HorizonComposer: Spatiotemporally Consistent Driving Video Editing with Enriched Traffic Semantics

Mauricio Soroco*
Simon Fraser University

Yuqiu Liu*
Simon Fraser University

Zaid Tasneem
NEC Labs America

Francesco Pittaluga
NEC Labs America

Abhishek Aich
NEC Labs America

Wuyang Chen
Simon Fraser University
wuyang@sfu.ca

Manmohan Chandraker
NEC Labs America

Ziyu Jiang
NEC Labs America
zjiang@nec-labs.com

Abstract

Instruction-guided video generative models offer a scalable solution for stress-testing autonomous driving systems by simulating diverse environmental conditions. However, driving scenes operate within a dense Operational Design Domain (ODD) governed by strict traffic semantics. When applied to global driving-condition edits, general-purpose instruction-guided video editors frequently corrupt these semantics, leading to structural hallucinations and "temporal shock." This work distills and protects traffic semantics during video editing at three levels. First, we construct a hybrid dataset by combining temporally aligned real-world videos with synthetic edits, balancing foundational scene structure with unambiguous appearance signals. Second, during training, we adapt advantage conditioning from offline reinforcement learning to align the model using dense rewards. We extract dense, pixel-level rewards from the hybrid data to serve as a localized signal and actively steer generation toward structurally reliable pixels. Finally, our model mitigates temporal shock by introducing thinking frames, a transition mechanism that seamlessly reconciles static image conditioning with dynamic video context. Extensive experiments demonstrate that our method prevents the corruption of multi-actor dynamics, establishing state-of-the-art spatiotemporal consistency, yielding consistent user preference (**39%** relative improvement in content preservation and **12%** in temporal consistency), and achieving a significant **5%** gain in downstream road line segmentation and **7%** gain in downstream multi-object detection.

1 Introduction

Autonomous driving systems must operate safely and reliably across a vast Operational Design Domain (ODD), encompassing diverse weather conditions, seasonal variations, and day-night cycles. To robustly evaluate perception and planning algorithms, we require scalable simulation environments capable of simulating diverse conditions for the exact same driving scene. However, collecting perfectly paired, real-world data across these variations is practically impossible due to the natural evolution of traffic patterns. Consequently, instruction-guided video generative models (VGMs)

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



Figure 1: HorizonComposer maintains complex traffic semantics during global appearance editing (rows 1-3) using a careful design across data (Sec 3.2), model (Sec. 3.4), and training (Sec. 3.3). Beyond global transformations, our fidelity in traffic semantics allows HorizonComposer to seamlessly compound global and local edits for comprehensive scene simulation (row 4).

have emerged as a promising proxy [41], offering the ability to controllably transform the global appearance of existing driving logs (e.g., weather or time) to stress-test autonomous algorithms.

Unlike object-centric videos, driving operation design domains are defined by the dense co-existence of numerous critical factors in a single frame: multiple independent actors, strict lane geometries, and complex infrastructure. This spatiotemporal density forms a coupled and constrained set of *traffic semantics*. Because of this inherent complexity, editing driving scenes requires a delicate balance between *Instruction Alignment* (altering global appearance) and strictly enforcing *ODD Preservation* (maintaining local scene structure). In practice, global appearance editing almost always corrupts local traffic semantics, leading to severe spatiotemporal inconsistencies. Obtaining precise video-editing pairs to supervise driving scene editing is challenging, leaving the training of instruction-guided video editors hampered by three critical bottlenecks:

1. Unpaired Supervision Leads to Semantic Corruption. Controlled real-world driving video (e.g., the same route captured in different weather) is rare and suffers from temporal unpairing due to dynamic traffic flow. Training naively on these unpaired videos misleads the model into believing that vehicle identities and locations can arbitrarily change during an appearance edit. We require a mechanism to definitively separate the desired global appearance shifts from the strict local structure that must be preserved.

2. Misaligned Synthetic Priors. Conversely, using baseline models to generate synthetic real-world editing pairs provides strong appearance signals but may introduce structural hallucinations or "creative additions" that violate the ODD. Because the supervision quality varies drastically across pixels and frames, a robust training paradigm cannot treat all regions equally.

3. Conditioning-induced Temporal Shock. Video generators benefit from image-conditioning signals that guide global appearance. However, in dense traffic scenes, these static reference images frequently suffer from actor misalignment and are not guaranteed to be consistent with the underlying video dynamics. Injecting these conditions directly into the generation pipeline leads to "temporal shock": unnatural transitions, structural warping, and flickering artifacts, which are much more serious than temporal inconsistency.

These bottlenecks motivate our core question:

> *How to distill rich and faithful traffic semantics into driving scene video editing and thus achieve spatiotemporal consistency?*

To address this question, we propose HorizonComposer, a three-pronged approach to inject and protect traffic semantics throughout driving video editing:

Data - Pairing Hybrid Videos. We construct a hybrid dataset by pairing temporally aligned real-world driving videos with synthetic edits to introduce controlled appearance variations. While real-world pairs preserve the foundational scene structure despite traffic inconsistencies, synthetic pairs provide unambiguous appearance signals.

Training - Aligned Semantics via Dense Rewards. We align our editor with the spatiotemporal density of driving scenes through offline reinforcement learning, extracting *dense, pixel-level rewards* from the hybrid data as a localized supervision signal that steers generation toward structurally reliable pixel states.

Model - Temporally Robust Transition. To mitigate temporal shock, we introduce *thinking frames*, a mechanism that provides a temporal transition between the static conditioning signal and the dynamic video content. By prepending thinking frames during inference, the model can reconcile undesired artifacts to smoothly incorporate the intended edit without corrupting multi-agent dynamics.

We summarize our concrete contributions below:

1. **Comprehensive Datasets for Driving Video Editing.** We construct a comprehensive editing dataset for instruction-guided driving video edits, featuring paired real and synthetic videos with corresponding editing masks and prompts. In total, we collect 86614 videos covering 172 categories of driving-scene semantics.
2. **Superior Driving Video Editing Performance.** HorizonComposer outperforms recent video editing methods for safety-critical structural fidelity on unseen driving scenarios, achieving up to 20% improvements in content preservation while maintaining competitive instruction alignment and photorealism.
3. **Consistent User Preference.** HorizonComposer consistently generated driving video edits preferred by real users (via voting), with a relative out-performance over the second-best method by 21 % overall.
4. **Benefits to Downstream Driving Tasks.** Including driving scene videos edited by HorizonComposer further boosts downstream road line segmentation performance by 5% and boosts downstream multi-object detection by up to 7.14%.

2 Related Work

Instruction-guided video editing. General-purpose video editors built on large generative backbones [51, 20, 31] and unified MLLM-diffusion frameworks [54, 40, 25, 27, 10] have made rapid progress, but they offer no native guarantee on the strict spatiotemporal traffic semantics that driving requires. The work most relevant to us is ChronoEdit [58], whose intermediate "reasoning" frames inspire our *thinking frames*; however, we adapt the idea to anchor multi-agent dynamics rather than single-image physics.

Driving scene synthesis and editing. A growing body of work generates driving footage from layouts or text [50, 60, 52, 17, 55, 18, 38, 9, 1, 41], while a parallel line edits real logs through 3D/4D priors, particle fields, or occupancy [53, 48, 37, 44, 39, 34, 46]. These approaches typically rely on explicit reconstruction or external simulators and still struggle to preserve original vehicle identities under appearance edits. We instead operate directly in the video domain and enforce ODD preservation through dense pixel-level rewards, avoiding any 3D/4D reconstruction overhead.

Reinforcement learning for diffusion. On-policy methods [36, 62, 2, 35] are prohibitive at video scale in terms of computing cost, while offline weighted regression [42] collapses to filtered imitation [22]; recent inference-time guidance methods [26, 16, 13] sidestep this but rely on scalar, sample-level signals that miss intra-frame structural variation. We extend this line with dense, pixel-aligned reward indicators that extract localized policy improvements from partially reliable hybrid data.

For detailed survey, see Appendix E for extended discussion.

3 HorizonComposer

3.1 Overview

HorizonComposer strictly distills and protects traffic semantics during video editing through interventions at the data, training, and model levels. Sec. 3.2 outlines our data pipeline, constructing a

hybrid dataset of real-world and synthetic video pairs to balance structure and appearance. Sec. 3.3 details our training paradigm, where we extract spatiotemporal rewards from our hybrid data for reinforcement learning. Sec. 3.4 introduces temporal conditioning mechanisms that buffer against misaligned reference frames, ensuring seamless appearance transitions and strict actor fidelity.

Preliminary. We instantiate our framework on top of Cosmos-Transfer2.5 [41], a diffusion transformer (DiT) that performs latent-space video denoising under multiple structured control branches. Given a noisy latent $\mathbf{z}_\tau$ at diffusion timestep τ, the base model ϵ_θ predicts a denoising velocity conditioned on a text prompt ℓ, a reference image $\mathbf{c}_{\text{img}}$, and a stack of spatial control signals $\mathbf{c}_{\text{ctrl}} = \{\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(K)}\}$ (e.g., depth, visual) injected through dedicated ControlNet-style branches. Of these, we retain two: a depth branch that anchors the geometric layout of the source clip, and a visual branch that takes the source clip itself as input to carry its appearance prior. We freeze the depth branch to strictly enforce structural fidelity, and finetune the visual branch to specialize it for driving-scene editing.

For a source driving clip $\mathbf{x}^s = \{\mathbf{x}^s_0, \ldots, \mathbf{x}^s_{T-1}\}$ and a target editing instruction ℓ^t, our goal is to produce an edited clip $\mathbf{x}^t$ that (i) aligns globally with ℓ^t and (ii) preserves the local traffic semantics of $\mathbf{x}^s$. We denote our editing model as $\mu_\theta(\mathbf{x}^t \mid \mathbf{o})$, where the observation $\mathbf{o} = (\mathbf{x}^s, \mathbf{c}^{(0)}_{\text{img}}, \mathbf{c}_{\text{depth}}, \ell^t)$ bundles the source clip, a reference image generated by an image editor [19], source depth sequence, and the target instruction.

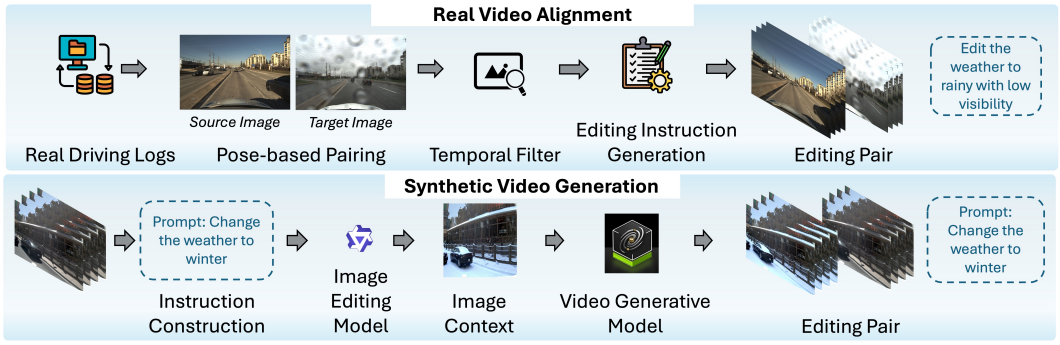


Figure 2: **Dataset Construction and Pseudo-Labeling Pipeline. (Top)** *Realistic Dataset Development* : Real-world data (Sec. 3.2.1) are matched via pose alignment, temporally filtered, and processed to generate editing instructions, forming real-world video pairs. **(Bottom)** *Pseudo Dataset Development* (Sec. 3.2.2): We synthesize driving scenes by editing the initial frame with an image prior, followed by a video generative model to propagate the edits and produce synthetic video pairs.

3.2 Distilling Semantics via Hybrid Video Pairing

3.2.1 Real-World Data Harvesting through Structural Alignment

We begin by introducing our real-world data pipeline, which produces high-quality, temporally consistent, and well-aligned video pairs from large-scale driving datasets. We build on Boreas [4], a dataset of multi-season driving logs collected over repeated routes. As illustrated in Figure 2, our pipeline transforms raw video logs into temporally aligned pairs spanning diverse environmental conditions through three stages:

1. *Pose-based Pairing.* We first pair frames from different traversals of the same route using camera pose. Specifically, we designate one log as the source video and, for each of its frames, retrieve the nearest frame from another traversal of the same route captured on a different day. This yields two viewpoint-matched sequences recorded under different weather, seasons, or times of day.
2. *Temporal Filtering.* However, viewpoint matching alone does not guarantee temporal consistency, since the two traversals exhibit independent traffic dynamics. For example, when approaching an intersection, the ego-vehicle in one traversal may stop at a red light while the other proceeds through. Naively pairing these sequences forces one clip to dwell on a near-duplicate frame until the stationary ego-vehicle resumes motion, producing pairs in which one side is effectively static. To address this, we apply a temporal filter that detects and removes both freeze-frame segments and segments with large timestamp gaps, then splits the remaining footage into fixed-length clips with consistent ego-motion and uniform duration.

