ReDreamer: Geometry-Grounded Memory for Real-Time Driving World Models

Anonymous authors
Paper under double-blind review

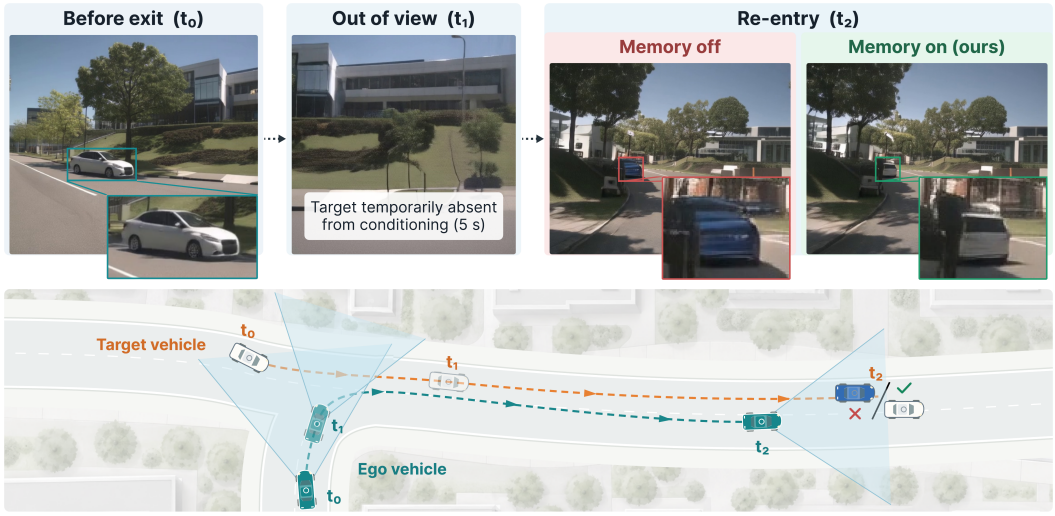


Figure 1: **A vehicle can reappear with a different appearance after leaving the view.** As the ego vehicle turns right, the target vehicle travels straight and temporarily leaves the view. Without appearance memory, it returns as a blue vehicle instead of the original white one, even though the source observations remain within the video context. Retaining past frames alone does not explicitly link the returning vehicle to its earlier appearance. ReDreamer stores the generated appearance and retrieves it upon re-entry, preserving the vehicle's white appearance. The two continuations use the same model weights and preceding video context.

Abstract

Video world models should generate evolving scenes in real time while preserving object appearance over time. Yet objects can reappear with altered appearances despite correct motion and geometry. The model reproduces where they are, but not what they looked like. Our key idea is to use the persistent object identities and geometry in scene controls as addresses for appearance memory. ReDreamer is a chunk-wise autoregressive video diffusion model that records generated appearances and retrieves them at current object locations through geometry-grounded attention. However, training with real-image references while reading generated appearances at inference introduces exposure bias in memory conditioning. We address this mismatch through rollouts that build and read their own memory, using distribution matching distillation to supervise continuations without forcing generated appearances to match the original dataset video. On our nuScenes-derived re-entry evaluation, ReDreamer improves our Re-entry Appearance Consistency (RAC) score by 10.5% over the strongest evaluated baseline. Enabling memory reads improves RAC by 12.0%. With four denoising steps per chunk, it reduces FVD by 5.7% relative to MagicDrive-V2 in our front-view evaluation and generates at 8.1 FPS.

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

1 Introduction

Video world models generate evolving scenes in response to actions or structured controls, offering environments for learning and simulation (Sun et al., 2026). Autoregressive generation and distillation increasingly enable low-latency video streams (Huang et al., 2025b; Sun et al., 2026). As cameras and objects move, parts of the scene become temporarily hidden. When an object becomes visible again, its appearance should remain consistent with earlier observations.

Maintaining this continuity becomes challenging when objects disappear from view and later return (Dou et al., 2026; Xu et al., 2026). Figure 1 shows a vehicle returning with a different appearance even though its earlier observations remain within the video context. Those frames provide appearance evidence, but using it requires associating the returning object with its earlier observations despite changes in position and visibility. We seek to make this connection explicit.

In autonomous driving, layout-conditioned video generators use road maps and vehicle boxes to control scene geometry, together with preceding frames to maintain temporal continuity (Mei et al., 2025; Ren et al., 2025). However, following these controls does not ensure that a returning vehicle retains its earlier appearance. The same trajectory and box sequence can describe vehicles of different colors and shapes, so geometric consistency alone cannot recover the appearance previously generated for an object. Persistent instance identities in driving scene controls offer an explicit link between observations of the same vehicle (Yang & Zhang, 2026; Yang et al., 2025b). Binding generated appearances to these identities would allow the model to retrieve the same object's visual history as its position and visibility change.

We introduce ReDreamer, a chunk-wise autoregressive video diffusion model that uses geometry-grounded memory to preserve object appearance through **re**-entry. Starting from an empty bank, it records object appearances from its own outputs. As objects move or return to view, their identities retrieve the corresponding records, and their projected regions route the stored features to the current image. This separates what the generator has already chosen from where the object appears now.

Learning to use this memory introduces a second challenge. Initial training uses real-image references, whereas deployment reads generated observations, creating exposure bias in memory conditioning (Huang et al., 2025b). We therefore train on causal rollouts that build and read their own memory. A generated appearance can satisfy the scene controls while differing from the dataset object, so reconstructing the original video would conflict with preserving that choice. Distribution matching distillation (DMD) instead supervises continuations under the generated records without requiring that paired reconstruction (Yin et al., 2024a).

To evaluate this ability, we construct controlled re-entry cases from nuScenes (Caesar et al., 2020) and introduce Re-entry Appearance Consistency (RAC), comparing each method's generated source and return observations. ReDreamer achieves the highest RAC among the evaluated driving generators, outperforming the strongest baseline by 10.5%. Enabling memory reads improves its score from 0.4506 to 0.5046. The four-step generator also achieves the lowest FVD in our front-view comparison and runs at 8.1 FPS on one H20. Table 1 summarizes its capabilities alongside representative generation and memory methods.

Our contributions are threefold:

- We present geometry-grounded object memory for causal driving world models, using structured scene controls to organize generated appearance records and retrieve them during subsequent rendering.
- We address exposure bias in object-memory conditioning through self-generated-memory rollouts and DMD, learning to continue the generator's appearance choices without reconstructing a fixed dataset video.
- We develop a controlled re-entry evaluation protocol and the RAC metric to measure whether returning objects retain their previously generated appearances.

