HORIZONFORGEV2: AGENTIC SCENE EDITING FOR LONG-TAIL DRIVING VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-tail scenarios, such as unusual road users and unexpected obstacles, remain particularly challenging for autonomous-driving systems and are important for both training and evaluation. They are also extremely rare in real-world data: of over 6.39 million miles of Waymo driving logs, only 0.03% qualify as long-tail scenarios (Xu et al., 2026). *Collecting long-tail scenarios directly is prohibitively expensive*, and some, such as an elephant standing in the middle of the road, cannot be staged safely at all. Driving simulation can instead generate these scenarios on demand. Recent methods combine the precise control of 3D scene representations with the photorealism of video generation. These methods *break down*, however, when the *inserted 3D asset falls outside the video generator's training distribution*: the video generator has rarely seen such objects during training and tends to distort or even remove them. *Image-editing* foundation models, in contrast, are trained on web-scale data and can insert diverse long-tail objects that fit the surrounding scene. Building on this observation, we present **HorizonForgeV2**, which *moves long-tail appearance generation from the video domain to the image domain*: image-editing models synthesize realistic keyframes, and a keyframe-conditioned video diffusion model (VDM) turns them into temporally coherent videos. Importantly, HorizonForgeV2 is trained only on existing driving videos, without collecting any new keyframe data. Edited keyframes can still be inconsistent with the 3D scene, and these errors propagate to the generated video. We therefore employ an agentic framework that inspects the intermediate keyframes and videos, locates failed edits, and iteratively refines generated videos with tool calling. Compared with HorizonForge (Wang et al., 2026a), HorizonForgeV2 raises the Operation Success Rate (OSR), a six-criterion score on a 1–10 scale, from 5.77 to 7.52 on our long-tail benchmark and from 6.47 to 7.75 on common driving-scene edits.

1 INTRODUCTION

Autonomous driving has made substantial progress, but safe operation requires systems that can handle diverse real-world scenarios. Training such systems needs data that cover both routine traffic and *long-tail scenarios*, which are difficult, costly, and sometimes unsafe to collect at scale. A practical alternative is to edit reconstructed 3D scenes of real drives, which keeps the real background and lets a rare object be placed and animated as requested. Methods such as HorizonForge (Wang et al., 2026a) learn to translate coarse renderings of edited 3D scenes into realistic driving videos. A coarse rendering, however, conveys the requested object's placement and motion far better than its appearance (Li et al., 2026b), so the video generator must infer the appearance from priors learned on driving videos. For objects rarely seen in such videos, such as animals, work vehicles, and construction equipment, the video generator can change their identity or remove them altogether (Fig. 1a). Training the video generator on more long-tail videos would help, but it requires the same scarce data that motivates scene editing in the first place (i.e., a "chicken-and-egg problem"). This leads to our question:

*How can we **automatically** generate **realistic and consistent** long-tail driving videos from edited 3D scenes without deliberately collecting or selecting long-tail training data?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

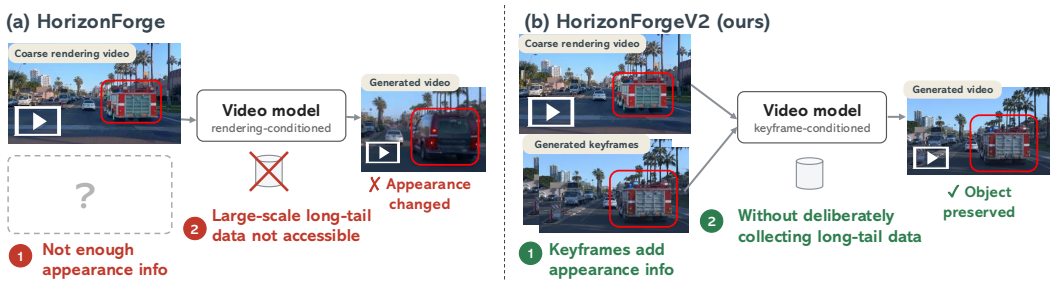


Figure 1: Keyframes supply the appearance that a coarse rendering lacks. (a) HorizonForge generates the video from the coarse rendering alone, which carries too little appearance information for a rare object, and the object is changed in the output. Adapting its video generator would require long-tail driving videos, which are scarce. (b) Edited keyframes add the missing appearance, and the keyframe-conditioned model learns to follow them from ordinary driving videos, without deliberately collecting long-tail data.

To answer this question, we **decouple appearance generation from video generation**: appearance of objects required by long-tail scenarios is first created in a *few keyframes* via *image editing*, and the video generator then extends it over time. This is because the generalizable visual prior in image editors is a natural source for these keyframes, and image editing can make the objects look realistic in individual coarse frames (Fig. 1b). The video generator then only needs to follow these references and connect them over time, instead of inventing the object's appearance from the coarse rendering alone.

Turning edited keyframes into a consistent driving video is still difficult. Rapid ego-motion changes the viewpoint substantially between frames, which makes it hard to preserve edited content across the video. Even strong image editors can produce keyframes that disagree with one another or with the coarse scene, and they may drop or alter small but safety-critical objects such as distant pedestrians and traffic signs. To handle such imperfect keyframes, we train a *keyframe-conditioned video diffusion model* (VDM) that turns them into coherent videos. During training, we sample keyframes from real driving videos and augment them to mimic the inconsistencies introduced by image editing, so that the VDM learns to follow the keyframes while remaining robust to their errors and noise. We also condition the VDM on the coarse rendering, which supplies the scene layout and the motion between keyframes, and on an optional depth input.

Despite our VDM's robustness, large keyframe deviations can still degrade video quality or cause the edit to fail. For example, if the image editor removes a pedestrian from a keyframe, the VDM may follow this keyframe and remove the pedestrian from nearby frames as well, so the pedestrian vanishes abruptly and the video becomes physically implausible and temporally inconsistent. Correcting such failures requires revising the keyframes or changing how the VDM uses them. We therefore introduce an **agentic framework** that coordinates the image editor and the VDM. The agent examines the intermediate keyframes and the generated video, forms a hypothesis about the cause of a failure, and takes a targeted action: it can revise the keyframe-editing prompt to avoid failure patterns observed in earlier rounds, or adjust the VDM's hyperparameters and the conditions it receives.

Our contributions are summarized as follows:

- We propose **HorizonForgeV2**, which generates long-tail driving videos from edited 3D scenes without deliberately collecting long-tail training data. It decouples appearance generation from video generation: an image editor creates the rare object's appearance in a few keyframes, and a keyframe-conditioned VDM, trained on ordinary driving videos to tolerate imperfect keyframes, propagates it through the video (Sec. 3.2).
- We introduce an agent that inspects intermediate keyframes and videos, attributes each failure to the keyframes or to video generation, and revises the responsible stage (Sec. 3.3).
- On our long-tail benchmark, HorizonForgeV2 reaches an OSR of 7.52, compared with 6.94 for the strongest baseline and 5.77 for HorizonForge, and it also leads on common driving-scene edits (7.75 vs. 6.49).