3. *Instruction Generation.* Finally, we use a vision-language model to extract scene-level attributes from each clip (e.g., weather, lighting, road layout), and convert the difference between source and target attributes into a natural-language editing instruction.

The resulting corpus consists of real-world video pairs in which the underlying scene structure is preserved while global appearance varies, providing strong supervision for structure-faithful driving-scene editing. More details can be found in App. F

3.2.2 Synthetic Appearance Generation

Collecting real-world videos along repeated routes is expensive, and even aligned traversals cannot preserve identical traffic dynamics. To complete this missing supervision, we turn to synthetic appearance-editing pairs. However, we find that available state-of-the-art video editing models often yield low-quality or unstable results on driving scenes. We therefore construct synthetic pairs using a two-stage pipeline inspired by CCEdit [15] (first edit a reference frame, then propagate the edit to the full video) but powered by stronger components: Qwen Image Editing [56] for first-frame editing, and Cosmos-Transfer2.5 [41] for video propagation.

For propagation, we feed both depth and visual signals into Cosmos-Transfer2.5 with branch weights of 0.9 and 0.1, respectively. The depth branch provides a strong geometric prior, while the visual branch contributes appearance cues from the source clip. We deliberately keep the visual weight small: a larger weight causes the model to over-anchor to the source appearance, producing videos that ignore the target edit entirely. We use Qwen here rather than Nano Banana for cost efficiency given the scale of training-data construction; at inference time and elsewhere in our pipeline, we use Nano Banana [19] as the reference image editor (see Sec. 3.4 and Sec. 4.1). Full implementation details and dataset statistics are provided in App. G and App. M, respectively.

It is worth noting that this synthetic generation pipeline closely mirrors the architecture of Horizon-Composer itself. Initializing from a framework that already performs a closely related task lets us inherit its pretrained knowledge and yields stronger generalization than training from scratch.

3.3 Semantics Alignment with Fine-grained Reward and Offline-RL Fine-tuning

Our hybrid paired dataset $\mathcal{D}$ (Section 3.2) provides rich traffic semantics supervision (App. K), but neither source is perfect. Real-world pairs preserve scene structure yet exhibit unaligned traffic between traversals, while synthetic pairs deliver clean appearance signals yet may hallucinate or distort safety-critical semantics like lane markings. Since these artifacts are spatially sparse yet appear in nearly every video, conventional sample-level filtering (i.e., discarding "bad" videos) would decimate the dataset or admits pervasive local corruption. To exploit reliable signals while suppressing artifacts, the supervision must be grounded at the pixel level. Section 3.3.1 describes how we construct dense reward maps that combine video-editing metrics with driving-specific semantic checks to score each pixel's reliability. Section 3.3.2 formulates training as offline reinforcement learning to maximize appearance changes in reliable regions while strictly protecting vulnerable traffic semantics.

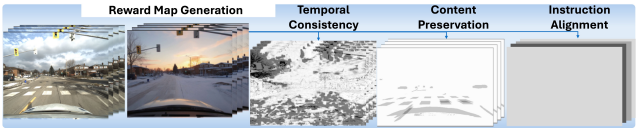


Figure 3: **Fine-grained Reward Map Generation** (Sec. 3.3.1): For each video pair, we combine frame-level video-editing metrics, pixel-level scene-change masks and driving-specific semantic checks to construct dense reward maps, which guide training toward reliable appearance edits while protecting safety-critical traffic semantics. Technical details are in App. H.

3.3.1 Constructing Fine-grained Traffic-Semantic Rewards

We construct a dense reward map $\mathbf{r} \in [-1, 1]^{3\times H\times W\times T}$ that assigns per-pixel reliability scores to each training pair along three complementary axes (Fig. 3): **Content Preservation**, **Temporal Consistency**, and **Instruction Alignment**. Aggregated together, these axes yield the improvement indicator I used in Sec. 3.3.2.

Content Preservation anchors traffic semantics by assigning low rewards to regions where source content is hallucinated or deleted. Rather than relying on simple pixel diffs, we utilize a SAM-based

scene change detector [30] to identify structural deviations. Also, we explicitly segment safety-critical elements like traffic lines, crosswalks, and vehicles using SAM3 [6], penalizing any geometric corruption. Furthermore, we employ a Vision-Language Model [12] to verify that original vehicle appearance is preserved.

Temporal Consistency measures structural fidelity across frames to penalize flickering and motion artifacts. We compute this using FF-α and FF-β [11], which track intensity-based consistency via backward-warped optical flow and motion orientation, respectively, relative to the source video.

Instruction Alignment evaluates how faithfully the generated frames reflect the target editing instruction ℓ^t. We compute a success rate using CLIP cosine similarity, comparing each frame against both the source and target prompts. Because this is a frame-level scalar metric, its value is broadcast spatially across all pixels for the given frame.

We defer per-metric definitions, prompts, and implementation details to App. H.

3.3.2 Offline Reinforcement Learning

While neither real-world nor synthetic pairs are perfect, we treat them jointly as off-policy rollouts of an imperfect behavior policy μ_{ref} defined by our hybrid dataset $\mathcal{D}$, and employ an offline RL approach to learn an improved policy μ_θ from this data. We define $\mathbf{o}$ as the conditioning inputs (i.e., the source video and text prompt) and the action $\mathbf{a} \equiv \mathbf{x}^t$ as the generated edited video clip.

CFGRL offline RL. We build on CFGRL [16], an offline RL framework that performs policy improvement via advantage conditioning. Under the standard regularized RL objective, the optimal improved policy admits $\hat{\mu}(\mathbf{a} \mid \mathbf{o}) \propto \mu_{\text{ref}}(\mathbf{a} \mid \mathbf{o}) \exp(A^{\mu_{\text{ref}}}(\mathbf{o}, \mathbf{a})/\beta)$, where $A^{\mu_{\text{ref}}}$ is the advantage of action $\mathbf{a}$ under μ_{ref}, and the β hyperparameter is the KL-regularization weight. Following [16, 22], a related product-policy formulation can be derived using $I = \mathbb{1}[A^{\mu_{\text{ref}}} > \epsilon]$ as

$$\hat{\mu}(\mathbf{a} \mid \mathbf{o}) \propto \mu_{\text{ref}}(\mathbf{a} \mid \mathbf{o}) \left(\frac{\mu_{\text{ref}}(\mathbf{a} \mid I, \mathbf{o})}{\mu_{\text{ref}}(\mathbf{a} \mid \mathbf{o})} \right)^{\beta}, \tag{1}$$

which avoids the need to explicitly model the advantage. The improved policy $\hat{\mu}$ is approximated by training μ_θ to represent both the unconditional and indicator-conditional distributions. Following [16], we randomly drop I during training to learn the unconditional prior.

$$\min_\theta \ \mathbb{E}_{\mathcal{D}} \Big[-\log \mu_\theta(\mathbf{x}^t \mid \mathbf{o}) - \alpha \log \mu_\theta(\mathbf{x}^t \mid I, \mathbf{o}) \Big]. \tag{2}$$

where α is a trade-off hyperparameter following [22]. At inference, sampling with $I = 1$ favors outputs with higher expected rewards from $\hat{\mu}$ than the behavior policy [16] (additional details in App. P).

Offline RL with dense signal. Standard CFGRL applies a single scalar indicator $I \in \{0, 1\}$ to gate an entire action. This is appropriate for low-dimensional action spaces with sparse, episodic rewards, but driving video editing inverts both assumptions: the action $\mathbf{x}^t$ is a high-dimensional pixel volume. We therefore extend the indicator to a dense, pixel-aligned mask $I \in \{0, 1\}^{H \times W \times T}$ derived from the reward map of Sec. 3.3 via $I_t(\mathbf{p}) = \mathbb{1}[\mathbf{r_t}(\mathbf{p}) > \epsilon]$. Each pixel is independently labeled positive or negative, allowing μ_θ to extract policy improvement signal from reliable regions of an otherwise noisy training example while learning to avoid the failure modes encoded in negative pixels. Crucially, this dense formulation requires no architectural change to Eq. 2, with the indicator simply broadcast to pixel resolution, making the approach scalable to higher resolutions, longer videos, or additional reward channels without modifying the underlying training objective.

Implementation. The dense indicator I is encoded as an additional control branch and injected through the visual control branch alongside the depth and reference-image conditions. Following [22], the log-likelihoods in Eq. 2 are realized as the standard flow-matching loss of the base model. During training, we drop I with probability 0.3 to retain the unconditional distribution. Having learned to associate $I = 1$ with high-quality structural preservation during training, the model is conditioned at inference by setting $I = \mathbf{1}$ globally, to treat the entire spatial volume as a high-quality target.

3.4 Thinking Frames for Smoothing Temporal Shock

While the reference image $\mathbf{c}_{\text{img}}$ is powerful for guiding global appearance transitions (e.g., shifting weather or time of day), those generated by general-purpose state-of-the-art editing methods such as Nano Banana [19] may introduce *actor misalignment and structural hallucinations*, due to the

dense co-existence of elements in driving scenes and less awareness of the traffic semantics. Naively instructing the driving video generation using these imperfect reference images often causes a "*temporal shock*," as the model struggles to immediately reconcile the static reference artifacts with the strict spatiotemporal dynamics of the traffic.

To mitigate this, we introduce **Temporal Thinking Frames**. Specifically,

1. Rather than forcing an abrupt transition from the reference image to the target video, we prepend a sequence of $N = 2$ thinking frames to the temporal dimension.
2. To align this temporal buffer with latent traffic semantics, we simultaneously prepend a temporally reflected sequence of the driving controls (e.g., depth or structural layouts). Concretely, if the initial control frames are denoted as $\mathbf{c}_{0:N-1}$, the prepended control segment is the temporal reversal $[\mathbf{c}_{N-1}, \ldots, \mathbf{c}_0]$. This reflected control prefix creates a smooth, temporally coherent "warm-up" period at the onset of generation without introducing externally synthesized, unverified structural data.
3. During these thinking frames, the model is forced to resolve the discrepancies between the image condition and the video condition. It gradually smooths out the actor misalignments present in the reference image, merging the desired global appearance into the rigid geometric constraints of the driving scene.
4. After the denoising process is complete, this prepended sequence is cropped.

Consequently, the final video retains its original length and strict semantic layout, having fully benefited from the stabilized appearance conditioning achieved during the internal thinking steps.

4 Experiments

4.1 Settings

We structure our experiments to answer our core question (Sec. 1): *How to distill rich traffic semantics into driving scene video editing and thus achieve spatiotemporal consistency?* We assess editing performance based on the preservation of traffic semantics across four perspectives:

1. **Temporal Consistency.** Following the evaluation protocol from [11], we evaluate FF-α $\downarrow$ and FF-β $\downarrow$, which quantify structural fidelity and motion alignment via optical flow.
2. **Operational Design Domain (ODD) Preservation.** We measure the cross-frame stability of foreground actors and background scenes using Subj.$\uparrow$ (DINO [7] features) and Bg.$\uparrow$ (CLIP [45] features). We measure road line IoU (Lane $\uparrow$) and car appearance preservation (Car $\uparrow$).
3. **Instruction Alignment.** We report Succ.$\uparrow$, the frame-level success rate of preferring the target over the source prompt, alongside CLIP$\uparrow$ for global text-video similarity and the multimodal large language model instruction alignment score following [32] (GPT$\uparrow$).
4. **Photorealism.** We employ Aes.$\uparrow$ (LAION [47] aesthetic score) and Img.$\uparrow$ (MUSIQ [29] imaging quality) to evaluate visual appeal and the absence of low-level visual distortions.

We build upon the video world model Cosmos-Transfer2.5 [41]. Reference image frames are generated using Nano Banana [19]. Additional details are in App. I.

In Sec 4.2 we show that our model edits driving videos consistently, preserving scene semantics per instruction. We evaluate on both withheld videos from our in-distribution datasets (Boreas [4], nuScenes [5] and Waymo [49]) and the out-of-distribution nuPlan [28] dataset. We explicitly select nuPlan test images that are non-overlapping with any nuScenes training geographic locations. Sec. 4.4 shows the importance of preserving traffic semantics for the downstream lane detection.

