DRIVENVS: GENERATIVE NOVEL VIEW SYNTHESIS FROM MONOCULAR DRIVING VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative novel view synthesis (NVS) methods for dynamic scenes typically reconstruct a full 4D representation of the scene and then generate refined video outputs using a video diffusion model (VDM), conditioned on the rendered views of the 4D structure. However, conventional VDMs require extensive training, exhibit significant memory inefficiency, and remain constrained to generating short video sequences. We present **DrivENVS**, an efficient alternative to VDM-based enhancement that employs a few-step image-diffusion model to sequentially refine coarse rendered views into a realistic and temporally coherent video. A key architectural novelty is an efficient conditioning mechanism that involves leveraging several reference frames selected from the source video using a reconstruction-aware strategy. Incorporating relative camera pose and capture time-step information for reference frames also provides useful spatio-temporal context. Across nuScenes, Waymo, and Argoverse 2, DrivENVS outperforms baseline methods on video quality metrics in challenging NVS settings, while also providing memory-efficient inference and enabling the generation of extended video sequences.

1 INTRODUCTION

Dashcam videos are a rich source of diverse traffic scenarios that are useful for developing autonomous driving systems. However, these videos are typically monocular, differ in calibration from multi-camera rigs mounted on autonomous vehicles (AVs) and capture the scene from a very different perspective, resulting in an embodiment gap for AVs (Wang et al., 2026). To reduce the gap, these videos must be re-rendered with a different calibration and from camera viewpoints along a new trajectory in the scene. This task requires a solution for novel view synthesis (NVS) from monocular videos for dynamic scenes – an actively explored research challenge.

Generative NVS is currently the dominant approach for this setting. Broadly, these approaches first run 4D reconstruction to recover scene representations that are used to render novel views. However, these views tend to have visual artifacts, and a video diffusion model (VDM) with geometric conditioning mechanisms is often used to fix the artifacts and generate the final photorealistic video. VDMs are expensive to train, need massive data curation efforts, and have high runtime memory usage. Furthermore, single-pass generation is limited to short sequences (typically ≈ 5 seconds), while multi-pass generation for longer videos often lacks spatiotemporal consistency.

Efficient image diffusion-based methods exist for fixing NVS artifacts (Wu et al., 2025; Liu et al., 2024; Wei et al., 2026). Specifically, Wu et al. (2025) proposes an effective reference mixing layer that enables reference image conditioning, enabling effective removal of rendered artifacts in novel views. However, these methods cannot handle dynamic scenes. In contrast, autoregressive methods (Zhang et al., 2026; Wang et al., 2026) can generate long videos of dynamic scenes but lack the conditioning mechanisms needed for geometry-grounded generative NVS.

In this paper, we show that the memory-intensive VDMs used in generative 4D NVS can be replaced by an image diffusion architecture that (1) generates *long, temporally coherent videos* via frame-wise generation with a fixed-length temporal context, (2) is grounded in 4D reconstruction but with a novel *reference-frame conditioning mechanism*, and (3) enables *memory efficient* inference.

We present **DrivENVS**, a spatiotemporally-coherent framework for generative 4D NVS in driving scenes. DrivENVS integrates a practical motion-aware 4D reconstruction front end with an au-

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.

toregressive image-diffusion neural enhancer. From a monocular video, our reconstruction stage, tailored for dynamic driving scenes, uses off-the-shelf foundation models to infer moving geometry, aggregate static geometry over time, and renders coarse views from a novel camera trajectory. The neural enhancer refines the coarse views by correcting rendering artifacts and filling in occluded regions. To exploit dense, smoothly varying views of the scene from many viewpoints common in driving videos, the enhancer not only conditions on coarse renderings and past generated frames for 4D consistency, but also incorporates novel conditioning mechanisms for leveraging source video frames, their poses, and timesteps, as a strong generative prior for spatio-temporal scene context.

This paper presents the following technical contributions:

1. A new **geometric multi-reference conditioning** mechanism based on using the 3D reconstruction to rank source video frames by their relevance to a specific target view and selecting top-k ($k \in [8, 16]$) frames for reference frame conditioning. We propose jointly processing the top-1 reference along with the target tokens to anchor scene structure and appearance. To exploit multiple reference frames without the quadratic cost of joint attention over all frames, we propose an efficient cross-attention mechanism to inject information from multiple posed reference frames into the diffusion backbone.
2. A **relative camera-time conditioning** for reference frames to explicitly encode each reference frame's pose and timestamp relative to the target view is introduced. These conditioning signals make the image-diffusion architecture geometrically and temporally-aware during appearance transfer from source frames at different timesteps, reduce the probability of inconsistent fusion of dynamic geometry, and encourages generated videos faithful to the source videos.

We evaluate DrivENVS on NUSCENES Caesar et al. (2020), WAYMO Sun et al. (2020), and ARGOVERSE 2 Wilson et al. (2023) across translated and side-facing novel camera trajectories. Under controlled training with identical training data and backbones, DrivENVS reduces FVD by $\mathbf{28.8}\%$ and LPIPS by $\mathbf{21.8}\%$ on average relative to DIFIX (Wu et al., 2025) and DIFFUSIONHARMONIZER (Zhang et al., 2026). Despite using a compact 2B image-DiT backbone rather than the 14B video-DiT of NEOVERSE (Yang et al., 2026), DrivENVS achieves competitive quality while reducing peak inference memory from 88 GB to 20 GB (greater than a factor of **4**) and supporting long-video generation, demonstrating a practical alternative to large VDMs for generative 4D NVS.

2 RELATED WORK

4D Reconstruction and NVS. Early NVS methods render novel views primarily by warping and blending nearby posed input views (Debevec et al., 1996; Buehler et al., 2001; Sinha et al., 2009; Chaurasia et al., 2013; Penner & Zhang, 2017; Hedman et al., 2018), but require accurate geometric proxies and densely sampled multi-view images for 3D reconstruction. Newer methods recover novel views by learning NeRFs (Mildenhall et al., 2020) or optimizing 3D Gaussians (3DGS) (Kerbl et al., 2023) per scene, or by training deep architectures (Riegler & Koltun, 2020; Wang et al., 2021) to generalize across scenes. 3DGS extensions can incorporate LiDAR, model traffic scenes and dynamic road actors (Yan et al., 2025; Hess et al., 2025; Yang et al., 2025), and handle dynamic scenes (Tonderski et al., 2024; Wu et al., 2024a). ViT-based feedforward architectures can nowadays predict depth, camera pose, scene geometry, and 4DGS representations from uncalibrated videos (Lin et al., 2025; Wang et al., 2025a; Nguyen et al., 2026; Yang et al., 2026), making dynamic NVS from monocular videos increasingly practical (Chen et al., 2025; Yu et al., 2025).

Image Diffusion and Generative NVS. Image diffusion models have been used to correct reconstruction artifacts and synthesize plausible content for unobserved regions (Liu et al., 2024; Wu et al., 2025; Fischer et al., 2025; Dong et al., 2026; Xu et al., 2026; Wei et al., 2026; Fan et al., 2025). These approaches range from efficient single- or few-step models (Wu et al., 2025; Parmar et al., 2024; Dong et al., 2026) to latent diffusion architectures augmented with geometric controls (Wei et al., 2026; Fan et al., 2025). Wu et al. (2025), in particular, jointly processes a rendered target view and a source reference, providing effective appearance guidance for single-frame refinement. However, independently refining each frame does not enforce temporal consistency, while extending joint attention to multiple references substantially increases computation and memory. Zhang et al. (2026) addresses temporal consistency by autoregressively conditioning the current frame on previously generated frames and introducing flow-based supervision. However, its conditioning is designed for

image harmonization rather than NVS across changing viewpoints and lacks reconstruction-derived visibility, cross-view correspondence, and relative camera-time cues. Wang et al. (2026) similarly generates videos autoregressively, translating monocular dashcam footage into a predefined multi-camera AV sensor suite. However, it does not use an explicit 4D reconstruction to ground inference. Moreover, its target-camera calibration is encoded by a model trained on a large, specifically curated dataset, tying generation to a predefined sensor rig. *Thus, existing image-diffusion methods either lack NVS-specific geometric reasoning or remain tied to a fixed target-camera rig.* DrivENVS addresses these limitations through explicit 4D grounding, geometry-aware multi-reference conditioning and relative camera-time conditioning, enabling generation along user-specified camera trajectories without the need to retrain for a new camera rig.

Video-Diffusion and NVS. Existing VDM-based methods fall into two categories. **The first group** uses video priors to refine novel-view sequences rendered from static 3D representations (De Lutio et al., 2026; Vuong et al., 2026; Yu et al., 2024; Liu et al., 2024). As coarse renderings encode geometry and camera motion, these methods can improve visual quality and multi-view consistency. *However, their static-scene formulation makes them unsuitable for time-varying geometry and dynamic 4D NVS*. **The second group** conditions VDMs on rendered views of a dynamic 4D reconstruction from target views (Ren et al., 2025; Zhu et al., 2026; Yang et al., 2026), while jointly generating multiple frames to encourage spatio-temporal consistency. Other conditioning signals include source video frames (Yu et al., 2025; Lin et al., 2026), point-cloud depth (Wang et al., 2025b), Plücker ray embeddings, and joint camera-time encodings (He et al., 2024; Wu et al., 2024b). *Despite their benefits, VDM backbones are expensive to train and deploy*. Bidirectional VDMs jointly denoise fixed-length spatiotemporal windows to reduce temporal inconsistency, resulting in high memory usage. Causal variants (De Lutio et al., 2026) support longer sequences but require additional distillation and remain computationally expensive. Training them also requires extensive synthetic or real multi-view footage (Bai et al., 2025), or per-scene preprocessing (Yu et al., 2025). DrivENVS instead uses an efficient 4D reconstruction method to curate training pairs directly from monocular videos and needs neither synchronized multi-view videos nor costly preprocessing.

3 METHOD

Given an input video $\mathcal{V} = \{(\mathbf{I}_i, \mathbf{K}_i, \mathbf{T}_i)\}_{i=1}^N$ containing N RGB frames, where $\mathbf{I}_i$, $\mathbf{K}_i$, and $\mathbf{T}_i$ denote the i-th frame, camera intrinsics, and extrinsic pose respectively, our method generates N RGB frames depicting the scene rendered from a user-specified camera trajectory denoted by $\{(\mathbf{K}'_i, \mathbf{T}'_i)\}_{i=1}^N$. First, a 4D scene pointcloud $\mathcal{P}$ is reconstructed from $\mathcal{V}$ and a coarse novel view $\widetilde{\mathbf{I}}_i$ is rendered from $\mathcal{P}$. Then, the neural enhancer f_θ refines the coarse view to yield the output frame,

$$\widehat{\mathbf{I}}_i = f_\theta \left(\widetilde{\mathbf{I}}_i, \mathbf{K}'_i, \mathbf{T}'_i, \mathcal{V}_i^r, \widehat{\mathbf{I}}_{<i} \right),$$

where, $\mathcal{V}_i^r \subseteq \mathcal{V}$ and $\widehat{\mathbf{I}}_{<i}$ denote reference frames and a few preceding generated frames respectively.

4D Reconstruction and Motion Analysis. Our 4D pointcloud $\mathcal{P}$ contains a static pointcloud $\mathcal{P}^s$ and dynamic points $\{\mathcal{P}_i^d\}_{i=1}^N$ where, $\mathcal{P}_i^d$ denotes dynamic points observed in the i-th frame. We use DA-3 (Lin et al., 2025) to first recover consistent depth maps from $\mathcal{V}$. Using the known calibration, we transform depth maps into per-frame pointclouds in world coordinates. Naively aggregating these pointclouds will produce motion trails for moving objects. Thus, we next segment out moving objects from the pointcloud and avoid fusing them across time. To segment moving object points, we use SAM 3 Carion et al. (2026) to segment and track vehicles and pedestrians in $\mathcal{V}$ and transfer the instance labels to the per-frame pointclouds. For each tracked instance, we project the per-frame 3D points associated with it into the bird's-eye-view (BEV) plane and compute centroids for the subset of points in each frame. We then calculate the geometric median of the set of centroids and subsequently compute the mean deviation between the centroids of the track and the track center. We classify the object as stationary when the deviation is $< \tau_{\text{mot}}$ or dynamic otherwise, where τ_{mot} is a threshold. Static scene and static object points are aggregated into $\mathcal{P}^s$, whereas dynamic points $\{\mathcal{P}_i^d\}_{i=1}^N$ remain time-indexed. To render the i-th novel view $\widetilde{\mathbf{I}}_i$ from $\mathcal{P}$, the points $\mathcal{P}^s \cup \mathcal{P}_i^d$ are fused on the fly and projected into the target view with standard z-buffering. This simple method enables good static scene completion while avoiding motion trails for moving objects.

