VIDEO DIFFUSION RENDERING FOR 3D ASSET INSERTIONS IN DRIVING SIMULATION

Anonymous authors
Paper under double-blind review

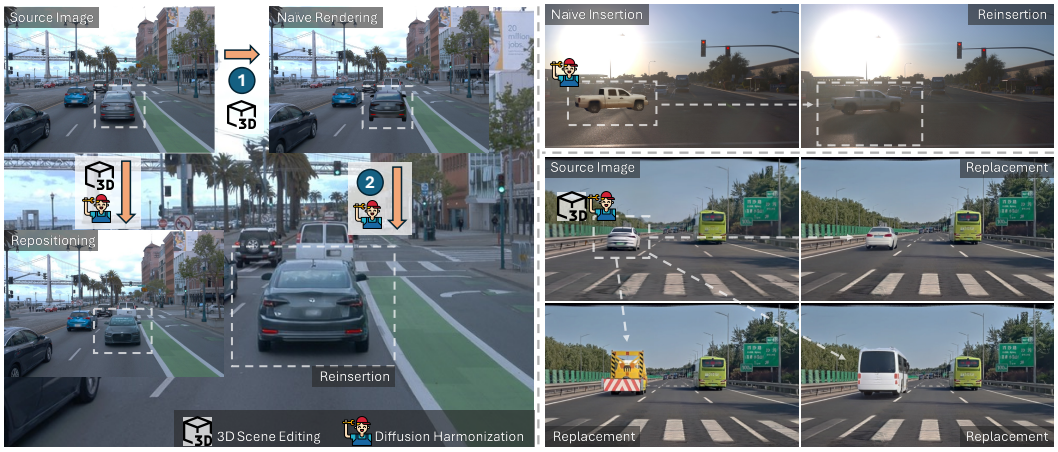


Figure 1: **Geometry-controlled editing with photorealistic rendering.** Our framework supports asset reinsertion, repositioning, and replacement by combining geometry-controlled 3D editing with diffusion-based rendering harmonization. *The left examples* show the transition from source images to naive renderings and our harmonized results for reinsertion and repositioning. *The upper-right examples* demonstrate robustness under over-exposed lighting, while *the lower-right examples* show generalization to real-world driving scenes and assets with diverse vehicle types, including buses and construction vehicles.

ABSTRACT

The growing availability of reusable 3D assets and reconstructed real-world scenes is opening new opportunities for scalable and controllable driving simulation. A key capability is asset insertion, where external objects are placed into existing scenes to create diverse and interactive scenarios. However, geometrically rendered insertions often remain visually implausible, suffering from appearance mismatch and limited scene interaction, especially when assets originate from different scenes. While frame-wise harmonization can improve visual realism, it often introduces temporal inconsistency in video outputs. We address this naive-to-real rendering problem with a one-step video diffusion harmonizer that maps naive insertion renderings to photorealistic and temporally consistent driving videos. The harmonizer is obtained by fine-tuning a pretrained video diffusion model, with temporally agnostic latent injection introduced to recover spatial details lost during 3D VAE decoding. To effectively train the harmonizer, we construct DriveInsert-3K, an offline paired video dataset built with a two-stage alignment procedure to provide clean naive-insertion/real-video supervision. At inference time, our model takes a naive insertion video from an upstream insertion pipeline and produces a harmonized video in a single denoising step. Experiments show that our approach improves visual realism and temporal consistency across diverse driving scenes and inserted assets while enabling efficient single-step inference.

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.

1 INTRODUCTION

Vision-centric autonomous driving systems require simulation environments that are both realistic and controllable. Recent advances in 3D asset generation and real-world scene reconstruction have made it increasingly practical to edit existing driving scenes instead of synthesizing entire scenarios from scratch. In this context, 3D asset insertion has emerged as a fundamental capability for driving simulation: external assets can be placed into reconstructed scenes to create diverse and interactive scenarios while preserving the original scene layout, camera motion, and background context. Such controllable insertions provide a scalable way to create object-level scenario variations grounded in real-world driving environments.

However, geometry-controlled asset insertion does not guarantee photorealistic rendering. Although a geometry-based insertion pipeline can specify the asset pose and render it along the original camera trajectory, the resulting frames often remain visually naive. The inserted asset may not share the same illumination, material appearance, shadowing, or local context as the surrounding scene, especially when it is generated or captured from a different scene. These discrepancies make the inserted object appear synthetic, limiting the realism and usefulness of the resulting simulation videos.

Bridging this rendering gap requires improving the rendered appearance while preserving the geometry produced by the 3D insertion pipeline. Recently, R3D2 Ljungbergh et al. (2025) addressed this problem by harmonizing 3D asset insertions in reconstructed scenes with image diffusion. While such image-level harmonization can improve the realism of individual frames, processing frames independently struggles to maintain temporal coherence, often leading to flickering or appearance variations over time. This limitation suggests that asset insertion harmonization should be addressed as a video-level rendering problem rather than independent frame-wise image translation.

To overcome this limitation, we propose a video diffusion harmonizer for 3D asset insertions in driving scenes. Given a naive insertion video, our model adapts a pretrained video diffusion model for single-step harmonization, enabling efficient inference while leveraging its temporal prior for cross-frame consistency. Adapting such a model to our setting, however, is non-trivial. Latent denoising may inadvertently alter clean background regions and disturb the original scene structure, while the autoencoding process of a 3D VAE introduces compression loss that degrades spatial fidelity. Therefore, we introduce a *temporally agnostic latent injection* strategy, which injects high-frequency cues from a pretrained 2D encoder into the 3D decoder in a temporally compatible manner. Since the injected 2D features and the 3D decoder features lie in different representational spaces, we further introduce a cross-modal fusion block to align and integrate them. Together, these components improve spatial fidelity while preserving temporal consistency, enabling photorealistic and coherent rendering of inserted assets.

Moreover, training such a harmonizer requires clean paired supervision between naive insertion renderings and real videos. In previous asset reinsertion pipelines, the distribution gap between scene Gaussians and generated 3D assets often leads to inaccurate pose alignment. As a result, the discrepancy between the naive insertion and the real video includes not only rendering artifacts but also geometric misalignment, which introduces noisy supervision for harmonization. To reduce this noise, we introduce a 3D-to-2D alignment strategy. We first align the asset Gaussians to the scene object in 3D space by optimizing the center, orientation, and scale. We then refine the alignment in 2D space by calibrating the rendered asset bounding box against the ground-truth object bounding box, yielding tighter image-space consistency. Following this, we construct DriveInsert-3K, an offline paired video dataset of 3,550 clips with precisely aligned naive-insertion/real-video pairs. This dataset provides clean supervision for learning naive-to-real rendering effects in asset insertion videos.

Extensive experiments demonstrate that our method improves visual realism and temporal coherence across diverse driving scenes and inserted assets, outperforming existing baselines in both quantitative metrics and qualitative comparisons while supporting efficient single-step inference. Even under challenging insertion settings, such as repositioning a vehicle with a 180-degree direction reversal, our harmonizer can still produce realistic outputs, as shown in Figure 1. Beyond producing simulation videos, the harmonized outputs can also serve as enhanced supervision for improving the underlying 3D representation. In summary, our key contributions are as follows:

- We present a video diffusion harmonizer that improves the realism and temporal coherence of 3D asset insertion videos with efficient single-step inference.