4.2 Main Results on Driving-Scene Video Editing

The careful design of our method across data (Sec. 3.2), training (Sec. 3.3) and model (Sec. 3.4) jointly leads to state-of-the-art traffic-relevant content preservation, particularly lane markings and vehicle appearance, and temporal consistency, while maintaining competitive instruction alignment and photorealism.

In Tab. 1, we compare HorizonComposer against the state-of-the-art editors CCEdit [15] and Uni-Video [54], as well as our synthetic generation pipeline (SyntheticPip; Sec. 3.2.2). To facilitate comparison across metrics with heterogeneous scales, we report each method's relative change with respect to SyntheticPip, the most competitive baseline. Raw scores with standard deviations are provided in Tab. 5 in the appendix.



(a) Sunny during a autumn daytime. (b) Foggy during a spring dawn. (c) Night.

Figure 4: **Qualitative Comparison of Driving Scene Editing.** We highlight key differences in semantic preservation and visual quality across methods. Target conditions are shown in text. Row order from top to bottom: input video, CCEdit, UniVideo, SyntheticPip, Ours. **(a)** SyntheticPip hallucinates a yellow traffic light from the initial image condition. **(b)** SyntheticPip introduces a new vehicle that suddenly disappears, breaking temporal consistency. **(c)** Only our method and UniVideo accurately retain the "STOP AHEAD" road marking. Overall, CCEdit regularly suffers from hazy artifacts and UniVideo distorts visual realism, whereas our approach maintains high-fidelity instruction alignment and strict content preservation.

Table 1: **Relative Improvement (%) over SyntheticPip.** We report percentage improvements computed from mean scores relative to SyntheticPip. For lower-is-better metrics, improvement is $(b - m)/b \times 100$; for higher-is-better metrics, it is $(m - b)/b \times 100$, where b and m denote baseline and model means. Positive values indicate improvement. We highlight **best** and runner-up positive improvements on In-Domain (ID) and Out-of-Distribution (OOD) scenarios. Raw scores with standard deviations are provided in Tab. 5 in the appendix.

	Temporal		Edit Align.			ODD Preservation				Quality	
Model	**FF-α ↓**	**FF-β ↓**	**Succ.↑**	**CLIP↑**	**GPT↑**	**Subj.↑**	**Bg.↑**	**Lane↑**	**Car↑**	**Aes.↑**	**Img.↑**
In-Domain (ID) Evaluation											
CC-Edit	**+18.16**	-78.85	-6.97	-4.46	-20.82	**+5.09**	**+2.94**	-65.95	-34.39	-2.01	-8.08
UniVideo	-54.30	-22.84	-13.02	-5.51	-21.39	+2.82	+0.92	-25.40	+2.73	**+0.82**	-12.57
SyntheticPip	–	–	–	–	–	–	–	–	–	–	–
Ours	+14.44	**+0.58**	-1.34	**+0.16**	-0.70	+1.70	+0.94	**+20.15**	**+7.46**	-1.49	-4.57
Out-of-Distribution (OOD) Evaluation											
CC-Edit	-1.74	-83.40	**+0.22**	**+2.69**	-16.58	**+5.46**	**+3.21**	-50.79	-32.00	-2.03	-5.99
UniVideo	-92.47	-22.69	-11.15	-1.39	-17.88	+1.65	+0.30	-29.19	+3.13	**+0.36**	**+4.33**
SyntheticPip	–	–	–	–	–	–	–	–	–	–	–
Ours	**+9.20**	**+1.47**	-4.52	-0.22	-1.25	+1.19	+0.77	**+7.43**	**+8.28**	+0.13	-3.10

Our framework emphasizes safety-critical structural fidelity. On in-domain data, our model improves temporal consistency given by FF-α, and improves content preservation with the best scores for safety-critical road markings and car appearance preservation. "Ours" is the only model able to maintain this improvement on out-of-distribution data, highlighting the generalizability of our methods. Alongside these structural gains, our method maintains competitive instruction alignment and image quality although SyntheticPip scores higher on some metrics in these dimensions. These quantitative gains are further strongly supported by qualitative analysis. As shown in Fig. 4, our approach successfully preserves safety-critical semantics such as specific vehicle appearances, road lines, crosswalks, and text markings. Baselines like SyntheticPip struggle with hallucinated traffic lights and disappearing vehicles. We show additional results in Section D.

Furthermore, our user study (Tab. 2) corroborates these findings. We polled 44 participants, each presented with 10 questions showing video outputs from each method on test examples randomly chosen from our combined In-Domain and Out-of-Distribution evaluation datasets. For each question, participants evaluated the outputs according to one randomly selected criterion: content preservation, instruction following, photorealism, or temporal consistency (see App. J for details). The study yielded 429 criterion-level responses, with overall preference computed by pooling responses across

Table 2: **User Study.** Preference rates across four evaluation criteria and overall preference pooled across criteria.

Model	Content Preservation	Instruction Following	Photorealism	Temporal Consistency	Overall
Ours	**46%**	**36%**	**43%**	**38%**	**40%**
SyntheticPip	33%	33%	33%	34%	33%
UniVideo	7%	7%	7%	7%	7%
CCEdit	5%	5%	6%	3%	5%
Tie	8%	20%	11%	19%	15%

criteria. Users preferred our model across all evaluated criteria, with relative increases in preference of approximately 39% in content preservation, 9% in instruction following, 30% in photorealism, and 12% in temporal consistency relative to SyntheticPip. These results suggest that human viewers perceive our edits as faithful to the prompt while better preserving the underlying scene content.

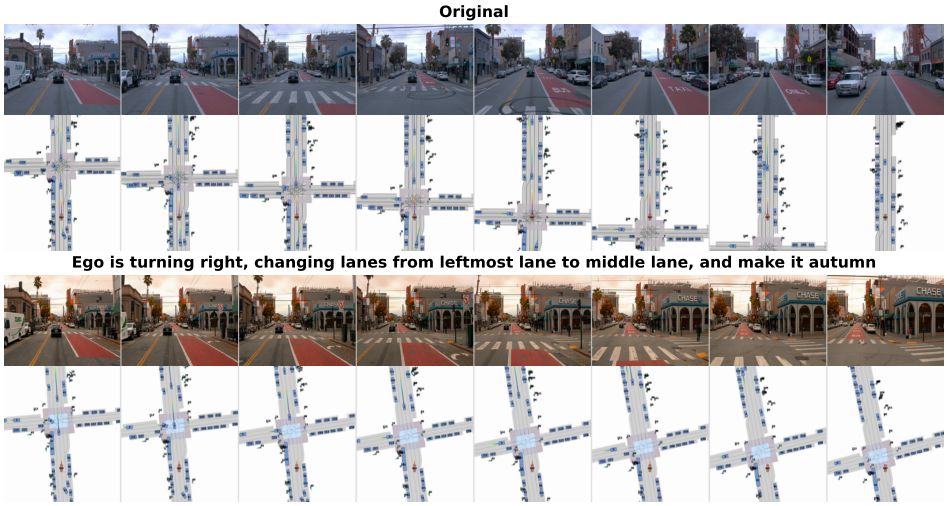


Figure 5: **Combining Local and Global Appearance Edits.** We apply HorizonComposer on top of HorizonForge [53] to combine an ego-motion edit with a global appearance edit. Given an initial scene, HorizonForge modifies the trajectory to synthesize a new vehicle maneuver (e.g., a right-turn lane change) and generates the corresponding base scene. HorizonComposer then applies global appearance edits–shifting the season to autumn while preserving the newly edited BEV trajectory and local semantics.

4.3 Extensions to Local Editing

While HorizonComposer is primarily designed for global driving-condition editing, its ability to preserve traffic semantics also makes it readily composable with localized editing approaches. To explore this extension, we build on HorizonForge [53, 21, 8], which offers local trajectory-level controls for modifying driving scenes. Given a source video together with a local editing instruction, for instance, altering the ego-vehicle motion or executing a lane change, HorizonForge first generates a locally edited video that reflects the intended structural modification. HorizonComposer is then applied to this intermediate result to impose the target global appearance condition while retaining the newly synthesized traffic layout. As illustrated in Figure 5, this pipeline achieves high-fidelity global-local editing that jointly captures trajectory adjustments and seasonal transformations. Additional video examples are provided in the supplementary material.

4.4 Downstream Task: Lane Marking Segmentation

We evaluate whether our traffic-semantic-preserving video edits benefit downstream perception tasks. Specifically, we conduct lane marking segmentation on the SeRM dataset [24, 23], which provides pixel-level annotations for 16 categories of lane markings, covering diverse road marking types critical for autonomous driving.

Table 3: Lane marking segmentation performance on SeRM.

Method	mIoU	Pixel Acc	mAcc
Original	37.48	98.76	50.04
+ SyntheticPip	38.45	98.81	53.05
+ Ours	**40.27**	**98.88**	**55.62**

We reconstruct full-resolution image–annotation pairs from SeRM and generate additional training data using both the proposed HorizonComposer and our best baseline SyntheticPip. Implementation details are provided in App. O. We fine-tune a PIDNet [59] on three training sets: (1) original data only, (2) original + SyntheticPip-generated data, and (3) original + our generated data.

We report mIoU, pixel accuracy, and mean accuracy (mAcc) following PIDNet. As shown in Tab. 3, synthetic data improves segmentation performance, and our method consistently outperforms SyntheticPip across all metrics. In particular, we improve mIoU from 37.48% to 40.27% and mAcc from 50.04% to 55.62%. The larger gain in mAcc indicates that our method better preserves fine-grained traffic semantics, leading to more effective supervision for safety-critical structures such as road markings.

4.5 Downstream Task: Multi-Object Detection

We evaluate whether videos generated by HorizonComposer benefit downstream object detection on BDD100K. We compare two fine-tuning settings initialized from the same pretrained checkpoint: (1) fine-tuning on 200 real BDD100K videos, and (2) fine-tuning on the same 200 real videos augmented with 200 additional videos generated by HorizonComposer.

Table 4: Downstream multi-object detection performance on BDD100K.

Training Data	mAP ↑	mAP@50 ↑	mAP@75 ↑
Real-200	0.167	0.316	0.154
Real-200 + Generated-200	**0.175**	**0.324**	**0.165**
Relative improvement	+4.79%	+2.53%	+7.14%

We report mean average precision (mAP), together with mAP at IoU thresholds of 0.5 (mAP@50) and 0.75 (mAP@75). As shown in Tab. 4, adding our generated videos consistently improves performance across all three metrics. Specifically, mAP increases from 0.167 to 0.175, mAP@50 from 0.316 to 0.324, and mAP@75 from 0.154 to 0.165, corresponding to relative improvements of 4.79%, 2.53%, and 7.14%, respectively. These results suggest that videos generated by HorizonComposer provide effective data augmentation for downstream perception tasks.

5 Conclusion

We present HorizonComposer, a driving-scene video editing framework for instruction-guided appearance editing. Our method distills and preserves traffic semantics through hybrid real/synthetic supervision, dense reward-conditioned offline reinforcement learning, and temporally robust thinking frames. HorizonComposer achieves strong spatiotemporal consistency while maintaining faithful instruction alignment. Experiments demonstrate improved preservation of safety-critical structures such as lane markings. The edited videos also improve downstream lane marking segmentation performance, highlighting the potential of traffic-semantic-preserving video editing for scalable autonomous driving simulation and data augmentation.

References

[1] Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025.

[2] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2024. URL `https://arxiv.org/abs/2305.13301`.

[3] Ollin Boer Bohan. Taehv: Tiny autoencoder for hunyuan video. `https://github.com/madebyollin/taehv`, 2025.

[4] Keenan Burnett, David J Yoon, Yuchen Wu, Andrew Z Li, Haowei Zhang, Shichen Lu, Jingxing Qian, Wei-Kang Tseng, Andrew Lambert, Keith YK Leung, Angela P Schoellig, and Timothy D Barfoot. Boreas:

A multi-season autonomous driving dataset. *The International Journal of Robotics Research*, 42(1-2): 33–42, 2023. doi: 10.1177/02783649231160195.

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[6] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL `https://arxiv.org/abs/2511.16719`.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. URL `https://arxiv.org/abs/2104.14294`.

[8] Wei-Jer Chang, Wei Zhan, Masayoshi Tomizuka, Manmohan Chandraker, and Francesco Pittaluga. Langtraj: Diffusion model and dataset for language-conditioned trajectory simulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26622–26631, 2025.

[9] Jiacheng Chen, Ziyu Jiang, Mingfu Liang, Bingbing Zhuang, Jong-Chyi Su, Sparsh Garg, Ying Wu, and Manmohan Chandraker. Autoscape: Geometry-consistent long-horizon scene generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25700–25711, 2025.