2 Related Work

Driving Scene Generation and Editing. Driving-scene diffusion models synthesized multi-view videos from maps, 3D boxes, and BEV layouts (Gao et al., 2024; Wen et al., 2024; Gao et al., 2025b; Wang et al., 2026b; Meng et al., 2026), and world models such as Cosmos (Agarwal et al., 2025; Alhaija et al., 2025; Ali et al., 2025) added spatial controls for driving simulation. Geometry-grounded methods extended recorded drives to novel viewpoints or editable scenes (Yan et al., 2025; Zhao et al., 2025; Yang et al., 2025; Zhu et al., 2025; Liang et al., 2025; Li et al., 2026b). Closest to our setting, HorizonForge (Wang et al., 2026a) and Dream4Drive (Zeng et al., 2026a) rendered explicit scene edits and translated the renderings into realistic videos, and SPVC (Li et al., 2026a) repaired artifacts in such renderings. In these pipelines, the video generator supplies the appearance of the edited object, which fails for objects rarely seen in driving data. We instead obtain this appearance from an image editor and use the video generator only to propagate it.

Frame-Guided Video Editing. Frame-guided editors propagated edited frames through keyframe translation, diffusion features, optical flow, or image-to-video generation (Yang et al., 2023; Geyer et al., 2024; Liang et al., 2024; Ouyang et al., 2024; Ku et al., 2024; Huang et al., 2025; Chen et al., 2026b), and reference-guided methods inserted objects into videos (Tu et al., 2025; Qi et al., 2026; Jin et al., 2026; Cao et al., 2026). Closest to ours, LoRA-Edit (Gao et al., 2025a), NOVA (Pan et al., 2026), and Frame Guidance (Jang et al., 2026) guided video generation with edited frames or sparse keyframes, and Liu et al. (2026) selected which keyframe to edit. These methods improved propagation from the supplied frames but did not revise the frames themselves, which are often incomplete or inconsistent for a rare object under fast ego-motion (Fig. 5). We train the video generator to tolerate such keyframes and let the agent revise them when they fail.

Agentic Visual Generation. Agents for image generation and editing orchestrated tools, verified intermediate results, and applied localized corrections (Wang et al., 2024; Shen et al., 2026; Zeng et al., 2026b; Zhang et al., 2026), and VISTA (Long et al., 2026) iteratively refined prompts for video generation. In driving, ChatSim (Wei et al., 2024) used LLM agents to edit reconstructed 3D scenes, and LangDriveCTRL (He et al., 2025) tuned diffusion strength and guidance after reviewing generated videos. The image agents worked on single images, and the video and driving agents revised prompts or global generation parameters. Our agent inspects the intermediate keyframes as well as the final video, attributes a failure to one stage, and revises that stage.

3 Method

3.1 Overview

As shown in Fig. 2, given a natural-language edit request r and the original driving video O, our goal is to generate a realistic and consistent video V that realizes the requested edit while preserving the surrounding scene. We first obtain a coarse rendering G of the edited scene, as described below. Unlike HorizonForge (Wang et al., 2026a), which refines G in a single diffusion pass, we generate V with an agentic framework built around a keyframe-conditioned VDM. The VDM lets image editors supply the appearance of rare objects through a few edited keyframes and propagates this appearance across the video while being robust to imperfect keyframes (Sec. 3.2). The agent inspects the intermediate keyframes and videos, diagnoses failures, and revises the keyframes or the VDM's settings (Sec. 3.3).

Preliminary: Coarse Rendering. We obtain G through explicit 3D scene editing, following LangDriveCTRL (He et al., 2025) and HorizonForge (Wang et al., 2026a). We reconstruct O with OmniRe (Chen et al., 2025b) into a scene graph with a static Gaussian background and dynamic object nodes. LangDriveCTRL's agents ground the objects referenced in r and apply the requested node operations, including removal, asset insertion, and trajectory generation with LangTraj (Chang et al., 2025). The edited scene graph is then rendered into G together with object masks. G specifies the scene layout and the requested object's placement and motion, but its appearance remains crude, especially for inserted assets.

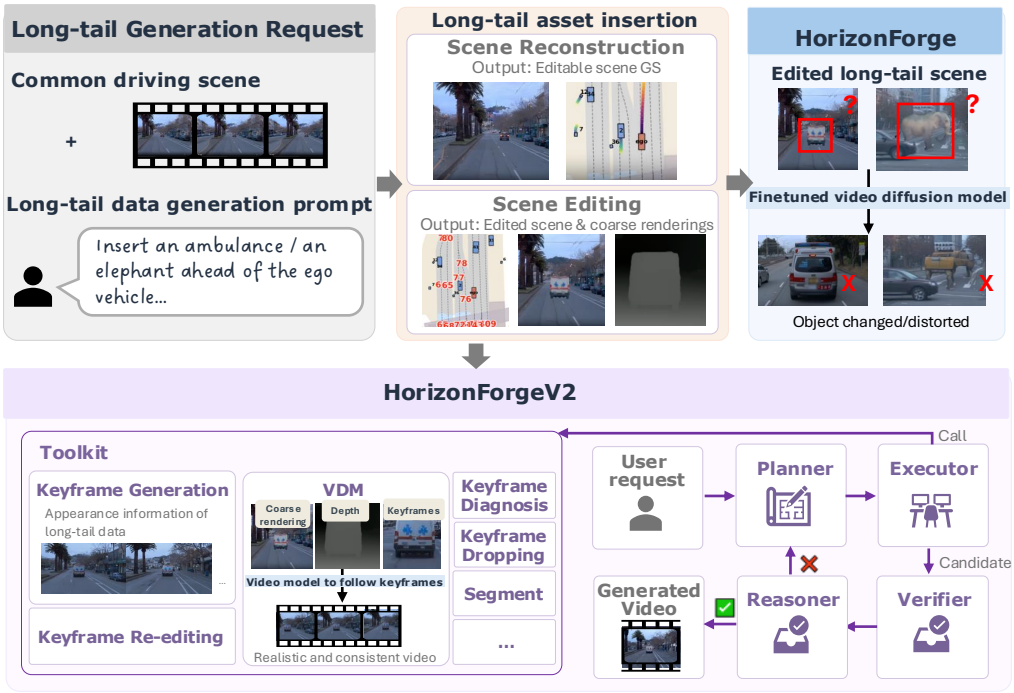


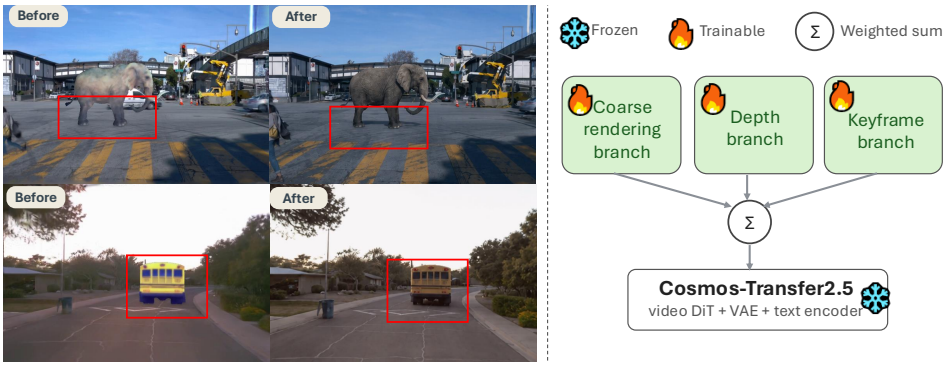
Figure 2: Overview of HorizonForgeV2. The edited 3D scene is rendered as a coarse video that fixes the object's placement and motion but not its appearance. Edited keyframes supply the appearance, and a keyframe-conditioned VDM carries it through the video (Sec. 3.2). The agent verifies each candidate and revises the keyframes or the generation settings (Sec. 3.3).

