# PIH: PHYSICAL-INTRINSIC HARMONIZER FOR EDITABLE NEURAL RECONSTRUCTION

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




Figure 1: PIH transforms artifact-prone frames from neural-reconstructed scenes into coherent simulation observations while enabling virtual-asset insertion and flexible illumination control.

## ABSTRACT

Editable neural driving scenes are essential for scalable simulation, yet neural reconstructions often produce severe artifacts under novel viewpoints and virtual-asset insertion. Existing harmonizers operating in RGB space learn to correct these heterogeneous failure patterns, making extension to new editing operations data-intensive. We reformulate the task as physical-intrinsic harmonization, using imperfect depth, normals, albedo, roughness, and illumination derived from the reconstructed scene as a compositional interface. This interface enables test-time edits by composing physical factors, eliminating the need for insertion-specific paired RGB correction data. To make this formulation practical for online simulation, we propose PIH, a one-step harmonizer trained directly with latent flow matching on the predicted velocity and perceptual supervision on its decoded clean endpoint. Mixed RGB-grounded and self-rollout histories provide temporal context during training to ensure temporal consistency and mitigate exposure bias. Human preference experiments across Waymo, nuPlan, and PandaSet show that PIH is preferred to the prior method in 77.41% of comparisons while using fewer paired training frames. PIH additionally enables illumination control.

## 1 INTRODUCTION

Scalable and robust driving policies require extensive simulation before real-world deployment (Yan et al., 2023), as well as abundant data for training. A useful simulator should support novel ego motions, the insertion or removal of traffic participants, and safety-critical scenario generation while preserving photorealism (Xiao et al., 2026; Wang et al., 2025). While neural reconstruction (Zhou et al., 2025; Chen et al., 2025c) has recently shown promising results in reconstructing driving scenes, its rendering fidelity often degrades when the reconstructed scene is edited.

These edits introduce two major forms of artifacts: *novel-view artifacts* and *object-insertion artifacts* (Fig. 1-top). Recent works (Zhang et al., 2026b; Ljungbergh et al., 2025) formulate the problem as image *harmonization*, where an imperfect rendering is mapped directly to a realistic one. The



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.

causes of appearance errors are multifaceted in RGB space, where lighting variations across scenes are entangled with geometry and material appearance. As a result, the same edit operation can produce different RGB failure patterns across scenes and lighting conditions, while visually similar artifacts may originate from different physical causes. RGB harmonizers must therefore learn these dependencies implicitly from paired examples. Extending such a model to new editing operations therefore requires increasingly broad coverage of heterogeneous RGB failure patterns (Zhang et al., 2026b), often involving substantial data-engineering effort and careful recalibration of the training mixture.

We argue that this scaling burden originates from formulating harmonization in the space of RGB failures. While appearance failures are combinatorial, their underlying physical scene states are compositional. For example, a dark region may arise from albedo, shading, missing illumination, or incomplete geometry, while similar boundary artifacts may stem from distinct geometric or lighting factors. These underlying physical factors can instead be represented explicitly and composed through physically grounded scene representations, including G-buffers that encode geometry and material properties together with representations of illumination (Pharr et al., 2016). We therefore propose PIH, a one-step **P**hysical-**I**ntrinsic **H**armonizer that uses intrinsics recovered during neural reconstruction (Chen et al., 2025b;a) as an interface from imperfect scene factors to the final observation. (Fig. 1-mid). Under this formulation, test-time edits can be specified by composing physical factors rather than collecting a separate degraded/clean RGB correction set for each operation. This interface naturally supports operations such as virtual-asset insertion and illumination control while retaining the original harmonization objective.

To make PIH practical for online simulation, each frame must be synthesized in a single denoising step. Unlike RGB harmonizers, which start from an imperfect rendering that already contains much of the desired appearance, PIH must synthesize the complete RGB observation from noise while preserving the scene structure specified by its physical conditions (He et al., 2025; Zeng et al., 2024). Our probing experiment in Fig. 5 shows that dense physical intrinsics establish coarse geometry and scene composition within the first few function evaluations (NFEs), while later evaluations mainly refine appearance. This suggests that physical intrinsics already provide a strong structural anchor for direct endpoint prediction. Based on this finding, we directly train this one-step harmonizer with latent flow matching on the predicted velocity and perceptual supervision on its decoded clean endpoint. To maintain temporal consistency during causal simulation, we further mix RGB-grounded and self-rollout histories during training.

We evaluate PIH on neural-reconstructed driving scenes from the Waymo Open Dataset (WOD) (Sun et al., 2020), nuPlan (Caesar et al., 2021), and PandaSet (Xiao et al., 2021). The benchmark covers reconstruction artifacts under original and shifted camera trajectories, zero-shot virtual-asset insertion, and compositions of insertion with viewpoint extrapolation. Compared with previous RGB harmonizers, PIH produces more photorealistic outputs using fewer than one-third of the paired training frames, without requiring insertion-specific paired supervision. Moreover, this physical-intrinsic representation naturally supports flexible illumination control (Fig. 1-bottom). We also conduct a human preference study to evaluate the perceptual realism of the generated results. The study shows that PIH is preferred over the prior RGB harmonizer in 77.41% of comparisons. To summarize:

- We formulate physical-intrinsic harmonization as a compositional interface for correcting imperfect editable neural reconstructions, simplifying the data curation process.
- We instantiate this formulation with a one-step diffuser under physical-intrinsic guidance, directly trained with latent flow matching and perceptual endpoint supervision.
- Quantitative, qualitative, and human evaluations across three datasets demonstrate that PIH improves photorealism across diverse settings while providing greater controllability.

## 2 Related Work

### 2.1 Neural Reconstruction and Harmonization

Neural radiance fields and Gaussian splatting support novel-view synthesis and editable driving scenes (Mildenhall et al., 2020; Kerbl et al., 2023; Wang et al., 2023; Zhou et al., 2025; Chen et al.,

2025c), but reconstructed views can contain artifacts. Generative enhancement methods address view-dependent reconstruction quality (Liu et al., 2024; Wu et al., 2025; Fischer et al., 2025; De Lutio et al., 2026), while driving-focused work studies online enhancement and asset insertion (Zhang et al., 2026b; Ljungbergh et al., 2025; Wang et al., 2026). Image and video harmonization generally learns correction in RGB or rendered appearance space (Tsai et al., 2017; Ke et al., 2022; Lu et al., 2022; Guo et al., 2024). PIH keeps harmonization as the task but exposes imperfect physical factors as an interface for test-time insertion without insertion-specific paired RGB correction data.

### 2.2 Physical-Intrinsic and Neural Rendering

Intrinsic and inverse-rendering methods separate geometry, material, and illumination for structured image formation (Li et al., 2020; Chen et al., 2025a;b; Wang et al., 2023). DiffusionRenderer (Liang et al., 2025a), RGB↔X (Zeng et al., 2024), ZeroComp (Zhang et al., 2025), and Spotlight (Fortier-Chouinard et al., 2026) use such factors for forward or inverse synthesis, relighting, and compositing; RenderFlow (Zhang et al., 2026a) and Ouroboros (Sun et al., 2025) study single-step rendering. These works motivate the representation choice. PIH instead uses noisy, potentially inconsistent factors recovered from editable reconstructions as conditioning for harmonization. PIH uses only past frames as temporal context, enabling causal inference.

### 2.3 Efficient Generative Models

Works like (Yin et al., 2024b;a) reduce the cost of iterative diffusion sampling (Lipman et al., 2023) through distribution matching. Consistency-based objectives (Song et al., 2023; Geng et al., 2024; 2025; Zheng et al., 2026) and rollout-aware training for video (Huang et al., 2025; Cui et al., 2026) further improve efficient generation and temporal stability. Perceptual Flow Matching (PFM) (Zhao et al., 2026) supervises decoded endpoints with perceptual signals; PIH combines this endpoint supervision with latent velocity matching, dense physical-intrinsic conditioning, and mixed RGB-grounded/self-rollout histories for native one-step causal harmonization.

## 3 Methods

Our objective is to build an online harmonizer that converts artifact-prone physical-intrinsic renderings from neural reconstruction into realistic, temporally consistent frames. This requires (1) a network that accepts an editable physical-intrinsic interface together with lighting conditions, and (2) a dedicated training scheme for single-step denoising with temporal consistency. We first describe the physical-intrinsic interface and network architecture, followed by the training scheme.

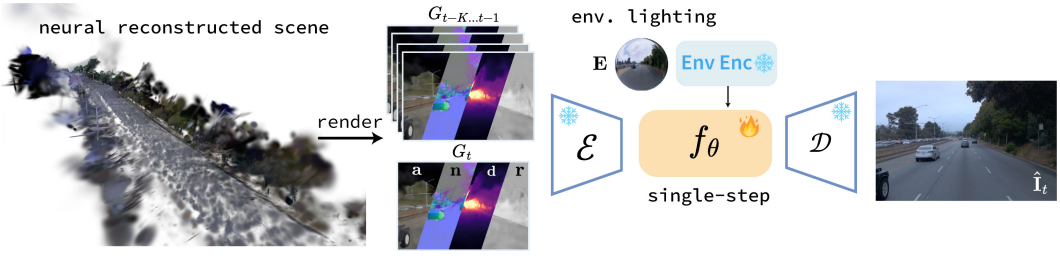



Figure 2: Overview of PIH. The model receives the current G-buffer renderings and lighting conditions from a neural-reconstructed scene, together with temporal context, and predicts the harmonized RGB latent in a single denoising step.

### 3.1 PIH: Physical-Intrinsic Harmonization

#### 3.1.1 Physical Intrinsic Representation

Following (Liang et al., 2025a; Zeng et al., 2024), we use the four G-buffers available in our reconstruction pipeline as physical-intrinsic representations. For an $F$-frame sequence of spatial resolution $H \times W$, where $F$ is the number of frames and $H, W$ are the image height and width, the

physical-intrinsic representation is:

$$\mathbf{G} = \{\mathbf{n}, \mathbf{d}, \mathbf{a}, \mathbf{r}\}, \tag{1}$$

where $\mathbf{n}$, $\mathbf{d}$, $\mathbf{a}$, and $\mathbf{r}$ denote the corresponding sequences of surface normals, normalized depth, albedo, and roughness. The buffers are pixel-aligned, editable under camera or asset changes, and may contain reconstruction noise or cross-asset inconsistencies. We encode them independently with the frozen variational autoencoder (VAE) encoder $\mathcal{E}$ and concatenate the resulting latents as dense conditioning; implementation details are given in Appendix D.1.