[10] Junyi Chen, Tong He, Zhoujie Fu, Pengfei Wan, Kun Gai, and Weicai Ye. Vino: A unified visual generator with interleaved omnimodal context, 2026. URL `https://arxiv.org/abs/2601.02358`.

[11] Yupeng Chen, Penglin Chen, Xiaoyu Zhang, Yixian Huang, and Qian Xie. Editboard: Towards a comprehensive evaluation benchmark for text-based video editing models, 2025. URL `https://arxiv.org/abs/2409.09668`.

[12] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

[13] Min Cheng, Fatemeh Doudi, Dileep Kalathil, Mohammad Ghavamzadeh, and Panganamala R. Kumar. Diffusion blend: Inference-time multi-preference alignment for diffusion models, 2026. URL `https://arxiv.org/abs/2505.18547`.

[14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *CVPR Workshop on the Future of Datasets in Vision*, volume 2, pages 1–8. IEEE, 2015.

[15] Ruoyu Feng, Wenming Weng, Yanhui Wang, Yuhui Yuan, Jianmin Bao, Chong Luo, Zhibo Chen, and Baining Guo. Ccedit: Creative and controllable video editing via diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6712–6722, 2024.

[16] Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Diffusion guidance is a controllable policy improvement operator. *arXiv preprint arXiv:2505.23458*, 2025.

[17] Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. *arXiv preprint arXiv:2310.02601*, 2023.

[18] Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrivedit: High-resolution long video generation for autonomous driving with adaptive control. *arXiv preprint arXiv:2411.13807*, 2024.

[19] Google DeepMind. Nano banana: Gemini native image generation models. `https://ai.google.dev/gemini-api/docs/image-generation`, 2025. Accessed: [Insert Date].

[20] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion, 2024. URL `https://arxiv.org/abs/2501.00103`.

[21] Yun He, Francesco Pittaluga, Ziyu Jiang, Matthias Zwicker, Manmohan Chandraker, and Zaid Tasneem. Langdrivectrl: Natural language controllable driving scene editing with multi-modal agents. *arXiv preprint arXiv:2512.17445*, 2025.

[22] Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a vla that learns from experience, 2025. URL `https://arxiv.org/abs/2511.14759`.

[23] Wonje Jang, Jhonghyun An, Sangyun Lee, Minho Cho, Myungki Sun, and Euntai Kim. Road lane semantic segmentation for high definition map. In *Proc. of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[24] Wonje Jang, Junhyuk Hyun, Jhonghyun An, Minho Cho, and Euntai Kim. A lane-level road marking map using a monocular camera. *IEEE/CAA Journal of Automatica Sinica*, Volume: 9, Issue: 1, 2022.

[25] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing, 2025. URL `https://arxiv.org/abs/2503.07598`.

[26] Luozhijie Jin, Zijie Qiu, Jie Liu, Zijie Diao, Lifeng Qiao, Ning Ding, Alex Lamb, and Xipeng Qiu. Inference-time alignment control for diffusion models with reinforcement learning guidance, 2025. URL `https://arxiv.org/abs/2508.21016`.

[27] Xuan Ju, Tianyu Wang, Yuqian Zhou, He Zhang, Qing Liu, Nanxuan Zhao, Zhifei Zhang, Yijun Li, Yuanhao Cai, Shaoteng Liu, Daniil Pakhomov, Zhe Lin, Soo Ye Kim, and Qiang Xu. Editverse: Unifying image and video editing and generation with in-context learning, 2026. URL `https://arxiv.org/abs/2509.20360`.

[28] Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, et al. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 629–636. IEEE, 2024.

[29] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5148–5157, 2021.

[30] Jaewoo Kim and Uehwan Kim. Towards generalizable scene change detection, 2025. URL `https://arxiv.org/abs/2409.06214`.

[31] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025. URL `https://arxiv.org/abs/2412.03603`.

[32] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation, 2024. URL `https://arxiv.org/abs/2312.14867`.

[33] Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks, 2024. URL `https://arxiv.org/abs/2403.14468`.

[34] Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, Shuchang Zhou, Li Zhang, Xiaojuan Qi, Hao Zhao, Mu Yang, Wenjun Zeng, and Xin Jin. Uniscene: Unified occupancy-centric driving scene generation, 2025. URL `https://arxiv.org/abs/2412.05435`.

[35] Tiancheng Li, Jinxiu Liu, Huajun Chen, and Qi Liu. Instructrl4pix: Training diffusion for image editing by reinforcement learning, 2024. URL `https://arxiv.org/abs/2406.09973`.

[36] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl, 2025. URL https://arxiv.org/abs/2505.05470.

[37] 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.

[38] Yifan Lu, Xuanchi Ren, Jiawei Yang, Tianchang Shen, Zhangjie Wu, Jun Gao, Yue Wang, Siheng Chen, Mike Chen, Sanja Fidler, et al. Infinicube: Unbounded and controllable dynamic 3d driving scene generation with world-guided video models. *arXiv preprint arXiv:2412.03934*, 2024.

[39] Sicheng Mo, Ziyang Leng, Leon Liu, Weizhen Wang, Honglin He, and Bolei Zhou. Dreamland: Controllable world creation with simulator and generative models, 2025. URL https://arxiv.org/abs/2506.08006.

[40] Chong Mou, Qichao Sun, Yanze Wu, Pengze Zhang, Xinghui Li, Fulong Ye, Songtao Zhao, and Qian He. Instructx: Towards unified visual editing with mllm guidance, 2025. URL https://arxiv.org/abs/2510.08485.

[41] NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical ai, 2026. URL https://arxiv.org/abs/2511.00062.

[42] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019. URL https://arxiv.org/abs/1910.00177.

[43] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22 (3):313–318, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882269. URL https://doi.org/10.1145/882262.882269.

[44] Chenghao Qian, Wenjing Li, Yuhu Guo, and Gustav Markkula. Weatheredit: Controllable weather editing with 4d gaussian field, 2025. URL https://arxiv.org/abs/2505.20471.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[46] Jan Philipp Schneider, Pratik Singh Bisht, Ilya Chugunov, Andreas Kolb, Michael Moeller, and Felix Heide. Neural atlas graphs for dynamic scene decomposition and editing, 2025. URL https://arxiv.org/abs/2509.16336.

[47] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 2022.

[48] Mauricio Soroco, Francesco Pittaluga, Zaid Tasneem, Abhishek Aich, Bingbing Zhuang, Wuyang Chen, Manmohan Chandraker, and Ziyu Jiang. Horizonweaver: Generalizable multi-level semantic editing for driving scenes. *arXiv preprint arXiv:2604.04887*, 2026.

[49] 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*, pages 2446–2454, 2020.

[50] Alexander Swerdlow, Runsheng Xu, and Bolei Zhou. Street-view image generation from a bird's-eye view layout. *IEEE Robotics and Automation Letters*, 2024.

[51] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL `https://arxiv.org/abs/2503.20314`.

[52] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023.

[53] Yifan Wang, Francesco Pittaluga, Zaid Tasneem, Chenyu You, Manmohan Chandraker, and Ziyu Jiang. Horizonforge: Driving scene editing with any trajectories and any vehicles. *arXiv preprint arXiv:2602.21333*, 2026.

[54] Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. Univideo: Unified understanding, generation, and editing for videos. *arXiv preprint arXiv:2510.08377*, 2025.

[55] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[56] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL `https://arxiv.org/abs/2508.02324`.

[57] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation, 2023. URL `https://arxiv.org/abs/2212.11565`.

[58] Jay Zhangjie Wu, Xuanchi Ren, Tianchang Shen, Tianshi Cao, Kai He, Yifan Lu, Ruiyuan Gao, Enze Xie, Shiyi Lan, Jose M. Alvarez, Jun Gao, Sanja Fidler, Zian Wang, and Huan Ling. Chronoedit: Towards temporal reasoning for image editing and world simulation, 2025. URL `https://arxiv.org/abs/2510.04290`.

[59] Jiacong Xu, Zixiang Xiong, and Shankar P Bhattacharyya. Pidnet: A real-time semantic segmentation network inspired by pid controllers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19529–19539, 2023.

[60] Kairui Yang, Enhui Ma, Jibin Peng, Qing Guo, Di Lin, and Kaicheng Yu. Bevcontrol: Accurately controlling street-view elements with multi-perspective consistency via bev sketch layout. *arXiv preprint arXiv:2308.01661*, 2023.

[61] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

[62] Yiran Zhao, Yaoqi Ye, Xiang Liu, Michael Qizhe Shieh, and Trung Bui. Imageedit-r1: Boosting multi-agent image editing via reinforcement learning, 2026. URL `https://arxiv.org/abs/2603.08059`.

[63] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Limitations

The efficacy of our dense reward guidance approach is inherently bounded by the accuracy of the underlying reward models. If the dense reward evaluator misclassifies structurally unreliable pixels such as for small, distant agents or under heavy occlusion, or due to neural evaluator hallucinations, the diffusion policy receives noisy improvement signals. Future work could explore joint or iterative training of the reward and diffusion models to mitigate this bottleneck.

HorizonComposer is primarily designed for global appearance edits. Its standalone capability for localized edits is therefore limited. Nevertheless, as demonstrated in Section 4.3, our framework can be composed with existing editing pipelines to support localized changes. Extending the framework to directly support localized editing is a direction for future work.

Our results demonstrate that strong ODD preservation and temporal consistency can be achieved while maintaining competitive instruction following and image quality. Our specialized reward signals primarily target key ODD elements, such as lane markings and vehicles. Extending these signals to broader scene classes is a natural direction for future generalization.

B Broader Impacts

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here. Positively, our method enables the generation of high-fidelity, spatiotemporally consistent driving edge-cases. This can be used to enrich simulation environments and train safer, more robust autonomous vehicle systems without the physical risks of real-world testing.

C Additional Videos

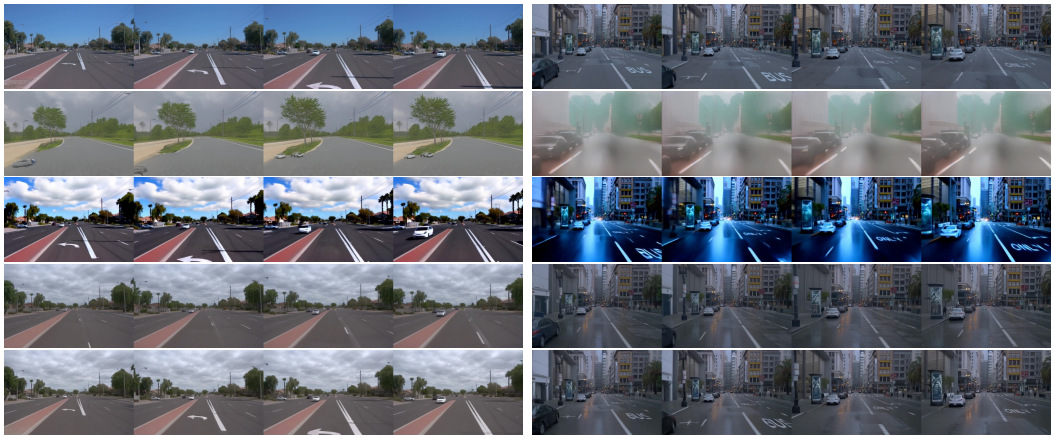
Additional video results are provided in the supplementary materials. The demo website can be accessed by opening index.html in the unzipped supplementary archive.

D Additional Results

In this section we demonstrate additional qualitative results in Fig. 6 and the raw quantitative results in Tab. 5.

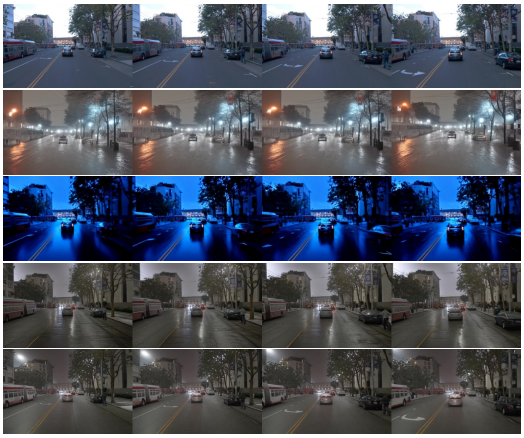
Table 5: **Quantitative Comparison on Driving Scene Editing.** We evaluate models across temporal stability, instruction alignment, ODD preservation, and overall video quality. Values indicate means with standard deviations below. We report **best** and runner-up mean performances on In-Domain (ID) and Out-of-Distribution (OOD) scenarios.