2 Related Work

Video world models. Driving video world models use maps and object geometry to specify how a scene should evolve (NVIDIA, 2025; Ren et al., 2025). To extend generation over time, models

condition on preceding frames (Mei et al., 2025). Causal attention and few-step distillation further enable low-latency continuation (Huang et al., 2025b; Yin et al., 2025; Zhao et al., 2026). Recent driving systems combine these temporal mechanisms with structured scene controls (Basant et al., 2026; Cheng et al., 2026). This provides the setting for studying appearance continuity as controlled objects move out of and back into view.

Table 1: **Capabilities of representative video generators.** Autoregressive denotes causal temporal attention. Layout control specifies scene or object geometry. Appearance memory retrieves object appearance from generated history. Checkmarks indicate reported capabilities, with real-time based on claims or reported generation speeds.

Method	Domain	Autoregressive	Real-time	Layout control	Appearance memory
MagicDrive-V2 (Gao et al., 2025)	Driving	✗	✗	✓	✗
Cosmos-Transfer1 (NVIDIA, 2025)[1]	Driving	✗	✗	✓	✗
InstaDrive (Yang et al., 2025b)	Driving	✗	✗	✓	✓
OmniDreams (Basant et al., 2026)	Driving	✓	✓	✓	✗
WorldDirector (Wang et al., 2026a)	Game	✓	✗	✓	✓
SlotMemory (Dou et al., 2026)	General video	✓	✗	✗	✓
RECAP-Forcing (Xu et al., 2026)	General video	✓	✗	✗	✓
ReDreamer (ours)	Driving	✓	✓	✓	✓

Object references and memory. Visual references complement geometric controls by specifying object appearance. Image-generation methods encode appearance separately from pose or route reference features to designated regions (Wang et al., 2025; Wu et al., 2024). Video and driving generators extend this approach through visual conditioning and spatial or subject-aware attention (Huang et al., 2025a; Liu et al., 2025; Wei et al., 2026; Zhan et al., 2026). These supplied references define what should be generated. Appearance memory instead records the choices already made during a rollout, so those observations must remain associated with the objects as their locations and visibility change.

Memory methods differ in how they connect current generation to the relevant past. Camera pose and point correspondences provide spatial links to earlier observations (Li et al., 2025; Song et al., 2026; Xiao et al., 2025), while optical flow can identify newly revealed features worth retaining (Xu et al., 2026). For a moving object, this connection must follow the object as its image position changes. SlotMemory organizes KV history into semantic slots (Dou et al., 2026). VideoTetris uses reference-frame attention to preserve object features (Tian et al., 2024), while StoryMem conditions later shots on selected keyframes from generated history (Zhang et al., 2025). WorldDirector brings such references into the current layout by placing remembered appearances into object boxes and encoding an appearance-conditioning video (Wang et al., 2026a). ReDreamer instead binds local appearance features to supplied instance identities and retrieves them directly through attention at current projected regions, without reconstructing that video.

Training with generated history. Using generated observations as conditions also changes the training problem. Distribution matching distillation provides distribution-level supervision for few-step generators (Yin et al., 2024a;b). Self Forcing applies such supervision to autoregressive rollouts conditioned on generated video history (Huang et al., 2025b), while causal consistency distillation supplies a few-step initialization (Zhao et al., 2026). Online memory pools have also been incorporated into training, with viewpoint-based retrieval in Matrix-Game 3.0 (Wang et al., 2026b). However, generated video history alone does not align an object-reference branch that is still trained with real-image references. We extend rollout training to instance-indexed appearance records: each generated continuation and its score predictions share the memory available when that continuation was produced.

[1]Cosmos-Transfer1 refers specifically to the Cosmos-Transfer1-7B-Sample-AV model (Ren et al., 2025).

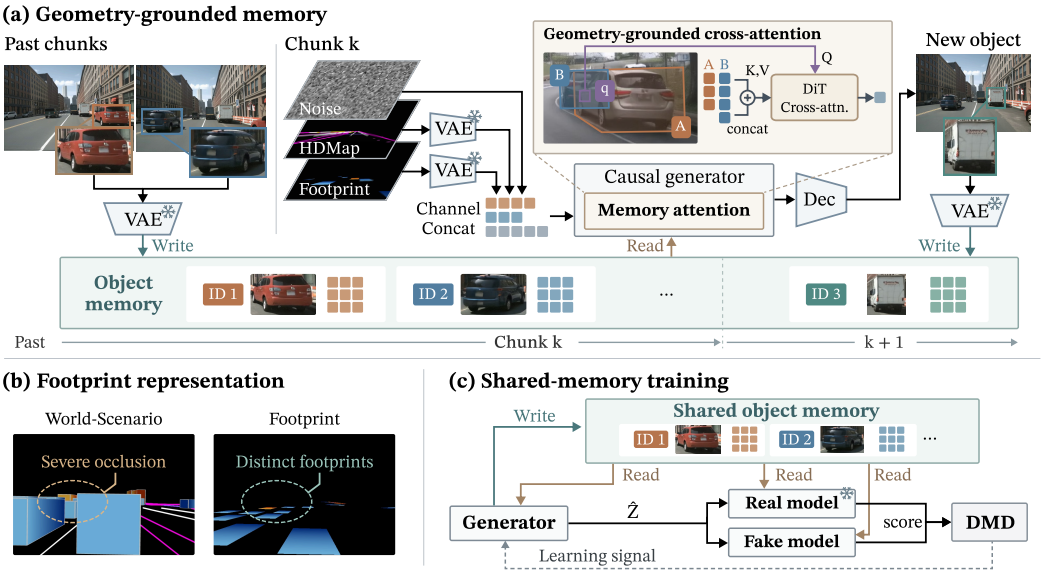


Figure 2: **Overview of ReDreamer. (a) Geometry-grounded memory.** Object records from past chunks guide the current generation through geometry-grounded attention. Queries in overlapping object projections attend jointly to the eligible memory tokens. Static-map and footprint latents are concatenated with noisy video latents. Newly generated objects supply records for later chunks. **(b) Footprint representation.** The same scene is rendered with world-scenario-style 3D cuboids (left) and projected ground footprints (right). Footprints reduce foreground occlusion of other control primitives, retaining more distinct object layouts in dense traffic. **(c) Shared-memory training.** The generator and both score models read the same memory from preceding generated chunks. The real–fake score difference supplies the DMD learning signal. Both score models receive the same noisy version of the clean-latent estimate $\hat{Z}$. This noising step is omitted for clarity.