#### 3.1.2 Lighting Conditions

We describe illumination with a high-dynamic-range (HDR) environment map $\mathbf{E} \in \mathbb{R}^{F\times H_{\text{env}}\times W_{\text{env}}\times 3}$, where $\mathbb{R}$ denotes real-valued tensors, $H_{\text{env}}$ and $W_{\text{env}}$ are the environment-map height and width, and the final dimension indexes RGB channels. We derive tone-mapped, log-intensity, and directional panoramic encodings. A lightweight U-Net environment encoder converts these maps into cross-attention tokens, separating non-aligned illumination from the pixel-aligned G-buffer; the exact encodings and adaptation layers are deferred to Appendix D.2.

#### 3.1.3 Model Architecture

**Network Design.** We use $\text{Concat}_C$ for channel-wise concatenation, where $C$ denotes the channel axis; $[\cdot]_t$ selects the output at the current frame, $\mathcal{D}$ is the frozen VAE decoder, a hat denotes a model prediction, and subscript $0$ denotes the clean endpoint. Our generative backbone $f_\theta$, with trainable parameters $\theta$, is initialized from Cosmos-Predict2.5 (Ali et al., 2025) as shown in Fig. 2. We retain its flow-based, bidirectional diffusion transformer (DiT) architecture (Peebles & Xie, 2023) and modify only the input projection and conditioning pathway. At frame $t$, the model receives the current noisy RGB latent $\mathbf{z}_{t,\tau}$, the current G-buffer latent $\mathbf{g}_t$, and the preceding $K = 4$ frames as temporal context. For each history frame $s < t$, with $s$ indexing a history frame, we pair the corresponding G-buffer $\mathbf{g}_s$ with a clean RGB history latent $\mathbf{h}_s$. Here $\mathbf{g}_s$ is the channel-concatenated VAE latent of the physical buffers at frame $s$, and $\mathbf{h}_s$ is a clean RGB latent. The scalar $\tau \in [0, 1]$ is the flow/noise coordinate, and $\sigma_\tau$ is the schedule's noise scale at that coordinate. Depending on the rollout setting, $\mathbf{h}_s$ is either the encoded 3D Gaussian Splatting (3DGS) RGB reference $\mathbf{z}_s^{\text{rgb}}$ or a previously generated prediction $\hat{\mathbf{z}}_{s,0}$. The current position instead pairs $\mathbf{g}_t$ with the noisy latent:

$$\mathbf{x}_s = \begin{cases} \text{Concat}_C[\mathbf{h}_s, \mathbf{g}_s], & s < t, \\ \text{Concat}_C[\mathbf{z}_{t,\tau}, \mathbf{g}_t], & s = t, \end{cases} \qquad \mathbf{X}_{t,\tau} = [\mathbf{x}_{t-K}, \ldots, \mathbf{x}_t]. \tag{2}$$

Here $\mathbf{X}_{t,\tau}$ is the temporal input window formed from the per-frame inputs $\mathbf{x}_s$. We expand the patch embedding for the concatenated latent channels and apply bidirectional attention over the window. Although attention is bidirectional within the input window, the window contains only the current and preceding frames, preserving causal inference. Conditioned on lighting tokens $\mathbf{c}_{\text{light}}$, the DiT predicts velocities for all positions; we supervise and retain only the current output:

$$\hat{\mathbf{v}}_{t,\tau} = \left[f_\theta(\mathbf{X}_{t,\tau}, \tau; \mathbf{c}_{\text{light}})\right]_t, \quad \hat{\mathbf{z}}_{t,0} = \mathbf{z}_{t,\tau} - \sigma_\tau \hat{\mathbf{v}}_{t,\tau}, \quad \hat{\mathbf{I}}_t = \mathcal{D}(\hat{\mathbf{z}}_{t,0}). \tag{3}$$

Here $[\cdot]_t$ selects the output at the current frame, and $\mathbf{c}_{\text{light}}$ denotes the encoded lighting tokens. After each prediction, $\hat{\mathbf{z}}_{t,0}$ is appended to the history queue for self-rollout; RGB-grounded rollout instead refreshes history with rendered 3DGS RGB frames. We apply the VAE independently per frame to preserve alignment between RGB latents and physical conditions (Appendix D.2).

### 3.2 Training

**Two-stage training.** We use a two-stage training procedure, while keeping $\mathcal{E}$ and $\mathcal{D}$ frozen throughout. Stage I learns the new physical-intrinsic conditioning layers from paired images with the latent flow matching loss $\mathcal{L}_{\text{FM}}$ in equation 4; only the cross-attention and patch-embedding layers are updated. Stage II directly trains the one-step predictor and extends image-only harmonization to temporally consistent video harmonization. It jointly trains all parameters with the hybrid latent–perceptual objective and mixes image, RGB-grounded video, and self-rollout video batches. Only the current position is supervised, exposing the model to both appearance-anchored references and autoregressive errors; no explicit teacher trajectory is used. The noise level in Stage I is sampled from the uniform distribution $\mathcal{U}(0, 1)$, whereas in Stage II we sample $\tau \sim \{0.99, 0.70, 0.40, 0.15\}$. The schedule maps each $\tau$ to the noise scale $\sigma_\tau$. During inference, we fix $\tau$ to $0.99$.

**Latent flow matching.** Following the flow matching objective, for the current target frame we let $\mathbf{z}_{t,0} = \mathcal{E}(\mathbf{I}_t^{\mathrm{gt}})$, where $\mathbf{I}_t^{\mathrm{gt}}$ is the ground-truth RGB frame, and draw Gaussian noise $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{\mathrm{id}})$. We sample an intermediate latent $\mathbf{z}_{t,\tau} = (1-\sigma_\tau)\mathbf{z}_{t,0} + \sigma_\tau \boldsymbol{\epsilon}_t$ and set the target velocity to $\mathbf{v}_{t,\tau} = \boldsymbol{\epsilon}_t - \mathbf{z}_{t,0}$. The network $f_\theta$ predicts the window-wise velocity field from the physical-intrinsic sequence and lighting condition, while $\hat{\mathbf{v}}_{t,\tau}$ denotes its output at the supervised current position. We optimize:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,\tau,\boldsymbol{\epsilon}_t}\left[\left\| f_\theta(\mathbf{X}_{t,\tau}, \tau; \mathbf{c}_{\mathrm{light}})_t - \mathbf{v}_{t,\tau} \right\|_2^2\right]. \tag{4}$$

This objective preserves the pretrained flow transport across sampled noise levels.

**Perceptual endpoint supervision.** Following the endpoint formulation of PFM, we directly supervise the clean prediction recovered from the model's predicted velocity:

$$\hat{\mathbf{z}}_{t,0} = \mathbf{z}_{t,\tau} - \sigma_\tau \hat{\mathbf{v}}_{t,\tau}, \qquad \hat{\mathbf{I}}_t = \mathcal{D}(\hat{\mathbf{z}}_{t,0}). \tag{5}$$

We combine LPIPS (Zhang et al., 2018) for local appearance with P-DINO (Ma et al., 2026) for global structure, denoting their losses by $\mathcal{L}_{\mathrm{LPIPS}}$ and $\mathcal{L}_{\text{P-DINO}}$, respectively. The exact patch-feature distance is given in Appendix D.3. Let $\mathcal{L}_{\mathrm{pre}}$ denote the combined decoded endpoint loss, let $\mathcal{L}$ denote the complete second-stage objective, and let $\lambda_{\mathrm{fm}}$ and $\lambda_{\mathrm{pre}}$ be scalar weights for the latent flow-matching and perceptual endpoint terms, respectively. The two losses are:

$$\mathcal{L}_{\mathrm{pre}} = \mathcal{L}_{\mathrm{LPIPS}} + \mathcal{L}_{\text{P-DINO}}, \qquad \mathcal{L} = \lambda_{\mathrm{fm}}\mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{pre}}\mathcal{L}_{\mathrm{pre}}. \tag{6}$$

The perceptual networks remain frozen; gradients from the endpoint losses pass through the decoded clean prediction to $f_\theta$. PIH combines latent flow matching on the predicted velocity with perceptual supervision on its decoded clean endpoint. The same predicted velocity supplies both terms: $\mathcal{L}_{\mathrm{FM}}$ supervises the latent velocity, while $\mathcal{L}_{\mathrm{pre}}$ supervises its decoded clean endpoint. The resulting hybrid objective keeps a single latent flow field and adds perceptual endpoint supervision, without introducing an explicit teacher or trajectory-distillation stage.

### 3.3 Inference

At inference time, editing is performed in the physical-intrinsic space. For virtual asset insertion, after obtaining the assets' model, we put it into the scene by assigning its coordinates. During rendering, we take the assets' RGB as albedo; the normal is estimated from the assets' point cloud, and roughness is from the average of all vehicles. Depth is naturally supported under 3DGS rasterization. Details can be found in Sec. F.2.

## 4 Experiments

In this section, we will first describe the data and evaluation protocol, then report qualitative and quantitative results, followed by preference evaluation and ablations.

### 4.1 Dataset

**Scene split and reconstruction.** We construct our dataset from the Waymo Open Dataset (WOD), PandaSet, and nuPlan. For training, we randomly select 200 scenes from WOD only. For testing, we use 55 disjoint WOD scenes, 39 PandaSet scenes, and 6 nuPlan-mini scenes. WOD contains 200 frames per scene, while PandaSet and nuPlan contain 80 and 300 frames per scene, respectively, resulting in more than 150k frames across sequences of different lengths and locations. We reconstruct every scene with InvRGB+L (Chen et al., 2025b) and estimate the environment lighting of each scene using LuxDiT (Liang et al., 2025b). To reproduce the imperfect geometry and appearance encountered in practical neural simulators, each 3DGS scene is optimized using only the front camera and deliberately stopped after 5,000 iterations.

