., 2024), causal streaming (Wang & Agapito, 2025; Chen et al., 2024b), and joint structure-and-pose inference (Wang et al., 2025a; Jiang et al., 2025). For driving, UniSplat (Shi et al., 2026) fuses multi-view and multi-frame information through a 3D latent scaffold. All of these produce appearance-entangled reconstructions whose Gaussian colors are locked to the input lighting.

**Appearance-conditioned view synthesis.** Block-NeRF (Tancik et al., 2022) and NeRF in the Wild (Martin-Brualla et al., 2021) learn per-image latent codes for illumination and exposure; WildGaussians (Kulhanek et al., 2024), SWAG (Dahmani et al., 2024), and DAVIGS (Lin et al., 2025) bring appearance embeddings and affine color transforms to 3D Gaussian Splatting. These methods fit a separate trainable code per training image by per-scene optimization and rely on dense multi-view coverage of the *same* static scene under *different* lighting. Driving logs are long forward-facing trajectories that rarely revisit a location, so repeated observations under different lighting are scarce, and per-scene codes are not directly reusable on unseen scenes. SpectralSplat differs in three ways: codes are encoded from relit exemplars to build a reusable library, color is factored into a zero-code base stream and an adapted stream, and cached temporal Gaussians are recolored under the current code. The contribution is thus the feed-forward and streaming integration of appearance factorization, not appearance conditioning per se.

**Generative relighting, stylization, and driving world models.** IC-Light (Zhang et al., 2025) and Uni-Renderer (Chen et al., 2025) impose light-transport consistency in 2D diffusion models, while MVInverse (Wu et al., 2025) decomposes scenes into intrinsic components for physics-based re-rendering; 3D stylization (Zhang et al., 2022; Liu et al., 2023) lifts 2D styles into radiance fields. 2D diffusion lacks multi-view consistency and inverse rendering yields high-frequency artifacts in unbounded scenes; our hybrid pipeline combines the two only to build paired training data. Text-conditioned driving video and world models such as GAIA-1 (Hu et al., 2023), DriveDreamer (Wang et al., 2024), MagicDrive (Gao et al., 2024a), and Vista (Gao et al., 2024b) can steer weather or time of day, but they generate video rather than recoloring an explicit, fixed 3D Gaussian reconstruction.

## 3 METHOD

SpectralSplat disentangles appearance from geometry within a feed-forward Gaussian Splatting framework (overview in Fig. 2). It augments the color pathway of the UniSplat (Shi et al., 2026) backbone without changing the geometry architecture. Once reconstructed, the Gaussian geometry is independent of the selected appearance code.

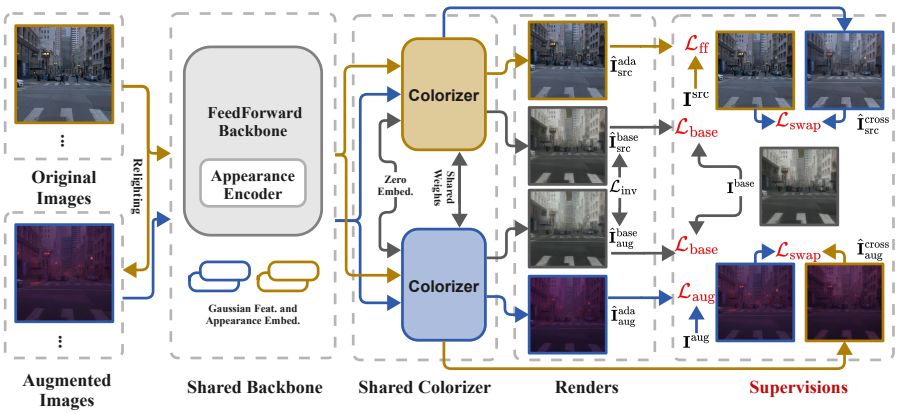



Figure 2: **Training pipeline.** Original and augmented inputs share the depth-derived scaffold, not all decoded Gaussian parameters, and produce separate features and appearance embeddings. Four losses encourage the factorization: $\mathcal{L}_{\text{inv}}$ (base invariance), $\mathcal{L}_{\text{aug}}$ (augmented reconstruction), $\mathcal{L}_{\text{swap}}$ (appearance swap), and $\mathcal{L}_{\text{base}}$ (both base renders anchored to source albedo). Both swap directions are shown; only one is sampled per training step.

### 3.1 Feed-Forward Gaussian Splatting Backbone

UniSplat uses a frozen Pi3 (Wang et al., 2025b) geometry transformer for dense 3D points and a fine-tuned DINOv2-S (Oquab et al., 2024) backbone for semantics, fused by a feature pyramid network (FPN) into per-pixel features $\mathbf{z}_i^{\text{img}} \in \mathbb{R}^{D_{\text{img}}}$; the points are voxelized and processed by a sparse 3D U-Net with temporal history fusion into per-voxel features $\mathbf{z}_j^{\text{vox}} \in \mathbb{R}^{D_{\text{vox}}}$. A dual-branch decoder predicts Gaussians: the *voxel branch* decodes $K$ Gaussians per voxel from $\mathbf{z}_j^{\text{vox}}$, and the *point branch* one Gaussian per pixel from $\mathbf{z}_i^{\text{img}}$ concatenated with the voxel feature sampled at its 3D position. Both predict position offset, scale, rotation, opacity, and a dynamic score as in UniSplat (App. A.1); only color prediction is factored.

### 3.2 Global Appearance Embedding

We derive a scene-level appearance code from the DINOv2 features already extracted by the backbone. The projected 2048-dimensional DINOv2 patch tokens of each camera are globally average-pooled and passed through a lightweight MLP encoder $\phi$ (App. A.2):

$$\mathbf{a}_k = \phi\left(\frac{1}{N_p}\sum_{n=1}^{N_p}\mathbf{f}_{k,n}\right), \quad \mathbf{a} = \frac{1}{|C_t|}\sum_{k\in C_t}\mathbf{a}_k \in \mathbb{R}^d, \tag{1}$$

where $\mathbf{f}_{k,n} \in \mathbb{R}^{2048}$ are the $N_p$ patch tokens of camera $k$, $C_t$ the cameras at the current timestamp, and $d$=64. We call $\mathbf{a}$ the *appearance code*; it is shared by all Gaussians in the scene, so every view renders the same appearance-conditioned 3D representation. During training $\mathbf{a}$ is computed from the input images; at inference it can be replaced by a stored code. The *appearance library* is built once by encoding relit frames, grouping them by generation condition (weather/time-of-day presets), and storing the per-condition centroid $\bar{\mathbf{a}}_c$ (the library code of condition $c$) and the global mean $\bar{\mathbf{a}}$. Selecting an entry re-renders the current reconstruction under that condition; entries can be linearly blended, and an optional strength $\gamma$ gives the code $\bar{\mathbf{a}} + \gamma(\bar{\mathbf{a}}_c - \bar{\mathbf{a}})$. Although UniSplat supervises on future frames ($t$+1), $\mathbf{a}$ is computed only from current-timestamp cameras and reused for future-frame renders, so the code captures only the global illumination state at time $t$; the disentanglement losses below are likewise applied only to current-frame renders. This design encodes the observation that appearance changes in driving scenes (tone, color cast, illumination mood) are predominantly global; spatially varying effects such as local shadows, headlights, and specularities are outside this global-code model (App. E).