	Temporal		Edit Align.			ODD Preservation				Quality	
Model	**FF-α ↓**	**FF-β ↓**	**Succ.↑**	**CLIP↑**	**GPT↑**	**Subj.↑**	**Bg.↑**	**Lane↑**	**Car↑**	**Aes.↑**	**Img.↑**
In-Domain Evaluation (Boreas, nuScenes, Waymo)											
CC-Edit	**3.2811** ± **0.0238**	0.9638 ± 0.0005	0.8685 ± 0.0128	0.2376 ± 0.0005	5.9436 ± 0.1320	**0.9454** ± **0.0002**	**0.9710** ± **0.0000**	0.1374 ± 0.0129	0.5103 ± 0.0118	0.4928 ± 0.0075	0.5667 ± 0.0030
UniVideo	6.186 ± 0.140	0.662 ± 0.002	0.812 ± 0.013	0.235 ± 0.001	5.901 ± 0.106	0.925 ± 0.001	0.952 ± 0.001	0.301 ± 0.003	0.799 ± 0.026	**0.507** ± **0.002**	0.539 ± 0.004
SyntheticPip	4.0092 ± 0.1992	0.5389 ± 0.0038	**0.9336** ± **0.0071**	0.2487 ± 0.0006	**7.5063** ± **0.0380**	0.8996 ± 0.0022	0.9433 ± 0.0006	0.4035 ± 0.0020	0.7778 ± 0.0070	0.5029 ± 0.0050	**0.6165** ± **0.0089**
Ours	3.4303 ± 0.0775	**0.5358** ± **0.0065**	0.9211 ± 0.0052	**0.2491** ± **0.0001**	7.4541 ± 0.0219	0.9149 ± 0.0008	0.9522 ± 0.0003	**0.4848** ± **0.0030**	**0.8358** ± **0.0037**	0.4954 ± 0.0025	0.5883 ± 0.0016
Out-of-Distribution Evaluation (nuPlan)											
CC-Edit	3.7547 ± 0.0607	0.9612 ± 0.0004	**0.9294** ± **0.0114**	**0.2364** ± **0.0003**	6.2439 ± 0.1059	**0.9503** ± **0.0006**	**0.9755** ± **0.0012**	0.2849 ± 0.0122	0.5387 ± 0.0161	0.5408 ± 0.0001	0.5884 ± 0.0039
UniVideo	7.103 ± 0.135	0.643 ± 0.002	0.824 ± 0.008	0.227 ± 0.002	6.146 ± 0.066	0.916 ± 0.002	0.948 ± 0.002	0.410 ± 0.005	0.817 ± 0.002	**0.554** ± **0.004**	**0.653** ± **0.009**
SyntheticPip	3.6904 ± 0.3114	0.5241 ± 0.0062	0.9274 ± 0.0021	0.2302 ± 0.0009	**7.4845** ± **0.0013**	0.9011 ± 0.0009	0.9452 ± 0.0012	0.5790 ± 0.0023	0.7922 ± 0.0060	0.5520 ± 0.0079	0.6259 ± 0.0187
Ours	**3.3507** ± **0.1156**	**0.5164** ± **0.0065**	0.8855 ± 0.0082	0.2297 ± 0.0004	7.3907 ± 0.0189	0.9118 ± 0.0008	0.9525 ± 0.0004	**0.6220** ± **0.0062**	**0.8578** ± **0.0019**	0.5527 ± 0.0050	0.6065 ± 0.0047



(a) Cloudy conditions during spring.

(b) Cloudy conditions during an autumn dusk.



(c) Rainy conditions during a winter night.



(d) Sunny conditions during a summer dusk.



(e) Cloudy conditions at night.



(f) Cloudy conditions during a winter daytime.

Figure 6: **Qualitative Comparison of Driving Scene Editing.** We highlight key differences in semantic preservation and visual quality across methods. Target conditions are shown in text. Row order from top to bottom: input video, CCEdit, UniVideo, SyntheticPip, Ours. **(a)** SyntheticPip cannot preserve left turn lane markings. **(b)** Ours preserves the "BUS ONLY" road marking that SyntheticPip removes. **(c)** Our method successfully preserves the road turn signal that SyntheticPip erroneously removes. **(d)** Our method successfully preserves the crosswalk and the red car, which SyntheticPip erroneously removes. **(e)** Our method preserves the clearance height of the overpass. **(f)** Only UniVideo and Ours preserve the solid white line and turn signal. Overall, CCEdit regularly suffers from hazy artifacts and UniVideo distorts visual realism, whereas our approach maintains high-fidelity instruction alignment and strict Operational Design Domain (ODD) preservation.

Table 6: **Ablation Study.** Impact of our methods across data (Sec. 3.2), training (Sec. 3.3) and model (Sec. 3.4) on temporal stability and ODD preservation on out-of-distribution data.

	Temporal		Edit Align.			ODD Preservation				Quality	
Model Variant	**FF-$\alpha\downarrow$**	**FF-$\beta\downarrow$**	**Succ.↑**	**CLIP↑**	**GPT↑**	**Subj.↑**	**Bg.↑**	**Lane↑**	**Car↑**	**Aes.↑**	**Img.↑**
SyntheticPip	3.4702	0.5285	**0.9289**	0.2308	7.4855	0.9018	0.9443	0.5774	0.7880	0.5465	**0.6126**
Finetune RGB only	3.3009	0.5505	0.9234	**0.2353**	7.4600	0.9038	0.9457	0.6006	0.8243	0.5456	0.5986
+ Depth Cond.	**3.1225**	0.5442	0.9227	0.2334	**7.4970**	0.9013	0.9433	0.5984	0.8308	0.5461	0.5985
+ Dense Rewards	3.3927	**0.5225**	0.9048	0.2307	7.4206	0.9011	0.9436	0.6266	**0.8595**	**0.5482**	0.6066
+ Temp. Aug. (Ours)	3.2185	0.5235	0.8762	0.2299	7.3836	**0.9126**	**0.9525**	**0.6281**	0.8558	0.5472	0.6030

Table 7: **Ablation Study (In-Distribution).** Impact of our methods across data (Sec. 3.2), training (Sec. 3.3) and model (Sec. 3.4) on temporal stability and ODD preservation on in-distribution data.

	Temporal		Edit Align.			ODD Preservation				Quality	
Model Variant	**FF-$\alpha\downarrow$**	**FF-$\beta\downarrow$**	**Succ.↑**	**CLIP↑**	**GPT↑**	**Subj.↑**	**Bg.↑**	**Lane↑**	**Car↑**	**Aes.↑**	**Img.↑**
SyntheticPip	3.8055	0.5427	0.9375	0.2491	7.4934	0.9004	0.9427	0.4029	0.7859	0.4981	**0.6067**
Finetune RGB only	3.6594	0.5716	**0.9569**	**0.2545**	**7.5885**	0.9073	0.9485	0.4133	0.7917	**0.5027**	0.5998
+ Depth Cond.	3.4901	0.5585	0.9489	0.2525	7.5848	0.9019	0.9450	0.4276	0.8008	0.5018	0.5932
+ Dense Rewards	3.6197	**0.5411**	0.9333	0.2500	7.4897	0.8987	0.9426	**0.4904**	0.8262	0.4975	0.5979
+ Temp. Aug. (Ours)	**3.3539**	0.5428	0.9155	0.2492	7.4297	**0.9156**	**0.9520**	0.4845	**0.8351**	0.4928	0.5873

D.1 Ablation: Robust Supervision & Conditioning

To understand how our method protects traffic semantics during editing, we ablate our design in Tab. 6 and Tab. 7. Our ablation reveals: **1) Standard finetuning is insufficient.** While the SyntheticPip baseline and our minimal RGB-only finetune achieve high Edit Alignment, they suffer from semantic corruption, exhibiting the poorest Lane and Car preservation scores. **2) Rewards protect localized semantics.** Introducing dense reward conditioning yields a massive improvement in ODD Preservation. Specifically, Lane and Car appearance preservation jump significantly across both ID and OOD splits, proving that our localized rewards successfully penalize structural hallucinations. **3) Thinking frames stabilize temporal dynamics.** Adding thinking frames ("Ours") yields the highest Subject and Background consistency while improving intensity-based temporal fidelity (FF-$\alpha\downarrow$). By buffering against temporal shock, our full model strictly maintains foundational scene structure. Crucially, these structural improvements do not compromise the intended edit. While automated metrics suggest a slight trade-off, qualitative and human evaluations (Sec. 4.2) confirm that our full method maintains competitive imaging and aesthetic quality, yielding edits that users perceive as highly instruction-aligned while preserving the underlying physical reality.

E Detailed Related Works

Instruction-Guided Image/Video Editing. Recent advancements in video generative models, such as [51, 20, 31], have significantly improved general-purpose video synthesis. Yet they can fall short in domains requiring precise, fine-grained control over object dynamics and physical consistency. To improve semantic control, a growing body of work focuses on unifying multimodal understanding and generation. These frameworks typically achieve this by interleaving text, image, and video tokens into a single unified sequence or by tightly integrating Multimodal Large Language Models (MLLMs) with Diffusion Transformers [54, 40, 25, 27, 10]. Other approaches seek to explicitly decouple structural and visual guidance or employ tuning-free temporal propagation to edit videos while preserving layout [57, 33, 15]. Conceptually, our framework shares the philosophy of decoupling structure and appearance; however, these general-purpose models cannot natively guarantee the strict spatiotemporal traffic semantics required for driving scenes. Closest to our temporal approach is ChronoEdit [58], which tackles physical consistency in image editing by hallucinating intermediate video frames as "reasoning tokens" to map out a plausible physical trajectory. Our proposed *thinking frames* shares this core philosophy. However, while ChronoEdit applies this to single-image edits, we utilize thinking frames to structurally anchor continuous multi-agent dynamics and insulate the generative pipeline against conditioning-induced temporal shock.

Driving Scene & Global Appearance Synthesis. The rising demand for scalable driving-scene data has spurred the development of driving scene simulation methods through generation [50, 60, 52, 17, 55, 18, 38, 9, 53, 48, 37]. Models such as the Cosmos family [1, 41] scale physical prediction and sim-to-real translation into unified frameworks. Yet, relying purely on massive, implicit data priors often fails to strictly preserve the Operational Design Domain (ODD) during targeted appearance edits. To enforce spatial consistency, many recent approaches incorporate explicit 3D/4D representations or physics-based priors. These include pipelines that combine 2D background editing with 4D weather particle fields [44], hybrid systems that pair element-wise pixel/object controls with physics simulators [39], unified occupancy-centric frameworks [34], and representations that decompose dynamic scenes into view-dependent neural graphs [46]. While these methods offer spatial conditioning to manipulate traffic semantics, they rely on external simulation engines, and nonetheless struggle to preserve original vehicle identities or scene structures without hallucinating new structures. In contrast, our approach operates directly in the video domain. By distilling traffic semantics through dense, pixel-level rewards and a hybrid data pairing strategy, we strictly enforce ODD preservation and achieve spatiotemporal consistency without the overhead of explicit 3D or 4D reconstruction.

Reinforcement learning for diffusion models. Adapting reinforcement learning to diffusion models typically relies on policy-gradient methods [36, 62, 2, 35] (e.g., PPO), which face prohibitive sampling costs for video, or offline weighted regression methods like AWR [42], which discard valuable data and degrade into filtered imitation learning [22]. Recent work [26, 16, 13, 22] addressed these issues by shifting policy improvement to inference via reward-conditioned guidance. Yet many rely on expensive auxiliary models or sample-level scalar indicators which fail to capture how structural reliability varies drastically *within* a video frame. We extend this line of work using dense, pixel-aligned reward indicators to extract localized policy improvements from partially reliable hybrid data.

F Real-World Dataset Construction

Expanding on Section 3.2.1, we provide additional details on the real-world data construction pipeline by walking through each of its modules.

Pose-based Pairing. Our pipeline starts with a pairing process that matches frames between two traversals based on spatial proximity and orientation similarity. Concretely, we designate one traversal as the source video and seek, for each source frame, the most pose-similar frame from another traversal of the same route. Formally, for each source frame I_{source} in the source sequence, we define its camera pose as $(\vec{x}_s, \phi_s, \theta_s, \psi_s)$, where $\vec{x}_s \in \mathbb{R}^3$ is the camera position and $(\phi_s, \theta_s, \psi_s)$ denote roll, pitch, and yaw. We then define a local temporal neighborhood $\mathcal{I}$ of candidate frames drawn from a second traversal of the same route, and select the paired target frame I_{target} by minimizing a pose-disparity distance following HorizonWeaver [48]:

$$\begin{aligned} I_{\text{target}} &= \arg\min_{I \in \mathcal{I}} \ \text{dist}(I, I_{\text{source}}), \\ \text{dist}(I_a, I_b) &= \left\|\vec{x}_a - \vec{x}_b\right\|_2 + \left|\phi_a - \phi_b\right| + \left|\theta_a - \theta_b\right| \\ &\quad + \left|\psi_a - \psi_b\right|, \end{aligned} \tag{3}$$

where $(\vec{x}_a, \phi_a, \theta_a, \psi_a)$ and $(\vec{x}_b, \phi_b, \theta_b, \psi_b)$ are the poses of frames I_a and I_b, respectively. In practice, we accept matches only when $\text{dist}(I_{\text{target}}, I_{\text{source}})$ falls below a fixed threshold, ensuring that paired frames share nearly identical viewpoints. This procedure is dataset-agnostic and applies to any driving dataset with sufficiently accurate pose estimates, including but not limited to Boreas, yielding paired image sequences that depict the same physical locations under different conditions.