- We introduce a temporally agnostic latent injection strategy that enhances spatial fidelity while preserving temporal consistency in causal 3D VAEs.
- We develop a 3D-to-2D alignment strategy and construct DriveInsert-3K, a high-quality paired video dataset of 3,550 clips that provides precise alignments and clean supervision for naive-to-real rendering.

2 Related Work

2.1 One-Step Diffusion Model

A prevalent strategy for accelerating diffusion models is to reduce the number of sampling steps. Common technical routes include rectified flow Liu et al. (2022; 2023), score distillation Zhou et al. (2024a); Wang et al. (2023); Yin et al. (2024), and adversarial training Lin et al. (2025); Zhang et al. (2024). Building on these advances, Image-to-Image Turbo Parmar et al. (2024) demonstrates how adversarial fine-tuning enables photorealistic image translation in a single step. In 3D scene rendering, DIFIX3D+ Wu et al. (2025) employs variational score distillation to enhance novel view synthesis quality while maintaining 3D consistency. Extending such acceleration techniques to the video domain, however, introduces the critical challenge of maintaining temporal coherence. The recently proposed DOVE Chen et al. (2025) addresses this by fine-tuning a pre-trained text-to-video model for one-step video super-resolution. Although it achieves improved coherence, its reliance on a 3D VAE for latent representation introduces noticeable compression artifacts, ultimately constraining the visual quality of the output.

2.2 Asset Insertion and Relighting

Seamless asset insertion and relighting are essential for realistic driving simulation, where inserted objects must be consistent with the surrounding scene in geometry, illumination, material appearance, and shadows. Traditional physically based rendering (PBR) methods Pharr et al. (2023) provide principled solutions, but require accurate geometry, material properties, and lighting estimation. Diffusion-based methods such as DiPIR Liang et al. (2024) combine generative priors with inverse rendering, yet still rely on asset material knowledge and complex optimization. Recent generative relighting and harmonization methods, including IC-Light Zhang et al. (2025) and DiffHarmony++ Zhou et al. (2024b), provide more flexible image-level alternatives, but they mainly focus on foreground appearance adaptation and do not explicitly address 3D geometry or temporal consistency in driving videos.

In autonomous driving, 3DGS-based methods such as OmniRe Chen et al. (2024) and SplatAD Hess et al. (2025) enable object-level scene manipulation through Gaussian node editing, providing strong 3D control. However, geometry-controlled insertion alone often suffers from inconsistent lighting, missing shadows, and limited realism. R3D2 Ljungbergh et al. (2025) further improves 3D asset insertion by using one-step image diffusion to generate realistic rendering effects. Nevertheless, its single-image processing paradigm lacks explicit temporal modeling, limiting its ability to maintain consistent asset appearance across video sequences. In contrast, our method addresses 3D asset insertion harmonization as a video-level rendering problem and leverages a pretrained video diffusion prior for temporally coherent single-step harmonization.

3 Method

Given a naive insertion video produced by a geometry-based 3D asset insertion pipeline, our goal is to generate a photorealistic and temporally coherent video while preserving the inserted asset geometry and scene structure. Figure 2 provides an overview of the pipeline. Our method builds on geometry-controlled 3D asset insertion and combines single-step video diffusion harmonization with clean paired supervision from aligned naive-insertion/real-video pairs. We train the harmonizer in two stages: a VAE adaptation stage that improves spatial fidelity through temporally agnostic latent injection, and a harmonization stage that refines naive-to-real rendering effects with lightweight LoRA modules.

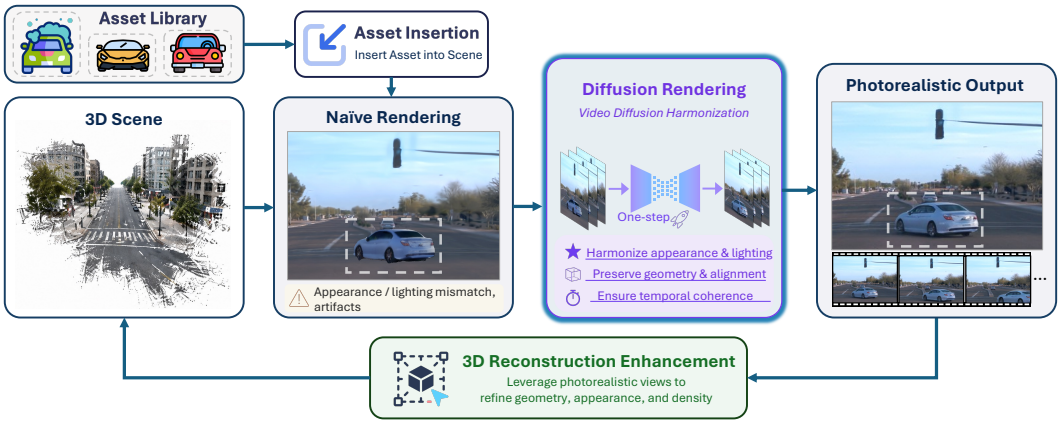


Figure 2: **Diffusion rendering for 3D asset insertion.** Starting from a geometry-controlled 3D asset insertion, the inserted asset is rendered as a naive video with appearance and lighting artifacts. Our diffusion rendering module harmonizes the naive rendering into a photorealistic and temporally coherent output while preserving the inserted geometry. The harmonized results can further provide supervision for enhancing the underlying 3D reconstruction.

3.1 Preliminaries: One-Step Latent Diffusion

Latent diffusion models perform denoising in a compressed latent space rather than directly in pixel space. Given a video x, a 3D VAE encoder maps it to a spatiotemporal latent representation z_0. During the forward diffusion process, Gaussian noise is added to z_0 at timestep t as

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $\bar{\alpha}_t$ denotes the cumulative noise schedule. A denoising network ϵ_θ is trained to predict the added noise:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{z_0,t,\epsilon}\left[\|\epsilon_\theta(z_t, t, c) - \epsilon\|_2^2\right], \tag{2}$$

where c denotes optional conditioning inputs.

Instead of sampling from pure noise through multiple denoising steps, one-step diffusion performs a single denoising operation from a fixed timestep. In our setting, the latent of a naive insertion video is treated as the input latent at a fixed timestep, and the denoising model predicts a harmonized latent that is later decoded into the output video. This formulation enables efficient naive-to-real rendering while preserving the structure provided by the 3D insertion pipeline.

3.2 Overall Framework

Our video diffusion harmonizer is built upon a pretrained text-to-video diffusion model, CogVideoX Yang et al. (2024). CogVideoX employs a causal 3D VAE to encode videos into a spatiotemporal latent space, and a Transformer denoiser ϵ_θ to perform latent-space denoising. This pretrained video prior provides strong temporal modeling capability, making it well-suited for harmonizing naive insertion videos without relying on auxiliary motion-specific modules such as optical flow.