3 Geometry-Grounded Object Memory

ReDreamer is a causal video generator built on Wan2.1-T2V-1.3B (Team Wan et al., 2025). It reads stored object appearances to generate each chunk, then records newly observed objects for subsequent chunks. For chunk k, let X_k and Z_k denote RGB frames and video latents, H_k the preceding video KV cache, and $\mathcal{M}_k$ the object-memory bank. Scene conditions C_k contain text, maps, instance identities and poses, and camera calibration.

Starting without input RGB or prior memory, the generator reads existing records before producing each chunk (Figure 2(a)):

$$Z_k = G_\theta(\epsilon_k; C_k, H_k, \mathcal{M}_k), \qquad X_k = \mathrm{Dec}(Z_k), \quad \epsilon_k \sim \mathcal{N}(0, I). \tag{1}$$

Here G_θ is the sampler and v_θ its flow predictor. Decoded observations update $\mathcal{M}_k$ only for subsequent chunks, separately from H_k.

3.1 Scene Conditioning

We adopt camera-aligned map and object rendering from Cosmos world-scenario (NVIDIA, 2026), separating static-map and dynamic-layout conditions into streams S_k and D_k. Separating them prevents object primitives from overwriting road information. For dynamic controls, we project each object's ground footprint instead of its full cuboid, reducing occlusion among control primitives in dense traffic (Figure 2(b)). Category and heading are encoded by color gradients, while footprint extent supplies ground-plane shape while the generator infers vertical appearance. Memory routing still uses full projected object hulls. These representations serve different purposes: footprints keep the layout readable, whereas full hulls make appearance features available across the image region occupied by the object, including its body above the ground. The frozen VAE encoder E encodes each

stream independently. Their clean 16-channel latents and noisy video latents form the 48-channel patch-embedding input:

$$U_k^\sigma = \text{PatchEmbed}(\text{Concat}_{\text{ch}}[Z_k^\sigma, E(S_k), E(D_k)]) \,. \tag{2}$$

3.2 Object Appearance Memory

We retain a fixed appearance record for each object as an anchor for subsequent generation. The record is created from its earliest generated observation with nonempty feature support, using the image region r_i defined by its projected box. Appendix A gives the registration and preprocessing rules.

To limit surrounding scene context, we decode first, isolate the object's region, and encode it independently. The aspect-preserving padded region $\bar{r}_i$ passes through the frozen Wan VAE up to its pre-head features:

$$\Phi_i = E_{\text{feat}}(\bar{r}_i) \in \mathbb{R}^{h_i \times w_i \times 384}. \tag{3}$$

The 384-dimensional features provide more bandwidth than the final 16-channel latents. Keeping local features rather than pooling them preserves detail for selective retrieval.

In padded reference-canvas coordinates, let u_p be the center of feature cell p, R_i the unpadded image region, and P_i^{src} the transformed source hull. We retain cells inside both regions:

$$\mathcal{S}_i = \{p : u_p \in R_i \cap P_i^{\text{src}}\}, \qquad F_i = \text{sg}(\Phi_i[\mathcal{S}_i]) \in \mathbb{R}^{|\mathcal{S}_i| \times 384}. \tag{4}$$

Here sg stops gradients. Readers normalize stored features as $A_i = \text{LN}(F_i)$.

Keeping the record fixed preserves the initial appearance anchor and avoids encoding the same object repeatedly. For newly registered objects $\mathcal{N}_k$,

$$\mathcal{M}_{k+1} = \mathcal{M}_k \cup \{i \mapsto F_i : i \in \mathcal{N}_k\}. \tag{5}$$

Later chunks reuse these features under the same instance ID. Deployment also caches their normalized tokens. Leaving the view removes an object from the current query regions, but does not delete its record. When it returns, its new projection reconnects generation to the same stored appearance.

3.3 Geometry-Grounded Memory Attention

Instance identity, geometry, and attention play complementary roles in retrieval. The instance ID selects a persistent record as the object moves, while its current projection restricts where that record can contribute. Within this region, attention selects local appearance features according to the current visual query. Thus, changes in image position do not require rediscovering the association between an object and its past appearance from video alone. Concatenate available tokens into A, and let $i(p)$ identify token p's object. A query patch Ω_j at representative frame $t(j)$ receives the mask

$$M_{jp} = \begin{cases} 0, & \text{area}(\Omega_j \cap P_{i(p),t(j)}) > 0, \\ -\infty, & \text{otherwise}, \end{cases} \tag{6}$$

where $P_{i,t}$ is the current projected object hull. The mask establishes object-level correspondence without fixing a pixel-to-pixel warp from the source observation. Each eligible query can select among that object's local features as its projected size and viewpoint change. Representative-frame alignment is specified in Appendix A.

Queries Q project normalized video hidden states, while keys K and values V project A. As in the region-restricted reference attention of MS-Diffusion (Wang et al., 2025), each head reads memory through

$$O = \text{softmax}_{\text{row}}\left(\frac{QK^\top}{\sqrt{d_h}} + M\right) V, \tag{7}$$

with head dimension d_h. A query patch can intersect several object projections. This geometric overlap leaves multiple plausible sources of appearance at the same query location. Rather than assigning it to a single object in advance, we include all eligible objects in one softmax, allowing its visual features to select the relevant appearance content.

Projected head outputs form a residual after causal self-attention and before text cross-attention. Queries without candidates bypass memory. Geometry thus constrains both which objects can contribute and which query–token pairs require computation (Appendix A.3).

4 Learning with Self-Generated Memory

Initial training uses real-image references, but deployment reads generated observations, creating exposure bias in memory conditioning (Huang et al., 2025b). Training must therefore use its own memory and preserve the appearance choices it contains: a generated red car should not be forced toward a blue dataset car. We use DMD under generation-time memory instead of paired reconstruction (Yin et al., 2024a), as illustrated in Figure 2(c).

Training has three stages: causal teacher forcing, consistency distillation, and adaptation with self-generated memory.