### 3.3 FACTORED COLOR PREDICTION

Each branch has a colorizer MLP whose weights are shared between its base and adapted evaluations. For the point branch, each Gaussian feature is the concatenation of the sampled voxel feature $\mathbf{z}_i^{\text{vox}}$ and the per-pixel FPN feature $\mathbf{z}_i^{\text{img}}$:

$$\mathbf{c}_i^{\text{base}} = \sigma\big(f_{\text{rgb}}^{\text{pt}}([\mathbf{z}_i^{\text{vox}};\, \mathbf{z}_i^{\text{img}};\, \mathbf{0}])\big), \tag{2}$$

$$\mathbf{c}_i^{\text{ada}} = \sigma\big(f_{\text{rgb}}^{\text{pt}}([\mathbf{z}_i^{\text{vox}};\, \mathbf{z}_i^{\text{img}};\, \mathbf{a}])\big), \tag{3}$$

where $\sigma$ is a sigmoid and $[\cdot;\cdot]$ denotes concatenation. The **base color** $\mathbf{c}^{\text{base}}$ (zero embedding) is trained to provide a canonical color reference; the **adapted color** $\mathbf{c}^{\text{ada}}$ matches the target appearance. The voxel and sky color heads concatenate the code with $\mathbf{z}_j^{\text{vox}}$ and $\mathbf{z}_i^{\text{img}}$, respectively. A zero vector rather than a learned canonical embedding gives the base color a fixed reference point. Rendering uses $\mathbf{c}^{\text{ada}}$; $\mathbf{c}^{\text{base}}$ serves as a disentanglement signal (Sec. 3.6).

### 3.4 APPEARANCE-ADAPTABLE TEMPORAL HISTORY

UniSplat accumulates Gaussians from previous timesteps in a temporal history buffer to improve coverage. Cached Gaussians keep their original colors, which causes inconsistencies when past and present frames were captured under different lighting. We make accumulated Gaussians *re-renderable* under the current appearance at recall time. For the voxel branch, temporal fusion already happens at the feature level inside the 3D U-Net, so voxel RGB is always predicted fresh with the current embedding. For the point branch, we cache the per-pixel FPN feature $\mathbf{z}_i^{\text{img}}$ with each static point Gaussian. When recalled, the geometry is transformed to the current frame, fresh voxel context is queried from the current scaffold, and the color is recomputed:

$$\mathbf{c}_i^{\text{hist}} = \sigma\big(f_{\text{rgb}}^{\text{pt}}([\mathbf{z}_i^{\text{vox,fused}};\, \mathbf{z}_i^{\text{img,cached}};\, \mathbf{a}^{\text{now}}])\big). \tag{4}$$

New and historical Gaussians are thus rendered under the same library condition instead of being locked to the appearance under which they were generated.

**Why condition the 3D representation instead of editing the inputs?** Relighting the input images and rerunning UniSplat is an intuitive baseline, but geometry, dynamic masks, and temporal history are then re-estimated for every target appearance, so output differences may stem from a changed reconstruction rather than from appearance alone. SpectralSplat holds geometry and cached point features fixed and changes only the color stream, so one reconstruction can be re-rendered under every library condition and historical Gaussians recolored consistently.

### 3.5 HYBRID MULTI-VIEW RELIGHTING

Training requires paired observations of the same scene content under different conditions, which forward-only driving captures do not provide. We generate such pairs, which also define the conditions of the appearance library, with a hybrid pipeline that combines physics-based intrinsic decomposition with a diffusion prior. MVInverse (Wu et al., 2025) provides per-pixel base color $\mathbf{A}_v$ and normals $\mathbf{N}_v$, from which a Lambertian re-rendering under a target light is computed; the shared lighting supplies a cross-view structural prior, though the reference is flat and artifact-prone. IC-Light (Zhang et al., 2025) is photorealistic but inconsistent across views when applied per image. We reconcile the two with *frequency-aware latent guidance*: high-frequency energy in the input latent defines a spatial mask that gives the physics-based reference more influence in smooth regions and less in textured regions. The estimated clean latent is blended toward this reference with a decaying guidance weight and passed to a DPM-Solver++ sampler (App. B, Eq. 6). Early steps emphasize the lighting reference; later steps allow more generative refinement. Figure 3 compares the hybrid pipeline with its components. IC-Light supplies paired training data and is never run at reconstruction or library-rendering inference.

### 3.6 PAIRED APPEARANCE SUPERVISION AND TRAINING OBJECTIVE

Original $\mathbf{I}^{\text{src}}$ and augmented $\mathbf{I}^{\text{aug}}$ images are processed by the shared backbone, yielding embeddings $\mathbf{a}^{\text{src}}$ and $\mathbf{a}^{\text{aug}}$ (Fig. 2). The point/voxel scaffold comes from the original depth predictions




Figure 3: **Relighting pipeline.** Physics rendering supplies a structural reference. In this example, independent IC-Light generation substantially changes scene content; the hybrid retains the road layout while changing its tone and appearance. This is a qualitative illustration, not a quantitative consistency or realism evaluation.

and is shared; FPN features, U-Net processing, embeddings, and Gaussian offsets, scales, rotations, and opacities are predicted separately for each input. This differs from substituting a code within one reconstruction, which leaves all Gaussian geometry fixed. Dynamic-mask supervision (binary cross-entropy) is applied to both renders; the base-color anchor $\mathcal{L}_{\text{base}}$ below is restricted to non-sky pixels, since MVInverse provides no sky albedo, while the other losses use the full image, so the appearance-conditioned sky Gaussians learn their adapted color from $\mathcal{L}_{\text{aug}}$ and $\mathcal{L}_{\text{swap}}$. **Base invariance:** $\mathcal{L}_{\text{inv}} = \|\hat{\mathbf{I}}^{\text{base}}_{\text{src}} - \hat{\mathbf{I}}^{\text{base}}_{\text{aug}}\|_2^2$. **Augmented reconstruction:** $\mathcal{L}_{\text{aug}} = \|\hat{\mathbf{I}}^{\text{ada}}_{\text{aug}} - \mathbf{I}^{\text{aug}}\|_2^2$. **Appearance-swap consistency:** at each step one of the two cross directions (augmented features with the original embedding, or vice versa) is rendered and supervised against the matching ground truth, $\mathcal{L}_{\text{swap}} = \|\hat{\mathbf{I}}^{\text{cross}} - \mathbf{I}^{\text{target}}\|_2^2$. **Base color consistency:** $\mathcal{L}_{\text{base}} = \sum_{q\in\{\text{src,aug}\}} \|\hat{\mathbf{I}}^{\text{base}}_{q} - \mathbf{A}^{\text{src}}\|^2_{2,M}$ anchors both base renders to the same fixed MVInverse base-color target from the source views (Sec. 3.5); $M$ denotes a mean over non-sky pixels. The full objective is

$$\mathcal{L} = \underbrace{\lambda_m \mathcal{L}_{\text{mse}} + \lambda_p \mathcal{L}_{\text{lpips}} + \lambda_d \mathcal{L}_{\text{dyn}} + \lambda_s \mathcal{L}_{\text{depth}}}_{\text{original feed-forward supervision } \mathcal{L}_{\text{ff}}} + \beta_1 \mathcal{L}_{\text{inv}} + \beta_2 \mathcal{L}_{\text{aug}} + \lambda_p \mathcal{L}^{\text{aug}}_{\text{lpips}} + \beta_3 \mathcal{L}_{\text{swap}} + \beta_4 \mathcal{L}_{\text{base}}, \tag{5}$$

with $\lambda_m$=5.0, $\lambda_p$=0.05, $\lambda_d$=0.05, $\lambda_s$=0.1, $\beta_1$=1.0, $\beta_2$=5.0, $\beta_3$=0.5, $\beta_4$=0.5. The original photometric terms supervise source adapted renders; $\mathcal{L}_{\text{aug}}$ and $\mathcal{L}^{\text{aug}}_{\text{lpips}}$ supervise the augmented adapted render with MSE and LPIPS. $\mathcal{L}_{\text{inv}}$ and $\mathcal{L}_{\text{base}}$ directly supervise base renders. Because the color heads share parameters, adapted-render losses also update weights used for base-color prediction. Paired supervision encourages the factorization; it does not guarantee feature-level independence or general identifiability.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets and metrics.** *Waymo Open Dataset* (Sun et al., 2020) has 798 training and 202 validation sequences with five surround cameras; following UniSplat (Shi et al., 2026) we evaluate *reconstruction* (input views at time $t$) and *novel view synthesis* (NVS, views at $t$+1). *nuScenes* (Caesar et al., 2020) has six cameras; following Shi et al. (2026) we split scenes into equally spaced bins (135,941 training, 30,080 validation) and evaluate on the first, last, and central frames of each bin. We report PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). App. C.1 distinguishes the quantitative and qualitative protocols.