The overall framework is illustrated in Figure 3. At inference time, given a naive insertion video x_{NI}, we first encode it into a spatiotemporal latent representation z_{NI} using the 3D VAE encoder. In parallel, we process x_{NI} with a pretrained 2D VAE encoder and cache intermediate features from selected layers for later injection. We regard z_{NI} as a degraded latent corresponding to a fixed diffusion timestep t and perform one-step denoising with ϵ_θ, producing the harmonized latent z_{DR}. Following prior works Ljungbergh et al. (2025); Wu et al. (2025), we use a relatively small timestep t=199 to balance harmonization strength and structural preservation. The 3D VAE decoder then reconstructs the harmonized output video x_{DR} from z_{DR}. During this decoding process, the cached high-frequency features from the 2D VAE are injected into selected layers of the 3D decoder through

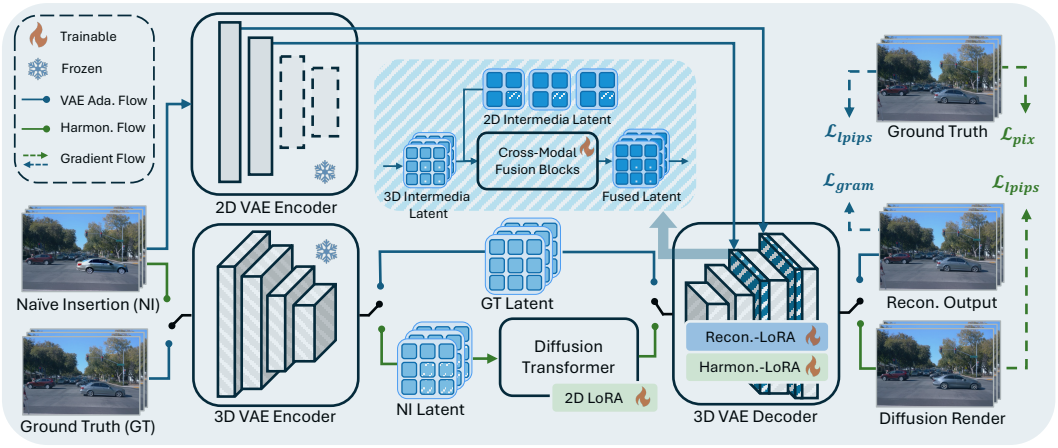


Figure 3: **Overview of the proposed video diffusion harmonizer.** Given a naive insertion video, our framework encodes the input with both a causal 3D VAE encoder and a pretrained 2D VAE encoder. The 3D latent is refined by a diffusion Transformer with LoRA adapters through one-step denoising, while intermediate 2D features are cached for decoder-side injection. During decoding, the cached 2D features are fused with 3D decoder features through cross-modal fusion blocks. The 3D VAE decoder reconstructs the output video, with Reconstruction-LoRA adapting the decoder to injected 2D features and Harmonization-LoRA refining naive-to-real rendering effects.

our temporally agnostic latent injection strategy, improving spatial fidelity while preserving temporal consistency.

3.3 VAE Adaptation Stage

A key challenge in adapting pretrained video diffusion models to asset insertion harmonization is the spatial fidelity loss introduced by the causal 3D VAE. To represent videos efficiently, causal 3D VAEs Yang et al. (2024); Yu et al. (2023); Wan et al. (2025) compress the input in both spatial and temporal dimensions, which may remove high-frequency details that are important for realistic rendering. Moreover, the latent denoising process may inadvertently perturb clean background regions, further degrading structural fidelity. This degradation is particularly noticeable in our setting, where subtle local cues such as object boundaries, material textures, shadows, and reflections determine whether an inserted asset appears natural in the scene.

3.3.1 Temporally Agnostic Latent Injection.

To address this issue, we introduce a *temporally agnostic latent injection* strategy that enhances spatial fidelity while preserving temporal causality. Our key observation is that temporal upsampling in D^{3D} is mainly performed in the early upsampling blocks, after which the decoder features become largely frame-aligned. In the last two upsampling blocks, each temporal feature slice primarily contributes to its corresponding output frame. This frame-aligned structure allows us to inject temporally agnostic 2D features into the decoder without disrupting causal temporal modeling.

We first encode each frame of the naive insertion video x_{NI} with a pretrained 2D VAE encoder E^{2D} to extract high-frequency intermediate features z^{2D}_{mid}. These features are extracted directly from the input video before diffusion denoising, and therefore provide clean spatial guidance for the decoder. Since the 2D encoder processes frames independently, these features are reshaped from the frame-as-batch format back into the temporal video dimension and injected into D^{3D} through Cross-Modal Fusion Blocks (CMFBs). Each CMFB fuses the 2D features with the corresponding 3D decoder features through concatenation and merging operations, enriching the decoder representation with detailed spatial cues without introducing additional temporal dependencies.

Since injecting 2D features changes the feature distribution expected by D^{3D}, we introduce trainable 3D causal LoRA adapters, denoted as Reconstruction-LoRA, into the decoder. These lightweight adapters modulate the decoder convolutional weights, allowing D^{3D} to adapt to the injected cross-modal features with minimal finetuning.

3.3.2 TRAINING OBJECTIVE.

During this stage, we decouple decoder adaptation from diffusion denoising and directly supervise the 3D decoder to reconstruct the ground-truth video. The 3D latent is taken from x_{GT} to isolate decoder reconstruction from diffusion denoising, while the 2D features are extracted from x_{NI} to match the inference-time input. Specifically, we feed the encoded ground-truth latent $E^{3D}(x_{GT})$ into the 3D decoder and inject the 2D intermediate features extracted from the naive insertion video x_{NI}:

$$x_{RO} = D^{3D}[E^{3D}(x_{GT}), z^{2D}_{mid}], \tag{3}$$

where z^{2D}_{mid} denotes the injected 2D features extracted by $E^{2D}(x_{NI})$. As illustrated in Figure 3, we optimize the reconstruction output x_{RO} using a combination of mean squared error loss $\mathcal{L}_{\text{MSE}}$ and perceptual loss $\mathcal{L}_{\text{LPIPS}}$ Zhang et al. (2018):

$$\mathcal{L}_{\text{vae}} = \mathcal{L}_{\text{MSE}}(x_{RO}, x_{GT}) + \lambda_1 \mathcal{L}_{\text{LPIPS}}(x_{RO}, x_{GT}), \tag{4}$$

where λ_1 is the weight for the perceptual loss. This stage adapts the decoder to the injected 2D features and improves reconstruction fidelity before diffusion harmonization.

3.4 HARMONIZATION TRAINING STAGE

After the VAE adaptation stage, the decoder is better adapted to the injected 2D features and can preserve more spatial details during reconstruction. However, this stage alone does not explicitly learn naive-to-real rendering effects such as shadows, specular reflections, and material consistency.