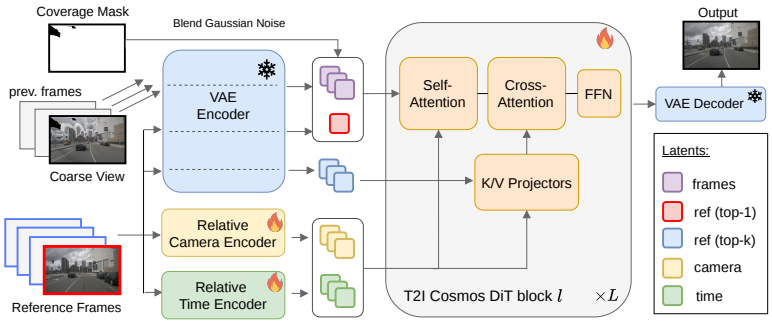

Figure 1: **Neural Enhancer Architecture.** The inputs are the coarse rendered view, a coverage mask, a set of reference frames, and previously generated frames. Our model exploits several posed reference images and injects latents derived from them into the self attention and cross attention layers of the diffusion backbone.

Training Data Curation via Double-Reprojection. To train the neural enhancer, we need coarse rendered novel views paired with ground truth images. Such image pairs are abundant in multi-view video footage, but unfortunately not in our monocular dynamic video footage. Inspired by (Yu et al., 2025; Ren et al., 2025) we design a driving-specific, two-pass double-reprojection strategy that leverages our motion-aware 4D reconstruction to construct pixel-aligned training pairs directly from monocular dynamic videos. For frame i, we sample a virtual target camera by perturbing its source camera pose and z-buffer-splat the corresponding point cloud $\mathcal{P}_i$ into the target view to obtain a visibility-aware rendering. Specifically, we compare per-point depth $d(p)$ for the point projected at p to $\mathbf{D}(q)$ for $q \in \mathcal{N}(p)$, where $\mathcal{D}$ is the target view's z-buffer and $\mathcal{N}$ denotes neighboring pixels of p. The points visible in the target view are projected back into the source camera view to produce a *virtual* coarse rendered view that is pixel-aligned with the source video frame. To fill holes in the coarse view by leveraging past and future frames, in a second pass, we project the static points $\mathcal{P}^s$ to fill as many holes as possible. A binary coverage mask $\mathbf{C}_i$, whose value is one at covered pixels and zero at holes, is also exported. These $(\mathbf{C}_i, \widetilde{\mathbf{I}}_i, \mathbf{I}_i)$ tuples are computationally efficient to compute, enabling practical dataset curation and preprocessing. Details are described in the Appendix A.3

3.1 Efficient Few-step Neural Enhancer

In this section, we present the design of our neural enhancer (Fig. 1), which builds on HARMONIZER (Zhang et al., 2026). However, HARMONIZER was designed mainly for enhancing synthetic imagery inserted into real videos. Thus, it was not explicitly designed to encode view-dependent effects and temporal changes in scene appearance observed across multiple video frames, a factor that is crucial for accurate novel view synthesis (NVS). Thus, in our neural enhancer architecture, we retain HARMONIZER's efficient T2I backbone, its design for temporal context conditioning for sequential frame generation, as well as the two-phase training schedule that first trains a single-frame enhancer, and then finetunes it along with temporal loss computed from preceding frames.

We propose three additions to make our neural enhancer more suitable for NVS. First, we replace the single-step image enhancer with a few-step enhancer based on *rectified flow*. Second, we propose a novel *geometric multi-reference conditioning* mechanism aimed at efficiently leveraging multiple relevant source video frames to transfer scene appearance from them to the target views. Third, we add *relative camera-time conditioning* to let the model exploit camera pose and time information for the reference views relative to the target view.

Few-step Enhancer using Rectified Flow. Most single-step diffusion models Wu et al. (2025); Zhang et al. (2026) directly predict the final image from a coarse image using a fixed diffusion timestep τ, but do not distinguish valid pixels from holes in the coarse image. De Lutio et al. (2026) proposed *opacity mixing* to blend Gaussian noise to video latents in unobserved regions and refined these blended latents using a video diffusion model. We adopt this idea to image latents and single frame generation, using our binary coverage mask $\mathbf{C}_i$ from point rendering as the opacity mask.

Let $\mathcal{E}$ denote the frozen VAE encoder, and $\widetilde{\mathbf{z}}_i = \mathcal{E}(\widetilde{\mathbf{I}}_i)$ and $\mathbf{z}_i = \mathcal{E}(\mathbf{I}_i)$ denote the coarse and clean latents, respectively. We resize the coverage mask $\mathbf{C}_i$ to the latent resolution, which we denote as $\mathbf{A}_i$, sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and then obtain the blended coarse latents $\mathbf{z}_i^{\text{init}} = \mathbf{A}_i \odot \widetilde{\mathbf{z}}_i + (1 - \mathbf{A}_i) \odot \boldsymbol{\epsilon}$.

Instead of single-step diffusion, we formulate our task of refining the blended latents $\mathbf{z}_i^{\text{init}}$ into clean latents $\mathbf{z}_i$ using *rectified flow* (Liu et al., 2023). The inputs to the rectified flow model $\mathbf{v}_\theta(\mathbf{z}, \tau, \mathbf{c})$ are the latent $\mathbf{z}$, timestep τ, and condition $\mathbf{c}$, and it predicts a velocity vector in latent space that is added to $\mathbf{z}$ to perform an iterative update. Training rectified flow is simpler when paired data is available. For each $\mathbf{z}_i^{init}$ in the source distribution, we know its matching $\mathbf{z}_i$ in the target distribution. Thus, we sample interpolated latents and velocities from ($\mathbf{z}_i^{init}$, $\mathbf{z}$) for a random $\tau \sim p_{\text{train}}$, as follows.

$$\mathbf{z}_{i,\tau} = (1-\tau)\,\mathbf{z}_i + \tau\,\mathbf{z}_i^{\text{init}}, \quad \mathbf{v}_i^* = \mathbf{z}_i^{\text{init}} - \mathbf{z}_i. \tag{1}$$

Following Albergo & Vanden-Eijnden (2022); Lipman et al. (2022); Liu et al. (2023), we supervise the predicted velocity for the interpolated latent samples by minimizing the following loss.

$$\mathcal{L}_{\text{RF}} = \mathbb{E}_{i,\tau,\boldsymbol{\epsilon}}\left[w(\tau)\,\|\mathbf{v}_\theta(\mathbf{z}_{i,\tau}, \tau, \mathbf{c}_i) - \mathbf{v}_i^*\|_2^2\right], \tag{2}$$

where in our case, $\mathbf{c}_i$ represents the reference frames, camera-time encodings, and temporal-context conditions, and $w(\tau)$ is the timestep-dependent loss weight. Each training sample supervises the flow for a single τ and multi-step updates are avoided at training. During inference, given the input latent $\mathbf{z}_i^{\text{init}}$, we sample four timesteps from $\tau = 1 \to 0$, predict velocities for those τ's and apply Euler integration steps to obtain the final latent which is then decoded by the VAE decoder.

Geometric Multi-Reference Conditioning. Exploiting source imagery to guide view synthesis is common in diffusion-based NVS Wu et al. (2025); Lin et al. (2026); Cao et al. (2026); Wang et al. (2025b), because these reference images act as a direct source for appearance transfer to the target view. Video models often use all source frames as reference via joint-attention to process source and reference tokens together across the DiT layers. However, using all frames as reference makes it harder for the model to find relevant tokens for a specific frame, hinders appearance transfer. Furthermore, the compute cost of joint self-attention increases quadratically with token count.

We tackle the first issue with a *reference frame selection strategy* that leverages geometric cues available in the precomputed 4D reconstruction. Specifically, we introduce a relevance score based on co-visibility of image pairs. The relevance of the j-th reference frame to the i-th target frame is defined as the number of 3D points in the j-th frame's pointcloud visible in the i-th view. These relevance scores are computed via point rendering and z-buffering steps during the initial scene reconstruction phase, and thus the rankings are precomputed before training and inference.

To efficiently handle multiple reference frames, the tokens from the most relevant (*top-1*) reference frame are added to joint self-attention and directly fused with source tokens, providing the model access to relevant appearance tokens. The remaining reference tokens are fused via cross-attention that allows adding many more tokens than would be possible with self-attention for a fixed compute budget. We obtain reference latents with the frozen VAE and map them into the target token space $\mathbf{F}_j = \mathcal{P}_\theta(\mathcal{E}(\mathbf{I}_j))$, with $\mathcal{P}_\theta$ the shared trainable patch embedder. The reference tokens are stacked into a matrix $\mathbf{F}$. The reference tokens supply keys and values via trainable KV projections, whereas the input tokens supply the queries. Given matrix $\mathbf{X} \in \mathbb{R}^{N_p \times d}$ denoting the target features of N_p d-dimensional tokens, we compute the feature update as follows (for each DiT block), d_k being the dimension of the key space.

$$\text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}; \quad \mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{F}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{F}\mathbf{W}_V. \tag{3}$$

Here, $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ denote Q, K, V weight matrices for cross attention layers. We fine-tune $\mathbf{W}_Q$ from the pretrained weights, while zero-initializing and learning $\mathbf{W}_K$ and $\mathbf{W}_V$ for reference latents. We also use PRoPE (Li et al., 2026) to encode relative pose between the target and reference views.

Relative Camera-Time Conditioning. Video diffusion models (VDMs) Lin et al. (2026); Yang et al. (2026); Vuong et al. (2026); Yu et al. (2025) process ordered sequences, use camera pose embeddings, and video VAEs, and thus learn temporal priors. In contrast, image diffusion models use independent conditioning per frame, and are unable to learn strong temporal priors. When a moving vehicle is seen in successive reference frames at different positions, the image models fail to treat it as a unique object, thus leading to ghosting and object duplication in the generated output.

To address this issue, we introduce relative camera and time embeddings to provide additional spatiotemporal context to the model, namely, (1) relative camera embeddings to encode each frame's

camera pose in the target camera's coordinates, and (2) relative time embeddings to encode its signed capture-time offset relative to the target frame. *This allows the enhancer to fuse multi-frame evidence while reasoning about both viewpoint and scene dynamics – a behavior that existing image diffusion NVS methods do not explicitly model*. We express the reference camera rays in the target camera's coordinate frame. For the j-th reference, the relative pose is $(\mathbf{T}'_i)^{-1}\mathbf{T}_j$, with relative rotation $\mathbf{R}_{ij}$ and translation $\mathbf{t}_{ij}$. For a pixel with homogeneous coordinates $\bar{\mathbf{u}}$, the 6D Plücker ray $\mathbf{r}_{ij}$, and relative camera embedding $\mathbf{X}^C_{ij}$ is computed as follows:

$$\mathbf{d}_{ij}(\mathbf{u}) = \mathbf{R}_{ij}\frac{\mathbf{K}_j^{-1}\bar{\mathbf{u}}}{\|\mathbf{K}_j^{-1}\bar{\mathbf{u}}\|_2}, \quad \mathbf{r}_{ij}(\mathbf{u}) = \begin{bmatrix}\mathbf{t}_{ij}\times\mathbf{d}_{ij}(\mathbf{u}) \\ \mathbf{d}_{ij}(\mathbf{u})\end{bmatrix}, \quad \mathbf{X}^C_{ij} = \mathbf{W}_C([\{\mathbf{r}_{ij}(\mathbf{u})\}_{\mathbf{u}\in p}])\,. \tag{4}$$

The six-dimensional Plücker rays of all pixels $\mathbf{u} \in p$ are concatenated into a vector in $\mathbb{R}^{6|p|}$ and projected by $\mathbf{W}^\ell_C \in \mathbb{R}^{d\times 6|p|}$ into one d-dimensional camera token. Applying this operation to all N_p image patches yields $\mathbf{X}^C_{ij} \in \mathbb{R}^{N_p\times d}$, matching the resolution and feature dimension of the DiT token sequence. These rays encode camera baselines and viewing directions in a shared frame. We flatten the rays within each image patch and map them to the DiT feature dimension through learned linear projections. Target tokens and the highest-ranked reference receive these embeddings before self-attention in each block, while the remaining references receive them before KV projection.