**Training pairs.** Each training example pairs physical-intrinsic buffers rendered from the under-trained reconstruction with the corresponding captured RGB frame. From every training scene, we retain all front-camera frames and the final 50 frames from each of the two side cameras. We additionally construct counterfactual pairs from the front-camera images by using DiffusionRenderer to estimate pseudo-G-buffers while retaining the captured image as the RGB target. These pseudo-intrinsic pairs broaden the material–illumination combinations seen during training and strengthen relighting supervision. After filtering invalid samples, the complete training set contains fewer than 100k paired frames, substantially fewer than the 350K training pairs used by DiffusionHarmonizer.

### 4.2 Evaluation Protocol

**Evaluation settings.** We organize the evaluation as a $2 \times 2$ design over two independent factors: camera trajectory (*original* or *novel*) and scene composition (*without* or *with* virtual-asset insertion). For *novel* trajectories, we randomly apply lateral shifts of 3m for different scenes. Asset insertion is not represented by paired examples during training. At evaluation time, we insert a vehicle by composing its physical attributes with the reconstructed scene buffers, e.g., modified albedo. In summary, the four settings are 1) original trajectory, 2) novel trajectory, 3) original trajectory with insertion, and 4) novel trajectory with insertion. All the experiments under Sec. 4.3 and Sec. 4.4 are conducted on Stage II models with single-step inference.

**Virtual assets.** We construct virtual assets from WOD. Following the asset-construction practice of R3D2 (Ljungbergh et al., 2025), we obtain virtual objects by applying SAM3D (Chen et al., 2026) to the WOD-OA (Shen et al., 2023). We retain only validated assets and evaluate insertion on the subset of test scenes for which a valid asset is available. For WOD, we insert the assets directly; for nuPlan and PandaSet, we transfer WOD assets by selecting the asset with the smallest bounding-box difference.

### 4.3 Qualitative Results

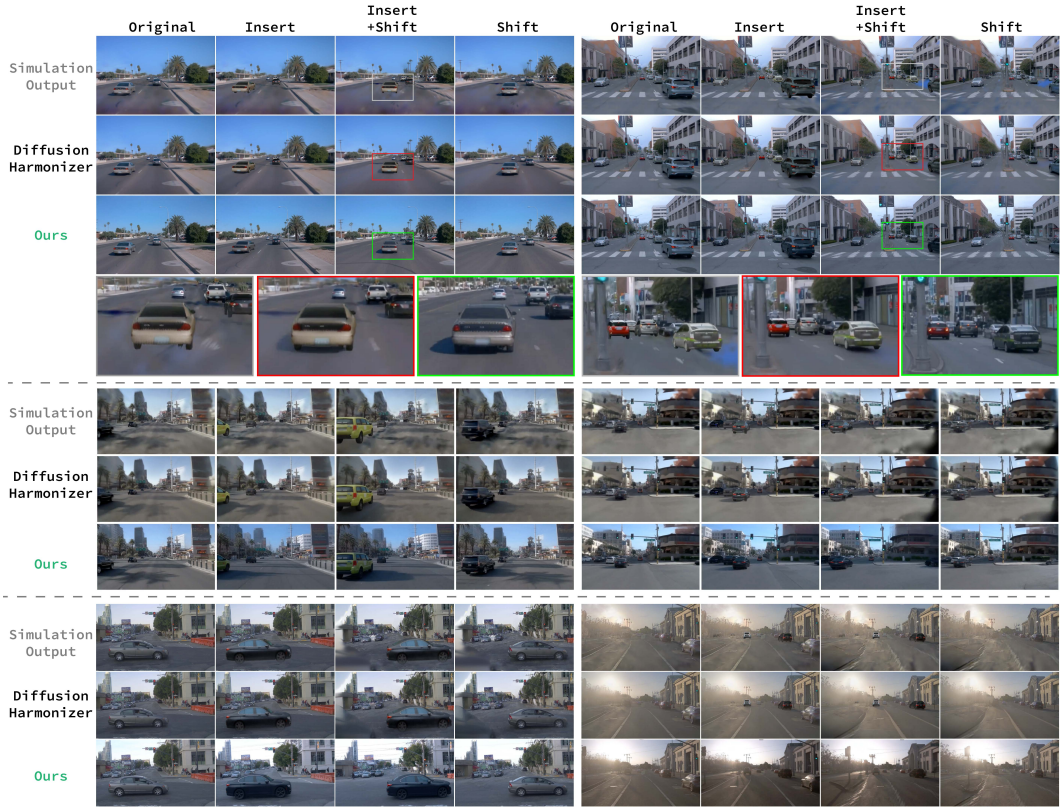



Figure 3: Qualitative harmonization results. From top to bottom: WOD, nuPlan, and PandaSet.

**Harmonization.** We visualize representative harmonization results on WOD, nuPlan, and PandaSet in Fig. 3 under the four settings described above. DiffusionHarmonizer removes some reconstruction artifacts and corrects appearance in easier cases, but degrades under more challenging conditions. In contrast, PIH better preserves simulator-defined structure while correcting appearance.

**Illumination Control.** The physical-intrinsic representation naturally enables illumination control. Fig. 4 shows five lighting conditions across five scenes. Quantitative evaluation is not available because ground-truth images under these counterfactual lighting conditions do not exist.

### 4.4 Quantitative Results

**WOD.** Tab. 1 evaluates the harmonization results without virtual asset insertion. PIH obtains the best value for every reported metric in both the original and novel trajectory columns, while the smaller margins on temporal and structural metrics under novel trajectories indicate that viewpoint extrapolation remains constrained by the underlying reconstruction and rendered physical intrinsics. For zero-shot virtual-vehicle insertion, Tab. 2 evaluates the harmonization results with virtual asset insertion. PIH again obtains the best value for every reported metric in both trajectory settings.

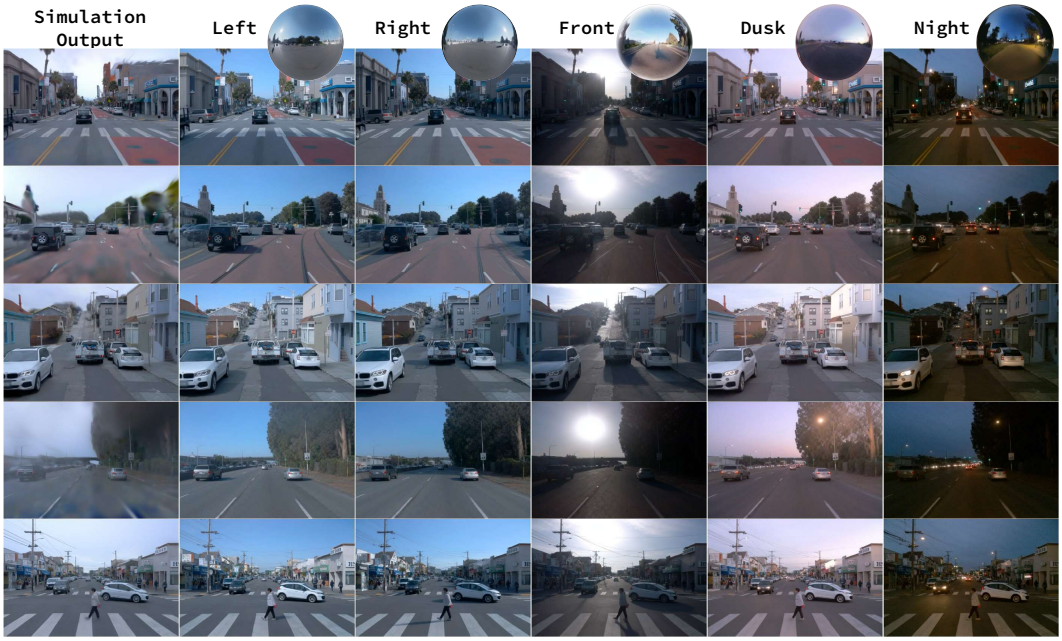



Figure 4: Qualitative illumination-control results across five scenes and five lighting conditions.

Table 1: Comparison of WOD harmonization results *without* virtual asset insertion. Bold and underlined values denote the best and second-best results.

| Method | Original Trajectory | | | | Novel Trajectory | | |
|---|---|---|---|---|---|---|---|
| | LPIPS↓ | FID↓ | FVD↓ | DINO-S↓ | FID↓ | FVD↓ | DINO-S↓ |
| SDEdit (SD 3) (Meng et al., 2022) | 0.4008 | 28.99 | 768.52 | 2.8505 | 49.51 | 1475.14 | 6.8635 |
| InstructPix2Pix (Brooks et al., 2023) | 0.4177 | 47.35 | 668.44 | 3.2117 | 86.27 | 1191.99 | 7.7316 |
| VACE-V2V (Jiang et al., 2025) | 0.4351 | 54.83 | 452.11 | 2.7672 | 104.77 | 1017.55 | 6.9973 |
| DiffusionHarmonizer (Zhang et al., 2026b) | 0.3535 | <u>19.87</u> | 261.10 | <u>1.9578</u> | 44.61 | 536.16 | 6.8946 |
| Difix3D (Wu et al., 2025) | <u>0.2910</u> | 20.32 | <u>232.82</u> | 2.0696 | <u>38.66</u> | <u>528.81</u> | 6.8596 |
| Ours | **0.2578** | **9.12** | **156.85** | **1.4459** | **26.34** | **503.07** | **6.5254** |

Table 2: Comparison of WOD harmonization results *with* virtual asset insertion.

| Method | Original Trajectory | | | | Novel Trajectory | | |
|---|---|---|---|---|---|---|---|
| | LPIPS↓ | FID↓ | FVD↓ | DINO-S↓ | FID↓ | FVD↓ | DINO-S↓ |
| SDEdit (SD 3) (Meng et al., 2022) | 0.4104 | 36.44 | 844.01 | 3.0947 | 54.15 | 1582.75 | 6.8800 |
| InstructPix2Pix (Brooks et al., 2023) | 0.4309 | 56.57 | 765.53 | 3.6115 | 89.97 | 1354.39 | 7.8406 |
| VACE-V2V (Jiang et al., 2025) | 0.4496 | 65.05 | 538.97 | 2.9653 | 110.47 | 1227.00 | 6.9609 |
| DiffusionHarmonizer (Zhang et al., 2026b) | <u>0.3696</u> | <u>27.65</u> | <u>326.93</u> | <u>2.2380</u> | <u>49.51</u> | <u>672.43</u> | <u>6.8920</u> |
| Ours | **0.3166** | **18.77** | **241.99** | **2.0322** | **31.32** | **459.68** | **6.5524** |

**nuPlan & PandaSet.** Tab. 3 evaluates harmonization results under nuPlan and PandaSet without virtual-asset insertion. PIH obtains the lowest FID on both datasets and trajectories, and is competitive on the remaining measures. Because camera styles differ across datasets, visually plausible enhancements may not always translate into uniformly improved reference-based metrics. For virtual-asset insertion, the inserted vehicle has no paired captured counterpart; we therefore report qualitative and VLM/human preference results instead of reference-based metrics.