To further enhance appearance realism, we train the harmonizer with pixel-space supervision while freezing the VAE adaptation modules, including CMFBs and Reconstruction-LoRA. We introduce two lightweight trainable components for harmonization. First, LoRA layers are inserted into the pretrained Transformer denoiser ϵ_θ, allowing the video diffusion prior to adapt to naive-to-real rendering. Second, we add a causal 3D LoRA module, denoted as Harmonization-LoRA, to the 3D VAE decoder. Unlike Reconstruction-LoRA, which adapts the decoder to injected 2D features, Harmonization-LoRA is optimized during the harmonization stage to refine rendering effects while preserving the causal temporal structure of the video decoder.

The model is trained to enhance the visual realism of the harmonized output x_{DR} generated from the naive insertion video x_{NI}. We optimize x_{DR} against the real video x_{GT} using LPIPS loss and Gram-matrix loss Reda et al. (2022):

$$\mathcal{L}_{\text{harm}} = \mathcal{L}_{\text{LPIPS}}(x_{DR}, x_{GT}) + \lambda_2 \mathcal{L}_{\text{Gram}}(x_{DR}, x_{GT}), \tag{5}$$

where λ_2 controls the Gram loss. The $\mathcal{L}_{\text{LPIPS}}$ term encourages perceptual similarity to the real video, while $\mathcal{L}_{\text{Gram}}$ matches second-order texture statistics computed from VGG-16 feature activations.

3.5 DATA PROCESSING PIPELINE

To train the harmonizer with clean naive-insertion/real-video supervision, we construct DriveInsert-3K, a high-quality paired video dataset with precisely aligned 3D asset insertions. The key challenge is to reduce supervision noise caused by pose mismatches between the inserted assets and the corresponding scene objects.

3.5.1 SUPERVISION NOISE FROM MISALIGNMENT.

A major source of supervision noise comes from the distribution mismatch between scene Gaussians and generated asset Gaussians. Scene Gaussians are optimized from multi-view observations of real driving videos, whereas asset Gaussians are produced by generative models trained on different data distributions. As a result, directly placing the generated asset at the original object position may lead to pose, scale, or orientation deviations. Such misalignment causes the discrepancy between the naive insertion and the real video to include not only rendering artifacts but also geometric errors, which corrupts the supervision for harmonization. Since our goal is to learn naive-to-real rendering effects rather than geometric correction, accurate asset-scene alignment is essential.

3.5.2 3D-TO-2D ALIGNMENT STRATEGY.

To reduce the misalignment caused by Gaussian distribution mismatch, we introduce a 3D-to-2D alignment strategy that combines coarse 3D matching with image-space refinement.

We first perform coarse alignment in 3D space. The Gaussian set of the generated asset is matched to the Gaussian subset corresponding to the original scene object. From this correspondence, we estimate a similarity transformation consisting of translation, rotation, and scale, placing the asset near the original vehicle in world space.

Although this step provides approximate geometric consistency, residual misalignment may still appear in the image plane. We therefore refine the alignment in 2D space. After rendering the coarsely aligned asset across the video, we compute the asset bounding box in each frame and compare it with the annotated bounding box of the original vehicle. A global affine transformation is then estimated from the averaged bounding-box differences across the video, adjusting displacement and scale uniformly across frames and improving frame-level spatial alignment.

3.5.3 DATA PAIR CURATION.

Using the proposed 3D-to-2D alignment strategy, we construct DriveInsert-3K for naive-to-real rendering supervision. For each real driving video, we create a naive insertion video x_{NI} by replacing a target vehicle with an aligned generated asset while preserving the original scene geometry and camera trajectory. The original video is used as the realistic reference x_{GT}, forming paired samples (x_{NI}, x_{GT}). Here, x_{NI} provides geometry-controlled asset insertion but lacks realistic rendering effects such as lighting interaction, shadows, and material consistency, while x_{GT} provides the desired real-video appearance.

Applying this pipeline to the Waymo Open Dataset Sun et al. (2020), we construct DriveInsert-3K with 3,550 carefully aligned video clips across diverse object categories, viewpoints, and lighting conditions. This dataset supports both the VAE adaptation stage and the harmonization training stage.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Implementation Details. Our video diffusion harmonizer is built upon the pretrained text-to-video model CogVideoX-1.5 Yang et al. (2024). To reduce inference overhead, we use a fixed prompt, "remove degradation," and pre-encode its text embedding. The training data consist of 9-frame clips at a resolution of 512×768. We train the model in two stages: VAE adaptation and harmonization training. Both stages are trained on 8 NVIDIA H200 GPUs with a total batch size of 8, using the AdamW optimizer Loshchilov et al. (2017) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Each stage runs for 10,000 iterations with a learning rate of 1×10^{-4}. The loss weights λ_1 and λ_2 are both set to 0.1. In the harmonization training stage, we activate the Gram loss after 2,000 steps and apply a constant warm-up schedule for the first 500 steps. For the 2D feature branch, we adopt AutoEncoderKL as the pretrained 2D VAE encoder.

Evaluation Metrics. We adopt multiple metrics to evaluate harmonization quality. For spatial fidelity, we report PSNR and SSIM Wang et al. (2004). For perceptual quality, we use LPIPS Zhang et al. (2018) and DISTS Ding et al. (2020). To assess temporal consistency, we compute the average flow warping error E^*_{warp} Lai et al. (2018); Sun et al. (2025), which measures frame-to-frame consistency after optical-flow-based warping. To evaluate overall realism, we compute video Fréchet Inception Distance (vFID) Wang et al. (2018) between generated videos and their ground-truth counterparts. Together, these metrics assess reconstruction accuracy, perceptual quality, temporal stability, and video-level realism.

Table 1: Quantitative comparison on our evaluation set. We report results on actor-centric crops and full-resolution frames. ↑ indicates higher is better; ↓ indicates lower is better.

Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	E^*_{warp} ↓	vFID↓	FPS↑
Actor-centric crops							
Naive Rendering	14.93	0.307	0.209	0.236	**0.801**	0.506	–
R3D2 Ljungbergh et al. (2025)	16.18	0.356	0.188	0.207	1.119	**0.442**	–
Ours	**17.43**	**0.390**	**0.179**	**0.183**	0.874	0.448	–
Full-resolution							
Naive Rendering	27.56	**0.921**	0.112	0.091	**7.423**	0.244	–
R3D2 Ljungbergh et al. (2025)	27.61	0.904	0.072	0.055	11.320	0.206	4.07
Ours	**28.60**	0.914	**0.069**	**0.054**	9.955	**0.178**	**4.54**