To encode relative time, we compute the signed temporal offset $\Delta t_{j\to i} = t^s_j - t^t_i$, where t^s_j and t^t_i are the reference and target frame timestamps and encode this offset using Fourier features:

$$\mathbf{X}^T_{ij} = \mathbf{W}_T\left(\begin{bmatrix}\cos(\boldsymbol{\omega_k}\Delta t_{j\to i}) \\ \sin(\boldsymbol{\omega_k}\Delta t_{j\to i})\end{bmatrix}\right),\ \omega_k = \exp\left(-\frac{k\log 10^4}{K_{\text{time}}-1}\right), k\in[0, K_{\text{time}})\,. \tag{5}$$

where K_{time} is the number of log-spaced Fourier frequencies and $\mathbf{W}_T$ is a learned linear projection. The resulting d-dimensional time embedding is broadcast across the N_p patch tokens of the j-th reference, yielding $\mathbf{X}^T_{ij} \in \mathbb{R}^{N_p\times d}$. We add each $\mathbf{X}^T_{ij}$ and $\mathbf{X}^C_{ij}$ both to the patch token features for the top-1 which are passed into the joint self-attention layers as well as to the patch token features for the top-k references which are passed into the key–value projectors for the cross-attention layers.

4 Experimental Evaluation

We evaluate DrivENVS in three settings. In the Lateral Shift setting, we generate videos from virtual camera positions obtained by shifting the source camera trajectory sideways. Specifically, we test six cases, where the virtual camera is moved 1 and 2 meters, to the left, right, and upward. In Stereo Shift, given calibrated stereo camera video, we evaluate our ability to generate one video from the other. In Side-View Synthesis, we evaluate side-facing camera videos generated from front-facing ones, with low overlap between source and target camera FoVs due to rotation.

Metrics. We evaluate Lateral Shift in terms of generative metrics (FID, KID) to compare distributions of generated and real video frames, and FVD to compare videos, as ground truth is unavailable. We also use VBench to assess temporal-consistency Huang et al. (2025). For Stereo Shift and Side-View Synthesis, aligned ground truth frames are available and used to obtain reconstructive metrics (PSNR, SSIM, LPIPS). See Appendix A.2. for more details

Datasets and Training. We evaluate on the nuScenes (Caesar et al., 2020), Waymo (Sun et al., 2020), and Argoverse 2 (Wilson et al., 2023) datasets, where the sequences (downsampled to 10 Hz) have 200, 150, and 75 frames respectively, with resolution 1024×576, 960×640, and 960×540 pixels, respectively. For all three datasets, we used 850 scenes for training and 150 scenes for testing. See Appendix for more details on data curation A.3 and training procedures A.1

4.1 Comparison to prior works

Baselines. We now report several comparisons with two closely related baselines – DiFix3D (Wu et al., 2025) and Harmonizer (Zhang et al., 2026). DiFix3D uses a single-step diffusion with a single reference image, while Harmonizer sequentially generates frames conditioned on past frames but lacks reference-frame conditioning. We retrained both baselines on identical data as our

Translation	Method	NuScenes				Waymo			
		FID ↓	KID ↓	FVD ↓	Motion ↑	FID ↓	KID ↓	FVD ↓	Motion ↑
None (0 m.)	Difix3D	1.472	0.0077	57.556	95.122	0.892	**0.0062**	48.179	95.899
	Harmonizer	3.412	**0.0075**	86.445	96.717	2.580	0.0063	69.431	96.623
	Ours	**1.239**	**0.0075**	**20.803**	**98.491**	**0.772**	**0.0062**	**13.282**	**98.699**
L, R, U (1 m.)	Difix3D	**4.934**	0.0058	132.731	95.043	**4.850**	**0.0058**	96.617	95.626
	Harmonizer	5.321	**0.0054**	139.622	96.691	5.876	0.0061	127.068	97.441
	Ours	6.020	0.0058	**106.489**	**98.216**	5.401	0.0063	**77.961**	**98.428**
L, R, U (2 m.)	Difix3D	**8.642**	0.0050	255.994	95.034	**8.659**	**0.0025**	196.433	95.531
	Harmonizer	9.784	**0.0039**	256.448	97.944	10.037	0.0030	239.721	97.402
	Ours	11.898	0.0078	**186.106**	**98.121**	9.937	0.0071	**150.522**	**98.303**

Table 1: **Lateral Shift Results.** Quality metrics for videos rendered from camera trajectories obtained by shifting the ego trajectory left (L), right (R), and up (U), by 0, 1 and 2 meters. Our method outperforms the baselines on two datasets by a wide margin on FVD and temporal metrics, and DIFIX has the best overall FID.



Figure 2: **Results (Waymo)**: Novel views rendered after shifting the source camera path in three directions.

method, after replacing their T2I Diffusion backbones with Cosmos-Predict2 T2I-DiT-2B NVIDIA (2025) to ensure a fair comparison. We also compare to the pretrained checkpoints for NEOVERSE (Yang et al., 2026).

Lateral Shift Results. Table 1 reports NVS results on 0, 1, and 2 m. camera shift. For FID, KID, and FVD, we pool the features from all left-, right-, and upward-shifted sequences before computing each metric. Motion Smoothness is averaged over all videos in the same combined pool. The 0 m. case has identical source and target cameras, and measures performance in the absence of parallax and disocclusions. The other two cases are progressively more difficult for modeling parallax and disocclusions. Here, our method consistently achieves the lowest FVD and highest VBench Motion scores. At 2 m., our FVD is 27.3% lower than DIFIX3D and 27.4% lower than HARMONIZER on nuScenes. On Waymo, the gains are 23.4% and 37.2%, respectively. Under large camera shifts, our FID and KID are not always better, showing that video optimization compromises image quality. However, FVD and motion metrics measure spatiotemporal coherence and are more relevant for high-quality video generation. See comparisons in Fig. 3 and detailed analysis in Appendix A.4.

Table 4 compares DrivENVS and the video-diffusion baseline NEOVERSE Yang et al. (2026) on the same 81-frame sequences under a 2 m. camera shift on NUSCENES. Our method reduces FID by 65.7% and FVD by 33.8%. It also reduces peak inference memory from 88 GB to 20 GB while generating at 3× higher resolution (1024×576 vs 560×336). Our model was trained using 4 GPUs for 4 days, compared to 32 GPUs for 5 days for NEOVERSE. As NEOVERSE can generate at most 81 output frames, we also generated 81 frames using DrivENVS, but our method can easily generate much longer sequences with inference memory footprint independent of sequence length. See Appendix A.4 for more details.

Stereo Shift. Table 2 reports NVS performance on Argoverse 2 stereo sequences, which enables rigorous evaluation using ground truth. Translating between the stereo cameras introduces horizontal

Method	Left → Right			Right → Left		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Difix3D	25.2681	0.8163	0.1597	25.3124	0.8160	0.1584
Harmonizer	24.7783	0.8128	0.1788	24.8514	0.8136	0.1765
Ours	**27.1940**	**0.8361**	**0.1427**	**27.1939**	**0.8372**	**0.1429**

Table 2: **Stereo Shift:** Ground truth-based accuracy metrics for stereo sequences (from Argoverse 2), where one sequence is generated from the other.

Method	Front → Right			Front → Left		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Difix3D	**18.3563**	0.5546	0.5353	**18.5171**	0.5510	0.5229
Harmonizer	18.0431	0.5448	0.5389	18.3622	0.5282	0.5423
Ours	17.6523	**0.6154**	**0.4136**	17.8557	**0.6023**	**0.4227**

Table 3: SIDE-VIEW SYNTHESIS: Accuracy metrics for front-right and front-left frames generated from the front camera frames on the nuScenes dataset.

Method	Quality		Resources	
	FID↓	FVD↓	Training Resources	Inference Memory↓
NeoVerse	40.41	312.38	32 GPUs / 5 days	88 GB
Ours	**13.85**	**206.88**	**4 GPUs / 4 days**	**20 GB**

Table 4: **Performance comparison** of our method vs. the pretrained models from NeoVerse on the nuScenes dataset (81 frames setting) for the 2 meter lateral camera shift NVS setting.

#	Ablation Components					Metrics			
	Time (Fourier)	*Camera (Plücker)*	*Top-8 Reference*	*Top-1 Reference*	*Temporal Fine-tuning*	FID↓	KID↓	FVD↓	Motion↑
(1)	✓	✓	✓	✓		**9.26**	0.0072	220.46	98.07
(2)				✓	✓	11.34	0.0093	178.65	98.13
(3)	✓	✓	✓		✓	12.12	0.0113	189.46	98.11
(4)			✓	✓	✓	10.97	0.0092	167.49	98.12
(5)	✓		✓	✓	✓	10.77	0.0082	164.22	98.12
(6)		✓	✓	✓	✓	10.61	0.0079	163.42	98.13
(7)	✓	✓	✓	✓	✓	9.94	**0.0071**	**150.52**	**98.30**

Table 5: **Effects of ablating** parts of our model on metrics.

parallax and disocclusions, requiring the model to preserve geometric alignment while completing regions visible only in the target view. Our method outperforms DIFIX3D and HARMONIZER on all metrics in this setting as well. PSNR improves over DIFIX3D by 1.88–1.93 dB and LPIPS improves by 9.8%–10.6%, while the higher gains are observed over HARMONIZER. Evaluating left-to-right and right-to-left also confirms that our model is not biased for a specific direction.

Side-View Synthesis. We generate side camera videos from the front on NUSCENES simulating a 55° camera rotation. Table 3 reports reconstructive metrics for both left and right side videos. Here, DIFIX3D has the best PSNR, although our method only trails by 0.66–0.70 dB. PSNR tends to favor baselines due to blur, whereas sharper frames get lower PSNR scores from small misalignment caused by depth and point cloud warping errors. However, our method excels on SSIM and LPIPS; achieving absolute SSIM gains of 0.051–0.061 and improving LPIPS by 19.2%–22.7%. DIFIX3D uses a single reference and HARMONIZER lacks reference-conditioning and struggles to accurately render objects not simultaneously visible in the source view, often generating plausible content but inconsistent with the source video. In contrast, our model gives better appearance transfer and generates more faithful novel views than the baselines.

4.2 ABLATION STUDIES / ADDITIONAL ANALYSIS

Our first ablation study (Table 5) evaluates model components in the Lateral Shift (2 m.) setting on WAYMO. Rows (2) and (4) show the impact of multi-reference conditioning and that additional references improve upon top-1 conditioning. Conversely, rows (3) and (7) show that using cross-attention alone without top-1 conditioning is ineffective. Although able to retrieve complementary local details, cross-attention uses reference tokens as key and value projections, limiting their interaction with the generated image. Top-1 conditioning instead jointly updates the target and reference in the DiT, providing iterative, bidirectional interactions that improve performance. Our design combines top-1 joint attention with the efficient multi-reference retrieval to avoid the cost of joint attention over all reference tokens. Rows (4)–(7) show that relative camera-time branches help. Camera conditioning disambiguates cross-view correspondence under viewpoint changes, while time conditioning distinguishes observations of dynamic objects captured at different moments, and their

# ref. frames	Quality			Memory / Speed		
	FID ↓	FVD ↓	Motion ↑	Train	Test	Time (fps)
1	10.93	173.20	98.12	44 GB	9 GB	1.67
4	**9.60**	157.85	98.26	89 GB	13 GB	0.83
8	9.94	**150.52**	**98.30**	137 GB	20 GB	0.63
16	10.19	153.57	98.29	184 GB	31 GB	0.45

Table 6: **Effect of ablating the number of reference frames on metrics, memory, and runtime.**

Reference frame selection strategy	Waymo (Lateral Shift 2m.)			nuScenes (Side View Synthesis)		
	FID ↓	FVD ↓	Motion ↑	PSNR ↑	SSIM ↑	LPIPS ↓
Curr + Prev 4 + Next 3	9.99	169.46	97.98	15.94	0.53	0.51
Curr + Prev 7	10.00	170.64	97.85	16.20	0.56	0.49
Curr + Next 7	**9.87**	168.95	98.16	15.54	0.52	0.54
3D Scene Aware	9.94	**150.52**	**98.30**	**17.79**	**0.61**	**0.42**

Table 7: **Effect of ablating the reference frame selection** strategy on metrics for two different NVS settings.