3.2 Keyframe-Conditioned Video Diffusion Model

A video generator that renders a long-tail object from a coarse rendering alone largely relies on having seen similar objects in driving videos. This dependence does not scale, because the long tail is open-ended: each new category, such as an elephant or an excavator, would need its own driving footage covering diverse viewpoints, motions, and lighting conditions, and such footage cannot be collected at reasonable cost. Images, in contrast, are abundant for almost any object, and *image-editing foundation models* trained on them can already insert such objects into driving frames with plausible appearance, scale, and lighting (Fig. 3a). We therefore let an image editor supply the object's appearance in a few keyframes and train the video generator only to follow these references over time. Following references is a general skill that can be learned from ordinary driving videos, without the need to collect new videos.

Keyframe branch. We build on Cosmos-Transfer2.5-2B (Ali et al., 2025) and attach a ControlNet-style branch that conditions the model on sparse keyframes. The backbone stays frozen to retain its video prior, and only the branch is trained. Each keyframe is encoded as a single image by the model's own autoencoder and placed at its time position in the latent sequence, with all other positions left empty. Therefore, the *number and placement of keyframes are flexible*, since they only determine which positions are filled, and the same model accepts anywhere from one to a dozen references. Because the autoencoder merges every four frames into one latent step, the latent position alone cannot tell which of the four frames a keyframe comes from. We resolve this by giving each latent step four keyframe slots, one per frame, so that each keyframe is placed at its exact frame.

Training with imperfect keyframes. We train the model on existing driving videos, using frames sampled from each clip as keyframes, and the clip itself as the target. Trained this way alone, the model would learn to copy the keyframes exactly, since they always match the target. At inference time, however, keyframes come from an image editor and may differ in color, shift slightly in position, be inconsistent with one another, or even alter the object itself. To close this gap, during training we perturb the keyframes while keeping the target video unchanged. We apply color and position jitter, either shared across all keyframes or drawn independently for each, and randomly



(a) (b)

Figure 3: Keyframe generation and VDM architecture. (a) Coarse-rendering keyframes from our long-tail benchmark are edited to provide realistic object appearance. (b) Our VDM uses trainable coarse-rendering, depth, and keyframe branches with a frozen video backbone.

replace one keyframe with real edits from a separate image editor (Wu et al., 2025a). We also vary the number of keyframes during training. The model thus learns to follow the keyframes where they are reliable and to tolerate their errors elsewhere. Table 7 lists the exact settings.

Auxiliary branches. Keyframes fix the appearance at a few moments but carry no information about what happens between them. Because the image editor may make mistakes, they can also contain wrong object states, such as a pedestrian's articulation or a vehicle's wheel angle that changes implausibly from one keyframe to the next. We therefore attach two more ControlNet-style branches, one for the coarse rendering and one for depth, to supply the missing temporal and structural information (Fig. 3b). As with the keyframe branch, only these branches are trained. The *coarse rendering* specifies the positions of the ego vehicle and every object at every frame, so motion between keyframes follows the edited 3D scene instead of being interpolated. The *depth* indicates where an object touches the road and how it is ordered relative to other traffic, which keeps inserted objects grounded. Both conditions are derived automatically from the training clips.

For the coarse-rendering training data, we follow HorizonForge (Wang et al., 2026a) to harvest noisy Gaussian splats and vehicle meshes from each scene and composite them into a rendering that mimics the artifacts of edited scenes at inference. Depth is estimated from the real videos with Video Depth Anything (Chen et al., 2025a). Since any of the three conditions can be unreliable for a given request, we drop each branch independently with probability 0.3 during training while keeping at least one branch active. The model thus learns to generate from any subset of the conditions and can rely on the remaining ones when one is missing or wrong. This also allows the agent to down-weight or withhold an unreliable condition without pushing the model out of distribution (Sec. 3.3).

3.3 Agentic Framework

Even with the keyframe-conditioned VDM, a single generation pass can still fail, and the failures differ from request to request. An inserted tow truck may float above the road, a pedestrian may vanish because one keyframe dropped it, or a rare object may fade into the coarse asset. These failures arise at different stages and need different fixes, and diagnosing them requires visual inspection of intermediate results that fixed rules cannot provide. We therefore employ an agent that compares the keyframes with the generated video, identifies the failing stage, and revises it.

Revision loop. The agent starts from an initial candidate video that the toolkit produces with default settings from k evenly spaced, jointly edited keyframes ($k = 6$ by default). Each subsequent round has four steps. The *planner* forms a hypothesis about which stage failed and selects actions. The *executor* applies these actions with the tools below and renders a new candidate. The *verifier* scores the candidate and describes its visible failures. The *reasoner* judges whether the revision helped and what to examine next, which informs the next planner. All steps can access the history of the current request, including previous candidates, their diagnoses, keyframe versions, and actions. The loop runs for a fixed number of rounds set by the render budget and returns the highest-scoring candidate. We implement the agent with the Codex CLI harness; the agent runs in a sandbox without network access and acts only through tools served over MCP.

Table 1: Tools available to the agent.

Type	Tool	Function
Inspection	`status`	Report remaining budget, keyframe versions, and all candidates with their scores
	`diagnose_keyframes`	Score each keyframe's alignment with G and flag inconsistent frames
	`segment`	Measure object presence, size, and count with SAM 3
	`compare_videos`	Compare two candidates side by side with a VLM
Keyframe	`gen_keyframes`	Jointly re-edit all keyframes with a revised instruction
	`edit_keyframe`	Re-edit a single keyframe, keeping the others
	`drop_keyframes`	Remove outlier keyframes
Generation	`run_vdm`	Render a new candidate with chosen keyframes and VDM settings
	`verify_video`	Re-score an existing candidate

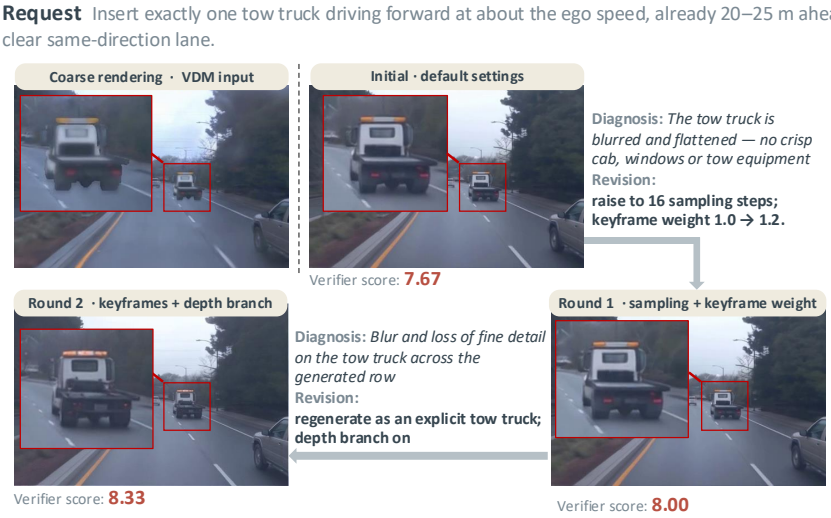


Figure 4: An example of agent reasoning and revision for long-tail video generation. The agent analyzes the cause of each failure and revises the keyframes or the video-generation settings across attempts to improve the final result.