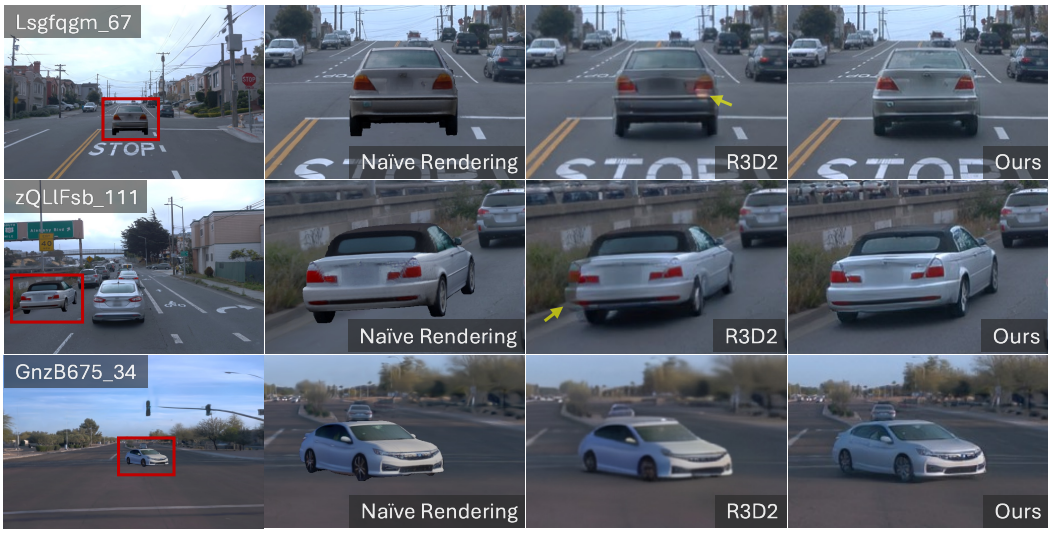


Figure 4: **Qualitative comparison of naive-to-real rendering results.** Naive rendering exhibits visible appearance artifacts and inconsistent shading around the inserted asset. R3D2 improves frame-level realism but may still produce local artifacts, blurred details, or lighting inconsistencies. In contrast, our method produces more photorealistic and scene-consistent results.

4.2 Experimental Results

4.2.1 Quantitative Comparison.

In Table 1, we report quantitative results on our validation set, covering both actor-centric crops and full-resolution frames. The actor-centric setting evaluates local rendering quality around the inserted asset, while the full-resolution setting measures global scene preservation and overall video realism.

As shown in the table, naive rendering preserves the inserted geometry but exhibits clear appearance discrepancies around the asset, leading to poor actor-centric perceptual scores. R3D2 improves rendering realism through image-level diffusion harmonization, but its frame-wise processing leads to weaker temporal consistency, as reflected by the larger warping error.

Our method achieves the strongest overall performance in perceptual realism and rendering fidelity. On actor-centric crops, it obtains the highest PSNR and SSIM and the lowest LPIPS and DISTS, demonstrating improved local rendering quality around the inserted asset. Although naive rendering achieves the lowest warping error due to minimal appearance modification, it lacks realistic lighting and material consistency. On full-resolution frames, our method achieves the best LPIPS, DISTS, and vFID, showing improved perceptual realism and video-level distribution alignment while preserving

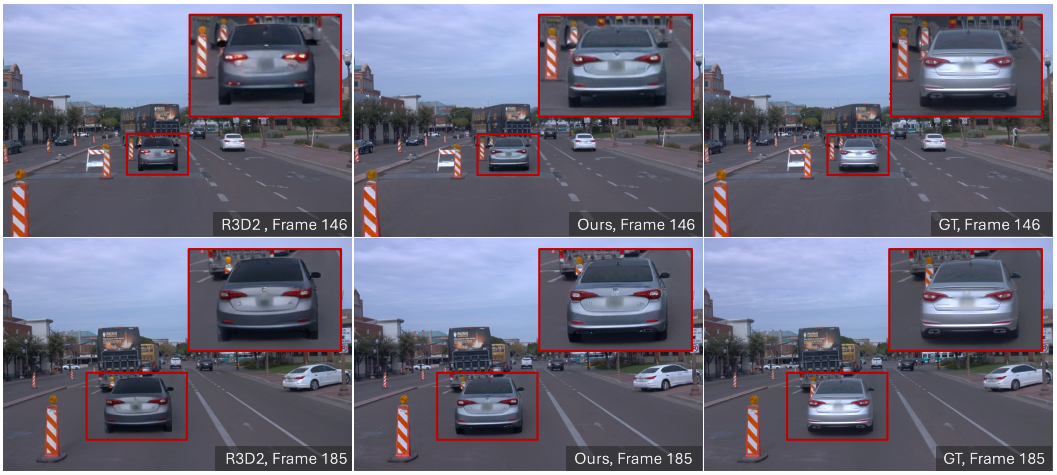


Figure 5: **Temporal consistency comparison.** We retrain R3D2 on DriveInsert-3K to reduce domain differences before evaluating temporal behavior. Despite using the same paired supervision, R3D2 may exhibit appearance variations across frames, including changes in vehicle shape, color tone, and reflectance. In contrast, our method maintains more stable asset appearance over time and produces results that remain closer to the ground-truth video.

global scene structure. Moreover, our method achieves slightly higher FPS than R3D2, indicating efficient single-step inference.

4.2.2 Qualitative Comparison.

Figure 4 presents qualitative comparisons among naive rendering, R3D2, and our method across challenging driving scenes. Naive rendering often produces clear visual discontinuities between the inserted vehicle and the surrounding environment, including missing shadows, inconsistent illumination, and boundary artifacts. R3D2 reduces some of these inconsistencies through diffusion-based frame-level harmonization, but it can still introduce local artifacts, blurred reflections, and unstable shading around the inserted asset.

In contrast, our video diffusion harmonizer generates more photorealistic and scene-consistent results. It better adapts the inserted asset to the surrounding illumination and road context, producing a more natural object appearance and local interactions.

4.2.3 Temporal Consistency Comparison.

Figure 5 compares the temporal consistency of R3D2 and our method across video frames. R3D2 improves individual frames but can vary the vehicle's shape, color, and reflectance across frames. Our video diffusion harmonizer uses temporal priors to maintain a more coherent appearance and produce results closer to the ground truth. Additional analyses, ablation studies, and qualitative results are provided in the appendices.

5 Conclusion

In this paper, we presented a single-step video diffusion harmonizer for photorealistic and temporally coherent rendering of 3D asset insertions in driving scenes. Our method improves naive 3D insertion renderings by leveraging a pretrained video diffusion prior, while preserving the geometry specified by the 3D insertion pipeline. To enhance spatial fidelity, we introduced a temporally agnostic latent injection strategy that injects high-frequency 2D features into causal 3D VAE decoding. We further developed a 3D-to-2D alignment strategy and constructed DriveInsert-3K to provide clean paired supervision for naive-to-real rendering. Extensive experiments demonstrate that our method improves appearance realism, scene consistency, and temporal stability across diverse driving scenarios. We hope this work provides a practical step toward controllable and realistic 3D asset insertion for driving simulation and data augmentation.

AI USE STATEMENT

We used generative AI tools to assist with language editing and improving the clarity of the presentation. All technical content, experimental results, numerical values, and citations were independently reviewed and verified by the authors. Generative AI was not used to design experiments, generate results, select baselines, or formulate scientific conclusions.

ETHICS STATEMENT

This work studies photorealistic insertion of 3D assets into driving videos for simulation and evaluation. The experiments use the Waymo Open Dataset and pretrained models described in the paper. Realistic video editing can also be misused to create misleading driving footage or synthetic sensor data. We encourage clear labeling and provenance of generated content, as well as responsible use of the underlying datasets and models.