Figure 3: **Results (nuScenes).** Comparison of DrivENVS with Difix3D and Harmonizer baselines with the camera shifted up 1 m. (row 1) and right 2 m. (row 2). Difix3D duplicates moving vehicles and distorts the road and sidewalk. Harmonizer produces ghosting artifacts above the bus (row 1) and blurry foliage (row 2).



Figure 4: **Comparison with NeoVerse.** Qualitative results under a 2 m upward camera shift. DRIVEENVS preserves source-scene geometry, vehicle identities, and road layout, whereas NEOVERSE produces visually plausible views but alters vehicle appearance and introduces structural changes to the road scene.

combination performs best. Finally, rows (1) and (7) show the benefit of temporal fine-tuning for video. Row (7) has the best overall results and confirms the impact of our architectural designs.

Table 6 analyzes the benefits and cost of using 1, 4, 8, and 16 reference frames. Four frames has the best FID, while eight frames has the best FVD and Motion metrics and the FID is competitive. Thus, we use 8 reference frames in our final model. Table 7 compares our 3D-aware selection strategy with other alternatives. Temporal proximity alone cannot guarantee target-view relevance: nearby frames may be redundant, occluded, or observe different surfaces under substantial camera translation or rotation. Our 3D-aware strategy instead selects informative frames and delivers improved FVD, Motion, PSNR, SSIM, and LPIPS over the alternatives in both experiments.

5 CONCLUSIONS

We have proposed an efficient few-step image diffusion-based architecture to enhance artifact-prone NVS rendered frames, and autoregressively generate temporally coherent videos from novel camera trajectories. This novel architecture is used within an efficient and scalable novel-view synthesis framework well suited for simulating novel videos from monocular egocentric driving videos. In our quantitative and qualitative evaluations, we demonstrate that the proposed neural enhancer consistently outperforms closely related generative baselines in terms of perceptual quality and temporal coherence in generated videos and also runs on a single GPU in a highly memory-efficient manner.

AI USE STATEMENT

Generative AI tools were used solely to refine author-written text for grammar, clarity, concision, and organization. They were not used to generate research ideas, design the method, implement code, conduct experiments, analyze data, produce results, create figures, or formulate scientific claims. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final content of this paper.

ETHICS STATEMENT

This work uses publicly released autonomous-driving datasets in accordance with their respective licenses and does not collect new personal data or involve human-subject experiments. Because DRIVENVS generates photorealistic content, its outputs may contain hallucinated or inaccurate scene details and should not be used as factual evidence or as direct input to safety-critical driving decisions without independent verification. Performance may also reflect geographic, environmental, and demographic biases present in the training datasets. We present this method for research, simulation, and data-generation purposes and encourage responsible use with explicit disclosure of synthesized content.

REPRODUCIBILITY STATEMENT

The main paper specifies the proposed architecture, conditioning mechanisms, training objectives, datasets, evaluation protocols, and baselines. The appendix provides additional details on the motion-aware 4D reconstruction pipeline, double-reprojection data curation, model optimization, temporal training curriculum, inference procedure, evaluation metrics, hyperparameters, and ablation settings. We also report training resources, inference memory, runtime, and dataset-specific preprocessing to support faithful comparison and reproduction of our results. Our source code, evaluation scripts, and trained checkpoints will be released upon publication.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. arXiv preprint arXiv:2209.15571, 2022.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. ReCamMaster: Camera-Controlled Generative Rendering from A Single Video. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 14834–14844, October 2025.

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 425–432, 2001.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11621–11631, 2020.

Wei Cao, Hao Zhang, Fengrui Tian, Yulun Wu, Yingying Li, Shenlong Wang, Ning Yu, and Yaoyao Liu. FreeOrbit4D: Training-free arbitrary camera redirection for monocular videos via foreground-complete 4d reconstruction. In Proceedings of ACM SIGGRAPH, 2026.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment Anything with Concepts. In International Conference on Learning Representations (ICLR), 2026.

Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In Proceedings of 1st international conference on image processing, volume 2, pp. 168–172. IEEE, 1994.

Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. ACM Transactions on Graphics, 32(3): 30:1–30:12, jun 2013. doi: 10.1145/2487228.2487238.

Kaihua Chen, Tarasha Khurana, and Deva Ramanan. Reconstruct, inpaint, test-time finetune: Dynamic novel-view synthesis from monocular videos. In Advances in Neural Information Processing Systems (NeurIPS), 2025.

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 ACM SIGGRAPH, 2026.

Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96), pp. 1996, New York, NY, USA, 1996. ACM Press. doi: 10.1145/247269.247270.

Yitong Dong, Qi Zhang, Minchao Jiang, Zhiqiang Wu, Qingnan Fan, Ying Feng, Huaqi Zhang, Hujun Bao, and Guofeng Zhang. One-shot refiner: Boosting feed-forward novel view synthesis via one-step diffusion. In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 3659–3667, 2026.

Lue Fan, Hao Zhang, Qitai Wang, Hongsheng Li, and Zhaoxiang Zhang. FreeSim: Toward free-viewpoint camera simulation in driving scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12004–12014, June 2025.

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), pp. 27702–27712, October 2025.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for text-to-video generation, 2024.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics (TOG), 37(6):1–15, 2018. doi: 10.1145/3272127.3275084.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11982–11992, 2025.

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. Advances in neural information processing systems, 30, 2017.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025. doi: 10.1109/TPAMI.2025.3633890.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics, 42(4):1–15, 2023. doi: 10.1145/3588438. URL https://doi.org/10.1145/3588438.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. Advances in Neural Information Processing Systems, 38:15984–16009, 2026.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647, 2025.

Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Yuancheng Xu, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, Paul Debevec, and Ning Yu. Vista4D: Video reshooting with 4d point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 32671–32682, June 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. arXiv preprint arXiv:2210.02747, 2022.

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 (NeurIPS), 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. 2023. doi: 10.48550/arXiv.2209.03003. URL https://arxiv.org/abs/2209.03003. arXiv preprint arXiv:2209.03003.

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 Proceedings of the European Conference on Computer Vision (ECCV), 2020. doi: 10.1007/978-3-030-58452-8_24. URL https://mlanthology.org/eccv/2020/mildenhall2020eccv-nerf/.

Phuc Nguyen, Anh N Nhu, and Ming C Lin. Openvo: Open-world visual odometry with temporal dynamics awareness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14208–14218, 2026.

NVIDIA. Cosmos-Predict2: World simulation model for physical ai, 2025. URL https://github.com/nvidia-cosmos/cosmos-predict2.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp. 11400–11410. IEEE, 2022.

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

Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. ACM Transactions on Graphics (TOG), 36(6):1–11, 2017.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3d-informed world-consistent video generation with precise camera control. In Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR), pp. 6121–6132, June 2025.

Gernot Riegler and Vladlen Koltun. Free view synthesis. In European conference on computer vision, pp. 623–640. Springer, 2020.

Sudipta Sinha, Drew Steedly, and Rick Szeliski. Piecewise planar stereo for image-based rendering. In 2009 International Conference on Computer Vision, pp. 1881–1888, 2009.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2443–2451, 2020.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In European conference on computer vision, pp. 402–419. Springer, 2020.

Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. NeuRAD: Neural rendering for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14895–14904, June 2024.

Khiem Vuong, Deva Ramanan, and Srinivasa Narasimhan. FixAnything: 3d-consistent rendering refinement via video generative priors. In European Conference on Computer Vision, 2026.

Jiahao Wang, Bo Sun, Yijing Bai, Vincent Casser, Songyou Peng, Zehao Zhu, Meng-Li Shih, Xander Masotto, Shih-Yang Su, Kanaad Parvate, et al. Sensor2sensor: Cross-embodiment sensor conversion for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 32093–32102, 2026.

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

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4690–4699, 2021.

Qitai Wang, Lue Fan, Yuqi Wang, Yuntao Chen, and Zhaoxiang Zhang. FreeVS: Generative view synthesis on free driving trajectory. In International Conference on Learning Representations (ICLR), 2025b.

Jiaxin Wei, Stefan Leutenegger, and Simon Schaefer. Gsfix3d: Diffusion-guided repair of novel views in gaussian splatting. In 2026 International Conference on 3D Vision (3DV), pp. 1051–1060. IEEE, 2026.

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

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20310–20320. IEEE, 2024a.

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.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T. Barron, and Aleksander Holynski. CAT4D: Create Anything in 4D with Multi-View Video Diffusion Models. arXiv:2411.18613, 2024b.

Yan Xu, Yixing Wang, and Stella X Yu. Novel view synthesis from a few glimpses via test-time natural video completion. Advances in Neural Information Processing Systems, 38:46541–46561, 2026.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. arXiv preprint arXiv:2104.10157, 2021.

Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street Gaussians for modeling dynamic urban scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025.

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. NeoVerse: Enhancing 4d world model with in-the-wild monocular videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 40340–40351, June 2026.

Ze Yang, Jingkang Wang, Haowei Zhang, Sivabalan Manivasagam, Yun Chen, and Raquel Urtasun. GenAssets: Generating in-the-wild 3d assets in latent space. In Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 22392–22403, June 2025.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. TrajectoryCrafter: Redirecting camera trajectory for monocular videos via diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 100–111, October 2025.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. arXiv preprint arXiv:2409.02048, 2024.

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 (CVPR), pp. 43494–43504, June 2026.

Liyuan Zhu, Shengyu Huang, Amrita Mazumdar, Tianye Li, Zan Gojcic, Gordon Wetzstein, Iro Armeni, Shalini De Mello, and Alex Trevithick. World from motion: Generative dynamic gaussian reconstruction from monocular video. arXiv preprint arXiv:2607.01202, 2026.

A APPENDIX

A.1 TRAINING PROCEDURE

We initialize the enhancer from the pretrained 2B parameters Cosmos-Predict2 T2I backbone (NVIDIA, 2025), fine-tune the entire DiT together with the newly introduced conditioning modules, and keep the VAE frozen. We use AdamW, weight decay 0.01, a constant learning-rate schedule, and gradient clipping at 1.0. We train on four GPUs for approximately four days using bfloat16 precision, a per-GPU batch size of one, and no gradient accumulation, giving an effective batch size of four. We fix the random seed to 42. We follow the two-stage training schedule similar to HARMONIZER (Zhang et al., 2026).

Stage 1. We train the neural enhancer on individual coarse renderings using only $\mathcal{L}_{\mathrm{RF}}$, allowing it to learn reconstruction-guided restoration and reference conditioning before introducing temporal context. The model is trained for $200{,}000$ optimizer steps with a learning rate of 2×10^{-5}.

Stage 2. Stage 2 is initialized from the Stage 1 checkpoint and fine-tuned for $100{,}000$ steps at 5×10^{-6}. We randomly draw temporal samples with probability $\alpha_{\mathrm{temp}} = 0.75$ mixed with single-frame samples. Each temporal sample contains the a coarse rendered view and $K = 4$ preceding source video frames. The preceding frames serve as teacher-forced context. Only the current rendering is corrupted, and $\mathcal{L}_{\mathrm{RF}}$ supervises only its prediction. Using some single-frame samples during stage 2 preserves our model's ability to enhance single frames when the preceding frames are unavailable.