**VLM and Human Evaluation.** We conduct a preference evaluation with a VLM and use the same presentation protocol for human raters in Tab. 8. For each held-out scene, we sample one frame and its paired 3DGS-rendered RGB reference, randomly permute the candidate order, and ask the evaluator to select the visually best enhancement while preserving the reference content. We use Qwen3-VL-235B (Bai et al., 2025) and report the mean win rate and standard deviation over five random seeds. The VLM and human rows provide complementary preference evidence for quality. Human preference results are reported as the mean $\pm$ sample standard deviation of participant-level win rates over all completed responses, while VLM results are computed analogously over five random seeds.

Table 3: Comparison of harmonization on nuPlan and PandaSet *without* virtual asset insertion.

| Dataset | Method | Original Trajectory | | | | Novel Trajectory | | |
|---|---|---|---|---|---|---|---|---|
| | | LPIPS↓ | FID↓ | FVD↓ | DINO-S↓ | FID↓ | FVD↓ | DINO-S↓ |
| nuPlan | SDEdit (SD 3) (Meng et al., 2022) | 0.4890 | 91.56 | 1931.31 | 2.7714 | 127.81 | 2558.91 | 5.6010 |
| | InstructPix2Pix (Brooks et al., 2023) | 0.4883 | 95.31 | 1064.84 | 2.6884 | 150.27 | 2227.89 | 5.8144 |
| | VACE-V2V (Jiang et al., 2025) | 0.5158 | 167.69 | 1167.08 | 3.1059 | 248.02 | 2093.91 | 5.9151 |
| | DiffusionHarmonizer (Zhang et al., 2026b) | 0.4166 | 62.92 | 508.16 | 1.8674 | 107.37 | 871.12 | **5.4045** |
| | Difix3D (Wu et al., 2025) | **0.3547** | 51.53 | 376.03 | 2.0039 | 89.23 | **614.08** | 5.4479 |
| | Ours | 0.3670 | **44.66** | **371.84** | **1.7117** | **82.53** | 824.59 | 5.4240 |
| PandaSet | SDEdit (SD 3) (Meng et al., 2022) | 0.4175 | 36.77 | 723.02 | 2.1771 | 53.23 | 1249.57 | 6.6957 |
| | InstructPix2Pix (Brooks et al., 2023) | 0.4452 | 32.49 | 536.20 | 1.9966 | 61.18 | 1084.74 | 6.9371 |
| | VACE-V2V (Jiang et al., 2025) | 0.4967 | 105.99 | 1372.08 | 2.9699 | 156.55 | 1888.74 | 7.1104 |
| | DiffusionHarmonizer (Zhang et al., 2026b) | 0.3350 | 20.91 | 387.36 | **1.3614** | 41.92 | 792.91 | **6.5038** |
| | Difix3D (Wu et al., 2025) | **0.3056** | 21.62 | **302.44** | 1.5305 | 40.29 | **542.85** | 6.8138 |
| | Ours | 0.3190 | **18.74** | 419.54 | 1.3719 | **37.31** | 705.82 | 6.7598 |

## 4.5 Ablation Study

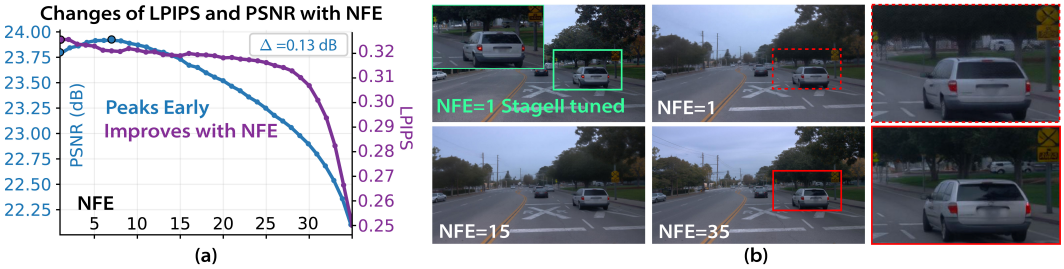



Figure 5: NFE probe of predicted clean endpoints under dense physical-intrinsic guidance. (a) shows NFE–fidelity metrics, and (b) visualizes images produced with different NFEs.

**Why One-Step Endpoint Learning?** We start by probing the NFE on our Stage I model. As shown in Fig. 5, the predicted clean image approaches peak pixel-level fidelity within the first ten NFEs, only 0.13 dB above the first prediction, while perceptual fidelity continues to improve at later steps. This pattern motivates the perceptual endpoint branch of the hybrid objective: dense physical intrinsics provide an early structural anchor, while later integration steps mainly refine appearance. We therefore train the one-step model directly with the objective in equation 6. As shown in Fig. 5(b) top left, the tuned one-step model can produce a visually faithful prediction.

**Which Perceptual Supervision Is Needed?** Given the need for perceptual endpoint supervision, we next ablate its individual components in Tab. 7. LPIPS alone provides better local perceptual similarity than P-DINO alone, whereas P-DINO alone gives the lowest FVD and DINO-S. Combining them achieves the best LPIPS and FID while remaining close to P-DINO on structure, indicating that local appearance and global semantic supervision are complementary rather than interchangeable. We also ablate the perceptual weight $\lambda_{\text{pre}}$ in Tab. 5.

**Do We Still Need a Regression Objective?** Given that $\mathcal{L}_{\text{pre}}$ provides perceptual supervision, do we still need a regression objective? If so, which regression objective is most effective? We ablate the regression objective in Tab. 6. Latent flow matching gives the best LPIPS and FVD. Moving flow matching or direct $x_0$ regression (same def. in JiT (Li & He, 2026)) to pixel space substantially worsens FVD, suggesting that preserving the pretrained latent transport is particularly important, which has been overlooked in prior works (Sun et al., 2025; Garcia et al., 2025; Zhang et al., 2026b). On the other hand, if the objective is in pixel space only, $x_0$ regression is better than flow matching, which is consistent with the findings of JiT (Li & He, 2026).

**RGB-grounding vs. Self-rollout.** LuxDiT provides an estimated illumination condition rather than measured scene lighting and can therefore retain a scene-dependent appearance bias, e.g., from camera style. Tab. 4 tests whether an RGB history reference can anchor this ambiguity. On the original trajectory, it improves performance. On the novel trajectory, grounding also improves FID and DINO-S, while self-rollout yields a lower FVD. Thus, RGB grounding primarily stabilizes scene appearance and structure, whereas a rendered reference can become temporally misaligned under a shifted camera path. This sensitivity also helps explain why cross-dataset LPIPS/FVD can remain relatively high.

Table 4: Different history sources on Waymo without insertion.

| History source | Original Trajectory | | | | Novel Trajectory | | |
|---|---|---|---|---|---|---|---|
| | LPIPS↓ | FID↓ | FVD↓ | DINO-S↓ | FID↓ | FVD↓ | DINO-S↓ |
| Self-rollout | 0.3075 | 15.27 | 210.43 | 1.8803 | 28.42 | **365.43** | 6.5920 |
| 3DGS RGB-grounded | **0.2578** | **9.12** | **156.85** | **1.4459** | **26.34** | 503.07 | **6.5254** |

Table 5: Ablation on $\lambda_{\text{pre}}$.

| $\lambda_{\text{pre}}$ | LPIPS↓ | FID↓ | FVD↓ | DINO-S↓ |
|---|---|---|---|---|
| 1.0 | 0.3103 | 14.82 | 215.16 | 1.8910 |
| 0.5 | **0.3044** | **14.63** | **215.06** | **1.8766** |
| 0.1 | 0.3062 | 16.31 | 270.16 | 1.8908 |

Table 6: Regression-objective ablation.

| Objective | Space | LPIPS↓ | FID↓ | FVD↓ | DINO-S↓ |
|---|---|---|---|---|---|
| Flow matching | Latent | **0.3104** | 14.82 | **215.16** | 1.8910 |
| Flow matching | Pixel | 0.3161 | 15.61 | 259.70 | **1.8606** |
| $x_0$ regression | Pixel | 0.3137 | 15.15 | 258.24 | 1.8761 |
| None (only $\mathcal{L}_{\text{pre}}$) | — | 0.3134 | **14.79** | 231.78 | 1.8790 |

Table 7: Ablation over $\mathcal{L}_{\text{pre}}$.

| $\mathcal{L}_{\text{LPIPS}}$ | $\mathcal{L}_{\text{P-DINO}}$ | LPIPS↓ | FID↓ | FVD↓ | DINO-S↓ |
|---|---|---|---|---|---|
| ✓ | | 0.3142 | 21.95 | 235.56 | 2.1770 |
| | ✓ | 0.3229 | 17.94 | **195.10** | **1.8669** |
| ✓ | ✓ | **0.3104** | **14.82** | 215.16 | 1.8910 |

Table 8: VLM/Human preference evaluation.

| Set | Evaluator | Ours | DiffusionHarmonizer |
|---|---|---|---|
| Waymo | VLM | 77.45%±1.17% | 22.55%±1.17% |
| | Human | 79.43%±16.18% | 20.57%±16.18% |
| nuPlan& PandaSet | VLM | 87.56%±1.22% | 12.44%±1.22% |
| | Human | 75.56%±24.78% | 24.44%±24.78% |
| All | VLM | 82.39%±0.91% | 17.61%±0.91% |
| | Human | 77.41%±20.97% | 22.59%±20.97% |




Figure 6: Design analysis under imperfect physical intrinsics: (a) direct endpoint fine-tuning, (b) albedo initialization, (c) illumination leakage, and (d) causal adaptation with Diffusion Forcing accumulates exposure bias over long horizons.