REPRODUCIBILITY STATEMENT

The main paper describes the one-step video diffusion harmonizer, temporally agnostic latent injection, the training objectives, and the construction of DriveInsert-3K. Section 4 specifies the training setup, evaluation metrics, and baselines. The appendix provides further details of dataset construction, ablations, and additional comparisons to support reproduction of the reported results. Code and model will be released publicly upon publication.

REFERENCES

Zheng Chen, Zichen Zou, Kewei Zhang, Xiongfei Su, Xin Yuan, Yong Guo, and Yulun Zhang. Dove: Efficient one-step diffusion model for real-world video super-resolution. *arXiv preprint arXiv:2505.16239*, 2025.

Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020.

Georg Hess, Carl Lindström, Maryam Fatemi, Christoffer Petersson, and Lennart Svensson. Splatad: Real-time lidar and camera rendering with 3d gaussian splatting for autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11982–11992, 2025.

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 170–185, 2018.

Ruofan Liang, Zan Gojcic, Merlin Nimier-David, David Acuna, Nandita Vijaykumar, Sanja Fidler, and Zian Wang. Photorealistic object insertion with diffusion-guided inverse rendering. In *European Conference on Computer Vision*, pp. 446–465. Springer, 2024.

Yiyuan Liang, Zhiying Yan, Liqun Chen, Jiahuan Zhou, Luxin Yan, Sheng Zhong, and Xu Zou. Driveeditor: A unified 3d information-guided framework for controllable object editing in driving scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5164–5172, 2025.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

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. *arXiv preprint arXiv:2506.07826*, 2025.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

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.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. MIT Press, 2023.

Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *European Conference on Computer Vision*, pp. 250–266. Springer, 2022.

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

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.

Yujing Sun, Lingchen Sun, Shuaizheng Liu, Rongyuan Wu, Zhengqiang Zhang, and Lei Zhang. One-step diffusion for detail-rich and temporally consistent video super-resolution. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in neural information processing systems*, 36:8406–8441, 2023.

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.

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*, pp. 26024–26035, 2025.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21469–21480, 2025.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024.

Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

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 *The Thirteenth International Conference on Learning Representations*, 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 conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhixing Zhang, Yanyu Li, Yushu Wu, Anil Kag, Ivan Skorokhodov, Willi Menapace, Aliaksandr Siarohin, Junli Cao, Dimitris Metaxas, Sergey Tulyakov, et al. Sf-v: Single forward video generation model. *Advances in Neural Information Processing Systems*, 37:103599–103618, 2024.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024a.

Pengfei Zhou, Fangxiang Feng, Guang Liu, Ruifan Li, and Xiaojie Wang. Diffharmony++: Enhancing image harmonization with harmony-vae and inverse harmonization model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 10592–10601, 2024b.

APPENDIX

This appendix is organized as follows:

- Additional ablation studies on VAE adaptation, DriveInsert-3K, and training configurations (Section A).
- Additional motivation and failure analysis of R3D2 (Section B).
- Temporal behavior analysis of the causal 3D VAE (Section C).
- Details of DriveInsert-3K construction and Gaussian distribution mismatch analysis (Section D).
- Cross-setting comparison with 2D driving scene editing (Section E).
- Qualitative video demos highlighting temporal consistency and controllable asset insertion (Section F).
- Limitations and broader impact discussion (Section G).

A ABLATION STUDY

A.1 VAE ADAPTATION STRATEGIES

Table 2: Ablation study on VAE adaptation strategies for video reconstruction. "Skip Con." denotes skip connections; "2D Inj." denotes injecting 2D encoder features.

Method	PSNR↑	LPIPS↓	vFID↓
2D VAE	30.30	0.040	0.112
2D VAE (w. Skip Con.)	32.70	0.043	0.106
3D VAE	33.98	0.041	0.080
3D VAE (w. 2D Inj.)	**35.07**	**0.039**	**0.079**

Table 2 analyzes the impact of different VAE adaptation strategies on reconstruction fidelity and perceptual realism. The 2D VAE baseline achieves the weakest overall performance, while switching to a 3D VAE substantially improves PSNR and vFID, indicating the benefit of spatiotemporal latent modeling for video reconstruction.

We further evaluate two feature-enhanced designs. Adding skip connections to the 2D VAE improves PSNR but does not improve LPIPS, suggesting that direct feature passing mainly benefits pixel-level reconstruction. In contrast, injecting 2D encoder features into the 3D decoder ("3D VAE w. 2D Inj.") achieves the best results across all metrics. This shows that our temporally agnostic latent injection restores fine spatial details while retaining the video modeling advantages of the 3D VAE.

A.2 DRIVEINSERT-3K ABLATION

Table 3 evaluates the effect of our aligned dataset on R3D2 training. Compared with the original R3D2 setting, training R3D2 on DriveInsert-3K improves PSNR from 27.61 to 28.31 and reduces vFID from 0.206 to 0.193, while maintaining the same LPIPS score. These results indicate that the aligned naive-insertion/real-video pairs provide cleaner supervision, improving reconstruction quality and video-level realism. This ablation validates the importance of dataset alignment for learning naive-to-real rendering effects.

A.3 ABLATION ON TRAINING CONFIGURATIONS

Table 4 evaluates the contribution of different training configurations. Using only the harmonization stage achieves a PSNR of 27.89 and a vFID of 0.206. Adding direct 3D skip connections degrades all metrics, suggesting that naive feature passing is not well suited to the causal 3D decoder. In contrast, combining VAE adaptation with harmonization training achieves the best performance across all metrics, improving PSNR to 28.60, LPIPS to 0.069, and vFID to 0.178. These results validate the importance of the VAE adaptation stage for improving reconstruction fidelity and video-level realism.

Table 3: Effect of DriveInsert-3K on R3D2 training.

Method	PSNR↑	LPIPS↓	vFID↓
R3D2	27.61	0.072	0.206
R3D2 (w. DriveInsert-3K)	**28.31**	0.072	**0.193**

Table 4: Ablation study on training configurations. Stage A denotes VAE adaptation, Stage H denotes harmonization training, and 3D Skip denotes direct 3D skip connections.

Configuration			PSNR↑	LPIPS↓	vFID↓
Stage H	Stage A	3D Skip			
✓	×	×	27.89	0.083	0.206
✓	×	✓	27.50	0.099	0.249
✓	✓	×	**28.60**	**0.069**	**0.178**

B Motivation of Our Work

Our work is motivated by two observations in existing asset reinsertion pipelines. First, image-level harmonization methods such as R3D2 process each frame independently, which can make it difficult to maintain stable asset appearance across video frames. This motivates our video-level formulation for temporally coherent naive-to-real rendering.

Second, existing reinsertion pairs may contain geometric misalignment between the inserted asset and the corresponding ground-truth object. As shown in Figure 6, such misalignment introduces supervision noise: the difference between the naive rendering and the real video reflects not only rendering artifacts but also pose and scale errors. This motivates our 3D-to-2D alignment strategy and the construction of DriveInsert-3K for cleaner paired supervision.