Tools. The agent acts through three types of tools (Table 1). *Inspection tools* gather evidence without changing any result. They report the current state of the request, check the keyframes for misalignment with G and for inconsistency with one another, and segment objects with SAM3 (Carion et al., 2026), so that questions such as whether an object is present or has shrunk are answered by measurement instead of free-form description. *Keyframe tools* revise the appearance references by regenerating all keyframes with a revised instruction, re-editing a single keyframe, or dropping outlier keyframes. Each revision creates a new keyframe version, and earlier versions remain available. *Generation tools* render a new candidate with the VDM from a chosen keyframe version, with adjustable sampling steps, guidance, seed, branch weights, and depth on or off. Each render counts toward the budget and is scored automatically by the verifier.

Example. Fig. 4 shows how the agent handles a request that inserts a tow truck 20–25 m ahead of the ego vehicle. The initial candidate places the truck in the correct lane (score 7.67), but the truck is blurred and flattened, without a crisp cab or windows. The agent attributes this to generation rather than to the keyframes, so it keeps the keyframes, raises the sampling steps to 16, and increases the keyframe weight from 1.0 to 1.2. The score rises to 8.00, but the truck still lacks fine detail throughout the video, which now points to the keyframes. The agent therefore regenerates the keyframes with an explicit tow-truck instruction and enables the depth branch. This candidate scores 8.33 and is returned as the final output. The two rounds fix different stages, first generation and then the appearance references, and each round improves the result because the agent first diagnoses which stage failed.



Figure 5: Qualitative comparison of long-tail video generation under the same edit instructions. Our method better preserves the requested objects and their appearance across frames.

4 Experiments

4.1 Experimental Setup

Benchmarks. Our long-tail benchmark contains 120 cases on 59 scenes from the Waymo Open Dataset (Sun et al., 2020) in four categories: 60 cases involve rare work and service vehicles, such as fire trucks, ambulances, school buses, tow trucks, dump trucks, cement mixers, and street sweepers; 30 involve construction zones with cones and barriers; 10 involve animals interacting with traffic; and 20 involve pedestrians. Pedestrians are common on the road but hard to edit, because they are articulated and occupy small image regions, and HorizonSuite (Wang et al., 2026a) does not cover them despite their importance for traffic safety. We also evaluate 109 common driving-scene edits from HorizonSuite, comprising 30 insertions, 9 removals, and 70 trajectory edits on 13 Waymo scenes.

Implementation. We train the keyframe-conditioned VDM on 2,492 clips randomly drawn from existing driving datasets: 1,883 from Alpamayo (Wang et al., 2025) and 609 from nuScenes (Caesar et al., 2020). We do not deliberately collect or select long-tail scenes for training. At inference, the agent uses the default $k = 6$ keyframes (Sec. 3.3) and a fixed budget of four video renders, including the initial candidate. Every request runs all four renders, and we report the highest-scoring candidate. We use `gpt-image-2` for keyframe editing and `gpt-5.6-luna` for the agent and verifier. Appendix A.3 gives further training and inference details.

Baselines. We compare against two groups of baselines. Video-refinement methods improve coarse scene renderings: HorizonForge (Wang et al., 2026a) and Difix3D (Wu et al., 2025b). Keyframe-guided methods use sparse visual references: NOVA (Pan et al., 2026), LoRA-Edit (Gao et al., 2025a), Partial Diffusion (Meng et al., 2022), and Frame Guidance (Jang et al., 2026). For the two inference-time methods, Partial Diffusion and Frame Guidance, we use HorizonForge as the base video generator. All baselines receive the same input video and prompt, and those that start from a coarse rendering receive the same coarse rendering video.

Table 2: Main results. Long-tail insertion is scored with the six insertion criteria; common driving-scene edits are taken from HorizonSuite (Wang et al., 2026a). All values are OSR on a 1–10 scale; higher is better. Real.: realism; Temp.: temporal consistency; Instr.: instruction alignment; Phys.: physical plausibility; Qual.: visual quality; Beh.: behavior alignment. For common edits we report the mean OSR, since removal uses its own six criteria; per-operation scores are given in Table 3 and Appendix B.

	Long-tail insertion							Common edits
Method	Real.	Temp.	Instr.	Phys.	Qual.	Beh.	Mean OSR	Mean OSR
NOVA (Pan et al., 2026)	5.08	6.16	7.21	5.96	4.79	5.78	5.83	5.98
LoRA-Edit (Gao et al., 2025a)	4.48	5.91	6.62	5.85	4.19	6.12	5.53	5.07
Partial Diffusion (Meng et al., 2022)	4.91	6.18	6.18	6.12	4.76	6.53	5.78	6.04
Frame Guidance (Jang et al., 2026)	4.88	6.11	6.68	6.03	4.59	6.21	5.75	5.68
Difix3D (Wu et al., 2025b)	5.91	7.15	8.03	7.02	5.58	7.95	6.94	6.49
HorizonForge (Wang et al., 2026a)	4.89	6.36	5.77	6.18	4.90	6.49	5.77	6.47
Ours	**6.69**	**7.61**	**8.55**	**7.66**	**6.28**	**8.33**	**7.52**	**7.75**

Table 3: Mean OSR by operation on the common-editing benchmark, scored with per-operation criteria.

Method	Insert.	Remov.	Traj.
HorizonForge	6.43	6.43	6.49
Ours (initial)	7.64	5.78	6.86
Ours	**8.30**	**7.30**	**7.57**

Table 4: Human validation on the long-tail benchmark (147 samples). Raters score instruction following, realism, and temporal consistency on a 5-point scale. Win rate is the fraction of samples in which a method receives the highest mean score over the three criteria, with ties split evenly.

Method	Instr. ↑	Real. ↑	Consist. ↑	Win Rate ↑
Difix3D	4.26	2.65	3.32	18.37%
HorizonForge	3.63	3.18	3.71	24.49%
Ours	**4.44**	**3.59**	**3.78**	**57.14%**

Evaluation metric. We report the six-criterion Operation Success Rate (OSR), a GPT-based rating of how well the generated video realizes the requested edit. For insertions and trajectory edits, the criteria are realism, temporal consistency, instruction alignment, physical plausibility, visual quality, and behavior alignment. Removals use six task-specific criteria: completeness, blending, consistency, realism, inpainting quality, and behavior alignment. The judge assigns an integer from 1 to 10 to each criterion, and we average the six values for each request and then over requests. For each candidate, the judge sees 5 frames sampled from the generated video and 5 frames from the coarse rendering at the same time indices. Appendix A.4 gives further details of the evaluation setting.

4.2 Main Results

Long-tail video generation. Table 2 shows that our method outperforms every baseline on every criterion. On the long-tail benchmark, it reaches a mean OSR of 7.52, a relative improvement of 8.36% over the strongest baseline, Difix3D (Wu et al., 2025b), and of 30.33% over HorizonForge (Wang et al., 2026a). The largest gains appear on the criteria that measure whether a rare object is rendered faithfully: relative to Difix3D, realism and visual quality improve by 13.20% and 12.54%, and relative to HorizonForge, instruction alignment and realism improve by 48.18% and 36.81%. The keyframe-guided baselines all stay at or below 5.83, and our method exceeds the best of them, NOVA (Pan et al., 2026), by 28.99%, as sparse references alone do not survive the viewpoint change caused by ego-motion. Fig. 5 shows the same trend qualitatively: our method keeps the requested object's identity and appearance as the viewpoint changes. Appendix B.3 breaks these results down by object category.

Common driving-scene edits. Our method also leads on common edits. On the 109 common edits from HorizonSuite, it reaches 7.75 OSR, a relative improvement of 19.41% over the strongest baseline, Difix3D (6.49), and of 19.78% over HorizonForge (6.47) on its own benchmark (Table 2). Table 3 shows that the improvement holds for every operation: relative to HorizonForge, OSR rises by 29.08% for insertion, 13.53% for removal, and 16.64% for trajectory edits. See Appendix B.4 for more details.