**Design Choices under Imperfect Physical Intrinsics.** We test several alternative designs inspired by single-step neural rendering. Directly applying the endpoint fine-tuning objective of (Sun et al., 2025) produces blurry predictions, likely because imperfect G-buffers admit multiple plausible RGB appearances and regression favors their average (Fig. 6(a)). Initializing the ODE from albedo following (Zhang et al., 2026a) also reduces robustness to lighting changes (Fig. 6(b)), as imperfect albedo may contain baked-in shading and illumination (Fig. 6(c)). Finally, adapting Diffusion Forcing (Chen et al., 2024) for causal inference accumulates exposure bias over long horizons (Fig. 6(d)). Because these variants are consistently inferior visually, we report them as qualitative design analyses rather than quantitative baselines.

## 5 Conclusion

PIH reframes neural-reconstruction harmonization with imperfect physical intrinsics and direct one-step endpoint learning. Across three driving datasets, PIH outperforms prior methods on quantitative metrics and is preferred to DiffusionHarmonizer in both VLM and human evaluations. The physical-intrinsic interface allows for compositional edits and illumination control, while the one-step endpoint learning enables faster simulation with a single NFE.

**Limitations and future work.** The running time is approximately 0.6 seconds per frame on a single H100 GPU due to additional computation on the VAE and environment light encoder. Future work may explore how to improve the inference efficiency for online deployment.

### Artificial intelligence (AI) use statement

During manuscript preparation, we used generative AI tools to assist with literature organization and with drafting and editing the Related Work, Conclusion, claim-qualification, and policy statements. We also used generative AI tools to generate data-analysis and visualization code (e.g., matplotlib scripts). We did not use generative AI tools to generate data, run experiments, or replace author verification of the technical claims and reported results. The authors reviewed and verified all AI-assisted text and remain responsible for the final manuscript, claims, data, code, and artifacts.

### Ethics statement

This work uses public driving datasets and includes a human preference evaluation of rendered outputs. We report aggregate preference rates and do not include participant identifiers or participant-level responses in the manuscript. Our institution has reviewed and approved this survey. To comply with the double blind review policy, we will not include this during reviewing process but will provided upon request.

### Reproducibility statement

The paper specifies the model inputs and physical-intrinsic buffers in Sec. 3, the scene splits and reconstruction procedure in Sec. 4.1, the training schedule in Appendix D.4, the evaluation protocol in Sec. 4.2, the metrics in Appendix D, and the preference protocol in Appendix E.3. Upon acceptance, we will release the complete code.

## References

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuPlan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Hongze Chen, Zehong Lin, and Jun Zhang. Gi-gs: Global illumination decomposition on gaussian splatting for inverse rendering. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 75713–75742, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/bc97207e3979d1cc23109db0be0e8ed2-Paper-Conference.pdf.

Xiaoxue Chen, Bhargav Chandaka, Chih-Hao Lin, Ya-Qin Zhang, David Forsyth, Hao Zhao, and Shenlong Wang. Invrgb+ l: Inverse rendering of complex scenes with unified color and lidar reflectance modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27176–27186, 2025b.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 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 *The Thirteenth International Conference on Learning Representations*, 2025c.

Jiaxing Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. In *International Conference on Learning Representations*, volume 2026, pp. 85802–85822, 2026.

Riccardo De Lutio, Tobias Fischer, Yen-Yu Chang, Yuxuan Zhang, Zhangjie Wu, Xuanchi Ren, Tianchang Shen, Katarína Tóthová, Zan Gojcic, and Haithem Turki. Artifixer: Enhancing and extending 3d reconstruction with auto-regressive diffusion models. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2026.

Tobias Fischer, Samuel Rota Bulò, Yung-Hsu Yang, Nikhil Keetha, Lorenzo Porzi, Norman Müller, Katja Schwarz, Jonathon Luiten, Marc Pollefeys, and Peter Kontschieder. FlowR: Flowing from sparse to dense 3d reconstructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Frédéric Fortier-Chouinard, Zitian Zhang, Louis-Etienne Messier, Mathieu Garon, Anand Bhattad, and Jean-François Lalonde. Spotlight: Shadow-guided object relighting via diffusion. In *2026 International Conference on 3D Vision (3DV)*, pp. 1185–1195. IEEE, 2026.

Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan De Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 753–762, 2025.

Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J. Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 75460–75482. Curran Associates, Inc., 2025. doi: 10.52202/085713-2534. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6d13e085b79d454da5910e4ca82a3d9d-Paper-Conference.pdf.

Zonghui Guo, Xinyu Han, Jie Zhang, Shiguang Shan, and Haiyong Zheng. Video harmonization with triplet spatio-temporal variation patterns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19177–19186, 2024.

Kai He, Ruofan Liang, Jacob Munkberg, Jon Hasselgren, Nandita Vijaykumar, Alexander Keller, Sanja Fidler, Igor Gilitschenski, Zan Gojcic, and Zian Wang. Unirelight: Learning joint decomposition and synthesis for video relighting. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 108435–108459. Curran Associates, Inc., 2025. doi: 10.52202/085713-3615. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/9bb159d886a92a8c3375dc6cbb383333-Paper-Conference.pdf.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025.

Zhanghan Ke, Chunyi Sun, Lei Zhu, Ke Xu, and Rynson WH Lau. Harmonizer: Learning to perform white-box image and video harmonization. In *European conference on computer vision*, pp. 690–706, 2022.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2472–2481. IEEE, 2020.

Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Zhi-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, and Zian Wang. Diffusionrenderer: Neural inverse and forward rendering with video diffusion models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Ruofan Liang, Kai He, Zan Gojcic, Igor Gilitschenski, Sanja Fidler, Nandita Vijaykumar, and Zian Wang. Luxdit: Lighting estimation with video diffusion transformer. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 5015–5046. Curran Associates, Inc., 2025b. doi: 10.52202/085713-0178. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/07b02efbb28aeb2d339d2b7c80f0adb3-Paper-Conference.pdf.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xi Liu, Chaoyi Zhou, and Siyu Huang. 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. In *Advances in Neural Information Processing Systems*, pp. 133305–133327, 2024.

William Ljungbergh, Bernardo Taveira, Wenzhao Zheng, Adam Tonderski, Chensheng Peng, Fredrik Kahl, Christoffer Petersson, Michael Felsberg, Kurt Keutzer, Masayoshi Tomizuka, et al. R3d2: Realistic 3d asset insertion via diffusion for autonomous driving simulation, 2025.

Xinyuan Lu, Shengyuan Huang, Li Niu, Wenyan Cong, and Liqing Zhang. Deep video harmonization with color mapping consistency. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, pp. 1232–1238, 2022.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Improving pixel diffusion with perceptual supervision. *arXiv preprint arXiv:2602.02493*, 2026.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

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.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Gaurav Parmar, Taesung Park, Srinivasa Narasimhan, and Jun-Yan Zhu. One-step image translation with text-to-image models. *arXiv preprint arXiv:2403.12036*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Matt Pharr, Greg Humphreys, Saty Raghavachary, and Pat Hanrahan. *Physically based rendering*. Elsevier, 2016.

Bokui Shen, Xinchen Yan, Charles R Qi, Mahyar Najibi, Boyang Deng, Leonidas Guibas, Yin Zhou, and Dragomir Anguelov. Gina-3d: Learning to generate implicit neural assets in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4913–4926, 2023.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

Shanlin Sun, Yifan Wang, Hanwen Zhang, Yifeng Xiong, Qin Ren, Ruogu Fang, Xiaohui Xie, and Chenyu You. Ouroboros: Single-step diffusion models for cycle-consistent forward and inverse rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10386–10397, 2025.

Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3789–3797, 2017.

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10748–10757, 2022.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Jiawei Wang, Haowei Sun, Xintao Yan, Shuo Feng, Jun Gao, and Henry X Liu. Terasim-world: Worldwide safety-critical data synthesis for end-to-end autonomous driving. *arXiv preprint arXiv:2509.13164*, 2025.

Yifan Wang, Francesco Pittaluga, Zaid Tasneem, Chenyu You, Manmohan Chandraker, and Ziyu Jiang. Horizonforge: Driving scene editing with any trajectories and any vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24895–24905, 2026.

Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8370–8380. IEEE, 2023.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 26024–26035, 2025.

Lingyu Xiao, Zexin Feng, and Xintao Yan. Long-term traffic simulation via structured autoregressive modeling. *arXiv preprint arXiv:2606.31209*, 2026.

Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, Yunlong Wang, and Diange Yang. PandaSet: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 3095–3101, 2021.

Xintao Yan, Zhengxia Zou, Shuo Feng, Haojie Zhu, Haowei Sun, and Henry X Liu. Learning naturalistic driving environment with statistical realism. *Nature communications*, 14(1):2037, 2023.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024b.

Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. RGB↔X: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705250. doi: 10.1145/3641519.3657445. URL https://doi.org/10.1145/3641519.3657445.

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 Conference on Computer Vision and Pattern Recognition*, 2018.

Shenghao Zhang, Runtao Liu, Christopher Schroers, and Yang Zhang. Renderflow: Single-step neural rendering via flow matching. *arXiv preprint arXiv:2601.06928*, 2026a.

Yuxuan Zhang, Katarína Tóthová, Zian Wang, Kangxue Yin, Haithem Turki, Riccardo de Lutio, Yen-Yu Chang, Or Litany, Sanja Fidler, and Zan Gojcic. Diffusionharmonizer: Bridging neural reconstruction and photorealistic simulation with online diffusion enhancer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43494–43504, 2026b.