**Baselines.** Feed-forward: UniSplat, MVSplat (Chen et al., 2024a), DepthSplat (Xu et al., 2025) on Waymo; PixelSplat (Charatan et al., 2024), MVSplat, Omni-Scene (Wei et al., 2025), UniSplat on nuScenes. Appearance disentanglement: WildGaussians (Kulhanek et al., 2024) (per-scene optimization).

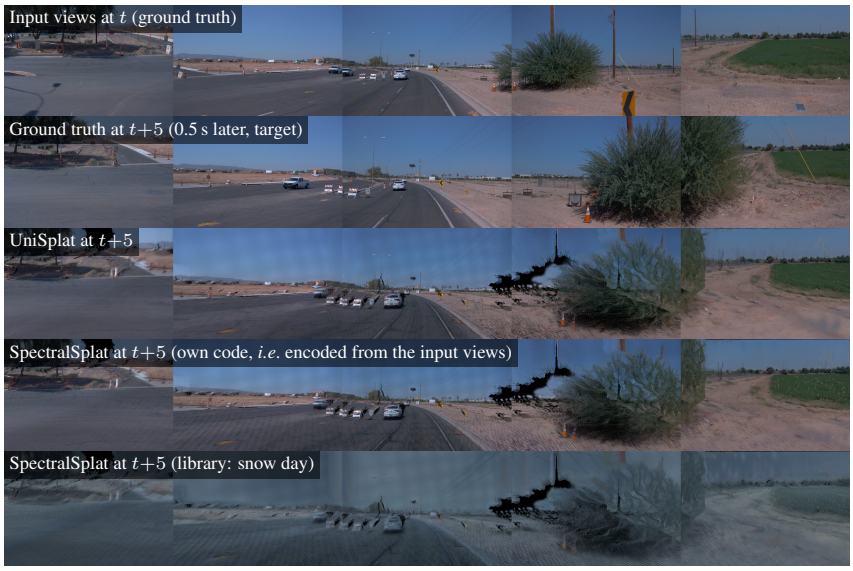



Figure 4: **Qualitative NVS.** *Top:* input views at $t$ (five cameras). *Below:* the predicted Gaussians rendered from the poses 0.5 s later ($t$+5; ego and lead car have moved) — ground truth, UniSplat, and SpectralSplat, which matches the baseline (19.1 vs. 19.0 dB here, no temporal history; Table 2 uses the $t$+1 benchmark protocol). A library code (bottom) changes only appearance on the same geometry. More segments in App. F.3.

**Implementation.** We keep UniSplat's hyper-parameters. Unlike UniSplat, we use LiDAR-derived scale and shift for Pi3's affine-invariant depth during training while still supervising the predictor; inference uses the learned predictor and needs no LiDAR. Relighting provides augmented training pairs (Sec. 3.5); our feed-forward models train on 64 A100 GPUs with batch size 64 for 10 epochs.

### 4.2 Reconstruction Quality

Table 2 shows reconstruction quality comparable to the published UniSplat results; the training-time depth-scaling difference means this comparison does not isolate the effect of appearance factorization alone. SpectralSplat matches or outperforms UniSplat on all Waymo reconstruction metrics and on NVS SSIM/LPIPS; the only exception is Waymo NVS PSNR, where the gap is 0.15 dB. The nuScenes column follows UniSplat's evaluation protocol. A separate matched-inference comparison on a Waymo subset reports small PSNR reductions across cameras (App. C.2). Figure 4 shows qualitative NVS half a second ahead of the input next to the UniSplat baseline.

Table 2: **Reconstruction quality.** Baseline metrics from (Shi et al., 2026). Best in **bold**, second best underlined.

| | Waymo | | | | | | nuScenes | | |
|---|---|---|---|---|---|---|---|---|---|
| | Recon. | | | NVS | | | Recon. + NVS | | |
| Method | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| PixelSplat (Charatan et al., 2024) | — | — | — | — | — | — | 21.51 | 0.616 | 0.372 |
| MVSplat (Chen et al., 2024a) | 24.94 | 0.80 | 0.23 | 22.04 | 0.68 | 0.34 | 21.61 | 0.658 | 0.295 |
| DepthSplat (Xu et al., 2025) | 25.38 | 0.76 | 0.26 | 23.86 | 0.70 | 0.31 | — | — | — |
| Omni-Scene (Wei et al., 2025) | — | — | — | — | — | — | 24.27 | 0.736 | 0.237 |
| UniSplat (Shi et al., 2026) | 29.58 | 0.86 | 0.17 | **25.98** | **0.76** | 0.24 | 25.37 | 0.765 | 0.246 |
| SpectralSplat | **29.99** | **0.87** | **0.14** | 25.83 | **0.76** | **0.23** | **25.75** | **0.792** | **0.227** |

### 4.3 Appearance Library Rendering

Figure 5 applies the library to a real Waymo frame: the scene is reconstructed once, and each row swaps in a per-condition centroid (library code) while geometry and cached features stay fixed. The same code controls appearance across all five cameras; App. F.1 shows all nine conditions and a day-to-night blend. Library construction and the synthetic-code visualization are described in




Figure 5: **Appearance library rendering.** A real Waymo frame (top, five cameras) is reconstructed once; each row below replaces the appearance code with a per-condition centroid (library code) while geometry and cached features stay fixed. The selected code changes only the color stream. All nine conditions (*e.g.* fog, golden hour, dusk) and a continuous day-to-night blend are shown in App. F.1.

Table 3: **Cross-appearance evaluation.** Rows: content-feature source; columns: appearance code. Bold diagonal: matched; off-diagonal: swapped.

| | | Src appearance | | | Aug appearance | | |
|---|---|---|---|---|---|---|---|
| | | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| Waymo | Src features | **29.99** | **0.875** | **0.143** | 23.42 | 0.728 | 0.439 |
| | Aug features | 21.21 | 0.672 | 0.426 | **27.52** | **0.862** | **0.199** |

App. D.1. Figure 1 illustrates recoloring accumulated Gaussians using synthetically relit sequence inputs. Conditioning on the code of an arbitrary *real* scene instead is not reliable (App. E). Figure 4 and App. F.3 apply library codes to novel views on fixed geometry. Supplementary videos show sequence rendering and code interpolation on fixed scenes.

### 4.4 Cross-Appearance Evaluation

On the Waymo validation set we use the first 10 frames of each sequence as input and generate augmented counterparts with the relighting pipeline, giving paired original and augmented images with shared geometry. **Protocol.** Given $\mathbf{I}^{\text{src}}$ and $\mathbf{I}^{\text{aug}}$, we render two *matched* and two *cross* (embedding swapped) combinations of content features and appearance embeddings, each compared with the ground truth of its target appearance; the matched-to-cross PSNR gap measures the reconstruction penalty when exchanging appearance codes.

Matched configurations (diagonal of Table 3) reach 29.99 and 27.52 dB PSNR; cross configurations still yield coherent renders (23.42 and 21.21 dB). Figure 10 (appendix) shows that base colors from the two inputs are similar under the invariance objective and that swapped renders transfer appearance while preserving geometry.