Table 5: VDM training ablation on the long-tail benchmark, without the agentic framework.

Setting	OSR ↑
HorizonForge	5.77
Full VDM	**6.58**
w/o augmentation	6.45
w/o aux. branches	6.00

Table 6: Long-tail OSR of the best candidate retained after each number of rendered candidates. Candidate 1 is the initial output before revision.

Candidates	1	2	3	4	6	8	10	12
HorizonForge	5.77 (single pass)							
Ours	6.58	7.11	7.31	7.52	7.75	7.86	7.93	**8.03**

Human study. We run a blind user study against the two strongest baselines, Difix3D (Wu et al., 2025b) and HorizonForge (Wang et al., 2026a), and collect 147 long-tail samples (details in Appendix A.4). Our method receives the highest score in instruction following, realism, and temporal consistency (Table 4), with the largest gain in realism, 12.89% over the best baseline. It is rated best in 57.14% of the samples, more often than both baselines combined (42.86%).

4.3 Ablations

VDM training. We ablate the two training choices of the keyframe-conditioned VDM (Sec. 3.2) without the agentic framework, scoring the initial candidate on the long-tail benchmark (Table 5). Removing the imperfect-keyframe augmentation lowers OSR from 6.58 to 6.45, consistent with the model copying editor errors into the video once it is no longer trained to tolerate them. Removing the auxiliary coarse-rendering and depth branches causes a larger drop, to 6.00, because the model loses both the motion between keyframes and the depth cues that ground inserted objects. Both components therefore contribute to performance.

Number of revision rounds. Table 6 varies the number of candidates the agent renders on the long-tail benchmark; as in all our experiments, every case runs the full number of rounds. A single candidate, the initial output before any revision, already surpasses HorizonForge by 14.04%. Each additional round then raises the retained score, from 6.58 to 7.52 at four candidates and to 8.03 at twelve, a gain of 22.04% over the initial candidate and 39.17% over HorizonForge. The gains diminish as the budget grows: the first three revisions contribute 0.94 of the 1.45-point improvement, and the next eight add the remaining 0.51. We use four candidates in the main experiments as a trade-off between quality and inference cost (Appendix B.5).

4.4 Downstream Evaluation

We further test whether the generated videos are useful as training data. Following the downstream protocol of HorizonForge (Wang et al., 2026a), we train a single-camera, car-only BEVFormer-tiny detector (Li et al., 2022) on 8,000 real Waymo training frames, optionally augmented with 9,600 frames from the generated long-tail videos, and evaluate it on 23,927 real Waymo validation frames. Each generated frame is labeled with the 3D boxes of its edited scene, which cover both the original cars and the inserted vehicles. Adding our videos raises AP (Average Precision) from 0.0579 to 0.0596, a 2.94% relative improvement over training on real data alone, while adding HorizonForge videos yields only 0.0587. The inserted long-tail vehicles, such as fire trucks, tow trucks, and school buses, broaden the range of vehicle appearances beyond the common cars in the real data.

5 Conclusion

We presented HorizonForgeV2, an agentic framework for generating realistic and consistent long-tail driving videos from edited 3D scenes. It decouples appearance generation from video generation: image-editing foundation models create the appearance of rare objects in a few keyframes, and a keyframe-conditioned video diffusion model, trained on ordinary driving videos to tolerate imperfect keyframes, propagates this appearance into coherent videos without deliberately collected long-tail training data. An agent further inspects the intermediate keyframes and generated videos, identifies the failing stage, and revises the keyframes or the video-generation settings. Experiments show that HorizonForgeV2 outperforms prior methods on both long-tail and common driving-scene edits, achieving the best results on both the OSR metric and the human study.

Ethics Statement

This work studies agentic scene editing for driving simulation. All experiments use publicly available driving datasets (Waymo Open Dataset, nuScenes, and Alpamayo) under their respective licenses, and no new driving data was collected. The human study involved only rating anonymized generated videos, and no personal information was collected. We therefore do not foresee specific ethical concerns.

Reproducibility Statement

The details of the method and algorithms are provided in Section 3, including the keyframe-conditioned video diffusion model (Section 3.2) and the agentic framework (Section 3.3). Datasets, benchmarks, baselines, and evaluation protocols are described in Section 4.1, with additional implementation details, training settings, and judge prompts in the Appendix. We will release code and trained checkpoints upon publication.

AI Use Statement

Large language and vision models are components of the proposed method: `gpt-image-2` edits keyframes, `gpt-5.6-luna` serves as the agent and verifier through the Codex CLI harness, Qwen-Image-Edit provides edited references during training, and Qwen3.6-27B serves as a second judge (Appendix B.1). LLMs were also used to polish the writing and to assist with code editing.

References

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

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.

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

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 *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11618–11628. IEEE, 2020.

Xiao Cao, Yansong Qu, Wen Xiao, Jiakui Hu, Heyuan Li, Jialun Liu, Zhiyong Huang, Xuelong Li, et al. Smart-insertion-v: Photorealistic video insertion via a closed-loop feedback dual-stream framework. *arXiv preprint arXiv:2605.23891*, 2026.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Wei-Jer Chang, Wei Zhan, Masayoshi Tomizuka, Manmohan Chandraker, and Francesco Pittaluga. Langtraj: Diffusion model and dataset for language-conditioned trajectory simulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26622–26631. IEEE, 2025.

Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22831–22840. IEEE, 2025a.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026a.

Xinyu Chen, Yuyi Qian, Jiang Lin, Shenyi Wang, Gao Wang, Zhiqiu Zhang, Jizhi Zhang, Mingjie Wang, Qiang Tang, Qian Wang, et al. Siminsert: Seamless video object insertion via regional sparse attention fusion. *arXiv preprint arXiv:2605.23245*, 2026b.

Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo De Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. In *International Conference on Learning Representations*, volume 2025, pp. 85508–85527, 2025b.

Chenjian Gao, Lihe Ding, Xin Cai, Zhanpeng Huang, Zibin Wang, and Tianfan Xue. Lora-edit: Controllable first-frame-guided video editing via mask-aware lora fine-tuning. *arXiv preprint arXiv:2506.10082*, 2025a.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *International Conference on Learning Representations*, volume 2024, pp. 22841–22860, 2024.

Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrive-v2: High-resolution long video generation for autonomous driving with adaptive control. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28135–28144. IEEE, 2025b.

Michal Geyer, Omer Bar Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *International Conference on Learning Representations*, volume 2024, pp. 1608–1620, 2024.

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.

Zhi-Lin Huang, Yixuan Liu, Chujun Qin, Zhongdao Wang, Dong Zhou, Dong Li, and Emad Barsoum. Edit as you see: Image-guided video editing via masked motion modeling. *arXiv preprint arXiv:2501.04325*, 2025.

Sangwon Jang, Taekyung Ki, Jaehyeong Jo, Jaehong Yoon, Soo Ye Kim, Zhe Lin, and Sung Ju Hwang. Frame guidance: Training-free guidance for frame-level control in video diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 60977–61003, 2026.

Hoiyeong Jin, Hyojin Jang, Junha Hyung, Jeongho Kim, Kinam Kim, Dongjin Kim, Huijin Choi, Hyeonji Kim, and Jaegul Choo. Insertanywhere: Geometrically grounded and optics-aware video object insertion. In *European Conference on Computer Vision*, pp. 38–54. Springer, 2026.

Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024.

Gen Li, Shu Han, Yun Xi Qiao, Hua Chen, Xuyang Dai, Bohan Li, Hao Zhao, and Chaojian Li. Spvc: Structured and panoptic video fixing for cross-dataset driving scene rendering. *arXiv preprint arXiv:2608.17420*, 2026a.

Jiusi Li, Jackson Jiang, Jinyu Miao, Miao Long, Tuopu Wen, Peijin Jia, Shengxiang Liu, Chunlei Yu, Maolin Liu, Yuzhan Cai, et al. Realistic and controllable 3d gaussian-guided object editing for driving video generation. In *2026 IEEE Intelligent Vehicles Symposium (IV)*, pp. 60–67. IEEE, 2026b.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision*, 2022.

Feng Liang, Bichen Wu, Jialiang Wang, Licheng Yu, Kunpeng Li, Yinan Zhao, Ishan Misra, Jia-Bin Huang, Peizhao Zhang, Peter Vajda, et al. Flowvid: Taming imperfect optical flows for consistent video-to-video synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8207–8216. IEEE, 2024.

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

Lin Liu, Zhihan Xiao, Haohang Xu, Rong Cong, Zhibo Zhang, Xiaopeng Zhang, and Qi Tian. Occlusion-aware physics-semantic keyframe selection for robust video editing. *arXiv preprint arXiv:2605.23192*, 2026.

Do Xuan Long, Xingchen Wan, Hootan Nakhost, Chen-Yu Lee, Tomas Pfister, and Sercan Ö Arik. Vista: A test-time self-improving video generation agent. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6021–6032, 2026.

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

Zijie Meng, Yufei Liu, Chengqian Ma, Zhiyu Li, Jiyuan Liu, Wenhua Nie, Bingcai Wei, Shuqin Chen, Weichen Xu, Jiquan Yuan, et al. Omnidrive: An llm-choreographed multi-agent world model with unified latent co-compression for multi-view driving video generation. *arXiv preprint arXiv:2606.17536*, 2026.

Wenqi Ouyang, Yi Dong, Lei Yang, Jianlou Si, and Xingang Pan. I2vedit: First-frame-guided video editing via image-to-video diffusion models. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Tianlin Pan, Jiayi Dai, Chenpu Yuan, Zhengyao Lv, Binxin Yang, Hubery Yin, Chen Li, Jing Lyu, Caifeng Shan, and Chenyang Si. Nova: Sparse control, dense synthesis for pair-free video editing. *arXiv preprint arXiv:2603.02802*, 2026.

Xia Qi, Peishan Cong, Yichen Yao, Ziyi Wang, Yaoqin Ye, and Yuexin Ma. Controllable video object insertion via multiview priors. *arXiv preprint arXiv:2604.14556*, 2026.

Lingteng Qiu, Xiaodong Gu, Peihao Li, Qi Zuo, Weichao Shen, Junfei Zhang, Kejie Qiu, Weihao Yuan, Guanying Chen, Zilong Dong, et al. Lhm: Large animatable human reconstruction model for single image to 3d in seconds. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14184–14194. IEEE, 2025.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026. URL `https://qwen.ai/blog?id=qwen3.6-27b`.

Shaocheng Shen, Jianfeng Liang, Chunlei Cai, Cong Geng, Huiyu Duan, Xiaoyun Zhang, Qiang Hu, and Guangtao Zhai. Agentic retoucher for text-to-image generation. *arXiv preprint arXiv:2601.02046*, 2026.

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

Yuanpeng Tu, Hao Luo, Xi Chen, Sihui Ji, Xiang Bai, and Hengshuang Zhao. Videoanydoor: High-fidelity video object insertion with precise motion control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-R1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

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

Zhechao Wang, Yiming Zeng, Lufan Ma, Zeqing Fu, Chen Bai, Dongshuo Yin, Ziyao Lin, and Cheng Lu. Drivepts: A progressive learning framework with textual and structural enhancement for driving scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3699–3708, 2026b.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37: 128374–128395, 2024.

Yuxi Wei, Zi Wang, Yifan Lu, Chenxin Xu, Changxing Liu, Hao Zhao, Siheng Chen, and Yanfeng Wang. Editable scene simulation for autonomous driving via collaborative llm-agents. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15077–15087. IEEE, 2024.

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*, pp. 6902–6912, 2024.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

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

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Kate Tolstaya, Sarah Tang, Brandyn White, et al. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3709–3718, 2026.

Yunzhi Yan, Zhen Xu, Haotong Lin, Haian Jin, Haoyu Guo, Yida Wang, Kun Zhan, Xianpeng Lang, Hujun Bao, Xiaowei Zhou, et al. Streetcrafter: Street view synthesis with controllable video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 822–832. IEEE, 2025.

Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–11, 2023.

Zeyu Yang, Zijie Pan, Yuankun Yang, Xiatian Zhu, and Li Zhang. Driving view synthesis on free-form trajectories with generative prior. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28083–28092. IEEE, 2025.

Kai Zeng, Zhanqian Wu, Kaixin Xiong, Xiaobao Wei, Xiangyu Guo, Zhenxin Zhu, Kalok Ho, Lijun Zhou, Bohan Zeng, Ming Lu, et al. Rethinking driving world model as synthetic data generator for perception tasks. In *International Conference on Learning Representations*, volume 2026, pp. 135507–135534, 2026a.

Ziyun Zeng, Hang Hua, and Jiebo Luo. Mira: Multimodal iterative reasoning agent for image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9563–9573, 2026b.

Yuxuan Zhang, Shijia Huang, and Liwei Wang. Intentedit: Multi-agent reasoning for intent-driven complex image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 8776–8785, June 2026.

Guosheng Zhao, Chaojun Ni, Xiaofeng Wang, Zheng Zhu, Xueyang Zhang, Yida Wang, Guan Huang, Xinze Chen, Boyuan Wang, Youyi Zhang, et al. Drivedreamer4d: World models are effective data machines for 4d driving scene representation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12015–12026. IEEE, 2025.

Zehao Zhu, Yuliang Zou, Chiyu Max Jiang, Bo Sun, Vincent Casser, Xiukun Huang, Jiahao Wang, Zhenpei Yang, Ruiqi Gao, Leonidas Guibas, et al. Scenecrafter: Controllable multi-view driving scene editing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6812–6822. IEEE, 2025.

A Implementation Details

A.1 Benchmark Construction

Each long-tail case specifies one object to insert, its placement, and its motion. The 120 cases cover 59 Waymo training scenes and four categories: 60 vehicle cases (fire truck, school bus, dump truck, tow truck, ambulance, cement mixer, street sweeper), 30 construction cases (roadwork props such as cones and barriers), 20 pedestrian cases placed on sidewalks, and 10 animal cases. All clips have 80 front-camera frames at 10 fps and 960×640 resolution. Rigid assets are prepared with SAM 3D Objects (Chen et al., 2026a), and inserted pedestrians with LHM (Qiu et al., 2025). The 109 common driving-scene edits reuse the requests, scenes, and coarse renderings released with HorizonForge (Wang et al., 2026a), so we do not regenerate their conditions.

A.2 Keyframe Editing and Refinement

Joint panel editing. We sample six frames from G at uniform time indices and arrange them into a single image panel on a two-column serpentine grid. A single `gpt-image-2` edit call processes the whole panel, so the editor sees every view at once and the object keeps one identity across time. The instruction depends on the object category and asks for a realistic appearance of the requested object while preserving its class, pose, and location together with the surrounding scene. The construction and pedestrian categories use repair-only instructions, which forbid adding or moving any object. The edited panel is then split back into six frames. The initial candidate uses the low image-editing quality setting, and the agent may raise it to medium during revision.