4.1 Generator Initialization

Following Causal Forcing, Stage 1 jointly trains the backbone and reference branch on nuScenes using flow matching, teacher-forced history, and real-image object references (Caesar et al., 2020; Lipman et al., 2023; Zhu et al., 2026).

Stage 2 retains these references and applies causal consistency distillation to obtain the four-step generator used to initialize Stage 3 (Song et al., 2023; Zhao et al., 2026). The first stage establishes reference-conditioned causal generation. The second reduces its sampling cost while retaining the conditioning interfaces. Stage 3 therefore starts from a few-step model that already reads object appearances.

4.2 Memory-Conditioned DMD (Stage 3)

Score-model preparation. The real and fake scores use bidirectional Wan2.1-T2V-14B and 1.3B models, respectively (Team Wan et al., 2025). Both are first fine-tuned on nuScenes with scene controls and real-image references through the generator's conditioning interfaces. DMD freezes the real model and initializes the fake model from its prior fit to the student's distribution.

Causal memory rollouts. Each rollout starts with empty memory. For chunk k, $B_k \subseteq \mathcal{M}_k$ contains the existing records of objects present in its controls. This snapshot remains fixed throughout denoising. A uniformly sampled step $s \in \{1, 2, 3, 4\}$ at noise level τ_s supplies the training estimate

$$\widetilde{Z}_{k,s} = Z_k^{\tau_s} - \tau_s v_\theta(Z_k^{\tau_s}, \tau_s; C_k, H_k, B_k)\,. \tag{8}$$

Only this prediction carries generator gradients. All four steps still complete to produce Z_k: the intermediate estimate supplies the loss, while the completed sample supplies the memory observation. This keeps the bank tied to completed generated observations, matching deployment even when the training loss is evaluated at an earlier denoising step. Its decoded observations are encoded and detached before entering later chunks' memory. Input states and historical caches are detached.

Fake-score adaptation. DMD requires a fake score that models the current student's distribution under its memory conditions. Switching to self-generated references changes that distribution, so we first freeze the generator and fit the fake score to $\widetilde{Z}_{k,s}$ under the records B_k that produced it. Joint DMD then alternates generator and fake-score updates with the same target construction (Huang et al., 2025b; Zhao et al., 2026).

Distribution matching under shared memory. All three models read the same features F_i, with role-specific normalization $A_i^{(r)} = \mathrm{LN}_r(F_i)$. Sharing B_k keeps the appearance condition fixed when comparing the real and fake scores. Their difference then evaluates the student under the same generated observations that guided its prediction, whereas using different references would also change the conditioning of the two distributions. This lets the training signal assess a continuation conditioned on the student's earlier appearance choices without prescribing a particular dataset continuation. Scores condition on $c_k = (C_k, B_k)$ and independently evaluate the current three-latent-frame chunk, whereas only the causal generator uses video history H_k.

Let $Y_k = \widetilde{Z}_{k,s}$ and $Y_k^\sigma = (1-\sigma)Y_k + \sigma\xi$, with fresh Gaussian noise ξ and scoring level σ sampled conditionally on s. The fake score learns these detached estimates through flow matching (Appendix B).

For generator updates, both scores are frozen and predict $\widehat{Y}_k^{(r)} = Y_k^\sigma - \sigma v_r(Y_k^\sigma, \sigma; c_k)$ without classifier-free guidance. With D scalar elements per chunk, DMD uses

$$a_k = \frac{1}{D}\|Y_k - \widehat{Y}_k^{\text{real}}\|_1, \qquad g_k = \frac{\widehat{Y}_k^{\text{fake}} - \widehat{Y}_k^{\text{real}}}{a_k}, \tag{9}$$

$$\mathcal{L}_G = \mathbb{E}_{k,s,\sigma,\xi}\left[\frac{1}{2D}\left\|Y_k - \text{sg}(Y_k - g_k)\right\|_2^2\right]. \tag{10}$$

Chunk losses are weighted equally. Current memory-reading parameters receive gradients, whereas historical writes do not. Deployment retains only the generator. Appendix B gives training objectives and initialization details.

5 Experiments

5.1 Experimental Setup

We train on nuScenes using 64 PPUs (approximately NVIDIA H20-level performance per PPU) and evaluate on its validation scenes (Caesar et al., 2020). ReDreamer uses four denoising steps per chunk and starts with an empty memory bank. Appendix C.3 gives the baseline configurations and generation settings.

Re-entry evaluation. To evaluate appearance retention across an interruption in visibility, we construct 60 cases from distinct nuScenes scenes through programmatic trajectory edits. Each target exits and returns to the view. Its identity and source/return times are fixed before generation. The edits change target motion and, where needed, ego motion while retaining road geometry and target identity. This makes the visibility interruption explicit in the controls rather than relying on naturally occurring returns.

Motivated by DrivingGen's agent-level appearance evaluation (Zhou et al., 2026) and MemoBench's disappear-and-reappear evaluation (Chen et al., 2026), we measure appearance retention across a controlled re-entry. We match detections to projected target boxes and compare each method's own generated source and return crops, x_i^{src} and x_i^{ret}, rather than the original dataset vehicle. Using DINOv3 ViT-L/16 features ϕ (Siméoni et al., 2025), we define Re-entry Appearance Consistency (RAC) as

$$z_i^a = \frac{\phi(x_i^a)}{\|\phi(x_i^a)\|_2}, \quad \text{RAC} = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}(z_i^{\text{src}})^\top z_i^{\text{ret}}, \quad a \in \{\text{src}, \text{ret}\}, \tag{11}$$

where $\mathcal{V}$ contains cases with valid observations at both times. Higher scores indicate greater appearance similarity. Matching rules and valid-pair counts are given in Appendix C.2.

Video-quality evaluation. We regenerate videos for every method on 150 nuScenes validation scenes using released model weights and the corresponding inference pipelines, without ground-truth RGB initialization. We evaluate one sequence per scene using its first 16 native front-view frames, and for multi-view methods, we retain only the front camera. FID and FVD compare estimated feature distributions, so their absolute values depend on the evaluated data and sample count (Heusel et al., 2017; Jayasumana et al., 2024; Unterthiner et al., 2019). Our front-view-only cohort differs from the larger multi-view collections used in published evaluations: *the relatively high FID/FVD values here are not directly comparable with those published scores*. We therefore recompute all methods under this evaluation protocol rather than import their reported numbers. We also report VBench imaging quality and motion smoothness (Huang et al., 2024), complementing distributional distances given known discrepancies between FID and perceived quality and FVD's content bias (Ge et al., 2024; Jayasumana et al., 2024). Implementation details are in Appendix C.3.