Together, these observations motivate our pipeline for geometry-aligned and temporally coherent diffusion rendering of 3D asset insertions in driving scenes.

C Understanding the Temporal Behavior of the 3D VAE

To better understand why direct 3D skip connections are not ideal for our setting, we analyze the temporal behavior of the causal 3D VAE using a 9-frame input clip. Table 5 summarizes the feature shapes across encoder and decoder stages. The encoder gradually compresses temporal support, while the decoder progressively restores frame-level outputs. This hierarchy allows us to inspect whether encoder and decoder features at the same temporal index carry compatible temporal information.

We first visualize feature maps from corresponding encoder and decoder layers. As shown in Figure 7, the activation map at $T = 7$ in Enc-1 shows responses not only at Frame 7 but also at nearby frames such as Frame 5. This indicates that encoder features can carry mixed temporal support. In contrast, the corresponding high-resolution decoder feature at $T = 7$ is mainly concentrated around Frame 7, suggesting that the decoder feature is more frame-aligned at this stage. Directly injecting encoder features into such decoder layers may therefore introduce temporally mixed information into frame-aligned representations, leading to frame-inconsistent artifacts.

We further observe that the high-resolution decoder stages, such as Dec-4 and Dec-3, are largely aligned with the output frame index. For example, at $T = 7$, the dominant activation corresponds to Frame 7. This observation motivates our temporally agnostic latent injection strategy: instead of passing temporally mixed 3D encoder features, we inject frame-wise 2D features extracted independently from each input frame into frame-aligned decoder layers. This design improves spatial fidelity while avoiding additional temporal dependencies.



Figure 6: **Geometric misalignment in existing reinsertion pairs.** Horizontal and vertical reference lines, together with arrows, highlight pose and scale misalignment between naively inserted assets and the corresponding ground-truth objects. The examples are taken from the supplementary material of R3D2 Ljungbergh et al. (2025).

Table 5: Feature shapes of the 3D VAE follow the $[C, T, H, W]$ layout for a 9-frame input clip. The encoder gradually compresses the spatial and temporal dimensions, while the decoder progressively restores them.

Encoder		Decoder	
Stage	Feature Shapes	Stage	Feature Shapes
Enc-1	$[128,\ 9,\ H,\ W]$	Dec-4	$[256,\ 9,\ H,\ W]$
Enc-2	$[128,\ 5,\ H/2,\ W/2]$	Dec-3	$[256,\ 9,\ H/2,\ W/2]$
Enc-3	$[256,\ 3,\ H/4,\ W/4]$	Dec-2	$[512,\ 5,\ H/4,\ W/4]$
Enc-4	$[256,\ 3,\ H/8,\ W/8]$	Dec-1	$[512,\ 3,\ H/8,\ W/8]$

D More Details about DriveInsert-3K

For each driving scene, we construct DriveInsert-3K in three main stages: scene reconstruction and object selection, asset generation and alignment, and background–foreground compositing, as shown in Figure 8.

D.1 Distribution Mismatch Between Scene Gaussians and Asset Gaussians

Although scene reconstruction provides object-centric Gaussians, their distributions can differ substantially from those of generated assets. As shown in Figure 9, raw point clouds from 3DGS exhibit irregular density patterns, and the extracted object Gaussians can be noisy and anisotropic. In contrast, asset Gaussians generated by Trellis Xiang et al. (2025) tend to be more uniformly distributed. This distribution mismatch makes direct alignment between the generated asset and the extracted object Gaussians challenging, since their centers, scales, and density patterns are often geometrically incompatible.

To address this issue, DriveInsert-3K uses a 3D-to-2D alignment strategy. We first perform coarse 3D alignment using the reconstructed scene geometry and camera poses, followed by temporally consistent 2D refinement based on bounding-box edges. This combination compensates for distribution disparities and yields more stable alignment across frames.

D.2 Scene Reconstruction and Object Selection

We begin with a ground-truth driving video and reconstruct its 3D Gaussian Splatting scene using SplatAD Hess et al. (2025). For each frame, we obtain per-object segmentation masks and temporal tracks using Grounding-SAM2 Ren et al. (2024). Among all tracked instances, we select the vehicle

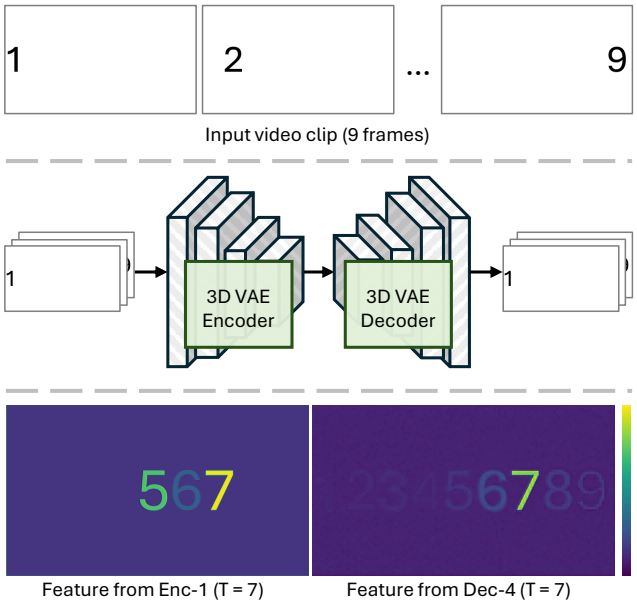


Figure 7: **Visualization of temporal behavior in the causal 3D VAE.** Top: input 9-frame video clip. Middle: encoder-decoder hierarchy. Bottom: activation maps from Encoder Stage 1 and Decoder Stage 4 at $T = 7$, obtained by averaging activations across channels. Encoder features may contain mixed temporal support, while high-resolution decoder features are more frame-aligned.

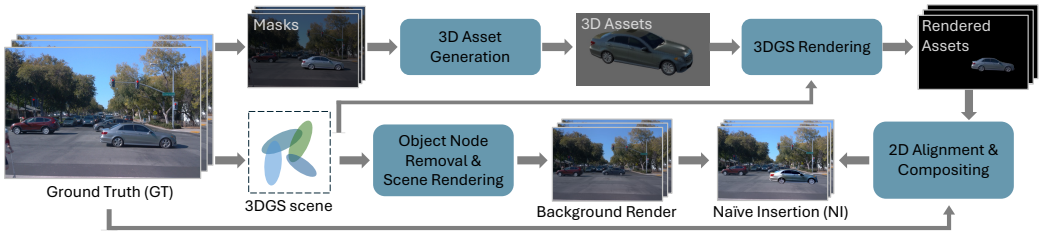


Figure 8: **DriveInsert-3K data generation pipeline.** Starting from ground-truth (GT) driving videos and their 3DGS scene reconstructions, we generate a 3D asset from the target vehicle and align it to the scene using a 3D-to-2D alignment strategy. After removing the original vehicle from the scene, the aligned asset is composited onto the rendered background to produce naive insertion (NI) frames, forming NI–GT training pairs for our harmonizer.

whose segmentation masks remain fully inside the image and have the largest average mask area. This selected vehicle serves as the reference object for constructing one paired naive-insertion/real-video sample. We then crop an image patch around the selected vehicle and use it as the reference for asset generation.