Zitian Zhang, Frédéric Fortier-Chouinard, Mathieu Garon, Anand Bhattad, and Jean-François Lalonde. Zerocomp: Zero-shot object compositing from image intrinsics via diffusion. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 483–494. IEEE, 2025.

Chuyang Zhao, Yifei Song, Hongfa Wang, Jianlong Yuan, Yuan Zhang, Siming Fu, Zhineng Chen, Huilin Deng, Haoyang Huang, and Nan Duan. Perceptual flow matching for few-step generative modeling. *arXiv preprint arXiv:2607.03524*, 2026.

Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large scale diffusion distillation via score-regularized continuous-time consistency. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 2582–2603, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0534abc9e6db91683d82186ef0d68202-Paper-Conference.pdf.

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 Transactions on Pattern Analysis and Machine Intelligence*, 2025.

# A VIDEOS

```
PIH/
|-- index.html
|-- style.css
|-- script.js
`-- assets/
    |-- images/
    `-- videos/
        `-- web/
            |-- teaser.mp4
            |-- waymo/
            |-- nuplan/
            |-- pandaset/
            `-- relighting/
                `-- grid/
```

The package is a self-contained webpage. `index.html` is the entry point, `assets/` contains the figures and browser-ready MP4 videos. To view the supplement, extract the package and open `PIH/index.html` in a modern browser. If the browser restricts local media playback, run `python3 -m http.server 8000` from inside `PIH/`

# B PRELIMINARIES

**Problem formulation.** Let $\mathbf{I}_t^{\mathrm{deg}}$ denote the frame rendered from a reconstructed driving scene at time $t$, and let $\mathbf{I}_t^{\mathrm{gt}}$ denote its photorealistic target. The rendered sequence may contain novel-view reconstruction errors as well as appearance, illumination, and shadow inconsistencies introduced by virtual assets. Existing online harmonizers directly learn an RGB-to-RGB mapping from $\mathbf{I}_t^{\mathrm{deg}}$ to $\mathbf{I}_t^{\mathrm{gt}}$ (Zhang et al., 2026b). In contrast, we condition the generation on the physical-intrinsic buffers $\mathbf{G}$ and the lighting condition $\mathbf{E}$. Our objective is to model the conditional distribution $p(\mathbf{I}_t^{\mathrm{gt}} \mid \mathbf{G}, \mathbf{E})$ while remaining robust to imperfect or compositionally mixed intrinsic buffers.

**Conditional flow matching.** For the current target frame, the frozen VAE encoder $\mathcal{E}$ gives $\mathbf{z}_{t,0} = \mathcal{E}(\mathbf{I}_t^{\mathrm{gt}})$. We draw Gaussian noise $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{\mathrm{id}})$, where $\mathbf{I}_{\mathrm{id}}$ is the identity matrix in latent space. Following flow matching, the intermediate state and target velocity are

$$\mathbf{z}_{t,\tau} = (1 - \sigma_\tau)\mathbf{z}_{t,0} + \sigma_\tau \boldsymbol{\epsilon}_t, \qquad \mathbf{v}_{t,\tau} = \boldsymbol{\epsilon}_t - \mathbf{z}_{t,0}, \tag{7}$$

where $\sigma_\tau \in [0, 1]$ is the schedule's noise scale. A conditional network $f_\theta$ is trained to predict the current flow velocity, $\hat{\mathbf{v}}_{t,\tau} = [f_\theta(\mathbf{X}_{t,\tau}, \tau; \mathbf{c}_{\mathrm{light}})]_t$, from the temporal input window and lighting tokens defined in the main text. Its prediction yields a clean latent estimate

$$\hat{\mathbf{z}}_{t,0} = \mathbf{z}_{t,\tau} - \sigma_\tau \hat{\mathbf{v}}_{t,\tau}, \qquad \hat{\mathbf{I}}_t = \mathcal{D}(\hat{\mathbf{z}}_{t,0}), \tag{8}$$

with frozen VAE decoder $\mathcal{D}$. Standard latent regression can favor an off-manifold average when only one or a few integration steps are used. PFM (Zhao et al., 2026) addresses this issue by supervising the decoded clean estimate through pretrained perceptual features. PIH adopts this endpoint-supervision principle as $\mathcal{L}_{\mathrm{pre}}$ and combines it with $\mathcal{L}_{\mathrm{FM}}$, which still supervises the single predicted latent velocity. The dense physical intrinsics condition the prediction and thereby constrain the geometry, material layout, and illumination of the endpoint.

# C TASK-LEVEL COMPARISON ACROSS DIFFERENT METHODS

Table 9 summarizes the task-level placement of PIH relative to representative artifact-removal, neural-rendering, and driving-scene editing methods. The comparison is organized by the capabilities that an online editable simulator must combine, rather than by individual architectural components. Artifact-removal methods such as FlowR, Difix3D, and ArtiFixer3D focus on correcting reconstruction artifacts, but do not expose dynamic object insertion and illumination control as a unified editing interface. In contrast, RenderFlow, DiffusionRenderer, and Ouroboros support physical

rendering or editing operations such as object composition and illumination manipulation, while their capability profiles do not target the full artifact-removal setting summarized here. HorizonForge covers artifact removal and dynamic object insertion with bidirectional temporal context, but does not combine these capabilities with explicit illumination control or one-NFE inference.

DiffusionHarmonizer is the closest task-level reference: it combines artifact removal, dynamic object insertion, past-only temporal context, and one-NFE inference. PIH instead extends this task regime with an explicit physical-intrinsic interface, through which the evaluated edits and lighting changes are composed before the final RGB observation is generated. This interface allows PIH to cover all five capabilities in the table while retaining past-only context, which is compatible with online simulation because only previously rendered or generated frames are available at time $t$.

The table is a capability summary rather than a claim that the listed methods cannot be extended to other settings. Its purpose is to make the gap addressed by PIH explicit: existing methods typically specialize in either reconstruction cleanup, physical editing, or online enhancement, whereas PIH uses physical factors as a common interface for the evaluated combination of these operations.

Table 9: Task-level comparison across different methods. A checkmark indicates that a method supports the corresponding capability.

| Methods | Artifacts removal | Dynamic object insertion | Illumination control | Temporal context | 1 NFE |
|---|---|---|---|---|---|
| FlowR (Fischer et al., 2025) | ✓ | | | Frame-wise | |
| Difix3D (Wu et al., 2025) | ✓ | | | Frame-wise | ✓ |
| ArtiFixer3D (De Lutio et al., 2026) | ✓ | | | Past-only | ✓ |
| HorizonForge (Wang et al., 2026) | ✓ | ✓ | | Bidirectional | |
| RenderFlow (Zhang et al., 2026a) | | ✓ | ✓ | Bidirectional | ✓ |
| DiffusionRenderer (Liang et al., 2025a) | | ✓ | ✓ | Bidirectional | |
| Ouroboros (Sun et al., 2025) | | ✓ | ✓ | Frame-wise | ✓ |
| DiffusionHarmonizer (Zhang et al., 2026b) | ✓ | ✓ | | Past-only | ✓ |
| PIH | ✓ | ✓ | ✓ | Past-only | ✓ |

# D Additional Method Details

## D.1 Latent G-buffer encoding

Each physical buffer is independently encoded by the frozen VAE encoder $\mathcal{E}$ and concatenated along the channel dimension. Let $h, w$ be the latent spatial height and width, and let $C_z$ be the latent channel dimension of one VAE-encoded buffer. For frame $t$, the latent G-buffer is

$$\mathbf{g}_t = \mathrm{Concat}_C[\mathcal{E}(\mathbf{n}_t), \mathcal{E}(\mathbf{d}_t), \mathcal{E}(\mathbf{a}_t), \mathcal{E}(\mathbf{r}_t)] \in \mathbb{R}^{h \times w \times 4C_z}, \tag{9}$$

The factor $4C_z$ corresponds to the four physical buffers. The resulting latent G-buffer remains pixel-aligned with the noisy image latent and is supplied as dense conditioning to the generative backbone.

## D.2 Lighting encoding and temporal alignment

Because the image VAE expects bounded low-dynamic-range inputs, we derive three complementary panoramic encodings from the HDR environment map. Reinhard tone mapping produces the low-dynamic-range map $\mathbf{E}_{\mathrm{ldr}}$, the log-intensity map $\mathbf{E}_{\log}$ retains high-intensity sources, and $\mathbf{E}_{\mathrm{dir}}$ stores the unit direction corresponding to every environment-map pixel. Let $E_{\max}$ denote the fixed intensity-normalization constant; the log-intensity encoding is

$$\mathbf{E}_{\log} = \frac{\log(\mathbf{E} + 1)}{E_{\max}}. \tag{10}$$

We encode each map with the frozen VAE encoder $\mathcal{E}$ and denote the concatenated latent by $\mathbf{h}_{\mathbf{E}}$:

$$\mathbf{h}_{\mathbf{E}} = \mathrm{Concat}_C[\mathcal{E}(\mathbf{E}_{\mathrm{ldr}}), \mathcal{E}(\mathbf{E}_{\log}), \mathcal{E}(\mathbf{E}_{\mathrm{dir}})]\,. \tag{11}$$

Let $S$ be the number of pyramid levels, let $k$ index a level, and let $\mathbf{h}_{\text{env}}^k$ denote its feature map. A lightweight environment encoder $\mathcal{E}_{\text{env}}$ extracts the feature pyramid

$$\mathbf{c}_{\text{env}} = \{\mathbf{h}_{\text{env}}^k\}_{k=1}^S = \mathcal{E}_{\text{env}}(\mathbf{h}_{\mathbf{E}}). \tag{12}$$

Its four levels contain $4096$, $1024$, $256$, and $64$ tokens with channel dimensions $320$, $640$, $1280$, and $1280$. Independent Linear–Gaussian Error Linear Unit (GELU)–Linear adaptors map them to the DiT width of $1024$; let $B$ denote the batch size. We retain the last two levels, yielding lighting tokens $\mathbf{c}_{\text{light}} \in \mathbb{R}^{B\times 320\times 1024}$ for cross-attention.

The original video tokenizer temporally compresses its input, which would mix neighboring physical conditions. We therefore apply the VAE independently to every RGB frame and intrinsic buffer. With $K = 4$ preceding frames, the input window $\mathbf{X}_{t,\tau}$ contains $K+1 = 5$ temporal latent positions, and temporal positional encodings preserve the one-to-one frame correspondence.