5.2 Quantitative Comparison

ReDreamer achieves the highest RAC among the evaluated driving generators (Table 2), improving over the strongest external baseline, UniMLVG, from 0.4568 to 0.5046 (10.5%). Since all methods receive spatial controls, the result highlights the value of retaining appearance in addition to specifying where an object should be. ReDreamer produces an 81-frame sequence in about 10 seconds on one H20, combining this appearance retention with few-step generation.

Table 2: **Re-entry appearance consistency and generation time.** ReDreamer exceeds the runner-up RAC by 10.5% while generating 81 frames in about 10 seconds on one H20. RAC averages valid source–return pairs. Evaluation and timing settings are detailed in Appendices C.2 and C.6.

Method	Resolution	Time/case (s)	RAC ↑
MagicDrive-V2 (Gao et al., 2025)	848×1600	1007	0.3900
UniMLVG (Chen et al., 2025)	256×448	1250	<u>0.4568</u>
DreamForge-DiT (Mei et al., 2025)	448×800	358	0.4399
X-Scene (Yang et al., 2025a)	224×400	76	0.4565
ReDreamer	320×480	10	**0.5046**

Table 3: **Video quality on 150 nuScenes validation scenes.** ReDreamer achieves the lowest FVD and highest imaging quality with four denoising steps per chunk. All methods are rerun and scored on the first 16 front-view frames. Published multi-view scores are not used. IQ/MS denote VBench imaging quality/motion smoothness.

Method	Steps	FID ↓	FVD ↓	IQ (%) ↑	MS (%) ↑
MagicDrive v1 (Gao et al., 2024)	20	**53.117**	644.045	<u>61.421</u>	96.983
MagicDrive-V2 (Gao et al., 2025)	30	59.954	<u>435.918</u>	60.281	98.347
UniMLVG (Chen et al., 2025)	50	<u>55.995</u>	562.527	57.793	97.440
DreamForge-DiT (Mei et al., 2025)	30	70.237	537.167	54.617	**98.472**
X-Scene (Yang et al., 2025a)	20	68.646	728.903	56.583	97.894
ReDreamer	4	56.613	**410.862**	**61.787**	<u>98.445</u>

Video quality. With four denoising steps per chunk, ReDreamer obtains the lowest FVD in Table 3, compared with 20–50 steps for the external methods. It ranks third in FID, first in VBench imaging quality, and second in motion smoothness. The agreement between strong video-distribution performance and high perceptual scores indicates that few-step generation preserves overall visual quality alongside object appearance retention, so the result is not confined to the targeted RAC metric.

5.3 Qualitative Comparison

Figure 3 compares each method's own generated history at the initial observation, an out-of-view moment, and re-entry. ReDreamer preserves the target's appearance across the visibility interruption, while the external methods change its color or vehicle shape.

5.4 Ablation Studies

Reading appearance memory. Table 4 ablates memory reading and compares the training-reference choices. Enabling reads raises ReDreamer's RAC from 0.4506 to 0.5046, a 12.0% increase. Both settings retain observation encoding and memory writes, so the change concerns access to stored appearance. FID changes little and FVD does not improve, despite the higher RAC. Thus, improved retention of individual objects need not lower a global distributional distance, motivating the targeted re-entry measure alongside overall video-quality evaluation.

Training with generated memory. The two DMD recipes start from the same consistency-distilled generator and both use self-generated memory at evaluation. Training with self-generated memory improves read-on RAC from 0.4569 to 0.5046, while reducing FID from 63.632 to 56.613 and FVD from 479.738 to 410.862. FID and FVD also improve with reads disabled, showing that the benefit extends beyond the immediate contribution of retrieved features to the adapted generator itself. The RAC gain from enabling reads is also larger after self-generated-memory training: 0.0540 versus 0.0206 for GT-reference training. This pattern is consistent with learning to use generated records more effectively, in addition to improving overall generation quality. Appendix C.5 describes the training procedures.

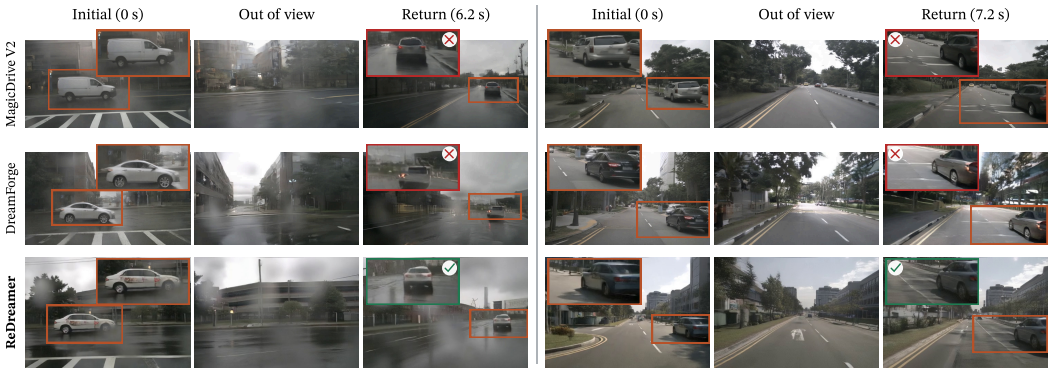


Figure 3: **Appearance retention after re-entry.** Within each case, rows show each method's own generated sequence. Columns show the initial observation, an out-of-view moment, and the return. ReDreamer preserves the target's appearance upon re-entry. Zoom in for details.

Table 4: **Ablation of memory reading and training-reference source.** Reading improves RAC, while self-generated-memory training improves both RAC and video quality. Off disables reads but retains encoding and writes. All variants use four steps and empty initial banks. FID/FVD use the validation protocol of Table 3.