D.3 Asset Generation and 3D-to-2D Alignment

Using the object masks, we identify the actor node in the 3DGS representation and extract the Gaussians associated with the selected vehicle, forming an object-centric Gaussian subset. On the asset side, we use Trellis Xiang et al. (2025) to generate a 3D vehicle asset from the cropped vehicle image. The generated asset is first rendered under the original camera poses to produce an initial asset sequence.

We then align the asset to the scene using the proposed 3D-to-2D alignment strategy. A coarse 3D alignment estimates a similarity transformation, including translation, rotation, and scale, to place the asset near the original vehicle in world space. Although this step provides approximate geometric

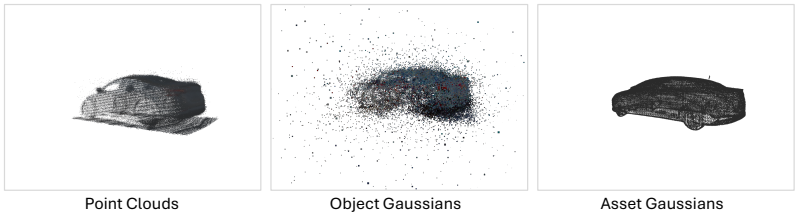


Figure 9: **Distribution mismatch between scene Gaussians and asset Gaussians.** Left: raw point clouds reconstructed from the driving scene. Middle: object Gaussians extracted from the 3DGS scene. Right: asset Gaussians generated by Trellis Xiang et al. (2025), which show a more uniform structure.

Table 6: Cross-setting comparison with DriveEditor. DriveEditor operates in a 2D editing setting and is included as a reference rather than a direct task-matched baseline.

Method	PSNR↑	LPIPS↓	DISTS↓	vFID↓
DriveEditor Liang et al. (2025)	26.06	0.188	0.133	0.293
R3D2 Ljungbergh et al. (2025)	27.61	0.072	0.055	0.206
Ours	**28.60**	**0.069**	**0.054**	**0.178**

consistency, residual misalignment may still appear in the image plane due to the distribution mismatch between scene Gaussians and generated asset Gaussians.

We therefore perform a second refinement step in 2D space. Specifically, we compare the rendered asset bounding boxes with the annotated bounding boxes of the original vehicle from the Waymo Open Dataset Sun et al. (2020). A global 2D affine transformation is estimated from the averaged bounding-box differences across frames, so that displacement and scale are adjusted consistently over the sequence. This refinement improves frame-level spatial alignment and reduces geometric noise in the resulting NI–GT pairs.

D.4 Background Rendering and Pair Construction

To obtain clean backgrounds, we remove the corresponding vehicle node from the 3DGS scene and re-render the scene. The aligned asset is then composited onto the rendered background frames to form the naive insertion video x_{NI}. This video preserves the original scene geometry and camera motion but lacks realistic rendering effects such as lighting interaction, shadows, and material consistency. Finally, we pair x_{NI} with the original real video x_{GT}, where x_{GT} serves as the realistic reference for naive-to-real rendering supervision.

We construct one paired sample for each selected target asset, which keeps the supervision focused on a single inserted object and avoids mixing multiple replacement effects in the same training pair. This data construction is used only for training. At inference time, our harmonizer is not restricted to this pipeline and can take any upstream naive insertion video as input. Because asset rendering is decoupled from the reconstructed background, the pipeline is compatible with different 3D asset generation methods.

E Comparison with 2D Driving Scene Editing

Our main experiments focus on 3D asset insertion harmonization, where the input is a geometry-controlled naive rendering produced by a 3D insertion pipeline. This setting differs from 2D driving scene editing methods, which typically operate directly in the image or video domain. While 2D editing methods are flexible and can modify visual content without explicit 3D reconstruction, they do not naturally provide explicit 3D asset control, multi-view geometry, or rendering under a prescribed camera trajectory. Conversely, 3D asset insertion provides precise geometry and camera control, but its naive renderings often suffer from lighting, material, and shadow inconsistencies. Our method

is designed for this latter setting: it improves the rendered appearance of geometry-controlled 3D insertions rather than directly editing pixels from scratch.

Many recent driving video editing methods do not release code or pretrained models, making direct comparison difficult. For completeness, we compare with DriveEditor Liang et al. (2025), a representative open-source method for driving scene editing. Since DriveEditor is not designed for our naive-to-real 3D insertion harmonization setting, this comparison should be interpreted as a cross-setting reference rather than a direct task-matched baseline. As shown in Table 6, DriveEditor can perform object-level visual editing, but it struggles to preserve the geometry and appearance required by our paired 3D insertion setting. In contrast, our method leverages the geometry provided by the 3D insertion pipeline and focuses on harmonizing the rendered appearance, leading to better perceptual quality and scene consistency.

F Video Demo

We provide supplementary videos to showcase our results on additional scenarios from the test set. Please refer to *Demo-SequentialComparison.mp4*, *Demo-zU5gAXQ5.mp4*, and *Demo-O0ipbuwo.mp4* in the supplementary material.

- **Demo-SequentialComparison.mp4** presents a sequential comparison across three vehicle categories, including truck, car, and bus. For each row, the naive insertion sequence and our harmonized result are shown side by side under the same video background. By inserting different 3D assets into the same scene, this demo shows that our method preserves a stable background, produces visually consistent asset insertions, and maintains temporal coherence across frames.
- **Demo-O0ipbuwo.mp4** compares naive insertion, R3D2, our method, and the ground truth over a video sequence. It highlights that our video diffusion harmonizer produces more stable asset appearance and better frame-to-frame consistency than image-level harmonization.
- **Demo-zU5gAXQ5.mp4** presents a challenging repositioning scenario where the front vehicle performs a U-turn, corresponding to the teaser example in the main paper. This demo illustrates controllable 3D asset editing with realistic and temporally coherent rendering under large pose changes.

G Limitations and Broader Impact

Limitations. Our method is designed to harmonize geometry-controlled but visually naive 3D asset insertions in driving videos, and relies on reasonably accurate upstream 3D reconstruction, asset placement, object masks, and camera calibration. Severe pose misalignment, heavy occlusion, extreme weather or lighting conditions, or asset categories far outside the training distribution may reduce appearance realism and temporal stability. The current training and evaluation are conducted on 9-frame clips at 512×768 resolution, so longer-horizon behavior and higher-resolution deployment require further validation.

Broader Impact. Photorealistic 3D asset insertion can improve autonomous-driving data coverage by creating diverse and safety-relevant scenarios without collecting risky real-world events. This capability may benefit simulation, robustness evaluation, and data augmentation for vision-centric driving systems. However, realistic video harmonization could also be misused to create misleading driving videos or synthetic sensor data. Careful documentation, provenance information, usage guidance, and evaluation protocols would help ensure that synthetic videos are used responsibly and distinguished from real sensor data.