**Interpreting synthetic supervision.** Training and this evaluation use pairs from the same relighting pipeline, so these numbers measure controlled disentanglement, not real illumination generalization. The shared code, $\mathcal{L}_{\text{inv}}$, and $\mathcal{L}_{\text{swap}}$ bias the model toward repeatable, multi-view-consistent factors rather than view-specific diffusion artifacts (Sec. 4.5).

### 4.5 Comparison with Optimization-Based Methods

We adapt WildGaussians to a single per-frame code shared across cameras with $d{=}64$, train it on the augmented images of the first 10 frames of the same Waymo sequences, and evaluate on those aug-

Table 4: **Augmented-target reconstruction** (Kulhanek et al., 2024). WildGaussians uses per-scene optimization; SpectralSplat uses feed-forward inference.

| Method | PSNR↑ | SSIM↑ | LPIPS↓ | Time* |
|---|---|---|---|---|
| WildGaussians | **30.42** | 0.843 | 0.296 | ∼31 min |
| SpectralSplat (ours) | 27.52 | **0.862** | **0.199** | **∼18 s** |

*Per scene, 10 frames; excludes offline training.

Table 5: **Ablation on Waymo (swapped configurations, 10-epoch budget for all rows).** Best in **bold**.

| | Src→Aug | | | Aug→Src | | |
|---|---|---|---|---|---|---|
| Model | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| Full | **23.42** | **.728** | **.439** | **21.21** | **.672** | **.426** |
| $-\mathcal{L}_{\text{swap}}$ | 14.58 | .520 | .488 | 14.33 | .497 | .498 |
| $-\mathcal{L}_{\text{inv}}$ | 23.04 | .711 | .473 | 20.56 | .627 | .500 |
| $-\mathcal{L}_{\text{base}}$ | 22.63 | .702 | .484 | 20.38 | .619 | .508 |




Figure 6: **WildGaussians vs. SpectralSplat.** Ground truth (top), WildGaussians (middle), and SpectralSplat (bottom), from the augmented-image evaluation. This is a qualitative reconstruction example; the aggregate comparison is in Table 4.

mented images (Table 4). Figure 6 shows a qualitative example from this evaluation. The reported per-scene processing times are ∼31 minutes for WildGaussians optimization and ∼18 seconds for SpectralSplat feed-forward processing; this comparison excludes our offline training and data generation. WildGaussians reaches higher PSNR (30.42 vs. 27.52), while SpectralSplat has better SSIM (0.862 vs. 0.843) and LPIPS (0.199 vs. 0.296). These metrics measure agreement with synthetic targets, not realism under actual weather.

### 4.6 Ablation Study

All ablations use the same 10-epoch budget as the full model and the cross-appearance protocol (Table 5). $\mathcal{L}_{\text{swap}}$ is the most critical term: removing it collapses cross-appearance PSNR from 23.42 to 14.58 dB (Src→Aug) and from 21.21 to 14.33 dB (Aug→Src). Removing $\mathcal{L}_{\text{inv}}$ or $\mathcal{L}_{\text{base}}$ consistently degrades all cross metrics, most visibly LPIPS (0.439→0.473/0.484). These ablations evaluate cross-appearance reconstruction, not the separate effect of temporal recoloring. The full objective is best in both directions.

## 5 Conclusion

We presented SpectralSplat, a feed-forward Gaussian Splatting method that reconstructs a driving scene once and controls its appearance through a learned library without changing the reconstructed geometry. Paired relighting supervision trains a canonical base-color output and an appearance-conditioned output, while feature caching allows accumulated Gaussians to be recolored under a selected code. The method retains comparable reconstruction quality on Waymo and nuScenes; Waymo experiments demonstrate cross-appearance rendering and reusable library conditions. The library provides scene-level appearance control, not physically complete weather simulation: arbitrary real-to-real transfer and spatially varying effects such as headlights and local shadows remain limitations (App. E). Spatially varying conditioning, validation against real captures under matched conditions, and evaluation in closed-loop robustness tests are directions for future work.

### AI USE STATEMENT

In this work, we used generative AI tools for the following tasks, in each case with the disclosure categories of the ICLR 2027 AI Policy for Authors in mind. **Writing.** We used large language models to aid and polish writing: grammar and wording suggestions, restructuring of paper text, and LaTeX formatting. All scientific claims, numbers, and tables come from the authors' own experiments and were verified by the authors. **Research ideation and execution.** We used AI assistants as a discussion and productivity aid during research execution: help with experiment-script debugging and analysis-utility code, and suggestions during ideation that were evaluated, validated, or discarded by the authors. The research questions, method design, experimental protocol, and all conclusions are the authors' own. We reviewed all AI-assisted work, checked AI-suggested text and code for correctness, and verified that AI assistance did not introduce claims or results not backed by the authors' experiments. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REFERENCES

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pp. 5470–5479, 2022.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV*, pp. 19697–19705, 2023.

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.

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*, pp. 19457–19467, 2024.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European conference on computer vision*, pp. 370–386. Springer, 2024a.

Zequn Chen, Jiezhi Yang, and Heng Yang. Pref3r: Pose-free feed-forward 3d gaussian splatting from variable-length image sequence, 2024b. URL https://arxiv.org/abs/2411.16877.

Zhifei Chen, Tianshuo Xu, Wenhang Ge, Leyi Wu, Dongyu Yan, Jing He, Luozhou Wang, Lu Zeng, Shunsi Zhang, and Ying-Cong Chen. Uni-renderer: Unifying rendering and inverse rendering via dual stream diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26504–26513, 2025.

Hiba Dahmani, Moussab Bennehar, Nathan Piasco, Luis Roldao, and Dzmitry Tsishkou. Swag: Splatting in the wild images with appearance-conditioned gaussians. In *European Conference on Computer Vision*, pp. 325–340. Springer, 2024.

Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *TPAMI*, 32 (8):1362–1376, 2009.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. MagicDrive: Street view generation with diverse 3d geometry control. In *International Conference on Learning Representations*, 2024a.

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

Quentin Herau, Nathan Piasco, Moussab Bennehar, Luis Roldao, Dzmitry Tsishkou, Cyrille Migniot, Pascal Vasseur, and Cédric Demonceaux. Moisst: Multimodal optimization of implicit scene for spatiotemporal calibration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1810–1817, 2023.

Quentin Herau, Moussab Bennehar, Arthur Moreau, Nathan Piasco, Luis Roldão, Dzmitry Tsishkou, Cyrille Migniot, Pascal Vasseur, and Cédric Demonceaux. 3dgs-calib: 3d gaussian splatting for multimodal spatiotemporal calibration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8315–8321, 2024a.

Quentin Herau, Nathan Piasco, Moussab Bennehar, Luis Roldao, Dzmitry Tsishkou, Cyrille Migniot, Pascal Vasseur, and Cédric Demonceaux. Soac: Spatio-temporal overlap-aware multi-sensor calibration using neural radiance fields. In *CVPR*, pp. 15131–15140, 2024b.

Quentin Herau, Nathan Piasco, Moussab Bennehar, Luis Roldão, Dzmitry Tsishkou, Bingbing Liu, Cyrille Migniot, Pascal Vasseur, and Cédric Demonceaux. Refining ground truth poses in autonomous driving datasets via neural rendering. *IEEE Robotics and Automation Letters*, 2026.