Temporal Filter. To ensure the quality and realism of the paired video data, the pipeline filters out segments containing "freeze frames"—periods where one ego-vehicle is stationary causing the second scene to be static for an extended duration. This is achieved by computing frame-to-frame differences (e.g., mean squared error) and splitting or discarding segments with excessive static content. The filtering step also detects and handles timestamp gaps, further improving temporal consistency. The filtered video pairs are then split into fixed-length segments with configurable overlap. This segmentation ensures that each data sample is of uniform duration, facilitating batch processing and model training. Segments shorter than a minimum threshold are discarded to maintain data quality.

Editing Instruction Generation. Finally, we use an image-based vision-language model (VLM) [12] to extract attributes for constructing the editing prompt. In Figure 7, we show the VLM prompt used to extract global scene attributes to construct the editing prompt and video caption.

```
You are an expert in autonomous driving, specializing in analyzing
    ↪ traffic scenes. You receive a series of traffic images from
    ↪ the perspective of the ego car. Your task is to describe the
    ↪ driving environment, focusing on weather, lighting, road
    ↪ layout, and environment.

It is essential that you strictly follow the rules and instructions
    ↪ below. Any deviation from the specified structure or format
    ↪ will result in an invalid output.

STRICTLY follow Rules:
 - You must strictly follow the dictionary structure provided above.
 - Only use the specified terms for weather, light, road layout, and
     ↪ environment. Do not create your own terms.
 - No additional information or categories should be added.
 - You should strictly follow these instructions. If an object or
     ↪ element is not visible or does not exist in the scene, set
     ↪ the value to 'None'. Ensure every field is filled with the
     ↪ appropriate value or 'None'.
 - STRICTLY ignore any text written on the image.


Output the result in the following dictionary format:

{
  "surrounding_info": {
    "weather": "[e.g., 'cloudy', 'sunny', 'rainy', 'fog', 'snowy']",
    "road_layout": "[Choose from: 'straight road', 'curved road',
        ↪ 'intersection', 'T-junction', 'ramp']",
    "environment": "[Choose from: 'city street', 'country road',
        ↪ 'highway', 'residential area']",
    "sun_visibility_conditions": "[Choose from: 'clear', 'foggy',
        ↪ 'low visibility', 'hazy']",
    "road_condition": "[Choose from: 'wet', 'icy', 'normal',
        ↪ 'debris', 'potholes']",
    "surface_type": "[Choose from: 'asphalt', 'gravel', 'dirt',
        ↪ 'concrete']",
    "surface_color": "[Choose from: 'light grey', 'dark grey',
        ↪ 'black', 'brown']",
    "time_of_the_day": "[Choose from: 'morning', 'midday',
        ↪ 'afternoon', 'night', 'dawn', 'dusk'.]",
    "precipitation_intensity": "[Choose from: 'none', 'light',
        ↪ 'moderate', 'heavy', 'torrential'.]",
    "precipitation_visibility_impact": "[Choose from: 'none', 'low',
        ↪ 'moderate', 'high']",
    "cloud_cover": "[Choose from: 'clear', 'light', 'moderate',
        ↪ 'heavy'.]
    }
}
```

Figure 7: Annotation pipeline prompt used for the VLM.

G Synthetic Appearance Generation Details

We construct synthetic appearance-editing pairs from unpaired driving videos using a two-stage generation pipeline. For each source clip, we first edit only the first frame according to a target condition, such as weather, season, or time of day. The edited frame replaces the original first frame to form a conditional input video, while the remaining frames are kept unchanged.

To construct editing prompts, we randomly sample 45-frame clips from Waymo [49], Boreas [4], and nuScenes [5] (10k clips for each dataset). For each clip, we use InternVL [63] to identify the original condition of the first frame, including weather, time of day, and season. The target condition is randomly sampled from the remaining valid conditions after removing the original condition and impossible combinations. The candidate condition lists are: time of day: {`None`, dusk, night, dawn, noon, daytime}; season: {`None`, summer, spring, winter, autumn}; and weather: {`None`, sunny, snowy, rainy, foggy, cloudy}, where `None` indicates that the corresponding attribute is not edited.

We then feed the conditional video and the source depth sequence into a fine-tuned Cosmos Transfer2.5 model with visual and depth control branches. For efficiency, we fine-tune the visual and depth branches using 480p videos with 45 frames. The edited first frame provides the target appearance

condition, while depth control helps preserve scene geometry and layout. We also use negative prompts to discourage object removal, hallucination, and structural deformation.

H Reward Map Construction

To construct our fine-grained reward map $\mathbf{r} \in [-1, 1]^{3\times H\times W\times T}$, we combine complementary video-editing metrics from Chen et al. [11] with custom, driving-specific semantic checks. We show our reward maps in Fig. 3. These signals identify which regions provide reliable supervision for the target edit and which regions contain safety-critical traffic semantics that must be protected. The video-editing metrics cover three dimensions: Temporal and Motion Consistency, Instruction alignment, and Subject and Scene Consistency.

Content Preservation. Synthetic videos may contain unexpected scene changes, such as object removal, hallucination, or structural deformation. If directly used as supervision, these artifacts may propagate into our model. To identify such unreliable regions, we adopt the generalizable scene change detection method from [30], which uses SAM features to produce scene change candidates and refines them with geometric-semantic mask matching. We generate dense masks marking regions where the synthetic target deviates structurally from the source.

Furthermore, we design two driving aware metrics for safety critical components. First we target traffic lines, crosswalks, and painted arrows. We segment them with SAM3 [6] and classify each pixel into one of four states: strictly preserved, hallucinated in the target, deleted in the target, or background. Second, we use SAM3 to extract vehicle masks in both the input and ground truth videos. We greedily match vehicle instances based on pixel Intersection-over-Union (IoU) and prompt a Vision-Language Model from [12] to determine if the original body paint is accurately preserved. Together, these spatial masks heavily penalize the Content Preservation reward channel wherever source geometry is corrupted. The prompts are provided in Sec. I.

Temporal Consistency. To measure structural fidelity across frames relative to the source video, we utilize FF-α and FF-β. FF-α measures intensity-based consistency using optical flow (l_i) extracted from the original video to backward-warp the edited frames. Due to scenarios with significant object displacement where warping becomes unreliable (low valid pixel percentage), we also use FF-β which compares the orientation of motion.

Instruction Alignment. To assess how well our model implements the requested edits while maintaining high-quality generation, we compute a *Success Rate* by measuring the CLIP cosine similarity of each edited frame f_i' against both the source prompt (p_s) and the target prompt (p_t). A frame is considered successful if its alignment to p_t is greater than to p_s. Since these are scalar metrics, their values are broadcast spatially across the frame to populate the Instruction Alignment channel of the dense reward map.

I Evaluation Protocol

To comprehensively evaluate our method, we assess editing performance across four distinct axes. We construct this protocol by combining general video-editing metrics [11], advanced multimodal large language model alignment scores [32], and our own driving-aware structural checks.

- **Temporal Consistency:** We quantify structural fidelity and motion stability using two complementary metrics. FF-α measures intensity-based consistency by backward-warping edited frames using the source video's optical flow, which is highly sensitive to minor structural changes. Since optical flow warping may be unreliable during large displacements, we also measure FF-β, which directly evaluates motion orientation consistency even when objects undergo significant positional changes.
- **Content & ODD Preservation:** To ensure the edit does not corrupt the underlying scene, we measure preservation at both the general and driving-specific levels. For general scene preservation, we track the cross-frame stability of foreground actors (Subj.) using DINO [7] features, and background environments (Bg.) using CLIP [45] features. For strict driving-aware preservation, we measure Road Line Preservation (Lane ↑) via spatial IoU of SAM3 masks. To quantify Car Appearance Preservation (Car ↑), we first use SAM3 to extract vehicle masks from both the input and generated videos. Next, we greedily match vehicle instances based on pixel Intersection-over-Union (IoU) and prompt InternVL to verify that the matched vehicles retain their original body paint. Specifically, we use the following prompt:

```
"Aligned crops of one vehicle:  first image from the original frame,
second from a generated frame.  Is the body paint the same color?  Ignore
lighting, white balance, and window reflections.  Reply with only JSON:
{"same":  true} or {"same":  false}"
```

- **Instruction Alignment:** We evaluate how faithfully the model executes the text prompt. The Success Rate (Succ.) calculates the frame-level frequency where the generated frame aligns better with the target prompt than the source prompt in CLIP latent space. We also report the global CLIP Similarity (CLIP) between the edited frames and the target text. Furthermore, following the evaluation framework in Ku et al. [32], we quantify complex instruction alignment using a multimodal large language model's judgment to capture nuanced text-to-video fidelity.
- **Photorealism:** We measure overall visual appeal using the LAION aesthetic predictor [47] (Aes.) and assess the absence of low-level visual distortions (e.g., blur, noise) using MUSIQ [29] (Img.).

J User Study Details

To evaluate our approach, we conducted a user study comparing HorizonComposer against three baselines: CC-Edit, UniVideo, and SyntheticPip. Test samples were randomly chosen from our combined in-distribution (ID) and out-of-distribution (OOD) evaluation sets.

For each survey instance, participants were presented with a source video, a target editing prompt, and the resulting videos generated by all four methods. To prevent bias, the order of the generated videos was randomized for each question. For each trial, participants evaluated the outputs according to one criterion selected at random from the 4 criteria below. If no method was a clear winner, users were instructed to select "Comparable." The criteria presented to the users were as follows:

> **Temporal Consistency**: Select the edit that has the best temporal consistency *(e.g., has the least sudden jumps, flickering, appearance changes, broken motion, etc.)*.
>
> **Content Preservation**: Select the edit that best preserves the original content *(e.g., preserves original car colours, road line patterns, traffic light states, building geometry, plants, etc.)*.
>
> **Instruction Following**: Select the edit that best follows the instruction *(e.g., most closely matches the target editing prompt)*.
>
> **Photorealism**: Select the edit that has the best photorealism *(e.g., realistic colours, lighting, natural appearance, etc.)*.

We polled 44 participants with each participant being provided 10 questions. Each question assessed one randomly selected criterion, yielding 429 responses in total. The overall scores were computed as the total preference pooled across criteria.

K Training and Inference Details

K.1 Training & Architecture Details

We instantiate our framework on top of Cosmos-Transfer2.5 [41], a diffusion transformer (DiT) that performs latent-space video denoising under multiple structured control branches. To achieve our goal of spatiotemporal consistency, we isolate two specific spatial controls: depth and visual (RGB). The depth branch is frozen to strictly preserve the geometric layout, using maps extracted via Depth Anything V2 [61]. Although originally designed as a de-blurring control, we repurpose the visual branch by providing sharp, unblurred inputs (see "Finetune RGB only" in Sec. D.1 for an isolated ablation of this branch). Finally, to guide the generation, both HorizonComposer and the SyntheticPip baseline are conditioned on an edited image context frame $\mathbf{c}_{\text{img}}$ (Sec. K.2), which is provided by external image editors at inference time.

We train this model with the standard diffusion objective parameterized via v-prediction on our hybrid dataset across 2×80GB NVIDIA H100 GPUs with a total batch size of 2 for 28000 steps with a temporal latent length of 12 frames at 480p resolution over 48 GPU hours. For optimization, we use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay of 1e-3) with a LambdaLR scheduler, a peak learning rate of 8.63e-5, and a 100-step warmup.

During inference we denoise for 35 steps with reward guidance scale 2.0 and text guidance scale 2.0. We use the FlowUniPCMultistepScheduler inference scheduler. During inference, we generate a total of 93 frames, comprising 85 edited video frames and 8 intermediate transition frames derived from 2 'thinking latents. We use the VAE from [3] which has a temporal compression factor of 4.