For temporal samples, we decode the predicted latent to obtain $\widehat{\mathbf{I}}_i$. A frozen RAFT model (Teed & Deng, 2020) estimates optical flow $\mathbf{F}_{i-k\to i}$ between each preceding clean frame and the current clean frame. We warp the predicted current frame into each preceding frame's coordinates and apply a Charbonnier consistency loss (Charbonnier et al., 1994):

$$\mathcal{L}_{\mathrm{temp}} = \frac{1}{K}\sum_{k=1}^{K} \rho\Big(\mathbf{M}_{i,k} \odot \Big[\mathcal{W}\Big(\widehat{\mathbf{I}}_i, \mathbf{F}_{i-k\to i}\Big) - \mathbf{I}^{s}_{i-k}\Big]\Big), \tag{6}$$

where $\mathcal{W}$ denotes backward warping and $\mathbf{M}_{i,k}$ excludes out-of-bounds samples. Stage 2 minimizes

$$\mathcal{L}_{\mathrm{stage2}} = \mathcal{L}_{\mathrm{RF}} + \mathbb{1}_{\mathrm{temp}}\lambda_{\mathrm{temp}}\mathcal{L}_{\mathrm{temp}}, \tag{7}$$

where $\mathbb{1}_{\mathrm{temp}}$ activates temporal supervision only for temporal samples. We increase λ_{temp} linearly to 0.5 over the first $2{,}000$ Stage-2 updates.

There is no optical flow computation during inference. The history consists of four recentmost output frames. The enhancer generates the first three frames without using any history, then generates subsequent frames autoregressively with the fixed-sized temporal context of four preceding frames.

Temporal Hyperparameters. Table 8 and Table 9 study the effect of temporal-sample probability α_{temp} and temporal-loss weight λ_{temp} on WAYMO on performance. Using temporal samples exclusively reduces FVD from 150.52 to 147.20, but increases FID from 9.94 to 10.59. Increasing λ_{temp} from 0.5 to 1.0 yields a similar trade-off: FVD improves from 150.52 to 147.12, whereas FID worsens from 9.94 to 10.95. We therefore set $\alpha_{\mathrm{temp}} = 0.75$ and $\lambda_{\mathrm{temp}} = 0.5$ to train our model.

A.2 EVALUATION METRICS

In this section, we describe the performance metrics and our evaluation protocol. We evaluate DrivENVS using two sets of metrics depending on whether real video frames are available for the target novel camera view. When ground truth real frames are unavailable, we measure the realism and temporal consistency of generated videos. However, when such frames are available, we measure frame-wise discrepancy between ground-truth and generated frames.

A.2.1 EVALUATION WITHOUT GROUND-TRUTH

For synthetically shifted camera trajectories, no image was captured at the target viewpoint. We therefore evaluate the distribution of generated frames and videos against real source-camera videos from the same test dataset. These metrics measure visual quality and temporal realism at the dataset

α_{temp}	FID↓	KID↓	FVD↓	Motion↑
25%	9.925	0.0075	160.753	98.214
50%	9.949	0.0072	155.489	98.285
75%	9.937	0.0071	150.522	98.303
100%	10.587	0.0086	147.198	98.391

Table 8: **Effect of the temporal-sample fraction** on WAYMO LATERAL SHIFT (2 M.).

λ_{temp}	FID↓	KID↓	FVD↓	Motion↑
0.05	9.005	0.0032	188.485	97.755
0.10	9.042	0.0032	179.451	98.002
0.50	9.937	0.0071	150.522	98.303
1.00	10.953	0.0091	147.123	98.411

Table 9: **Effect of the temporal-loss weight** on WAYMO LATERAL SHIFT (2 M.).

level; they do not measure pixelwise agreement with an unavailable target view. We compute distributional scores from features pooled across the evaluated sequences, rather than averaging independently computed per-scene scores. When reported, left, right, and upward trajectories are also evaluated as separate pools.

Fréchet Inception Distance (FID). We use Clean-FID Parmar et al. (2022) to extract Inception features Heusel et al. (2017) from individual generated and real frames. FID compares the means and covariance matrices of their feature distributions through the Fréchet distance. It therefore measures whether the generated frames collectively resemble real driving images in appearance and content. We pool frame features before computing the reported score; lower FID is better.

Kernel Inception Distance (KID). We compute KID from 2048-dimensional Inception features extracted with TorchMetrics. For each of 100 independently sampled subsets, we draw 20 real and 20 generated frames and compute an unbiased maximum mean discrepancy using a third-degree polynomial kernel. We report the mean subset score, with sampling fixed by a random seed of 42. KID evaluates agreement between frame-feature distributions, with lower values indicating better agreement. Its TorchMetrics feature pipeline is separate from the Clean-FID pipeline used for FID.

Fréchet Video Distance (FVD). We use the VideoGPT Yan et al. (2021) FVD implementation. Each frame is resized while preserving its aspect ratio, center-cropped to 224×224, and normalized before processing. The network produces a 400-dimensional video-level logit vector for each sequence. Unless otherwise specified, we treat each complete generated sequence and its corresponding real source sequence as one clip, then compute the Fréchet distance between the pooled generated and real video-feature distributions. Unlike the framewise FID and KID scores, FVD evaluates appearance and temporal structure jointly; lower is better.

Motion Smoothness. We evaluate generated videos using the Motion Smoothness metric of VBench Huang et al. (2025). Its pretrained interpolation model predicts intermediate frames from neighboring generated frames and compares those predictions with the frames actually generated. The resulting interpolation error is converted into a per-video smoothness score; we average these scores across videos and report them on a 0–100 scale. Higher scores indicate more consistent frame-to-frame motion. This metric does not require a ground-truth target-view video and should be interpreted as a measure of temporal smoothness, rather than motion or geometric accuracy.

A.2.2 GROUND TRUTH-BASED EVALUATION

For stereo and side-view synthesis, we compare each generated frame with the corresponding image captured by the target camera. If the two images have different resolutions, we resize the generated image to the target resolution using bilinear interpolation. We compute the following scores for each frame pair and report the mean over all evaluated pairs.

Peak Signal-to-Noise Ratio (PSNR). We compute PSNR from the mean squared RGB error between the generated and target images, with pixel intensities normalized to $[0, 1]$. The score is measured in decibels: higher PSNR indicates smaller pixelwise error.

Structural Similarity (SSIM). We compute SSIM independently on the three RGB channels and average their scores. Local image statistics are estimated with an 11×11 Gaussian window of standard deviation 1.5, using the standard SSIM stability constants $(0.01)^2$ and $(0.03)^2$ for images in $[0, 1]$. SSIM measures local agreement in luminance, contrast, and structure; higher values are better.

Learned Perceptual Image Patch Similarity (LPIPS). We compute LPIPS with the pretrained feature network, comparing the generated and target images through their learned feature responses.

TorchMetrics normalizes the $[0, 1]$ RGB inputs for this network. LPIPS captures perceptual differences that may not be reflected by pixelwise error; lower values indicate closer perceptual agreement.

A.3 Implementation Details: 4D Reconstruction pipeline

Our 4D reconstruction pipeline produces coarse views for the neural enhancer. From a calibrated monocular video, it aggregates stationary points into a fused static point cloud and retains moving object points as time-specific dynamic point clouds. For each target camera pose and timestamp, the static and time-specific dynamic geometry is rendered into a coarse novel view and coverage mask.

Input and 4D Representation. Given a calibrated monocular video $\mathcal{V} = \{(\mathbf{I}_i, \mathbf{K}_i, \mathbf{T}_i)\}_{i=1}^{N}$, where $\mathbf{I}_i$, $\mathbf{K}_i$, and $\mathbf{T}_i$ denote the RGB image, camera-intrinsic matrix, and extrinsic pose at timestep i, respectively, we reconstruct a static point cloud $\mathcal{P}^s$ and a dynamic point cloud $\mathcal{P}_i^d$ for each timestep. The resulting motion-aware 4D representation is denoted as:

$$\mathcal{P} = \big(\mathcal{P}^s, \{\mathcal{P}_i^d\}_{i=1}^{N}\big). \tag{8}$$

The static component aggregates stationary observations across the sequence, whereas the dynamic component preserves moving objects only at their corresponding timesteps, avoiding duplicated geometry along object trajectories.

Metric Multiview Reconstruction. We use DA3 (Lin et al., 2025) to estimate dense depth maps $\widehat{\mathbf{D}}_i$, confidence maps, and camera poses from the input video. After converting both the predicted and calibrated poses to camera-to-world form, we have $\widehat{\mathbf{T}}_i = [\widehat{\mathbf{R}}_i \mid \widehat{\mathbf{t}}_i], \mathbf{T}_i = [\mathbf{R}_i \mid \mathbf{t}_i]$. Because these transformations map camera coordinates into the world coordinate system, $\widehat{\mathbf{t}}_i$ and $\mathbf{t}_i$ are the predicted and calibrated camera translations, respectively. We then estimate Umeyama transformation jointly over the complete camera trajectory:

$$(s^\star, \mathbf{Q}^\star, \mathbf{b}^\star) = \underset{s>0,\, \mathbf{Q}\in \mathrm{SO}(3),\, \mathbf{b}\in\mathbb{R}^3}{\operatorname{argmin}} \sum_{i=1}^{N} \Big\| \mathbf{t}_i - \Big(s\mathbf{Q}\widehat{\mathbf{t}}_i + \mathbf{b} \Big) \Big\|_2^2. \tag{9}$$

The recovered scale converts every predicted depth map into metric units, $\mathbf{D}_i = s^\star \widehat{\mathbf{D}}_i$. We retain the calibrated poses $\mathbf{T}_i$, placing all reconstructed frames in the *same metric world coordinate* system. Let Ω_i denote 80% pixels retained after confidence filtering. For each $\mathbf{u} = (u_x, u_y) \in \Omega_i$, we define its homogeneous image coordinate, back-project it into the world coordinate system and assign each point its corresponding RGB value produces the metric colored point cloud as:

$$\begin{aligned} \bar{\mathbf{u}} &= [u_x, u_y, 1]^\top, \\ \mathbf{p}_i(\mathbf{u}) &= \mathbf{R}_i \left(\mathbf{D}_i(\mathbf{u})\, \mathbf{K}_i^{-1} \bar{\mathbf{u}} \right) + \mathbf{t}_i, \\ \mathcal{P}_i &= \left\{ \big(\mathbf{p}_i(\mathbf{u}),\, \mathbf{I}_i(\mathbf{u}) \big) \;\middle|\; \mathbf{u} \in \Omega_i \right\}. \end{aligned} \tag{10}$$

Motion Segmentation and Rendering. We use SAM 3 Carion et al. (2026) tracks to separate stationary and moving geometry. For each tracked instance, we compute its bird's-eye-view (BEV) centroid across the frames in which it is observed and measure its average displacement from the track's geometric-median center. Tracks with displacement below $\tau_{\mathrm{mot}} = 2.0\ m.$ are treated as stationary, while the remaining tracks are kept timestep-specific. We aggregate all stationary points into a global static cloud $\mathcal{P}^s$ and combine it with the dynamic cloud $\mathcal{P}_i^d$ when rendering timestep i. Source frames are ranked by the number of points that remain visible from the target camera.

Here, SplitByMotion assigns background and stationary-instance points to $\mathcal{P}_i^s$, and moving-instance points to $\mathcal{P}_i^d$. ZBufferProject projects a point cloud with z-buffer visibility $\tau_z = 0.3$ and returns its rendered image, depth map, and binary coverage mask. RelevanceScore calculates the number of source points which are co-visible with the target view. This procedure is used during testing to render novel viewpoints from a monocular video. Detailed implementation is presented in Algo. 1.