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

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

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

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

Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jiaqi Lin, Zhihao Li, Binxiao Huang, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Xiaofei Wu, Fenglong Song, and Wenming Yang. Decoupling appearance variations with 3d consistent features in gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Kunhao Liu, Fangneng Zhan, Yiwen Chen, Jiahui Zhang, Yingchen Yu, Abdulmotaleb El Saddik, Shijian Lu, and Eric P Xing. Stylerf: Zero-shot 3d style transfer of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8338–8348, 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7210–7219, 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*, pp. 405–421, 2020.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pp. 4104–4113, 2016.

Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, pp. 501–518, 2016.

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.

Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs, 2024. URL https://arxiv.org/abs/2408.13912.

Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *TOG*, 25(3):835–846, 2006.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021.

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 *CVPR*, pp. 2446–2454, 2020.

Stanislaw Szymanowicz, Chrisitian 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*, pp. 10208–10217, 2024.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8248–8258, 2022.

Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pp. 78–89. IEEE, 2025.

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

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NIPS*, pp. 27171–27183, 2021.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards real-world-driven world models for autonomous driving. In *European Conference on Computer Vision*, 2024.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. $\pi^3$: Permutation-equivariant visual geometry learning, 2025b. URL https://arxiv.org/abs/2507.13347.

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

Dongxu Wei, Zhiqi Li, and Peidong Liu. Omni-scene: Omni-gaussian representation for ego-centric sparse-view scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Xiangzuo Wu, Chengwei Ren, Jun Zhou, Xiu Li, and Yuan Liu. Mvinverse: Feed-forward multi-view inverse rendering in seconds. *arXiv preprint arXiv:2512.21003*, 2025.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Jiezhi Yang, Khushi Desai, Charles Packer, Harshil Bhatia, Nicholas Rhinehart, Rowan McAllister, and Joseph E Gonzalez. Carff: Conditional auto-encoded radiance field for 3d scene forecasting. In *European Conference on Computer Vision*, pp. 225–242. Springer, 2024.

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)*, 2023.

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

Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In *ICLR*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

# APPENDIX

## A ARCHITECTURE AND APPEARANCE MODULES

### A.1 BACKBONE ARCHITECTURE DETAILS

We provide additional details on the UniSplat (Shi et al., 2026) backbone that are relevant to understanding the feature dimensions used by our appearance modules.

**Feature Extraction.** Pi3 (Wang et al., 2025b) predicts dense 3D point maps from multi-camera inputs. DINOv2-S (Oquab et al., 2024) extracts 384-dimensional patch tokens, which are projected to a 2048-dimensional space via a learned linear layer and combined with Pi3's intermediate representations through a multi-scale FPN, producing per-pixel image features of dimension $D_{\text{img}}$=256.

**Voxel Branch.** The predicted 3D points are voxelized onto a regular grid; each occupied voxel aggregates coordinates, input RGB, and projected FPN features. A sparse 3D U-Net with encoder-decoder skip connections processes the voxels, incorporating temporal history fusion via ego-motion warping and sparse addition. The output per-voxel features have dimension $D_{\text{vox}}$=32, from which $K$=2 Gaussians per voxel are decoded by a geometry MLP.

**Point Branch.** For each pixel, the per-pixel FPN feature ($D_{\text{img}}$=256) is concatenated with a voxel feature ($D_{\text{vox}}$=32) sampled from the nearest occupied voxel in the fused scaffold, yielding a $(D_{\text{vox}}+D_{\text{img}})$-dimensional feature vector. A geometry MLP decodes position offset, scale, rotation quaternion, opacity, and dynamic score.

### A.2 APPEARANCE MODULE DETAILS

**Appearance Encoder.** The appearance encoder $\phi$ is a three-layer MLP ($2048 \to 256 \to 256 \to d$, $d$=64) with LayerNorm and GELU activations. The projected 2048-dimensional DINOv2 patch tokens for each camera are globally average-pooled before being passed through $\phi$. Per-camera embeddings are averaged over all cameras at the current timestamp to produce the global embedding $\mathbf{a}$.

**Color MLPs.** Each factored color MLP (point, voxel, sky) is a two-layer MLP with hidden dimension 128 and ReLU activations. Input dimensions: $D_{\text{vox}} + d = 96$ (voxel), $D_{\text{vox}} + D_{\text{img}} + d = 352$ (point), $D_{\text{img}} + d = 320$ (sky).

## B RELIGHTING PIPELINE DETAILS

**Intrinsic decomposition.** Given multi-view images and camera poses, MVInverse (Wu et al., 2025) estimates per-pixel base color $\mathbf{A}_v$ and surface normals $\mathbf{N}_v$, from which we compute a Lambertian re-rendering $\hat{\mathbf{I}}_v^{\text{mv}}$ under a target lighting direction. Shared lighting provides a cross-view prior, but this reference lacks photorealistic high-frequency detail (*e.g.* specularities, sky texture), so it is used only as a structural guide for the generative stage.

**Generative refinement.** We refine $\hat{\mathbf{I}}_v^{\text{mv}}$ with IC-Light (Zhang et al., 2025), a relighting diffusion model built on Stable Diffusion (Rombach et al., 2022). Applied independently per view, IC-Light is photorealistic but breaks multi-view consistency.

**Frequency-aware latent guidance.** The implementation uses frequency content to set a *spatial guidance mask*, rather than replacing only a low-frequency band at every step. Let $\mathbf{z}_{\text{img}}$ be the VAE encoding of the input image and $\mathbf{h} = \mathbf{z}_{\text{img}} - \mathcal{G}_\sigma(\mathbf{z}_{\text{img}})$ its high-frequency residual under Gaussian blur. We compute $e = \log(1 + \text{mean}_c\, \mathbf{h}^2)$ per spatial location and normalize it between its 5th and 95th percentiles, clipping to $[0, 1]$. The texture mask is one minus this normalized energy: it is large in smooth regions and small around detail. When a sky mask is available, its downsampled value is combined with the texture mask by a pointwise maximum, giving $\mathbf{M}_{\text{guide}}$; without an input latent, guidance is spatially uniform. Writing $\mathbf{z}_{\text{ref}} = \mathcal{E}(\hat{\mathbf{I}}^{\text{mv}})$ and $\hat{\mathbf{z}}_0$ for the estimated clean latent, the update is

$$\tilde{\mathbf{z}}_0 = (1 - w(\tau)\mathbf{M}_{\text{guide}}) \odot \hat{\mathbf{z}}_0 + w(\tau)\mathbf{M}_{\text{guide}} \odot \mathbf{z}_{\text{ref}}, \tag{6}$$

where the configured cosine schedule is $w(\tau) = \frac{\lambda}{2}[1 - \cos(\pi\tau/T)]$ and the diffusion noise timestep $\tau$ decreases during sampling. This blends the full latent spatially; it does not guarantee unchanged high-frequency content. The generation configuration specifies DPM-Solver++(2M) with Karras noise levels, 10 steps, classifier-free guidance scale 1.0, $\lambda = 0.5$, and latent blur $\sigma = 2.0$. The guided clean-latent estimate is passed directly to this solver; the implementation also supports deterministic DDIM (Song et al., 2021) by recomputing the noise direction from the guided estimate.

**Lambertian re-rendering.** To synthesize a target lighting environment defined by global direction $\boldsymbol{\ell}_w \in \mathbb{R}^3$, color $\mathbf{c} \in \mathbb{R}^3$, intensity $s$, and ambient term $k$, we compute Lambertian shading per view. The light direction is transformed to camera coordinates via $\boldsymbol{\ell}_v = R_v \boldsymbol{\ell}_w$, yielding the shading map $\mathbf{S}_v = \max(0, \mathbf{N}_v \cdot \boldsymbol{\ell}_v)$. The physics-based reference is:

$$\hat{\mathbf{I}}_v^{\text{mv}} = \mathbf{A}_v \odot (s \cdot \mathbf{c} \odot \mathbf{S}_v + k), \tag{7}$$

where $\odot$ denotes the Hadamard product.