Color adjustment. Image editing can also shift the global exposure and color of a keyframe. For each edited keyframe I_k and color channel c, we estimate a gain $a_{k,c}$ and a bias $b_{k,c}$ by matching its background pixels, excluding the inserted-object mask, to the original video frame at the same time, and apply

$$J_{k,c}(\mathbf{p}) = a_{k,c}\, I_{k,c}(\mathbf{p}) + b_{k,c}, \tag{1}$$

where p denotes a pixel location. After video generation, we estimate a single affine correction of the same form over eight sampled output frames and apply it to the whole video.

A.3 Video Model Training

Architecture. The backbone is Cosmos-Transfer2.5-2B (Ali et al., 2025): a 28-block DiT of width 2048 with 16 attention heads, an autoencoder with 16 latent channels and temporal and spatial compression factors of four and eight, and a frozen text encoder. Each of the three control branches has four trainable blocks injected at backbone layers 0, 7, 14, and 21. The backbone, autoencoder, and text encoder stay frozen, and only the branches and their input embedders are trained.

Data and schedule. We train the video diffusion model (VDM) on 93-frame windows, which correspond to 24 latent time steps. Optimization uses AdamW with a base learning rate of 10^{-4}, $\beta = (0.9, 0.999)$, weight decay 10^{-3}, and linear warmup over 100 steps. The loss is the rectified-flow velocity prediction objective of the base model. The final checkpoint is trained in two stages: a 480p model trained for 10k iterations, and a 720p model initialized from it and trained for another 10k iterations on four H100 GPUs.

Condition perturbation. Table 7 gives the exact settings of the perturbations described in Sec. 3.2. We replace at most one keyframe per training sample with an edited reference. The edited keyframe then disagrees with the other keyframes and with the coarse rendering, while the target video agrees with them. The model thus learns to treat inconsistent content that appears in only one keyframe as an editing error, and to follow content that is consistent across other keyframes.

A.4 Evaluation Protocol

Automatic rating. OSR uses `gpt-5.6-luna` as a fixed judge with one request per video (`reasoning_effort=low`, at most 400 completion tokens). The judge receives five frames sampled at uniform time indices from the generated video, five frames of the coarse rendering at the same indices, and a text prompt containing the edit request. When a mask is available, the edited object is marked with a green box for insertion and trajectory edits, and the original location of a

Table 7: Exact settings of the training-time condition perturbations.

Condition	Setting
Keyframe appearance	Brightness, contrast, and saturation jitter of magnitude 0.25, hue jitter 0.09, and translation of up to 10% of the shorter image side with replicated borders. Parameters are shared across a sample's keyframes with probability 0.5 and drawn independently otherwise.
Edited references	Replace keyframes with offline Qwen-Image-Edit (Wu et al., 2025a) addition or removal edits with probability 0.7 when an edit is available for the sampled window. At most one keyframe per sample is edited. The target video is unchanged.
Keyframe count	Sample between 1 and 12 distinct keyframe positions among the 24 latent time steps. Validation uses eight. Inference uses six.
Partial depth	Use full depth with probability 0.3. Otherwise retain depth only inside the union of two to twenty tracked instance regions and invalidate the rest.
Missing conditions	Drop each of the three branches independently with probability 0.3, restoring one uniformly chosen branch if all three are dropped.

removed object with a red box. The judge returns an integer from 1 to 10 for each of six criteria, and the OSR of a request is their mean. Insertion and trajectory requests use realism, temporal consistency, instruction alignment, physical plausibility, visual quality, and behavior alignment, while removal requests use completeness, blending, consistency, realism, inpainting quality, and behavior alignment. Because the rubrics differ across operations, we average only the per-request OSR across operations and never the individual criteria. An empty or unparsable response is retried once with a larger token budget, and no retry is triggered by the value of a score. The prompt template is given below, and Table 8 lists the criteria verbatim. To confirm that the results do not depend on a single evaluator, we also report a second judge from a different model family (Appendix B.1) and a human study below.

> You are evaluating a generated driving video for `{operation}` quality.
> `{task line}`
> `{box line, included only when a mask is available}`
> Images 1-5 are frames of the GENERATED video. Images 6-10 are frames of the COARSE REFERENCE video (the Gaussian-splat scene composite that the generation was conditioned on) at the same times; it defines where the edited subject should be and how it and the rest of the scene move.
> Please rate the generated video on a scale of 1-10 on each of: `{six criteria}` (Table 8)
> Provide ONLY six numbers from 1 to 10 separated by spaces, in this order, as your response.

The task line is "The goal was to insert: `{text}`" for insertion, "The goal was to remove: `{text}`" for removal, and "The goal was: `{subject}` is `{behavior}`" for trajectory edits, where `{subject}` is "the marked vehicle" ("the other vehicle" when no box is drawn) or "the ego vehicle (the camera car)". The box line reads, for example, "The inserted object is marked with a GREEN bounding box in the generated frames."

Human study. We collect 147 samples in total. For each sample, a participant sees the original video and the edit instruction, followed by the results of the three methods in a random order that changes across samples. Participants are not told which method produced which video and are not shown any automatic score. The three criteria are defined as follows: *instruction following*, whether the requested edit is realized with the correct object, placement, and motion; *realism*, whether the edited object and the scene look like real driving footage; and *temporal consistency*, whether the object keeps its identity and appearance across frames without flickering or vanishing. Beyond realism, our method improves instruction following by 4.23% over Difix3D and is on par with HorizonForge in temporal consistency (3.78 vs. 3.71). Averaged over the three criteria, our method scores 3.94, against 3.51 for HorizonForge and 3.41 for Difix3D.

Table 8: Criteria in the OSR judge prompt, quoted verbatim. The abbreviations in Table 2 correspond to criteria 1–6 of insertion: realism (Real.), temporal consistency (Temp.), instruction alignment (Instr.), physical plausibility (Phys.), visual quality (Qual.), and behavior alignment (Beh.).

	Insertion	Removal	Trajectory
1	*Realism*: Does the inserted object look realistic and well-integrated?	*Completeness*: Is the object fully removed from the marked area?	*Realism*: Does the edited driving behavior look realistic and well-integrated with the scene?
2	*Consistency*: Is the object consistent across frames?	*Blending*: Is the filled region seamless with its surroundings?	*Consistency*: Is the scene consistent across frames?
3	*Alignment*: Does it match the text description?	*Consistency*: Is the removal consistent across frames?	*Alignment*: Does the motion match the described behavior?
4	*Physical plausibility*: Does it follow realistic motion and positioning?	*Realism*: Does the scene look natural, without artifacts?	*Physical plausibility*: Does the vehicle follow realistic motion and positioning?
5	*Visual quality*: Is the video free of visible artifacts or inconsistencies?	*Inpainting quality*: How well is the removed region filled in?	*Visual quality*: Is the video free of visible artifacts or inconsistencies?
6	*Behavior alignment*: Do the object's position, motion state and trajectory, and the motion of the scene, match the coarse reference video?	*Behavior alignment*: Do the remaining actors and the camera motion match the coarse reference video?	*Behavior alignment*: Do the edited vehicle's path, speed and heading, and the motion of the rest of the scene, match the coarse reference video?