Double-Reprojection Training Pairs. To construct pixel-aligned training pairs, we perturb the source camera and project $\mathcal{P}_i$ into the resulting virtual view (Left 2m./Right 2m./Up 2m./Down

Algorithm 1 Motion-Aware 4D Reconstruction and Rendering

Input: Calibrated video $\mathcal{V} = \{(\mathbf{I}_i, \mathbf{K}_i, \mathbf{T}_i)\}_{i=1}^{N}$, per-frame point clouds $\{\mathcal{P}_i\}_{i=1}^{N}$, tracked masks $\{\mathbf{M}_{i,m}\}$, target cameras $\{(\mathbf{K}'_i, \mathbf{T}'_i)\}_{i=1}^{N}$, motion threshold τ_{mot}, and number of references K

for each tracked instance m **do**
  Compute BEV centroids $\{\mathbf{b}_{i,m}\}_{i \in \mathcal{T}_m}$
  $\bar{\mathbf{b}}_m \leftarrow \text{median}(\{\mathbf{b}_{i,m}\}_{i \in \mathcal{T}_m})$
  $\delta_m \leftarrow \frac{1}{|\mathcal{T}_m|} \sum_{i \in \mathcal{T}_m} \|\mathbf{b}_{i,m} - \bar{\mathbf{b}}_m\|_2$
  Label m as stationary if $\delta_m < \tau_{\text{mot}}$, otherwise dynamic
end for
for $i = 1, \ldots, N$ **do**
  $(\mathcal{P}_i^s, \mathcal{P}_i^d) \leftarrow \textsc{SplitByMotion}(\mathcal{P}_i, \{\mathbf{M}_{i,m}\}_m)$
end for
$\mathcal{P}^s \leftarrow \bigcup_{i=1}^{N} \mathcal{P}_i^s$
for $i = 1, \ldots, N$ **do**
  $(\widetilde{\mathbf{I}}_i, _, \mathbf{C}_i) \leftarrow \textsc{ZBufferProject}(\mathcal{P}^s \cup \mathcal{P}_i^d, \mathbf{K}'_i, \mathbf{T}'_i)$
  for $j = 1, \ldots, N$ **do**
    $s_{ij} \leftarrow \textsc{RelevanceScore}(\mathcal{P}_j, \mathbf{K}'_i, \mathbf{T}'_i)$
  end for
  $\mathcal{V}_i^r \leftarrow \textsc{TopKReferences}(\mathcal{V}, \{s_{ij}\}_{j=1}^{N}, K)$
end for
return $\mathcal{P}^s$, $\{\mathcal{P}_i^d\}_{i=1}^{N}$, and $\{\widetilde{\mathbf{I}}_i, \mathbf{C}_i, \mathcal{V}_i^r\}_{i=1}^{N}$

Algorithm 2 Double-Reprojection Training-Pair Construction

Input: Calibrated video $\mathcal{V} = \{(\mathbf{I}_j, \mathbf{K}_j, \mathbf{T}_j)\}_{j=1}^{N}$, per-frame point clouds $\{\mathcal{P}_j\}_{j=1}^{N}$, static cloud $\mathcal{P}^s$, target index i, dynamic mask $\mathbf{M}_i^d$, depth tolerance τ_z, and number of references K

$\mathbf{T}_i^v \leftarrow \textsc{Perturb}(\mathbf{T}_i)$
$(_, \mathbf{D}_i^v, _) \leftarrow \textsc{ZBufferProject}(\mathcal{P}_i, \mathbf{K}_i, \mathbf{T}_i^v)$
$\mathcal{P}_i^{\text{vis}} \leftarrow \textsc{VisiblePoints}(\mathcal{P}_i, \mathbf{D}_i^v, \mathbf{K}_i, \mathbf{T}_i^v, \tau_z)$
$(\widetilde{\mathbf{I}}_i, _, \mathbf{C}_i) \leftarrow \textsc{ZBufferProject}(\mathcal{P}_i^{\text{vis}}, \mathbf{K}_i, \mathbf{T}_i)$
$\Omega_i^{\text{fill}} \leftarrow \left\{ \mathbf{u} \;\middle|\; \mathbf{C}_i(\mathbf{u}) = 0,\ \mathbf{M}_i^d(\mathbf{u}) = 0 \right\}$
Project $\mathcal{P}^s$ into $(\mathbf{K}_i, \mathbf{T}_i)$ and fill valid pixels in Ω_i^{fill}
Update $\widetilde{\mathbf{I}}_i$ and $\mathbf{C}_i$
for $j = 1, \ldots, N$ **do**
  $s_{ij} \leftarrow \textsc{VisibleCount}(\mathcal{P}_j, \mathbf{K}_i, \mathbf{T}_i)$
end for
$\mathcal{V}_i^r \leftarrow \textsc{TopKReferences}(\mathcal{V}, \{s_{ij}\}_{j=1}^{N}, K)$ (exclude target i)
return $(\mathbf{C}_i, \widetilde{\mathbf{I}}_i, \mathbf{I}_i, \mathcal{V}_i^r)$

1M.). Only points visible after virtual-view z-buffering are retained and reprojected back to the original camera. This produces a degraded rendering that remains pixel-aligned with the clean source image $\mathbf{I}_i$. We then fill uncovered non-dynamic regions using the aggregated static cloud $\mathcal{P}^s$.

VISIBLEPOINTS retains points whose projected depth agrees with the virtual-view z-buffer within a local neighborhood and tolerance τ_z. The resulting coarse image therefore preserves visibility, disocclusion, and point-rendering artifacts while remaining exactly aligned with the clean target $\mathbf{I}_i$. Therefore, this procedure produces four training samples for each image in the dataset. See Algo. 2 for more details.

A.4 Detailed Quantitative Results

Comparison with VDM-Based NeoVerse. Tables 10 and 11 compare both methods using the same 81-frame evaluation. Across all camera shifts, DrivENVS substantially improves FID and FVD while maintaining comparable motion scores and generating 3.1× as many pixels as NeoVerse. As a VDM, NeoVerse generates the entire 81-frame video jointly in a single forward

Shift	Method	Backbone	Generation Quality		
			FID ↓	FVD ↓	Motion ↑
0 m.	NeoVerse	Wan2.1-T2V-14B	14.884	206.130	**98.481**
	Ours	Cosmos-Predict2 T2I-2B	**1.656**	**20.321**	98.450
1 m.	NeoVerse	Wan2.1-T2V-14B	27.634	255.757	**98.396**
	Ours	Cosmos-Predict2 T2I-2B	**7.095**	**112.837**	98.201
2 m.	NeoVerse	Wan2.1-T2V-14B	40.409	312.384	**98.315**
	Ours	Cosmos-Predict2 T2I-2B	**13.851**	**206.879**	98.119

Table 10: **Comparison with NEOVERSE on NUSCENES.** We evaluate 81-frame rollouts under target-camera translations of 0, 1, and 2 m. NeoVerse generates at 560×336 resolution, our method generates at 1024×576.

Method	Training Resources	Inference Cost	
		FPS ↑	Peak Mem. (GB) ↓
NeoVerse	32 GPUs / 5 days	**6.750**	88.463
Ours	**4 GPUs / 4 days**	0.717	**19.392**

Table 11: **Training resources and inference cost.** Inference throughput (FPS) is normalized over a complete 81-frame video by dividing the number of generated frames by the total rollout time. Peak GPU memory is measured during the same rollout. NeoVerse generates at 560×336 resolution, ours generates at 1024×576.

pass. Although this provides higher throughput (6.750 FPS), its output length is fixed: all frames must reside in one spatiotemporal latent, and generating beyond 81 frames is not supported. Extending this horizon would further increase the temporal-token and attention footprint beyond its already substantial 88.5-GB peak memory. In contrast, DRIVENVS autoregressively refines one target frame at a time using fixed-size reference and history windows. It therefore requires only 19.4 GB of peak memory, $4.6\times$ less than NEOVERSE, while supporting rollouts beyond 81 frames without increasing peak memory. This flexibility comes at lower throughput (0.717 FPS), as frames are generated sequentially. DRIVENVS also uses substantially fewer training resources, reducing the training budget from 32 GPUs for 5 days to 4 GPUs for 4 days. These results demonstrate that our autoregressive design enables long and high-resolution video generation even on a single 20-GB GPU.

Detailed Lateral-Shift Results. Table 12, Table 13, Table 14, and Table 15 provide a fine-grained analysis of each direction in the LATERAL SHIFT benchmark across WAYMO and NUSCENES and displacement magnitudes. Furthermore, additional video metrics from VBench Huang et al. (2025) are also reported. DRIVENVS consistently improves video-level temporal quality across left, right, and upward shifts, achieving the best FVD and Motion Smoothness in all four settings. These gains persist at (2 m.), demonstrating that the improvement is consistent across shift directions and remains robust under larger viewpoint changes. These tables are supplementary to Table 1.

A.5 QUALITATIVE RESULTS

We provide **additional qualitative demo videos** in our supplementary materials

Fig. 5 presents side-view synthesis results for NUSCENES. For each target frame, DRIVENVS retrieves geometrically overlapping references from the source front-view video to recover appearance details and content revealed by the viewpoint change. The synthesized views closely preserve the target-side scene layout, vehicles, pedestrians, and background structure despite the large camera displacement.

The figures present qualitative results for LEFT, RIGHT, and UP camera shifts of (1 m.) and (2 m.) on NUSCENES (Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, Fig. 11) and WAYMO (Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 16, Fig. 17). Each comparison shows the original frame,

Lateral Shift (1 m) – nuScenes										
Method	**Split**	**FID**	**KID**	**FVD**	**Motion**	**Subject**	**Background**	**Overall**	**Aesthetic**	**Imaging**
DIFIX3D	Left	5.240	0.0076	149.662	95.964	85.265	91.375	8.666	50.371	52.664
	Right	5.811	0.0081	165.616	95.955	85.445	91.318	9.484	50.213	52.818
	Up	12.396	0.0065	198.196	95.285	85.638	91.847	7.338	53.032	56.373
	All	4.934	0.0058	132.731	95.043	85.427	91.474	8.631	50.993	53.670
HARMONIZER	Left	5.761	0.0077	169.483	96.900	86.117	92.143	8.811	50.417	52.836
	Right	5.880	0.0081	150.540	96.892	86.327	92.166	9.478	50.277	53.375
	Up	15.746	0.0064	263.839	96.266	86.774	92.827	8.087	53.286	56.975
	All	5.321	0.0054	139.622	96.691	86.363	92.327	9.011	51.099	54.095
DRIVENVS	Left	6.349	0.0078	131.522	98.055	87.069	92.263	9.675	50.352	53.395
	Right	6.382	0.0079	114.151	98.047	87.247	92.223	10.510	50.341	53.961
	Up	16.728	0.0063	218.978	98.393	87.359	92.786	8.293	52.643	57.527
	All	6.020	0.0058	106.489	98.216	87.209	92.382	9.632	50.934	54.662

Table 12: **Detailed results for LATERAL SHIFT (1 m) on nuSCENES.**

Lateral Shift (2 m) – nuScenes										
Method	**Split**	**FID**	**KID**	**FVD**	**Motion**	**Subject**	**Background**	**Overall**	**Aesthetic**	**Imaging**
DIFIX3D	Left	9.132	0.0045	277.663	95.965	87.004	92.299	9.749	49.392	51.021
	Right	9.380	0.0052	291.752	95.961	86.908	92.080	10.693	49.298	51.398
	Up	20.534	0.0145	398.547	95.245	87.228	92.404	8.753	51.930	54.860
	All	8.642	0.0050	255.994	95.034	87.025	92.244	9.849	50.001	52.136
HARMONIZER	Left	10.366	0.0044	304.446	97.883	86.561	92.039	9.959	49.096	50.965
	Right	9.794	0.0039	269.942	97.874	86.809	92.100	10.818	49.314	52.052
	Up	28.173	0.0171	446.559	98.139	87.681	92.757	8.501	53.415	56.644
	All	9.784	0.0039	256.448	97.944	86.938	92.244	9.909	50.274	52.812
DRIVENVS	Left	12.448	0.0081	224.474	98.053	86.150	92.123	9.838	49.008	51.353
	Right	11.106	0.0086	193.400	98.040	86.606	92.256	10.825	49.372	52.501
	Up	23.133	0.0094	381.384	98.340	86.993	92.410	8.667	52.084	56.965
	All	11.898	0.0078	186.106	98.121	86.534	92.245	9.909	49.925	53.206

Table 13: **Detailed results for LATERAL SHIFT (2 m) on nuSCENES**.

the coarse 4D-reconstruction rendering, and the corresponding DRIVENVS output. Although the reconstruction establishes the target-view geometry, it contains holes, splatting artifacts, and incomplete disoccluded regions. DRIVENVS consistently removes these artifacts and completes missing content while preserving scene layout, appearance, and viewpoint-dependent structure across both datasets, all shift directions, and the larger (2 m.) displacement.