### D.3 Perceptual endpoint distance

LPIPS compares multi-level features of $\hat{\mathbf{I}}_t$ and $\mathbf{I}_t^{\text{gt}}$; we denote this loss by $\mathcal{L}_{\text{LPIPS}}$:

$$\mathcal{L}_{\text{LPIPS}} = \text{LPIPS}(\hat{\mathbf{I}}_t, \mathbf{I}_t^{\text{gt}}). \tag{13}$$

For global structure, let $\mathcal{P}$ be the set of corresponding patch indices, let $p \in \mathcal{P}$ index a patch, and let $\phi_{\text{DINO}}^p(\cdot)$ be the feature of patch $p$ from the last layer of a frozen DINOv2-B encoder (Oquab et al., 2024). P-DINO averages the cosine distance between corresponding patch features; we denote this loss by $\mathcal{L}_{\text{P-DINO}}$, where $|\mathcal{P}|$ is the number of patches and $\cos(\cdot,\cdot)$ denotes cosine similarity:

$$\mathcal{L}_{\text{P-DINO}} = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\left[1 - \cos\Big(\phi_{\text{DINO}}^p(\hat{\mathbf{I}}_t), \phi_{\text{DINO}}^p(\mathbf{I}_t^{\text{gt}})\Big)\right]. \tag{14}$$

Together, these feature distances define the perceptual endpoint term $\mathcal{L}_{\text{pre}}$ in the hybrid objective above; they supervise only the current decoded frame, while the mixed history batches expose the model to image-conditioned and self-rollout contexts.

### D.4 Training schedule

Stage I optimizes the latent velocity-matching term $\mathcal{L}_{\text{FM}}$ for 10K iterations using all paired images, with a global batch size of 32 and a learning rate of $2 \times 10^{-4}$. Stage II runs for 1.4K iterations with a learning rate of $3.2 \times 10^{-5}$ and all parameters of $f_\theta$ trainable; it optimizes the hybrid objective defined in Sec. 3.2 with $\lambda_{\text{fm}} = 1$ and $\lambda_{\text{pre}} = 0.5$. The first 500 iterations use image batches to warm up the perceptual endpoint term $\mathcal{L}_{\text{pre}}$; during the remaining 900 iterations, batches are sampled as image-only, RGB-grounded video, or self-rollout video with probabilities $0.50$, $0.25$, and $0.25$, respectively. Image batches use the complete training set, while both video modes use temporally contiguous front-camera sequences. RGB-grounded batches use encoded 3DGS RGB references as history, whereas self-rollout batches recursively reuse each $\hat{\mathbf{z}}_{s,0}$; in both modes, the loss is applied only at the current frame.

## E Evaluation Details

### E.1 Baselines

We use the same baseline suite as DiffusionHarmonizer (Zhang et al., 2026b): SDEdit (Meng et al., 2022), InstructPix2Pix (Brooks et al., 2023), VACE-V2V (Jiang et al., 2025). The inference protocol for these baselines is follow the same as described in DiffusionHarmonizer. Additionally we report Difix3D (Wu et al., 2025) for artifact removal tasks. We evaluate all methods on the same scene splits, camera trajectories, frame sampling, and virtual-asset compositions. All baseline models are evaluated using the officially released checkpoints.

### E.2 Metrics

LPIPS, FID, and FVD measure paired perceptual error, frame-level realism, and spatiotemporal realism, respectively (Zhang et al., 2018; Heusel et al., 2017; Unterthiner et al., 2018). Without

paired novel-view ground truth, FID and FVD are distributional rather than frame-aligned measures. DINO-S (Tumanyan et al., 2022; Parmar et al., 2024) measures preservation of scene structure; scores are multiplied by 100. We use these scores for artifact removal and the approximately matched Waymo insertion setting, while cross-dataset insertion is assessed with qualitative comparisons and VLM/human preference evaluation. For VLM/human evaluation, we use win rate as the final assessment.

### E.3 VLM and Human Preference Evaluation

We evaluate visual preference on held-out scenes from the *replace* set. For each scene, we use the same frame for the RGB reference and all candidate outputs. Specifically, we probe the frame count of the RGB and candidate videos, use the minimum frame count as a safe upper bound, and sample one frame index with a fixed seed. Candidate labels are independently and deterministically permuted for every scene, while the mapping from labels to methods is kept only in the metadata and evaluation manifest.

**VLM evaluation.** For the VLM evaluation, each item contains one reference image and the candidate images in the order specified by their anonymous labels. We use Qwen3-VL-235B (Bai et al., 2025) with five independent sampling seeds and ask it to return exactly one candidate label. For each seed and dataset split, the win rate of a method is the fraction of valid scene-level predictions assigned to that method. We average the win rates across the five seeds and report their sample standard deviation; invalid or failed model responses are excluded from the corresponding seed-level denominator.

The multimodal message first presents the text "Reference input image:" and the reference image, and then each available candidate preceded by "Candidate [label]:" in label order. It ends with the following prompt (with the candidate label list adapted to the item):

**VLM evaluation Prompt**

> You are evaluating AI-generated image enhancements.
> The reference image shows the original scene. Candidate images A, B, C, and D are enhanced versions of the same scene. Select the candidate that provides the best visual quality while remaining faithful to the reference.
> Consider: (1) realistic and physically consistent lighting and shadows; (2) absence of reconstruction or processing artifacts; (3) natural and harmonious colors; and (4) preservation of scene geometry and textures.
> Return only the label of the best candidate. Return only one label: A, B, C, or D.

Originally, four models has been tested, but the win rate on two of them (InstructPix2Pix and SDEdit) are 0%, so we only report the remaining two models (Ours and DiffusionHarmonizer) in the main paper, as well as in the human study.

**Human study.** An example of the user-study interface is shown in Fig. 7. Participants are recruited through the Prolific platform, which provides access to a diverse, globally distributed pool of evaluators. They are shown the reference image followed by anonymized candidate images and select their preferred candidate in a Google Form. We obtained 21 complete responses for Waymo (47 scenes per response) and 23 complete responses for the combined nuPlan–PandaSet split (45 scenes per response). For each participant, we first compute a response-level win rate as the fraction of evaluated scenes for which the selected candidate corresponds to a given method. We then report the mean and sample standard deviation of these response-level win rates across all completed responses, without participant-level filtering or trimming. Specifically, the Waymo and nuPlan–PandaSet statistics are computed over 21 and 23 responses, respectively, and the All row is computed over all 44 response-level win rates. Therefore, the reported standard deviation reflects inter-participant variation in preference, rather than a confidence interval or standard error. For the VLM evaluation, we analogously report the mean and sample standard deviation of the win rates across five independent random seeds; thus, the evaluator unit is a human response for the human study and a random seed for the VLM study.

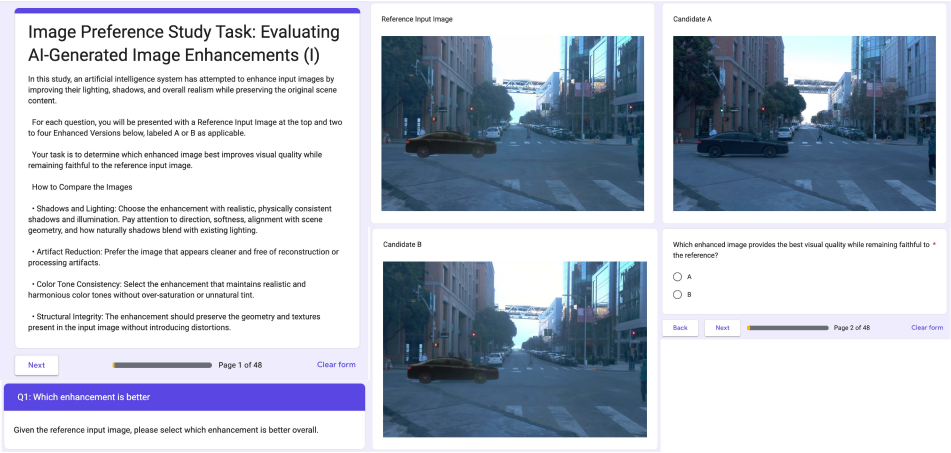



Figure 7: Example of a user-study interface. Evaluators see the reference image and candidate images in a randomized order, and select one candidate that best matches the reference.