Training references	Memory reads	RAC ↑	FID ↓	FVD ↓
GT reference	Off	0.4363	63.843	477.990
GT reference	On	<u>0.4569</u>	63.632	479.738
Self-generated memory	Off	0.4506	<u>56.886</u>	**406.524**
Self-generated (ours)	**On**	**0.5046**	**56.613**	<u>410.862</u>

Memory-reading overhead. On one H20, we measure six fixed 81-frame clips per setting after warm-up using BF16 and batch size 1 (Table 5). With memory enabled, ReDreamer generates at 8.085 FPS with a mean first-chunk latency of 1.030 seconds. Reading adds 13.4% to pipeline time while leaving peak memory and first-chunk latency nearly unchanged, so its main cost is incurred during subsequent generation rather than initial response. This measures the incremental cost of retrieval, since both settings retain encoding and writing. Appendix C.6 gives the timing protocol.

Table 5: **Ablation of memory-reading overhead on one H20.** Reading adds 13.4% to pipeline time with nearly unchanged first-chunk latency and peak memory.

Resolution	Memory reads	FPS ↑	First-chunk latency (s) ↓	Peak allocated memory (GiB) ↓
320×480	Off	9.167	1.025	19.474
320×480	On	8.085	1.030	19.347

6 Conclusion

We presented ReDreamer, a causal driving world model that retains generated object appearances across interruptions in visibility. Geometry-grounded memory connects current objects to their past observations, while self-generated-memory rollouts and DMD adapt the generator to the records it encounters during generation. Our experiments demonstrate improved appearance retention alongside competitive video quality with four-step causal generation. Together, these findings show how structured scene controls can support both spatial control and persistent appearance, allowing a generator's previous visual choices to inform its subsequent outputs. Appendix E discusses limitations and future directions.

AI USE STATEMENT

We used generative AI tools to assist with literature review, method and experimental design, code implementation and debugging, scenario-editing scripts and result interpretation, manuscript drafting, translation and revision, and figure design. The authors personally checked all AI-assisted content, including cited sources, methodological explanations, key code, experimental results, and their presentation in figures and tables. The authors take responsibility for the final text, claims, and research artifacts.

ETHICS STATEMENT

Our experiments use existing nuScenes data and programmatically edited scene controls for driving simulation research. The source data remain subject to the nuScenes dataset terms and CC BY-NC-SA 4.0 license. Improved visual quality and appearance consistency do not by themselves establish the safety of a driving system. Using generated scenes for safety-critical evaluation requires independent validation.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the memory mechanism and three-stage training procedure, with implementation and training details in Appendices A and B. Section 5.1 and Appendix C specify the data construction and evaluation protocols. The memory conditions, training comparisons, and runtime measurements are detailed in Appendices C.5 and C.6. The supplementary materials include anonymized source code, training configurations, instructions for data preparation, training, inference, and evaluation, and qualitative comparison and failure-case videos.

REFERENCES

Aarti Basant, Amlan Kar, Despoina Paschalidou, Fangyin Wei, Francesco Ferroni, Guillermo Garcia Cobo, Haithem Turki, Huan Ling, Jaewoo Seo, James Lucas, Jay Zhangjie Wu, Jialiang Wang, Jonathan Lorraine, Jun Gao, Kai He, Katarina Tothova, Kevin Xie, Michał Tyszkiewicz, Qi Wu, Riccardo de Lutio, Ruilong Li, Sanja Fidler, Seung Wook Kim, Tianchang Shen, Tianshi Cao, Tobias Pfaff, William Lew, Xindi Wu, Xuanchi Ren, Yifan Lu, Yuxuan Zhang, Zan Gojcic, and Zian Wang. NVIDIA OmniDreams: Real-Time Generative World Model for Closed-Loop Autonomous Vehicle Simulation, 2026. arXiv:2606.03159.

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 (CVPR)*, 2020.

Haoyu Chen, Kaichen Zhou, Hang Hua, Kaile Zhang, Jingwen Qian, Wufei Ma, Haonan Chen, Chunjiang Liu, Yizhou Zhao, Xiaoyuan Wang, Weiyue Li, Alan Yuille, Paul Pu Liang, and Yilun Du. MemoBench: Benchmarking World Modeling in Dynamically Changing Environments. In *Computer Vision – ECCV 2026*, volume 17067 of *Lecture Notes in Computer Science*, pp. 658–676. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-37174-4_38.

Rui Chen, Zehuan Wu, Yichen Liu, Yuxin Guo, Jingcheng Ni, Haifeng Xia, and Siyu Xia. UniMLVG: Unified Framework for Multi-view Long Video Generation with Comprehensive Control Capabilities for Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25453–25463, 2025.

Ke Cheng, Hanqiao Ye, Lei Shi, Yahui Liu, Yunhan Shen, Jingtao Dong, Zhenke Wang, Wenxuan Ao, Weixiang Xu, Kaining Huang, and Shuhan Shen. M^4world: A multi-view multimodal driving world model for interactive object manipulation and minute-long streaming, 2026. arXiv:2607.14005.

Weijia Dou, Hui Li, Jiahao Cui, Lei Zhou, Jingdong Wang, and Siyu Zhu. SlotMemory: Object-Centric KV Memory for Streaming Long-Video Generation, 2026. arXiv:2605.31033.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28135–28144, 2025.

Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the Content Bias in Frechet Video Distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7277–7288, June 2024.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent Video Diffusion Models for High-Fidelity Long Video Generation, 2023. arXiv:2211.13221v2.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Binyuan Huang, Yuqing Wen, Yucheng Zhao, Yaosi Hu, Yingfei Liu, Fan Jia, Weixin Mao, Tiancai Wang, Chi Zhang, Chang Wen Chen, Zhenzhong Chen, and Xiangyu Zhang. SubjectDrive: Scaling Generative Data in Autonomous Driving via Subject Control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3617–3625, 2025a. doi: 10.1609/aaai.v39i4.32376.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. In *Advances in Neural Information Processing Systems*, volume 38, pp. 167283–167308. Curran Associates, Inc., 2025b. doi: 10.52202/085713-5576.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive Benchmark Suite for Video Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, 2024.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking FID: Towards a Better Evaluation Metric for Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9307–9315, June 2024.

Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. VMem: Consistent Interactive Video Scene Generation with Surfel-Indexed View Memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25690–25699, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Lijie Liu, Tianxiang Ma, Bingchuan Li, Zhuowei Chen, Jiawei Liu, Gen Li, Siyu Zhou, Qian He, and Xinglong Wu. Phantom: Subject-consistent video generation via cross-modal alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14951–14961, 2025.

Jianbiao Mei, Tao Hu, Xuemeng Yang, Licheng Wen, Yu Yang, Tiantian Wei, Yukai Ma, Min Dou, Botian Shi, and Yong Liu. DreamForge: Motion-Aware Autoregressive Video Generation for Multi-View Driving Scenes, 2025. arXiv:2409.04003v4.