Lateral Shift (1 m) – WAYMO										
Method	**Split**	**FID**	**KID**	**FVD**	**Motion**	**Subject**	**Background**	**Overall**	**Aesthetic**	**Imaging**
DIFIX3D	Left	6.012	0.0065	110.413	95.632	90.960	93.608	8.547	53.402	61.960
	Right	6.047	0.0064	109.328	95.615	90.801	93.547	6.839	53.204	61.783
	Up	11.636	0.0065	190.066	95.631	90.267	93.502	7.736	54.667	62.868
	All	4.850	0.0058	96.617	95.626	90.009	93.552	7.707	53.757	62.207
HARMONIZER	Left	6.798	0.0068	158.381	97.541	91.922	93.625	8.544	53.089	62.050
	Right	6.790	0.0063	138.011	97.531	91.753	93.273	6.857	53.129	61.899
	Up	13.768	0.0062	212.122	97.550	92.359	93.220	7.745	54.275	63.205
	All	5.876	0.0061	127.068	97.441	92.011	93.373	7.715	53.498	62.384
DRIVENVS	Left	6.324	0.0067	99.528	98.440	92.597	94.926	9.207	52.123	66.493
	Right	6.412	0.0063	88.982	98.403	92.423	94.700	7.579	51.958	66.287
	Up	12.865	0.0064	161.939	98.440	92.026	94.787	8.508	53.251	67.208
	All	5.401	0.0063	77.961	98.428	92.582	94.804	8.432	52.444	66.663

Table 14: **Detailed results for LATERAL SHIFT (1 m) on WAYMO.**

Lateral Shift (2 m) – WAYMO										
Method	**Split**	**FID**	**KID**	**FVD**	**Motion**	**Subject**	**Background**	**Overall**	**Aesthetic**	**Imaging**
DIFIX3D	Left	9.181	0.0020	188.888	95.565	91.641	91.355	8.873	52.504	60.753
	Right	9.655	0.0023	209.499	95.533	91.269	91.226	7.061	51.990	60.118
	Up	21.423	0.0111	403.154	95.496	92.265	91.172	8.190	53.872	61.930
	All	8.659	0.0025	196.433	95.531	91.725	91.251	8.041	52.789	60.934
HARMONIZER	Left	10.116	0.0020	247.638	97.425	91.526	91.050	8.743	52.339	60.645
	Right	10.330	0.0021	258.171	97.381	91.135	91.007	7.008	52.062	60.108
	Up	25.560	0.0139	443.109	97.399	92.399	91.148	8.114	53.769	62.071
	All	10.037	0.0030	239.721	97.402	91.686	91.068	7.955	52.723	60.941
DRIVENVS	Left	9.946	0.0055	164.965	98.338	91.080	92.541	9.443	51.309	65.304
	Right	10.856	0.0057	168.237	98.267	90.530	92.156	7.895	50.889	64.443
	Up	24.201	0.0064	320.914	98.304	91.916	92.460	8.984	52.625	66.349
	All	9.937	0.0071	150.522	98.303	91.175	92.385	8.774	51.607	65.365

Table 15: **Detailed results for LATERAL SHIFT (2 m) on WAYMO.**

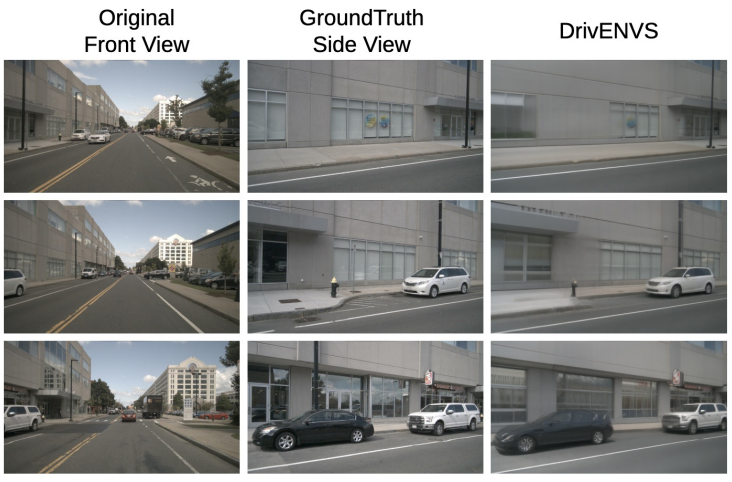

Figure 5: **Side-View Synthesis:** Results for Front-to-Left camera synthesis on NUSCENES.