# C Evaluation Protocols and Per-Camera Results

## C.1 Evaluation Protocols

The quantitative tables and qualitative examples serve different purposes; their targets should not be conflated.

| Result | Inputs, targets, and comparison scope |
|---|---|
| Reconstruction (Table 2) | Current views at $t$ and novel views at $t{+}1$ on Waymo; the nuScenes protocol is described in Sec. 4.1. Baseline numbers are taken from UniSplat's publication, not from our separate per-camera rerun. |
| Cross-appearance (Table 3) | Source/relit pairs from the first 10 frames of Waymo validation sequences. Each matched or swapped render is scored against the image with its target appearance; these use predicted depth scale/shift and no temporal history. |
| WildGaussians (Table 4) | Augmented-image targets for the first 10 frames per sequence. The image override replaces both inputs and reconstruction targets; Fig. 6 shows an example from this evaluation. |
| Per-camera (Table 6) | A separate 40-segment Waymo subset with both checkpoints rerun using predicted depth scale/shift and no temporal history; it is not a camera-wise decomposition of the published-baseline comparison. |
| Qualitative NVS (Figs. 4, 13) | Cameras at $t{+}5$ (0.5 s ahead), without temporal history. Library edits change color on the same reconstructed geometry; they are not scored against unobserved real-weather targets. |
| Supplementary videos | Sequence clips show current-view reconstruction without temporal history. The code-interpolation clip fixes a frame and camera poses within each scene and varies only the appearance code. Neither replaces the $t{+}1$ benchmark. |

The training-time depth-scaling difference described in Sec. 4.1 also means the headline reconstruction comparison does not isolate appearance factorization alone. The WildGaussians timing compares per-scene optimization with feed-forward processing, excluding offline model training and pair generation.

## C.2 Per-Camera Breakdown

Table 6 breaks reconstruction and NVS PSNR down by camera for SpectralSplat and the UniSplat baseline on 40 Waymo validation segments, with predicted depth scale/shift and no temporal history for either model. SpectralSplat has slightly lower PSNR on every camera in this comparison; side cameras have the highest reconstruction PSNR in this subset, while the front camera has the highest NVS PSNR.

Table 6: **Per-camera PSNR (dB) on 40 Waymo validation segments** (one frame each, no temporal history, LiDAR-free scale prediction), so absolute values are not comparable with Table 2. The gap between SpectralSplat and UniSplat is uniform across cameras (0.14–0.44 dB); no camera is disproportionately affected by the factored color pathway.

| Task | Method | Front | Front-left | Front-right | Side-left | Side-right |
|---|---|---|---|---|---|---|
| Recon. ($t$) | UniSplat | 30.13 | 29.55 | 29.21 | 31.35 | 31.46 |
| | SpectralSplat | 29.80 | 29.21 | 28.90 | 30.95 | 31.02 |
| NVS ($t$+1) | UniSplat | 27.80 | 26.06 | 25.91 | 25.61 | 25.67 |
| | SpectralSplat | 27.66 | 25.87 | 25.75 | 25.47 | 25.46 |

# D Appearance Library and Transfer

## D.1 Appearance Library Construction

The library contains per-condition means of encoded relit frames, not separately optimized condition vectors. Its construction pool is the augmented Waymo validation split, with five cameras resized to $518 \times 350$ per frame. Thus these visualizations do not establish transfer to a wholly disjoint library-construction split. The quantitative cross-appearance protocol uses codes encoded from each image pair, not the library centroids. The library rendering metadata records 240 frame codes, distributed as follows; these are frame counts, not counts of independent scenes.

| Condition | Frames | Condition | Frames | Condition | Frames |
|---|---|---|---|---|---|
| Clear day | 78 | Clear dusk | 12 | Clear golden | 22 |
| Clear night | 36 | Fog day | 22 | Rain day | 16 |
| Rain golden | 12 | Rain night | 10 | Snow day | 32 |

Generation conditions are grouped by prompt keywords (snow, rain, fog, otherwise clear) and by light elevation: below $0^\circ$ is night, $[0, 5)^\circ$ dusk, $[5, 20)^\circ$ golden, and $20^\circ$ or above day; an explicit "night" prompt takes precedence. These are coarse generation labels, not measured real-weather classes. A condition must have at least two encoded frames to retain its centroid. The generation script assigns one sampled condition per frame from lighting and prompt-fragment choices, rather than one fixed condition per sequence. The same centroid is reused across target scenes. Figure 7 visualizes the distribution of codes from relit images; it is not an evaluation of real-illumination separation. Figure 9 demonstrates code interpolation by rendering intermediate codes on fixed geometry. The supplementary interpolation videos extend this to piecewise-linear blends through all nine library conditions on three fixed scenes.

## D.2 Cross-Scene Transfer and Real-to-Real Conditioning

Figure 8 conditions four real target scenes on codes extracted from single relit exemplars of five *other* Waymo segments (one IC-Light output each, shown in the top row). The renders illustrate reuse of an exemplar code across these target scenes without changing their reconstructed geometry; the library of Sec. 3.2 averages many such exemplars per condition, rather than relying on one potentially atypical exemplar.

**Real-to-real conditioning (negative result).** Conditioning on the code of an arbitrary *real* scene does not reliably transfer its appearance. In our qualitative inspection, real-condition codes are poorly separated, whereas relit codes are better separated and produce appearance changes when substituted into a real-scene reconstruction. This observation does not isolate the cause of the real-to-real failure. A plausible explanation is that the content path already carries lighting information, allowing reconstruction without encoding it separately in the appearance code. A real reference scene can still be used through its coarse condition label, which selects the corresponding library entry.

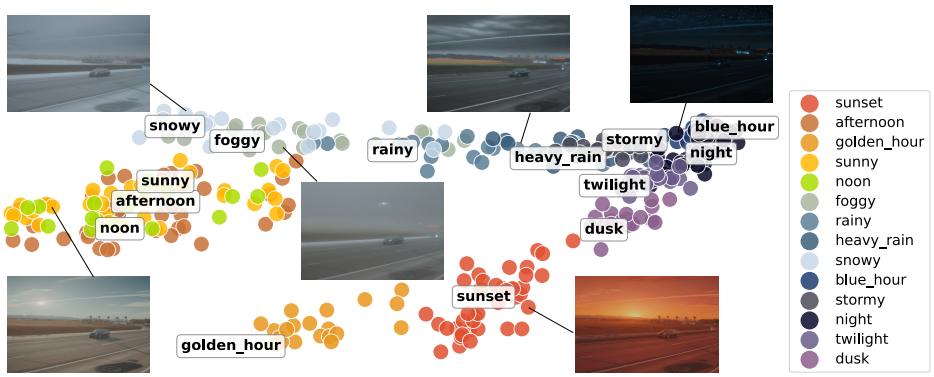



Figure 7: **t-SNE of appearance codes of relit frames**, colored by the IC-Light relighting preset used to generate them. The finer preset labels shown here are not additional library entries. The projection visualizes clustering of synthetic appearance codes; distances in this plot do not establish physical illumination similarity or guarantee smooth interpolation. Rendered code interpolation is shown in Fig. 9.



Figure 8: **Cross-scene transfer of exemplar codes.** Each row renders the same real target scene under codes extracted from relit exemplars of *other* Waymo segments (top row; IC-Light outputs, not real captures). *Column 1*: original render. *Column 2*: base color ($\mathbf{a}=\mathbf{0}$). *Columns 3–7*: the exemplar's condition (top row) transfers to every target scene while geometry is preserved, showing that a single code works across scenes; this is the mechanism the per-condition library averages over.