K.2 Image Context Robustness via Augmentation

During training, we apply an image-context augmentation to the reference frame used for conditioning. Specifically, we set the augmentation mode to Poisson blending [43] and apply it with probability 0.75; with the remaining 0.25 probability, the selected reference frame is left unchanged. When augmentation is triggered, we sample a random temporal frame from the video to edit as the donor frame. A spatial mask is generated using a random ellipse, and the donor content is composited into the target context frame via Poisson blending. This produces a locally edited but photometrically coherent reference image that preserves global scene structure while introducing controlled local appearance variation.

The objective is to increase robustness of reference-image conditioning by exposing the model to realistic, spatially localized context perturbations rather than only clean frame-level supervision. Because the blend is constrained by an irregular elliptical mask and solved with seamless cloning, the augmented context remains visually plausible while still encouraging the model to tolerate partial mismatches between reference appearance and target trajectory.

L Dataset Impacts

In Tab. 8 and Tab. 9, we show the impacts of our dataset construction. While real-world data carries groundtruth information about imaging quality and instruction alignment, due to temporal misalignments and traffic changes it cannot supervise safety critical car appearance preservation. Results show that training on hybrid datasets shows drastic improvements in temporal consistency and traffic semantic preservation. These gains far outweigh the minor performance drops observed in instruction alignment.

Table 8: **Ablation on Data Mixture.** We evaluate training exclusively on synthetic edits (Pseudo-only) versus real-world data (Real-only), compared to our hybrid approach. Results are computed on the OOD test data. Best results are **grey bold**; runner-ups are light grey.

	Temporal		Edit Align.			ODD Preservation				Quality	
Data	**FF-α↓**	**FF-β↓**	**Succ.↑**	**CLIP↑**	**GPT↑**	**Subj.↑**	**Bg.↑**	**Lane↑**	**Col.↑**	**Aes.↑**	**Img.↑**
Ours (Pseudo-only)	3.3004	0.5402	**0.9048**	**0.2342**	**7.4230**	0.9119	0.9520	0.5615	0.8296	0.5348	0.5975
Ours (Real-only)	3.4974	**0.5215**	0.8998	0.2309	7.3903	**0.9137**	**0.9540**	0.6246	0.8337	**0.5536**	**0.6135**
Ours	**3.2185**	0.5235	0.8762	0.2299	7.3836	0.9126	0.9525	**0.6281**	**0.8558**	0.5472	0.6030

Table 9: **Relative Improvement (%) over Hybrid Data.** We report percentage improvement compared to our hybrid dataset for temporal stability, instruction alignment, ODD preservation, and overall video quality showing **best** and runner-up improvements. Results are computed on the OOD test data.

	Temporal		Edit Align.			ODD Preservation				Quality	
Data Variant	**FF-α↓**	**FF-β↓**	**Succ.↑**	**CLIP↑**	**GPT↑**	**Subj.↑**	**Bg.↑**	**Lane↑**	**Car↑**	**Aes.↑**	**Img.↑**
Ours (Pseudo-only)	-2.54	-3.20	**+3.27**	**+1.85**	**+0.53**	-0.07	-0.06	-10.61	-3.06	-2.27	-0.91
Ours (Real-only)	-8.67	**+0.37**	+2.69	+0.43	+0.09	**+0.13**	**+0.16**	-0.56	-2.59	**+1.16**	**+1.74**
Ours	–	–	–	–	–	–	–	–	–	–	–

M Dataset Statistics

M.1 Synthetic Paired Videos

We construct a synthetic paired driving-video editing dataset from Boreas [4], nuScenes [5], Waymo [49], and nuPlan [28]. Unlike the original autonomous-driving datasets, which mainly provide raw driving logs or perception/planning annotations, our dataset provides explicit editing supervision. Each sample contains a 45-frame source clip, an edited target clip, a natural-language instruction, and a first-frame reference image. As summarized in Table 10, our dataset converts existing driving logs into paired source/target videos with controllable appearance-editing targets.

Table 10: Comparison between original autonomous-driving datasets and our generated video-editing dataset. Existing datasets provide raw driving logs or perception/planning annotations, while our dataset provides paired source/target videos with explicit editing supervision.

Dataset	Native unit	Native length	Paired videos	Target diversity	Supervision
Boreas [4]	Driving logs	Variable-length routes	No	Natural weather/season variation	Raw sensor data, poses
nuScenes [5]	1,000 scenes	20s per scene	No	Natural urban conditions	3D detection/tracking annotations
Waymo [49]	1,150 segments	20s / 200 frames per segment	No	Natural driving conditions	2D/3D perception annotations
nuPlan [28]	Driving logs	Large-scale variable-length logs	No	Multi-city driving scenarios	Planning annotations
Ours	Editing pairs	45 frames per clip	**Yes**	**Weather, time, season**	**Edited target video, reference image**

Table 12: Distribution by edited attribute for synthetic training data, in-domain test data, and OOD test data splits. Rows denote original conditions and columns denote target conditions. Clips with `None` for the corresponding attribute are excluded.

Train

Weather

Orig.→Tgt.	sunny	cloudy	rainy	snowy	foggy
sunny	–	2708	2687	2358	2672
cloudy	1800	–	2144	1688	2134
rainy	714	801	–	578	750
snowy	145	200	206	–	178
foggy	171	182	187	149	–

Time of Day

Orig.→Tgt.	day	noon	dawn	dusk	night
day	–	4794	4822	4806	4345
dawn	34	37	–	50	27
dusk	426	426	427	–	364
night	530	560	568	554	–

Season

Orig.→Tgt.	spring	summer	autumn	winter
spring	–	312	435	471
summer	2198	–	2235	2239
autumn	3247	2572	–	3348
winter	1040	946	1135	–

In-Domain Test

Weather

Orig.→Tgt.	sunny	cloudy	rainy	snowy	foggy
sunny	–	20	15	13	13
cloudy	17	–	9	8	8
rainy	7	3	–	4	3
snowy	0	2	1	–	0
foggy	0	1	0	1	–

Time of Day

Orig.→Tgt.	day	noon	dawn	dusk	night
day	–	27	27	27	27
dusk	7	1	1	–	0
night	8	0	0	0	–

Season

Orig.→Tgt.	spring	summer	autumn	winter
spring	–	4	4	4
summer	9	–	19	11
autumn	10	20	–	13
winter	8	6	5	–

OOD Test

Weather

Orig.→Tgt.	sunny	cloudy	rainy	snowy	foggy
sunny	–	26	23	25	24
cloudy	24	–	2	0	1

Time of Day

Orig.→Tgt.	day	noon	dawn	dusk	night
day	–	30	30	30	28
dusk	2	0	0	–	1

Season

Orig.→Tgt.	spring	summer	autumn	winter
spring	–	19	1	4
summer	27	–	29	23
autumn	2	9	–	2

Table 11: Video counts in our generated driving-video editing dataset.

Split	Boreas	nuScenes	Waymo	nuPlan	Total
Train	9,406	9,496	9,446	–	28,348
ID Test	50	50	50	–	150
OOD Test	–	–	–	150	150

Table 11 reports the dataset scale. We generate 28,348 training clips from Boreas, nuScenes, and Waymo, and reserve held-out clips from the same datasets for in-domain testing. We use nuPlan only for out-of-distribution testing to evaluate cross-dataset generalization. For training, the first-frame reference image is generated with Qwen Image Editing [56] for cost efficiency at scale. For both in-domain and OOD testing, the reference image is generated with Nano Banana Pro [19].

The editing instructions cover global appearance changes over weather, time of day, and season. Table 12 shows the training-set, ID and OOD testing set transition distribution for each edited attribute. Rows denote the original condition and columns denote the target condition. Clips with `None` for a given attribute are excluded from that attribute matrix, since that attribute is unchanged while another attribute is edited.

M.2 Real-World Paired Videos

In addition to synthetic paired videos, we also use real-world pose-paired videos from Boreas [4]. These pairs are constructed by matching repeated-route traversals according to camera pose (Sec. F). We employ a bidirectional training strategy where the source and target roles within each pair are interchangeable. Consequently, the 33,750 unique pairs yield 67,500 directional training examples.

To report the condition-transition distribution in Table 13 we normalize condition labels into the same taxonomy used by the synthetic dataset: `clear`→`sunny`, `morning`→`dawn`, `afternoon`→`daytime`, and `evening`→`dusk`.

M.3 Overall Editing Coverage

Table 14 summarizes how many clips edit each condition type. A single clip may edit multiple attributes simultaneously, so the weather, time-of-day, and season counts are not mutually exclusive.

Table 13: Condition-transition distribution of the real-world pose-paired Boreas data. Rows denote source conditions and columns denote target conditions. Counts only include clips with normalized non-`None` labels for the corresponding attribute.

Real-world Paired Dataset

Weather

Orig.→Tgt.	sunny	cloudy	rainy	snowy	foggy
sunny	–	9641	1520	3506	578
cloudy	9641	–	984	2183	325
rainy	1520	984	–	390	57
snowy	3506	2183	390	–	40
foggy	578	325	57	40	–

Time of Day

Orig.→Tgt.	day	noon	dawn	dusk	night
day	–	8116	4439	889	825
noon	8116	–	4216	806	800
dawn	4439	4216	–	558	550
dusk	889	806	558	–	89
night	825	800	550	89	–

Season

Orig.→Tgt.	spring	summer	autumn	winter
spring	–	2933	3613	5277
summer	2933	–	2500	4149
autumn	3613	2500	–	4261
winter	5277	4149	4261	–

Note: Some clips keep a given attribute unchanged or have missing condition annotations, so they are excluded from the corresponding transition matrix.

Table 14: Number of videos editing each condition type. Counts are not mutually exclusive because one clip may edit multiple attributes.

Dataset	Weather	Time of Day	Season	Total Videos
Train-Boreas	7510	7529	7106	9406
Train-nuScenes	7559	7643	7139	9496
Train-Waymo	7590	7598	7018	9446
ID Test-Boreas	42	42	40	50
ID Test-nuScenes	42	41	40	50
ID Test-Waymo	42	42	40	50
OOD-nuPlan	125	121	120	150
Train-Real-World-Boreas	38448	42576	45466	57,966
Total	**61358**	**65592**	**66969**	**86614**

The synthetic splits are edited by construction, while the real-world Boreas parquet pairs may contain unchanged or missing condition attributes.

Table 15 reports how many attributes are edited simultaneously. Most synthetic videos edit two or three attributes, providing compound appearance changes such as snowy winter dawn or rainy daytime autumn. For `Train-Real-World-Boreas`, 9,534 of 67,500 videos have no valid condition change or contain missing annotations; therefore, its total edited count is smaller than its total number of videos.

Table 16 further shows the target-value distribution across all synthetic and real-world paired data. The resulting dataset covers all target weather, time-of-day, and season categories, while preserving the natural imbalance inherited from real driving logs.

N Reward Impacts

Tab. 17 shows that combining the temporal consistency, instruction alignment, and content preservation reward channels during training yields the strongest overall generalization to out-of-distribution datasets. Crucially, this combined approach *simultaneously* maximizes both temporal stability and content preservation, while maintaining stable imaging quality and competitive instruction alignment. The corresponding raw scores are provided in Tab. 18.

O Downstream Experiment Details

The SeRM dataset only provides annotations for cropped bottom-road images, but we aim to study segmentation performance on full-resolution images. We first project the annotations onto full-resolution images (Section O.1), then generate edited video clips (Section O.2), and finally perform training (Section O.3).

O.1 SeRM Full-Resolution Pair Reconstruction

The original SeRM release provides two disjoint image sources: cropped training images with pixel-wise lane-marking annotations, and full-resolution mapping frames recorded across three routes: Sangam, Ilsan, and Sangam_Large. The training images have resolution 320×1280, while the mapping frames have resolution 672×1280. Each training crop corresponds to the bottom region of

Table 15: Number of conditions (weather, season, and time of day) edited simultaneously. `1-cat.`, `2-cat.`, and `3-cat.` indicate whether one, two, or all three attributes are changed.

Dataset	1-cat.	2-cat.	3-cat.	Total Videos
Train-Boreas	1097	3879	4430	9406
Train-nuScenes	1122	3903	4471	9496
Train-Waymo	1086	3960	4400	9446
ID Test-Boreas	4	18	28	50
ID Test-nuScenes	3	21	26	50
ID Test-Waymo	3	20	27	50
OOD-nuPlan	16	52	82	150
Train-Real-World-Boreas	10696	26016	21254	57966
Total	**14027**	**37869**	**34718**	**86614**

Table 16: Number of edits per target subtype across all paired data.

Weather

Target	sunny	cloudy	snowy	rainy	foggy
Count	18171	17115	10974	8265	6833

Time of Day

Target	noon	dawn	day	dusk	night
Count	19813	15638	15276	7809	7056

Season

Target	winter	spring	autumn	summer
Count	20079	18656	14536	13698

a full mapping frame, starting at row $y = 352$. We reconstruct full-resolution image–annotation pairs by recovering this correspondence and padding the cropped labels back to the full image canvas.