Figure 6: **Lateral Shift Qualitative Results (nuScenes Left 1 m.)**

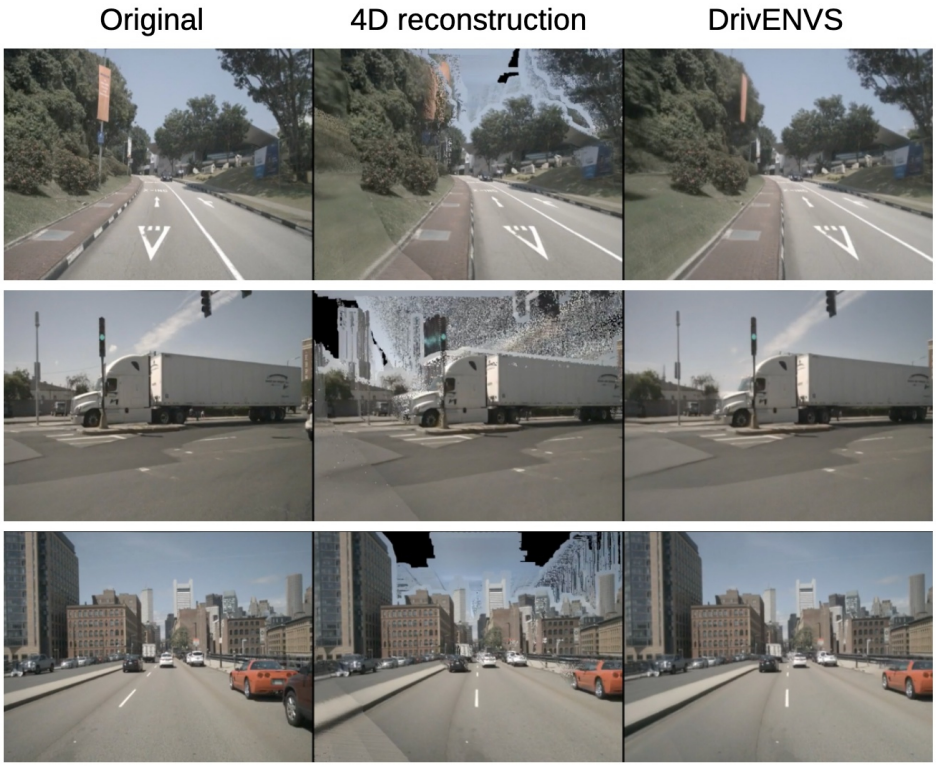

Figure 7: **Lateral Shift Qualitative Results (nuScenes Left 2 m.)**

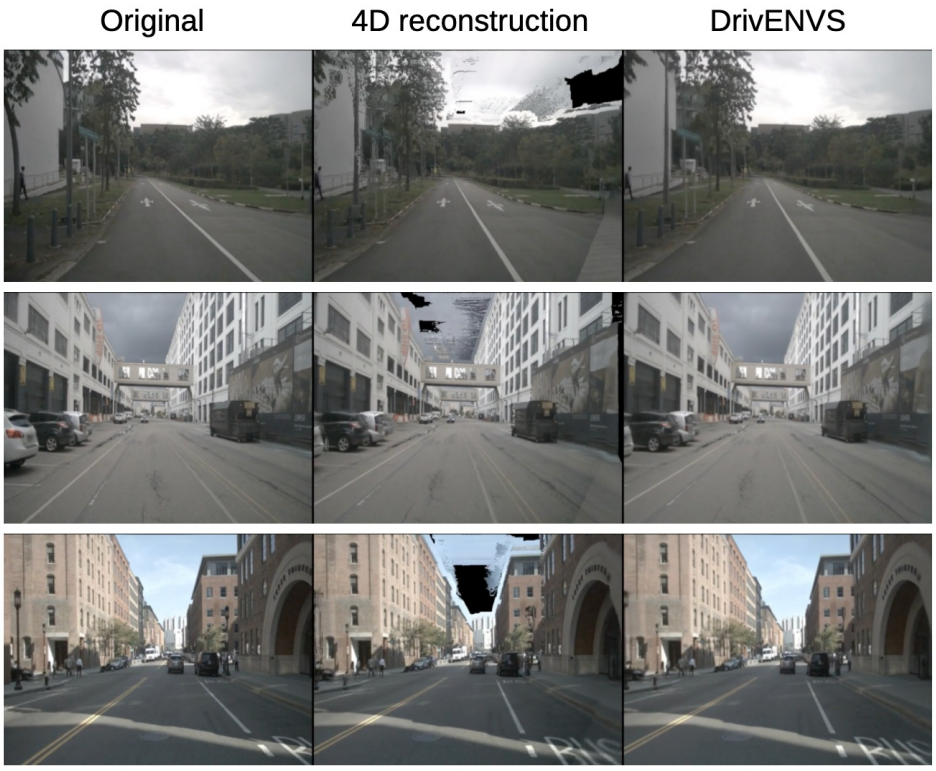

Figure 8: **Lateral Shift Qualitative Results (nuScenes Right 1 m.)**



Figure 9: **Lateral Shift Qualitative Results (nuScenes Right 2 m.)**



Figure 10: **Lateral Shift Qualitative Results (nuScenes Up 1 m.)**

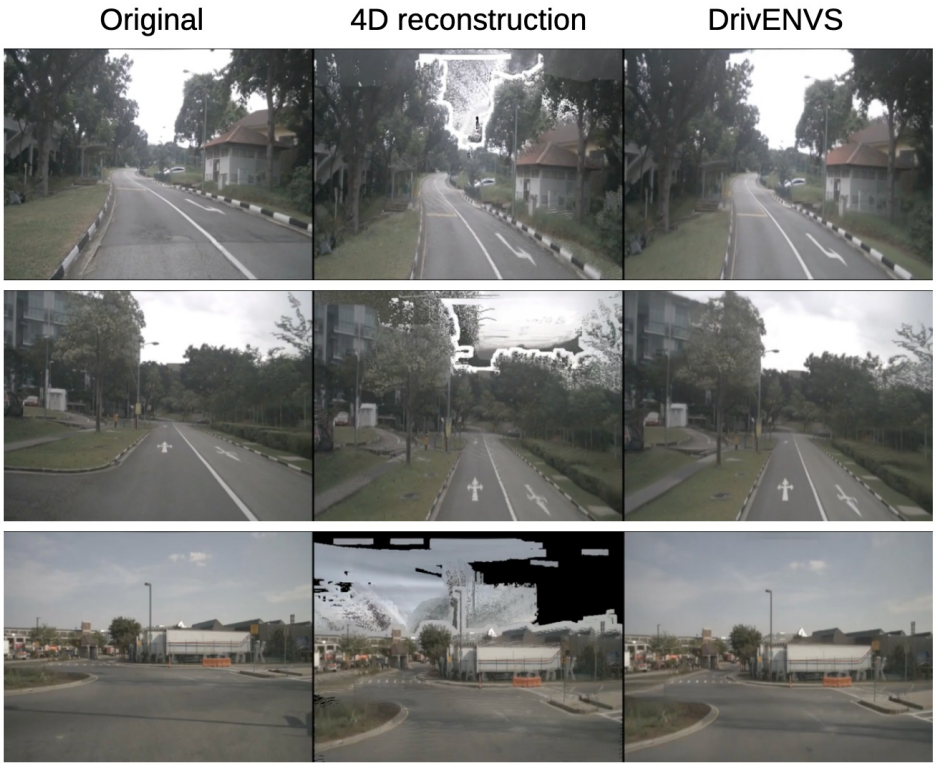

Figure 11: **Lateral Shift Qualitative Results (nuScenes Up 2 m.)**

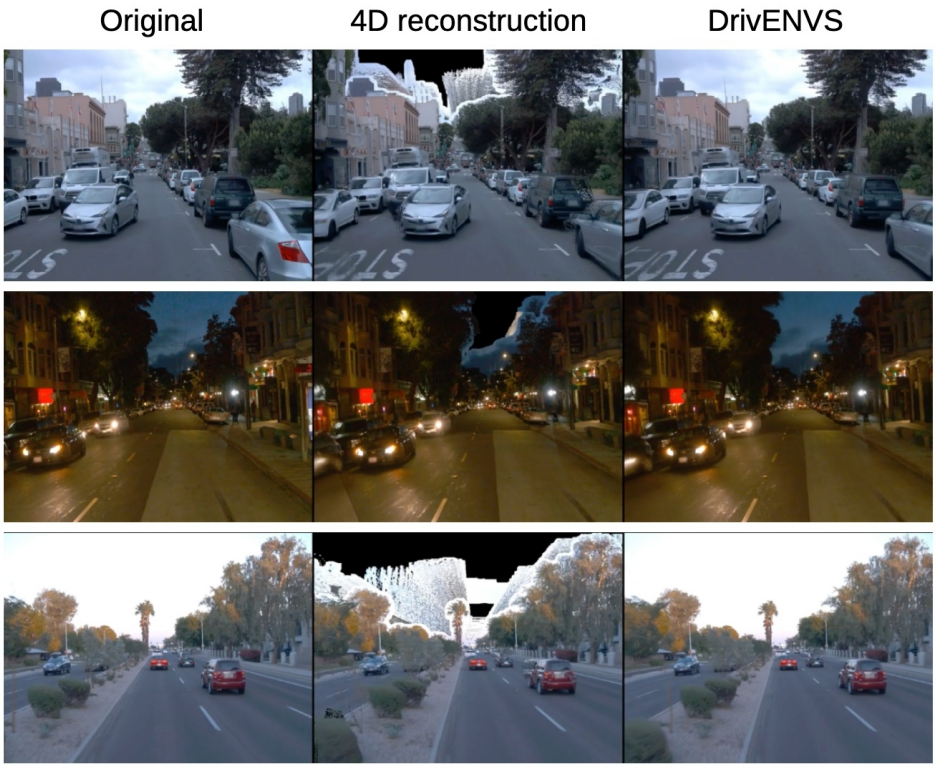

Figure 12: **Lateral Shift Qualitative Results (Waymo Left 1 m.)**

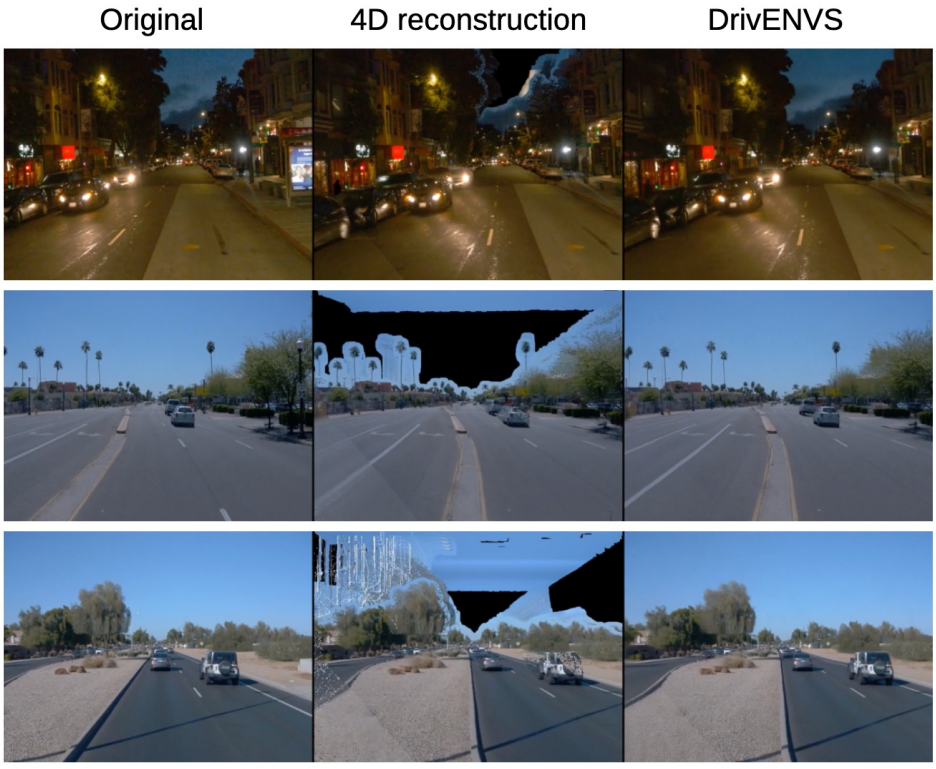

Figure 13: **Lateral Shift Qualitative Results (Waymo Left 2 m.)**



Figure 14: **Lateral Shift Qualitative Results (Waymo Right 1 m.)**

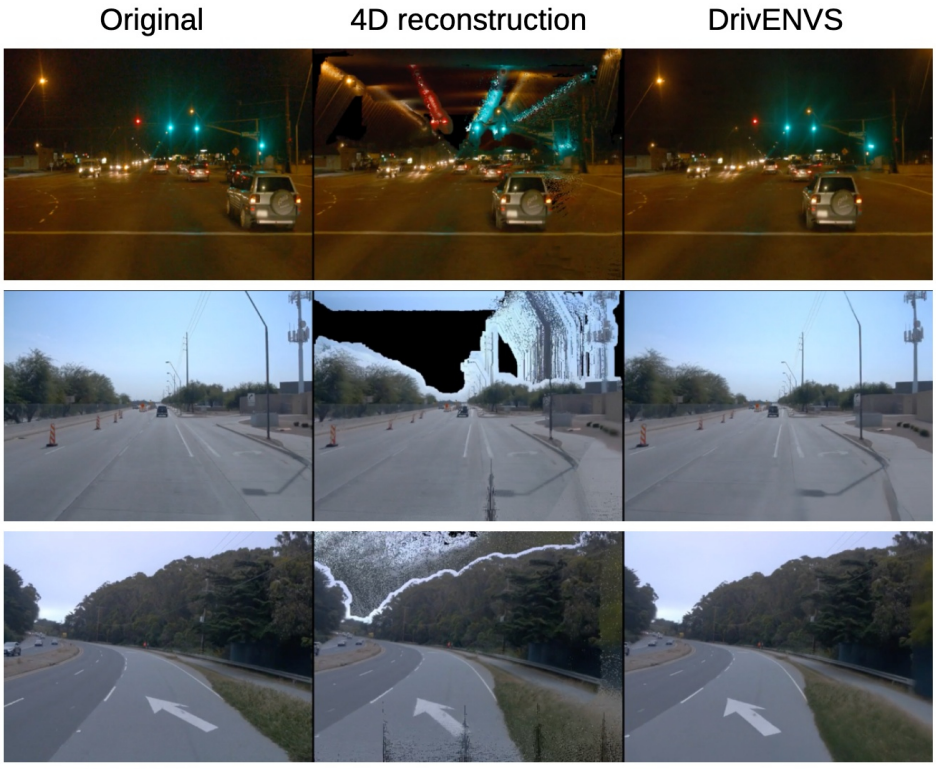

Figure 15: **Lateral Shift Qualitative Results (Waymo Right 2 m.)**

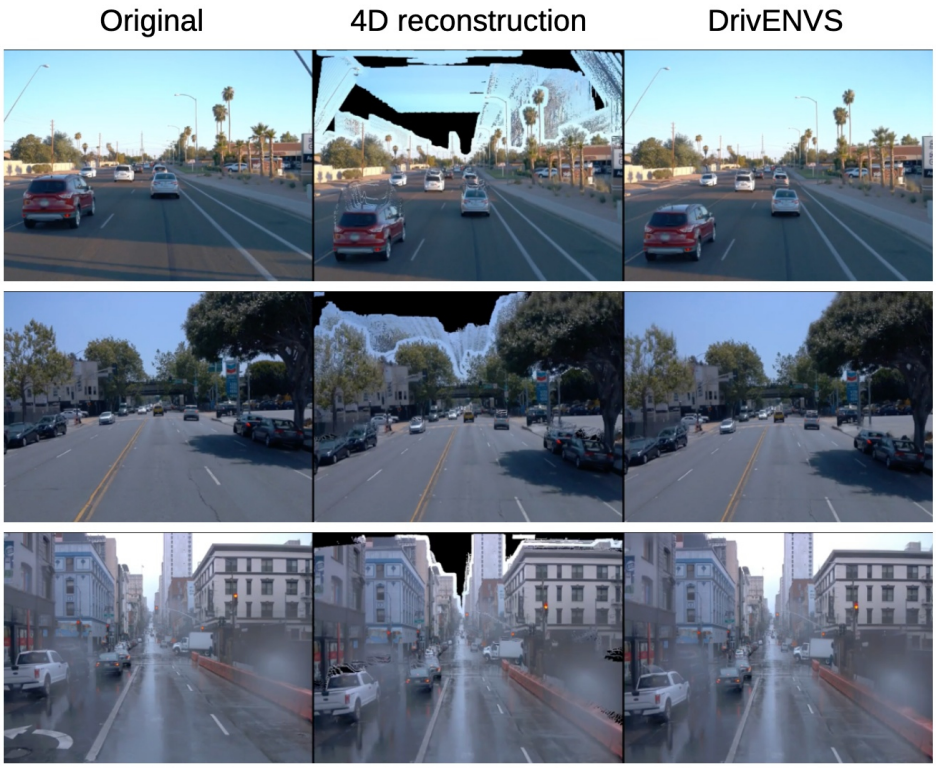

Figure 16: **Lateral Shift Qualitative Results (Waymo Up 1 m.)**

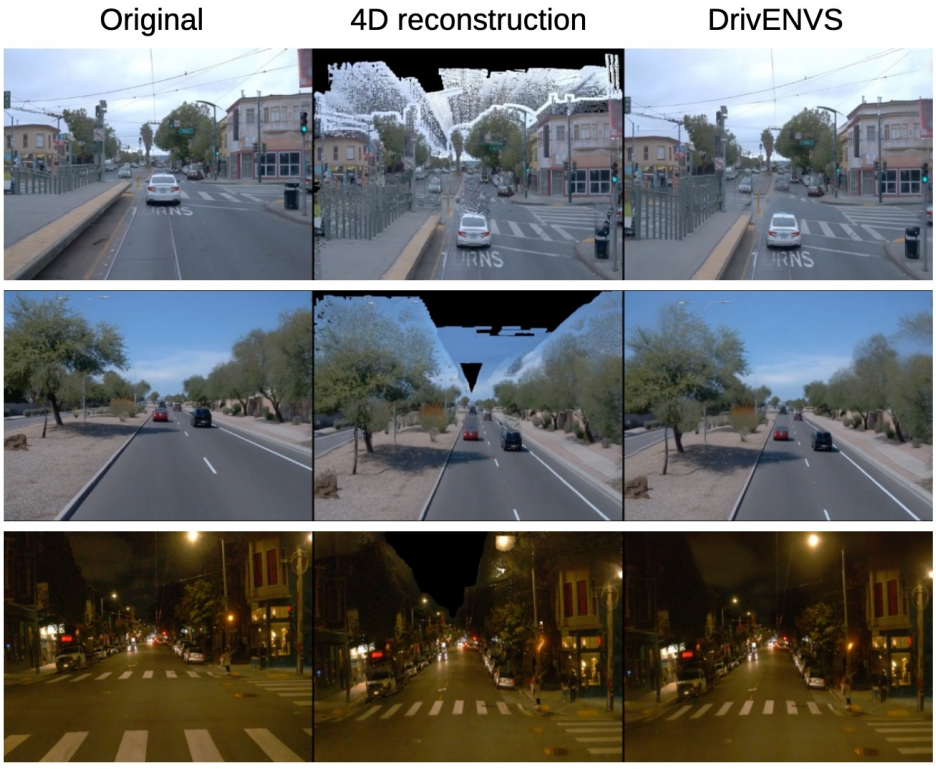

Figure 17: **Lateral Shift Qualitative Results (Waymo Up 2 m.)**