# E LIMITATIONS

**Library, not open-domain transfer.** The library covers the scene-level conditions produced by our relighting pipeline. Conditioning on the code of an arbitrary *real* scene, *i.e.* real-to-real transfer of that scene's specific appearance, does not work reliably (App. D.2). Content features may retain lighting information, allowing reconstruction to rely on this path rather than routing lighting through $\mathbf{a}$. We observe poorly separated real-condition codes, whereas synthetic codes are separated under paired swap supervision. The encoder extracts codes from relit exemplars without fitting a separate vector for each exemplar. Learned condition vectors could also be interpolated; our distinction is exemplar-based encoding, not an exclusive ability to blend codes. We do not claim an empirical advantage over a learned-vector alternative.

**Global code.** The shared 64-D embedding cannot represent spatially varying effects (headlights, local shadows, specular highlights, non-uniform weather). Typical failures include missing emitters under a night code, persistent lit signs under a day code, uniform casts, and softer renders; a per-camera breakdown is in App. C.2.

# F ADDITIONAL QUALITATIVE RESULTS

## F.1 APPEARANCE LIBRARY RENDERING

Figure 9 shows all nine library conditions and the day-to-night blend using the paper checkpoint.




Figure 9: **Full appearance library and continuous blending** (front camera). *Rows 1–2:* the real input and all nine library conditions (per-condition centroids of relit exemplars). *Row 3:* linear blend $(1-\alpha)\bar{\mathbf{a}}_{\text{clear day}} + \alpha\bar{\mathbf{a}}_{\text{clear night}}$; mean image brightness decreases monotonically from 0.376 to 0.119 along the sweep. Geometry is fixed throughout.

## F.2 CROSS-APPEARANCE RESULTS

Figure 10 defines the row layout for the cross-appearance evaluation: source and augmented ground truth, matched adapted renders, base colors from both inputs, and the two swapped renders.




Figure 10: **Cross-appearance results on Waymo (full).** *Rows 1–3:* source ground truth, adapted render, and base color. *Rows 4–6:* same for the augmented condition. Base colors from both inputs are similar despite their different appearances. *Rows 7–8:* swapping the appearance embedding transfers appearance while preserving geometry.

Figures 11 and 12 show the same protocol on additional Waymo segments.

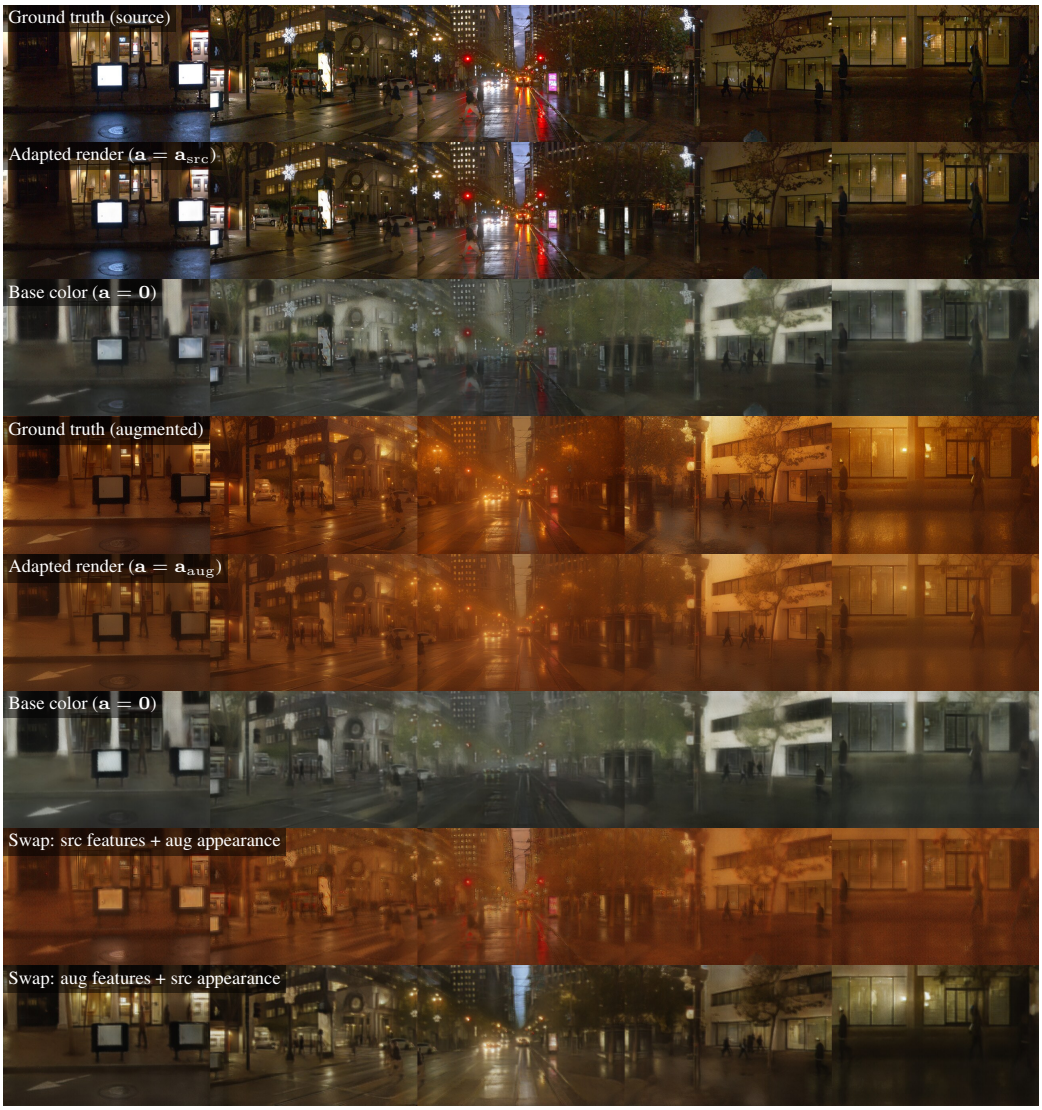



Figure 11: **Cross-appearance results on a second Waymo segment.** Rows as in Fig. 10.