## F Additional Qualitative Results

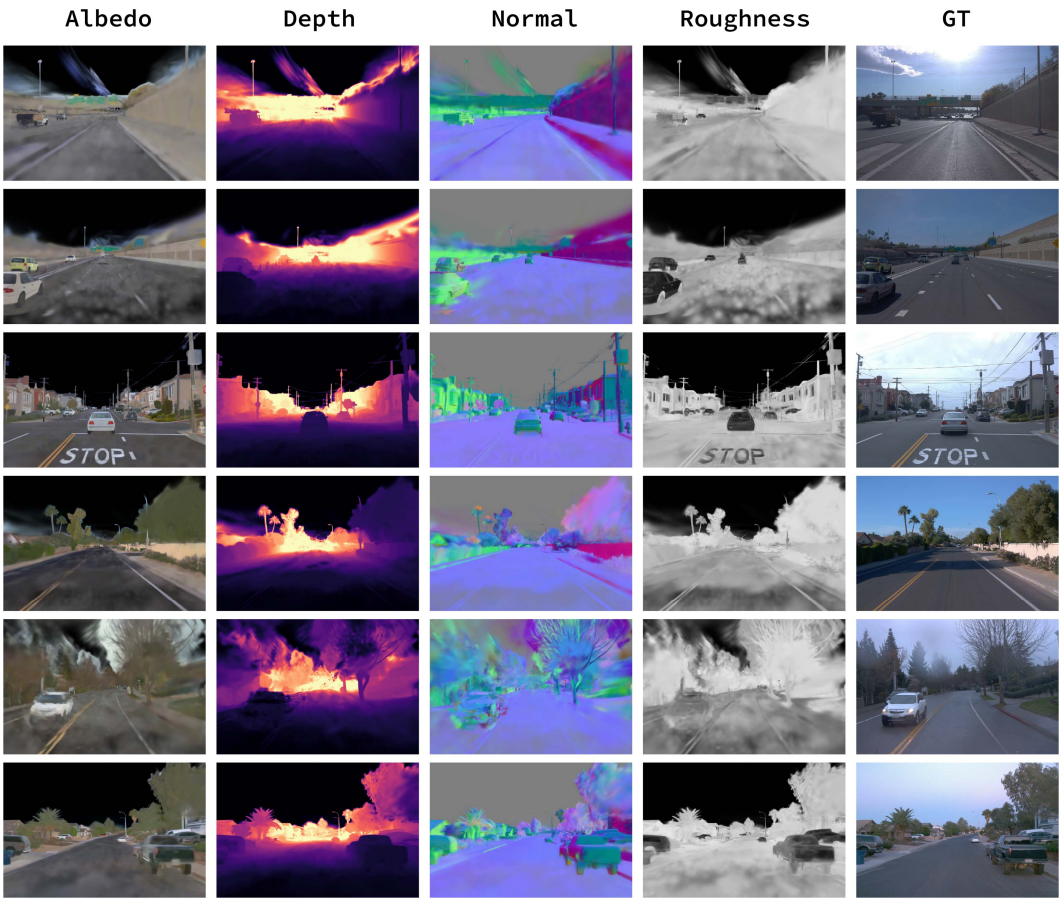



Figure 8: Illustration of training pair. From left to right are albedo, normal, depth, roughness, and GT. The examples shown here are directly rendered from neural reconstructed scenes. The full training set additionally contains clean pseudo-G-buffer pairs generated by DiffusionRenderer.

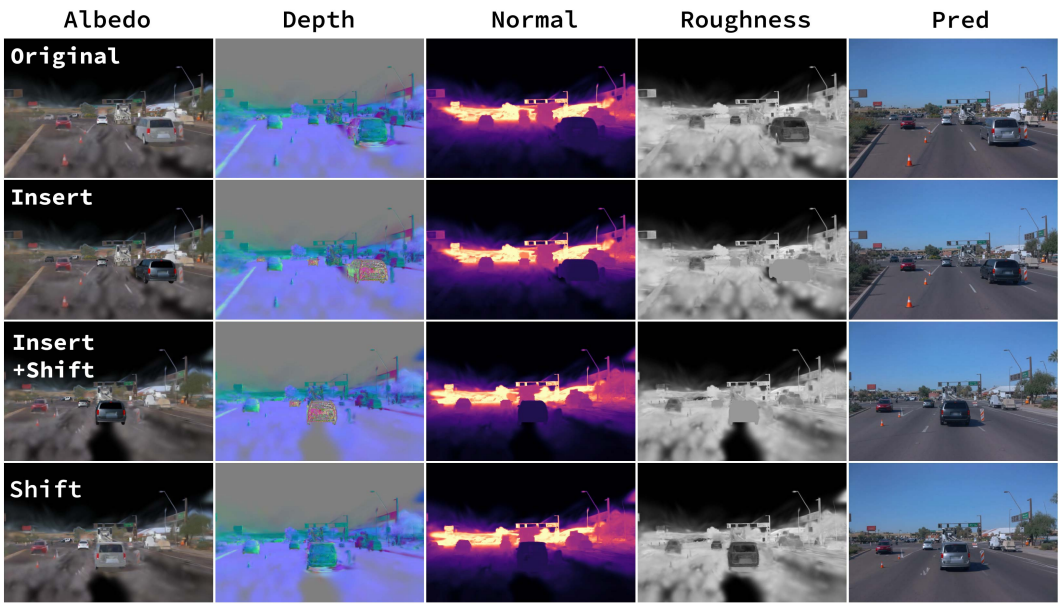



Figure 9: Illustration of editing in the rendered scene. From left to right are albedo, normal, depth, roughness, and prediction. Each row shows different tasks as described in Sec. 4.2.

### F.1 Illustration of training pair

We visualize the training pair in Fig. 8. The training pairs are directly rendered from the neural reconstructed scene without any editing or post-processing. The quality of the training pairs varies, from extremely noisy to relatively clean. Moreover, unlike common practice (Zhang et al., 2026b; Wu et al., 2025) that constructs pairs in three steps: first build a clean neural scene using the original trajectory, then laterally shift the camera to generate a set of novel-view images, which are then used to construct another neural scene, from which it renders the training pairs by shifting the camera back to the original trajectory. Our training pairs are directly rendered from the neural reconstructed scene without any editing or post-processing, which is more efficient; however, it may not be robust to lane lines under the shift setting.

### F.2 Illustration of dynamic object insertion

We visualize how to manipulate the neural scene under four different tasks in Fig. 9. From top to bottom, the four rows show the different tasks as we described in Sec. 4.2. For tasks with asset insertion, we naively treat the color from the virtual asset as the albedo. The normal is calculated from the depth map of the virtual asset, and the roughness is an average value of all the present virtual assets. It can be clearly seen that apart from the first task, all other tasks are out of the distribution of the training pairs. Moreover, the model is not sensitive to normal and roughness, as we can see that although the values are obviously not accurate, the model can still generate a reasonable result. This is because the model is trained with imperfect physical intrinsics, which makes it robust to imperfect physical intrinsics during inference.

### F.3 Additional qualitative comparisons

We additionally provide more qualitative comparisons on Waymo, nuPlan, and PandaSet in Fig. 10, Fig. 11, Fig. 12 and Fig. 13. For Waymo, we demonstrate a wider comparison with both harmonization baseline and image/video editing baselines. For nuPlan, we show a long-horizon simulation up to 300 frames; although our model uses history as reference, it does not suffer from long-horizon exposure bias. For PandaSet, we show a wider spectrum of scenarios with dense traffic and complex urban scenes. Throughout all three datasets, we can see that our method can generate more realistic and physically consistent results than DiffusionHarmonizer under various scenarios.

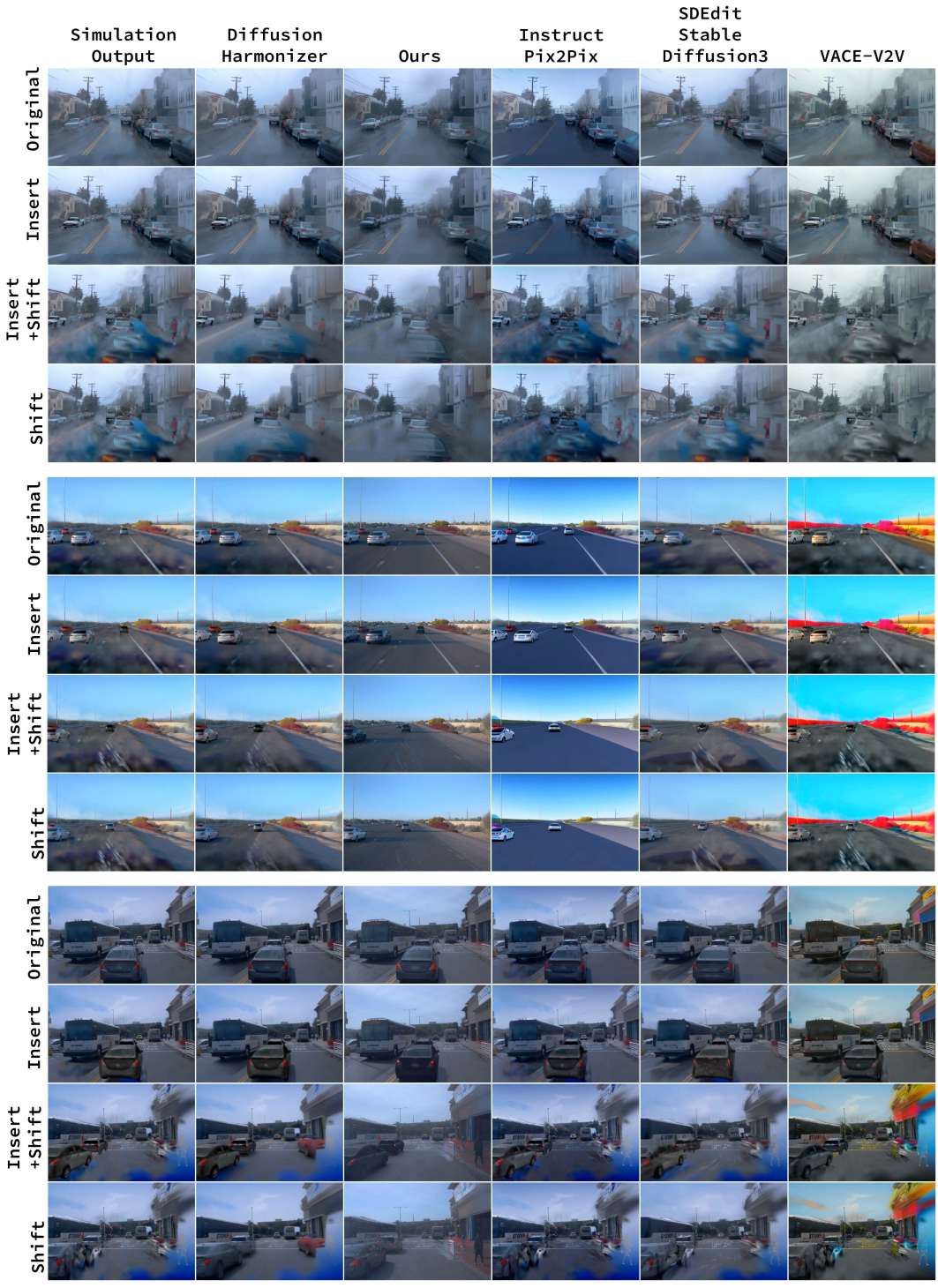



Figure 10: Additional qualitative comparisons on Waymo.




Figure 11: Additional qualitative comparisons on nuPlan. We present a sequence of long-horizon simulation up to 30s (10HZ, 300 times AR).




Figure 12: Additional qualitative comparisons on PandaSet. (Part1/2)




Figure 13: Additional qualitative comparisons on PandaSet. (Part2/2)