NVIDIA. Cosmos-Transfer1: Conditional World Generation with Adaptive Multimodal Control, 2025. arXiv:2503.14492.

NVIDIA. World Simulation with Video Foundation Models for Physical AI, 2026. arXiv:2511.00062v2.

Xuanchi Ren, Yifan Lu, Tianshi Cao, Ruiyuan Gao, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, Seung Wook Kim, Jun Gao, Laura Leal-Taixe, Mike Chen, Sanja Fidler, and Huan Ling. Cosmos-Drive-Dreams: Scalable Synthetic Driving Data Generation with World Foundation Models, 2025. arXiv:2506.09042.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. arXiv:2508.10104.

Chenxi Song, Yanming Yang, and Chi Zhang. World in World: Explore the World with World Models, 2026. arXiv:2609.11548.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 32211–32252. PMLR, 2023.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. WorldPlay: Towards Long-Term Geometric Consistency for Real-Time Interactive World Modeling. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

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. arXiv:2503.20314.

Ye Tian, Ling Yang, Haotian Yang, Yuan Gao, Yufan Deng, Jingmin Chen, Xintao Wang, Zhaochen Yu, Xin Tao, Pengfei Wan, Di Zhang, and Bin Cui. VideoTetris: Towards Compositional Text-to-Video Generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 29489–29513. Curran Associates, Inc., 2024. doi: 10.52202/079017-0928.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards Accurate Generative Models of Video: A New Metric & Challenges, 2019. arXiv:1812.01717v2.

Hanlin Wang, Hao Ouyang, Qiuyu Wang, Wen Wang, Qingyan Bai, Ka Leong Cheng, Yue Yu, Yixuan Li, Yihao Meng, Zichen Liu, Yanhong Zeng, Yujun Shen, and Qifeng Chen. WorldDirector: Building Controllable World Simulators with Persistent Dynamic Memory, 2026a. arXiv:2607.02517.

Xierui Wang, Siming Fu, Qihan Huang, Wanggui He, and Hao Jiang. MS-Diffusion: Multi-subject Zero-shot Image Personalization with Layout Guidance. In *International Conference on Learning Representations*, volume 2025, pp. 95118–95146, 2025.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, Yidan Xietian, Jiangbo Pei, Liang Hu, Boyi Jiang, Hua Xue, Zidong Wang, Haofeng Sun, Wei Li, Wanli Ouyang, Xianglong He, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-Game 3.0: Real-Time and Streaming Interactive World Model with Long-Horizon Memory, 2026b. arXiv:2604.08995.

Yujie Wei, Xinyu Liu, Shiwei Zhang, Hangjie Yuan, Jinbo Xing, Zhekai Chen, Xiang Wang, Haonan Qiu, Rui Zhao, Yutong Feng, Ruihang Chu, Yingya Zhang, Yike Guo, Xihui Liu, and Hongming Shan. DreamVideo-Omni: Omni-Motion Controlled Multi-Subject Video Customization with Latent Identity Reinforcement Learning, 2026. arXiv:2603.12257.

Ziyi Wu, Yulia Rubanova, Rishabh Kabra, Drew A. Hudson, Igor Gilitschenski, Yusuf Aytar, Sjoerd van Steenkiste, Kelsey R. Allen, and Thomas Kipf. Neural Assets: 3D-Aware Multi-Object Scene Synthesis with Image Diffusion Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 76289–76318. Curran Associates, Inc., 2024. doi: 10.52202/079017-2429.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WorldMem: Long-term Consistent World Simulation with Memory. In *Advances in Neural Information Processing Systems*, volume 38, pp. 49632–49652. Curran Associates, Inc., 2025. doi: 10.52202/085713-1659.

Haiyang Xu, Zheng Ding, and Zhuowen Tu. RECAP-Forcing: Retaining Content Appearances for Long Video Generation, 2026. arXiv:2608.26671.

Yu Yang, Alan Liang, Jianbiao Mei, Yukai Ma, Yong Liu, and Gim Hee Lee. X-Scene: Large-Scale Driving Scene Generation with High Fidelity and Flexible Controllability. In *Advances in Neural Information Processing Systems*, volume 38, pp. 104415–104451. Curran Associates, Inc., 2025a. doi: 10.52202/085713-3487.

Zhuoran Yang and Yanyong Zhang. ConsisDrive: Identity-Preserving Driving World Models for Video Generation by Instance Mask. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 156328–156341, 2026.

Zhuoran Yang, Xi Guo, Chenjing Ding, Chiyu Wang, Wei Wu, and Yanyong Zhang. Instadrive: Instance-aware driving world models for realistic and consistent video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25410–25420, 2025b.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved Distribution Matching Distillation for Fast Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 47455–47487. Curran Associates, Inc., 2024a. doi: 10.52202/079017-1505.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623, 2024b.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From Slow Bidirectional to Fast Autoregressive Video Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974, 2025.

Yifan Zhan, Zhengqing Chen, Qingjie Wang, Zhuo He, Muyao Niu, Xiaoyang Guo, Wei Yin, Weiqiang Ren, Qian Zhang, and Yinqiang Zheng. Composing Driving Worlds through Disentangled Control for Adversarial Scenario Generation. In *Computer Vision – ECCV 2026*, volume 17071 of *Lecture Notes in Computer Science*, pp. 467–487. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-37016-7_25.

Kaiwen Zhang, Liming Jiang, Angtian Wang, Jacob Zhiyuan Fang, Tiancheng Zhi, Qing Yan, Hao Kang, Xin Lu, and Xingang Pan. StoryMem: Multi-shot Long Video Storytelling with Memory, 2025. arXiv:2512.19539.

Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal Forcing++: Scalable Few-Step Autoregressive Diffusion Distillation for Real-Time Interactive Video Generation, 2026. arXiv:2605.15141.