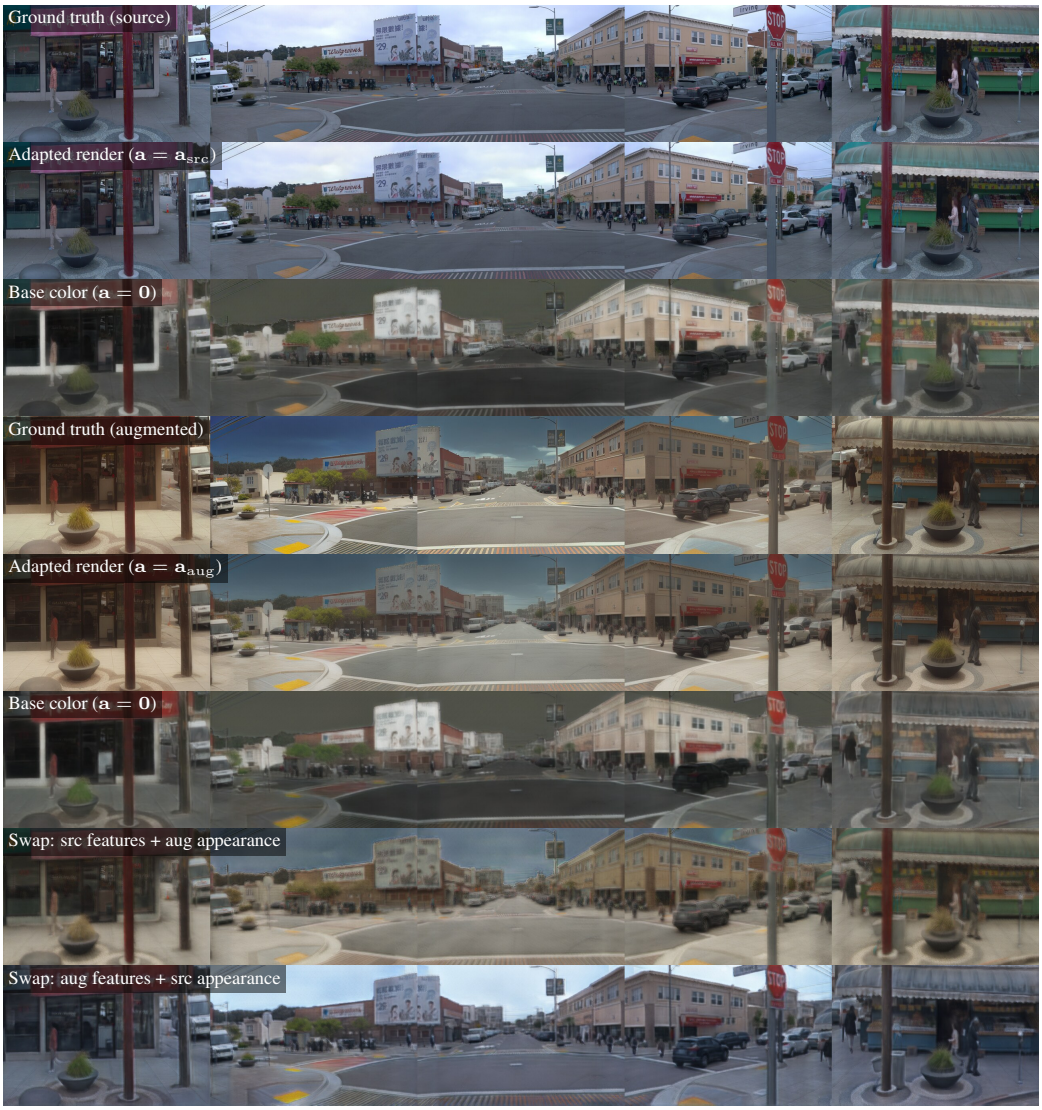



Figure 12: **Cross-appearance results on a third Waymo segment.** Rows as in Fig. 10.

### F.3 Qualitative Novel View Synthesis

Figure 13 complements Fig. 4 with two segments in which the ego vehicle is moving, rendered from the camera poses five frames (0.5 s) ahead of the input and without temporal history. We use $t{+}5$ rather than the $t{+}1$ protocol of Table 2 so that the viewpoint change is visible; at 10 Hz the next frame is only a fraction of a metre away, and many validation segments are recorded while stopped at intersections. In a scan of 20 segments, ego displacement over 0.5 s ranges from near zero to 10.1 m, with larger motion generally reducing PSNR for both methods; SpectralSplat stays within 0.4 dB of UniSplat on every segment in that scan. The displayed first segment has PSNR 20.4 vs. 20.3 dB. The second example was regenerated in a separate qualitative run with a clear-golden code; its matched-code PSNR is 18.7 vs. 18.6 dB, and these example values do not replace the earlier aggregate scan. The library code changes only the appearance of the synthesized view while geometry and dynamic objects are unchanged; library-conditioned renders are visibly softer than the own-code render, which may reflect the mismatch between these content features and the substituted code, and on some night scenes a daytime code fails altogether.

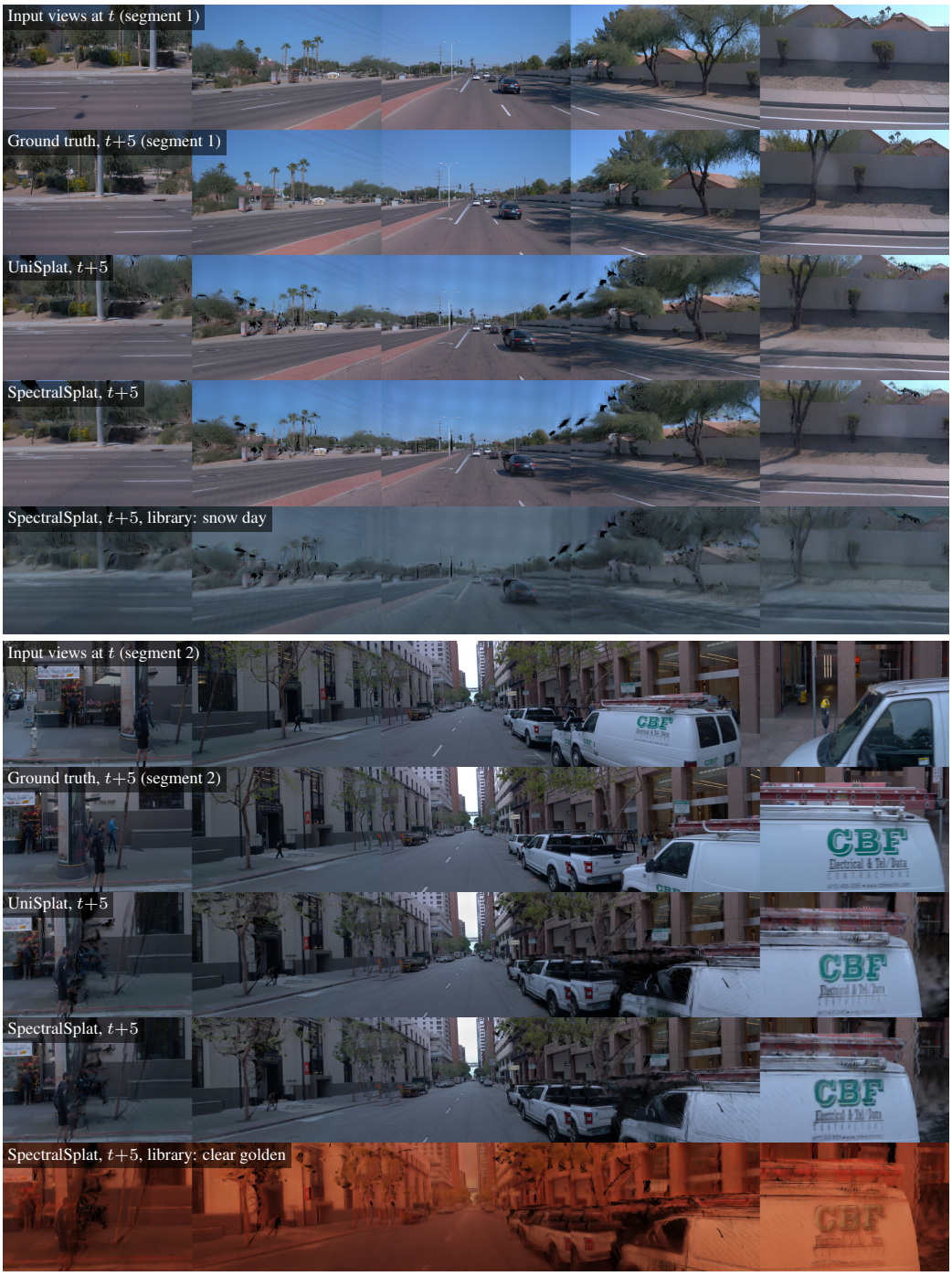



Figure 13: **Novel view synthesis 0.5 s ahead ($t+5$) with ego-motion.** Segment 1: arterial road, the ego vehicle advances 5.8 m and the surrounding cars move. Segment 2: downtown street, 3.2 m of ego-motion with strong parallax on the parked vehicles. For each segment: input views at $t$, the $t+5$ ground truth, UniSplat, SpectralSplat with its own code, and SpectralSplat with a library code on the same geometry. Regions that were not visible from the input views (image borders, behind the van) are left empty by both methods, since neither hallucinates unobserved content.