We first obtain an approximate train-to-full-frame correspondence using compact image features. For every training image and candidate full frame, we crop the bottom 320 rows when needed, downsample the image to 96×24 grayscale, and flatten it into a 2,304-dimensional feature vector. We compute an average perceptual hash for each image and retrieve the top candidate full frames by Hamming distance. These candidates are then re-ranked by the mean-squared error of z-score-normalized features, yielding an initial mapping from each training crop to a full-resolution frame.

We then verify the mapping using exact pixel-level hashing. All full mapping frames are concatenated into a single ordered sequence following the route order Sangam $\rightarrow$ Ilsan $\rightarrow$ Sangam_Large. For each full frame, we decode the image into raw RGB bytes and crop the bottom region using the fixed crop window $(x, y, w, h) = (0, 352, 1280, 320)$. We compute an MD5 hash for this decoded crop and compare it with the MD5 hash of each decoded training image. Matching is performed with a forward temporal cursor, so accepted matches must respect the original recording order. This produces a verified mapping where every retained training crop has an exact pixel match in the full-frame sequence.

Finally, we export the feasible subset as full-resolution image–annotation pairs. For each verified match, we copy the 672×1280 full RGB frame as the image. We construct the corresponding 672×1280 semantic label map by placing the original 320×1280 annotation into rows 352–671 of a zero-initialized canvas. The top 352 rows are assigned label 0, indicating unlabeled/background regions. We also save the verified mapping file to keep the route, frame index, and filename provenance for each reconstructed pair.

After this preprocess, we get about 100 clips with annotations.

O.2 Video Clip Construction and Target Assignment

Starting from the verified full-resolution SeRM pairs, we partition the ordered sequence into non-overlapping 93-frame clips at 10 fps, corresponding to approximately 9.3 seconds per clip. A trailing partial clip is retained only if it contains at least 46 frames; otherwise it is discarded.

We assign each clip a target appearance condition sampled from weather, season, and time-of-day attributes. The prepared clips are then used for video generation for both SyntheticPip and our model. For SyntheticPip, we generate videos following Sec. 3.2.2.

Fig. 8 visualizes the generated augmentation data used for downstream training, where our method better preserves lane markings while SyntheticPip often distorts or removes them.

Exclude 8 thinking frames for each clip, we get $100 * 85 = 8500$ frames. Because global editing should not change the scene, the generated frames can share the same annotation with the original frames.

Table 17: **Rewards: Relative Improvement over Baseline.** We evaluate the impact of individual components (content, instruction, temporal) compared to our full model and the SyntheticPip baseline. We report percentage improvement on OOD data compared to the SyntheticPip baseline. **Best** and runner-up improvements are highlighted.

	Temporal		Edit Align.			ODD Preservation				Quality	
Variant	**FF-α↓**	**FF-β↓**	**Succ.↑**	**CLIP↑**	**GPT↑**	**Subj.↑**	**Bg.↑**	**Lane↑**	**Car↑**	**Aes.↑**	**Img.↑**
SyntheticPip (baseline)	–	–	–	–	–	–	–	–	–	–	–
only content	+5.55	+0.15	-3.39	**+0.45**	-0.20	+1.15	**+0.89**	+7.70	+7.85	-0.29	-2.01
only instruction	+4.84	+0.53	-4.12	+0.06	-0.77	+1.21	+0.88	+8.14	+7.98	**+0.87**	-0.59
only temporal	+5.19	+0.13	-3.33	+0.06	-1.41	**+1.23**	+0.84	+7.79	+8.73	+0.35	-1.18
Ours	**+7.25**	**+0.94**	-5.67	-0.37	-1.36	+1.19	+0.87	**+8.78**	**+8.61**	+0.14	-1.57

Table 18: **Rewards: Component Ablation.** We evaluate the impact of individual components (content, instruction, temporal) compared to our full model and the SyntheticPip baseline on OOD data. Best results are **grey bold**; runner-ups are light grey.

	Temporal		Edit Align.			ODD Preservation				Quality	
Variant	**FF-α↓**	**FF-β↓**	**Succ.↑**	**CLIP↑**	**GPT↑**	**Subj.↑**	**Bg.↑**	**Lane↑**	**Col.↑**	**Aes.↑**	**Img.↑**
SyntheticPip	3.4702	0.5285	**0.9289**	0.2308	**7.4855**	0.9018	0.9443	0.5774	0.7880	0.5465	**0.6126**
only content	3.2777	0.5277	0.8973	0.2318	7.4703	0.9122	**0.9527**	0.6219	0.8498	0.5449	0.6003
only instruction	3.3022	0.5257	0.8906	0.2309	7.4279	**0.9127**	**0.9527**	0.6244	0.8509	**0.5512**	0.6090
only temporal	3.2901	0.5278	0.8980	0.2309	7.3800	**0.9129**	0.9522	0.6223	**0.8568**	0.5483	0.6054
Ours	**3.2185**	**0.5235**	0.8762	0.2299	7.3836	0.9126	0.9525	**0.6281**	**0.8558**	0.5472	0.6030

O.3 Training

We use PIDNet [59] pretrained on CityScapes [14] as our initial model. Then we finetuned it under following 3 settings: (1) only original preprocessed 8500 frames; (2) preprocessed 8500 frames and generated 8500 frames from SyntheticPip with original annotation; (3) preprocessed 8500 frames and generated 8500 frames from our method with original annotation. Low quality generation results which remove or distort lane markings may misguide the model, as results shown in Fig. 8.

P Offline Reinforcement Learning

While neither real-world nor synthetic pairs are perfect, we treat them jointly as off-policy rollouts of an imperfect behavior policy μ_{ref} defined by our hybrid dataset $\mathcal{D}$, and drawing on the offline RL perspective of CFGRL, we use reward-conditioned fine-tuning to learn an editing policy μ_θ from this data. We define $\mathbf{o}$ as the conditioning inputs (i.e., the source video and text prompt) and the action $\mathbf{a} \equiv \mathbf{x}^t$ as the generated edited video clip.

CFGRL offline RL. We build on CFGRL [16], an offline RL framework that performs policy improvement via advantage conditioning. Under the standard regularized RL objective,

$$J(\mu_\theta) = \mathbb{E}_{\tau\sim p(\tau|\mu_\theta)}\Big[\sum_t \gamma^t r(\mathbf{o}_t, a_t)\Big] - \beta\mathbb{E}_{\mathbf{o}\sim p_\mu(\mathbf{o})}[D_{\text{KL}}(\mu_\theta(\mathbf{a}\mid\mathbf{o}) \,\|\, \mu_{\text{ref}}(\mathbf{a}\mid\mathbf{o}))] \tag{4}$$

the optimal improved policy can be viewed as a probabilistic (multiplicative) adjustment to the reference and takes the form $\hat{\mu}(\mathbf{a}\mid\mathbf{o}) \propto \mu_{\text{ref}}(\mathbf{a}\mid\mathbf{o})\exp(A^{\mu_{\text{ref}}}(\mathbf{o},\mathbf{a})/\beta)$, where $A^{\mu_{\text{ref}}}$ is the advantage of action $\mathbf{a}$ under μ_{ref}, and β is a hyperparameter [16]. Following [16, 22], the product policy can similarly be defined as $\hat{\mu}(\mathbf{a}\mid\mathbf{o}) \propto \mu_{\text{ref}}(\mathbf{a}\mid\mathbf{o})p(I \mid A^{\mu_{\text{ref}}}(\mathbf{o},\mathbf{a}))^\beta$ where $p(I \mid A^{\mu_{\text{ref}}}(\mathbf{o},\mathbf{a})) = f(A^{\mu_{\text{ref}}}(\mathbf{o},\mathbf{a}))/\int f(A^{\mu_{\text{ref}}}(\mathbf{o},\mathbf{a}'))d\mathbf{a}'$ is the probability of any action improving over μ_{ref} for a non-negative monotonically increasing function f of the advantage. Under this formulation, the idealized policy, $\hat{\mu}$ results in an improvement over μ_{ref}.

Bayes rule gives $p(I \mid A^{\mu_{\text{ref}}}(\mathbf{o},\mathbf{a})) \propto \frac{\mu_{\text{ref}}(\mathbf{a}|I,\mathbf{o})}{\mu_{\text{ref}}(\mathbf{a}|\mathbf{o})}$ leading to an alternative form of the improved policy

$$\hat{\mu}(\mathbf{a}\mid\mathbf{o}) \propto \mu_{\text{ref}}(\mathbf{a}\mid\mathbf{o})\left(\frac{\mu_{\text{ref}}(\mathbf{a}\mid I,\mathbf{o})}{\mu_{\text{ref}}(\mathbf{a}\mid\mathbf{o})}\right)^\beta, \tag{5}$$

This can be written in terms of an improvement indicator $I = \mathbb{1}[A^{\mu_{\text{ref}}} > \epsilon]$, which avoids the need to explicitly model the advantage. The improved policy $\hat{\mu}$ is approximated by training μ_θ to represent



Figure 8: Qualitative comparison on SeRM lane marking segmentation. Each column shows one target scene, with rows corresponding to the original image, ground-truth annotation overlay, SyntheticPip-augmented training data, and our augmented data.

both the unconditional and indicator-conditional distributions. We apply insights from classifier-free guidance and randomly dropping I during training to learn the unconditional and indicator-conditional distributions

$$\min_{\theta} \mathbb{E}_{\mathcal{D}}\Big[-\log \mu_\theta(\mathbf{x}^t \mid \mathbf{o}) - \alpha \log \mu_\theta(\mathbf{x}^t \mid I, \mathbf{o})\Big]. \tag{6}$$

where α is a trade-off hyperparameter following [22]. At inference, we sample with $I = 1$ to favor outputs associated with higher reward.

Q Reward Granularity

To isolate the effect of dense reward conditioning, we train a scalar-reward baseline by aggregating the temporal-consistency, content-preservation, and instruction-alignment signals used by Horizon-Composer. We compare this baseline with our dense reward conditioning on the in-distribution evaluation set. As shown in Tab. 19, dense reward conditioning improves structural fidelity, with relative gains of 18.81% in lane preservation and 3.51% in car preservation. It also improves temporal consistency measured by FF-β and visual quality measured by Aes. and Img., while the remaining metrics exhibit small differences.

Table 19: Dense versus scalar reward conditioning on the in-distribution evaluation set. Relative changes are computed with respect to the scalar-reward baseline; negative changes indicate improvements for FF-α and FF-β.

Method	FF-α ↓	FF-β ↓	Succ. ↑	CLIP ↑	Lane ↑	Car ↑	Subj. ↑	Bg. ↑	Aes. ↑	Img. ↑
Scalar reward	**3.3313**	0.5619	**0.9205**	**0.2494**	0.4078	0.8068	0.9141	**0.9528**	0.4795	0.5785
Dense reward (ours)	3.3539	**0.5428**	0.9155	0.2492	**0.4845**	**0.8351**	**0.9156**	0.9520	**0.4928**	**0.5873**
Relative change	+0.68%	-3.40%	-0.54%	-0.08%	+18.81%	+3.51%	+0.16%	-0.08%	+2.77%	+1.52%

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction sections explicitly state the main contribution (Comprehensive driving dataset, reward guidance during training, temporal augmentation frames) and the results sections back up each of these claims with empirical evidence.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We add a limitations section in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper is empirical; it introduces no theorems nor formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide our training setting details in the appendix. We also provide details on our dataset construction and reward map construction in the methods and appendix sections.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Due to company restrictions.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We provide the data statistics, network architecture, and all training and inference hyperparameters in section 2, section 3 and the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report error bars on our results in Table 5 . In addition, we evaluate our method across 2 datasets with 150 examples each and repeated over multiple ablation experiments. The consistent improvements observed across these diverse settings suggest that the reported gains are unlikely to be due to random variation.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the Appendix we give our training GPU hours.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses only synthetic simulation data and public driving scene images involving no personal or sensitive information, so it aligns with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This is discussed in the Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All employed code and datasets used in this work are properly cited, and their licenses permit use for research purposes.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We provide the user study details and all prompts in the appendix. The videos were randomly sampled from the evaluation set and results discussed in Section 4 and the Appendix. No compensation was involved.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: While we conducted a crowdsourced user study to evaluate visual preferences, we did not seek formal IRB approval. The study consisted solely of a standard A/B visual comparison task of generated driving videos. It posed minimal to no risk to participants, did not involve deception, and collected no Personally Identifiable Information (PII) or demographic data. Participants were fully informed of the task's nature prior to accepting it

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.