Table 9: Scores from a second judge on the 120 long-tail cases, on a 1–10 scale. Higher is better. Real.: realism; Bg.: background consistency; Instr.: editing instruction following.

Method	Real.	Bg.	Instr.	Mean
NOVA (Pan et al., 2026)	6.68	9.04	6.63	7.45
LoRA-Edit (Gao et al., 2025a)	5.35	8.78	6.05	6.73
Partial Diffusion (Meng et al., 2022)	6.31	8.95	5.88	7.04
Frame Guidance (Jang et al., 2026)	6.36	8.90	6.41	7.22
Difix3D (Wu et al., 2025b)	6.87	**9.12**	6.92	7.63
HorizonForge (Wang et al., 2026a)	6.62	9.02	5.66	7.10
Ours	**8.01**	9.05	**7.78**	**8.28**

B Additional Results

B.1 Scores from a Second Judge

Automatic ratings from a single VLM judge can carry that model's own preferences. To check that our improvements hold under an independent evaluator, we re-score all 120 long-tail videos with a judge from a different model family, Qwen3.6-27B (Qwen Team, 2026), using its own three-criterion rubric: realism, background consistency, and editing instruction following. This judge shares neither the model nor the prompt with OSR. Table 9 shows that the two judges agree on the leading methods: our method ranks first with a mean of 8.28, followed by Difix3D at 7.63, a relative improvement of 8.52%, and it improves on HorizonForge by 16.62%. The gains again concentrate on the criteria that measure whether the rare object is rendered as requested: realism rises from 6.87 to 8.01 (+16.59%) and instruction following from 6.92 to 7.78 (+12.43%) over the best baseline, and instruction following improves by 37.46% over HorizonForge. All methods preserve the background well, scoring between 8.78 and 9.12, and our method stays within 0.07 of the best, so inserting the rare object does not come at the cost of the surrounding scene. These results indicate that the improvements in Table 2 are consistent across judges from different model families and are not specific to the OSR judge.

Table 10: Mean OSR by inserted-object category on the long-tail benchmark.

Method	Work vehicle	Construction	Pedestrian	Animal
NOVA (Pan et al., 2026)	6.15	5.87	4.48	6.45
LoRA-Edit (Gao et al., 2025a)	6.14	5.28	3.49	6.68
Partial Diffusion (Meng et al., 2022)	6.20	6.12	4.16	5.43
Frame Guidance (Jang et al., 2026)	6.36	5.58	3.60	6.88
Difix3D (Wu et al., 2025b)	7.43	7.03	5.26	7.08
HorizonForge (Wang et al., 2026a)	6.41	5.72	3.90	5.75
Ours	**8.10**	**7.55**	**5.82**	**7.32**

Table 11: Mean OSR by request type on the common-editing benchmark. Removal is scored with its own six criteria.

Method	Ego dir.	Ego lane	Ego speed	Other dir.	Insertion	Removal
HorizonForge (Wang et al., 2026a)	6.69	6.97	6.85	5.90	6.43	6.43
Difix3D (Wu et al., 2025b)	6.62	7.50	7.40	6.07	6.16	5.93
Ours	**7.54**	**7.65**	**8.00**	**7.38**	**8.30**	**7.30**

B.2 Qualitative Results

We provide a website in supplementary materials that shows more cases for both the long-tail and common driving-scene benchmarks, each with the coarse rendering.

B.3 Results by Object Category

The long-tail benchmark mixes categories of very different difficulty: work vehicles resemble the cars a driving-video model has seen, whereas pedestrians are small and articulated and animals are nearly absent from driving footage. We therefore report results per category (Table 10) to check that the gains are not driven by the easier cases. Our method is the best in all four categories, and its advantage over HorizonForge grows with difficulty: the relative improvement is 26.37% for work vehicles and 27.30% for animals, rising to 31.99% for construction zones and 49.23% for pedestrians. Our method improves over HorizonForge across all four categories, with particularly large gains on pedestrians, where small object size make appearance preservation challenging.

B.4 Results by Request Type

The common-editing benchmark mixes edits that change very different parts of a scene: some move the ego camera, while others add, remove, or redirect an object. We break the results down by request type (Table 11) to see where the gains come from. Our method is the best on all six types. The largest improvements appear where an object must be rendered from a viewpoint that the coarse rendering only sketches: relative to HorizonForge, OSR rises by 29.08% for insertion and 25.08% for other-vehicle direction changes. The three ego edits move the camera but leave the objects in place, so the coarse rendering is already close to the target, and the improvements are smaller but still consistent, from 9.76% to 16.79%.

B.5 Training and Inference Cost

Training the final 720p checkpoint used 430 H100 GPU-hours across its two stages, 170 for the 480p model and 260 for the 720p model, excluding foundation-model pretraining, data preparation, and offline keyframe generation. Table 12 reports the per-case inference cost over the 229 cases of our main evaluation. Revision raises the recorded model-API cost from \$0.024 to \$0.115 per case, for the gains reported in Table 6. The reported cost covers the image editor, the judge, and the planner and reasoner calls, and excludes unpriced executor usage.

Table 12: Inference cost per case, averaged over the 229 cases of the main evaluation. The full agent includes the initial candidate.

	Initial candidate	Full agent
Recorded model-API cost (USD)	0.024	0.115

B.6 Adaptive Revision versus Repeated Sampling

Because image editing is stochastic, independently re-edited keyframes can differ substantially in quality, so simply sampling several candidates and keeping the best one is already a strong strategy. To measure this effect, we compare the agent with repeated sampling, which repeats the initial keyframe-editing and video-generation procedure without feedback, changing only the image-editing sample and the video-model seed. Both methods start from the same first candidate and keep the highest-scoring candidate under the same verifier. Repeated sampling alone raises OSR from 6.58 to 7.52 at eight candidates, a 14.29% improvement that confirms the large variance of keyframe generation (Table 13).

The agent nevertheless improves on repeated sampling at every budget, and the gap widens as more candidates are rendered, from 0.12 points at two candidates to 0.34 at eight. It is also more efficient: the agent reaches 7.52 with four candidates, a score repeated sampling needs eight candidates to reach.

Further analysis shows that the benefit of feedback is concentrated on hard cases. Table 14 groups the long-tail cases by the score of the shared first candidate. On the 30 cases whose first candidate scores below 6, the agent leads by 1.01 points, a 17.78% relative improvement and more than five times its lead on the 64 intermediate cases (0.19). In these cases the first attempt has failed badly, so resampling rarely produces a good result; the agent instead diagnoses which stage caused the failure and revises it. On the 26 cases that already score 8 or above, the two methods perform on par (within 0.05), since little room for improvement remains. The two methods share a candidate budget but not their total compute, because the agent also spends model calls on diagnosis and may change editing and generation settings.

Table 13: Agent versus repeated sampling on the long-tail benchmark. Both methods share the first candidate, and each column reports the best candidate retained within that budget.

Candidates	1	2	3	4	5	6	8
Repeated sampling	6.58	6.99	7.17	7.27	7.36	7.41	7.52
Agent	6.58	7.11	7.31	7.52	7.65	7.75	**7.86**
Difference	0.00	+0.12	+0.14	+0.25	+0.29	+0.34	+0.34

Table 14: OSR with eight candidates, grouped by the score of the shared first candidate. The groups cover the full long-tail benchmark.

Initial OSR	Cases	Repeated sampling	Agent	Difference
< 6	30	5.68	**6.69**	+1.01
$[6, 8)$	64	7.96	**8.15**	+0.19
≥ 8	26	**8.58**	8.53	-0.05