Yang Zhou, Hao Shao, Letian Wang, Zhuofan Zong, Hongsheng Li, and Steven L. Waslander. DrivingGen: A Comprehensive Benchmark for Generative Video World Models in Autonomous Driving. In *International Conference on Learning Representations*, volume 2026, pp. 103502–103524, 2026.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal Forcing: Autoregressive Diffusion Distillation Done Right for High-Quality Real-Time Interactive Video Generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

A Implementation Details

A.1 Scene Control Rendering

The static and dynamic streams use the scene-conditioning path in Section 3.1. In each frame, static map elements are projected with semantic colors for lane markings and road boundaries. For a bottom-face vertex p in object coordinates, its image location u is given by

$$\lambda \begin{bmatrix} u \\ 1 \end{bmatrix} = K_t [I_3 \; 0] (T_t^{\text{cam}})^{-1} T_{i,t}^{\text{obj}} \begin{bmatrix} p \\ 1 \end{bmatrix}, \tag{12}$$

where K_t is the camera intrinsic matrix, T_t^{cam} and $T_{i,t}^{\text{obj}}$ are camera-to-world and object-to-world transforms, and λ is camera depth. Class-specific colors interpolate from rear to front along the object's longitudinal axis, following the heading-color convention of Cosmos world-scenario (NVIDIA, 2026). Overlapping footprints are drawn from far to near using their mean camera depth. Dynamic footprint colors interpolate in the object-local ground plane using the inverse planar projection. Polygons crossing the camera near plane are clipped before rasterization. The reference-image encoder described in Equation 3 uses the frozen pre-head network, while both scene-control streams use the final VAE latents.

A.2 Object Memory Representation

The writer checks the earliest finite, positive-area projected region in each decoded chunk and defers registration to a later chunk if its retained token set is empty. Regions are padded onto a 160×160 reference canvas with their aspect ratios preserved. Training stores detached pre-normalization features F_i. Each role computes its own $\text{LN}_r(F_i)$. With fixed deployment weights, the bank stores the source RGB region and caches the generator-normalized tokens. Feature and query normalization use separate learned LayerNorm modules. There are no additional source-coordinate positional encodings or object-ID embeddings: IDs determine candidate membership, and attention retrieves local content. The reference output projection is initialized near zero.

For visual hidden state u_j, the head outputs in Equation 7 are combined as

$$u'_j = u_j + \text{Concat}[o_j^{(1)}, \ldots, o_j^{(n_h)}] W_O + b_O. \tag{13}$$

This residual follows causal self-attention and precedes text cross-attention and the feed-forward network. Queries with no eligible records receive no residual.

A.3 Sparse Routing and Generation Settings

The implementation stores each query's eligible object indices in a sparse list. Each index refers to a variable-length token record. For query j, let $\mathcal{I}_j$ contain its eligible stored objects and let N_i be the number of tokens in object i's record. The number of query–token pairs evaluated by memory attention is

$$E_{\text{attn}} = \sum_{j=1}^{L} \sum_{i \in \mathcal{I}_j} N_i, \tag{14}$$

where L is the number of visual queries. Dense attention over the same available records would evaluate $L \sum_i N_i$ pairs. Equation 14 counts attention pairs, rather than total runtime: feature projections, geometry processing, and memory writing have separate costs. Overlapping projections contribute all eligible objects to one softmax, while queries with empty candidate sets receive no memory residual.

Table 6 summarizes the architecture and generation settings. The memory bank is local to a continuous rollout. An object that leaves the image has no eligible visual queries but retains its record for later reuse under the same track ID. Separate sequences start with fresh banks.

B Training Details

Table 7 summarizes the objectives, conditioning sources, and initialization of the three training stages.

Table 6: **Memory and generation configuration.** A persistent object record is separate from the generator's temporal KV cache. Training and main evaluations use 320×480 clips.

Setting	Value
Generator blocks / width / heads	$30/1536/12$
Latent patch size	$1 \times 2 \times 2$
Input channels	48: noisy video, static control, dynamic control, 16 each
Training clip	81 RGB frames / 21 latent frames
Chunk size	Three latent frames, nine initial RGB frames, twelve thereafter
Evaluation resolution	320×480
Denoising evaluations	Four per chunk
Guidance	CFG 0 (conditional-only)
Reference canvas	160×160, aspect ratio preserved
Reference features	Frozen VAE pre-head, 384 channels, native spatial grid
Record retention	First valid observation, write once per track ID
Reference token count	Variable, selected by content rectangle and projected hull
Initial history / memory	Empty / empty
Temporal context	Full history within each evaluated clip, no sink, clean-context KV refresh after each chunk
Geometry–latent alignment	Local RGB indices $(0, 4, 8)$ initially, $(3, 7, 11)$ thereafter

Table 7: **Training stages and conditioning sources.** The first two stages learn reference-conditioned causal generation. Stage 3 trains with memory built from generated observations.

	Stage 1	Stage 2	Stage 3
Objective	Flow matching	Causal consistency	Memory-conditioned DMD
Generator	Causal 1.3B	Causal 1.3B	Causal 1.3B
Video history	Dataset video	Dataset video	Generated video
Appearance source	Dataset objects	Dataset objects	Generated objects
Initialization	Wan2.1-T2V-1.3B	Stage 1 generator	Stage 2 generator EMA

B.1 Reference-Conditioned Causal Training

In Stage 1, following Causal Forcing, we train the causal generator with teacher-forced history to use object references through the routed attention path (Zhu et al., 2026). Training crops come from observations of the same vehicle in real videos. The source selector prioritizes observations outside the target window and falls back to observations inside it only when no outside observation exists. Candidates are ranked by crop area, annotated visibility, distance from the image boundary, and isolation from other projected objects, using no target RGB or prediction errors. Let $\mathcal{R}_k = \{i \mapsto A_i\}$ denote the available training reference features, Z_k^0 a clean target latent, and H_k^{GT} the teacher-forced history. With Gaussian noise ϵ and noise level σ, training uses the flow-matching objective (Lipman et al., 2023)

$$\begin{aligned} Z_k^\sigma &= (1-\sigma)Z_k^0 + \sigma\epsilon, \\ \mathcal{L}_{\mathrm{FM}} &= \mathbb{E}\Big[w(\sigma)\left\|v_\theta(Z_k^\sigma, \sigma; C_k, H_k^{\mathrm{GT}}, \mathcal{R}_k) - (\epsilon - Z_k^0)\right\|_2^2\Big], \end{aligned} \tag{15}$$

where w is the noise-level weighting of the training scheduler and the squared error is averaged over target elements. The backbone and routing projections are optimized jointly, while the VAE encoders remain frozen. The self-generated-memory stage below retains this encoding and routing interface while replacing dataset appearance references with the current generator's completed observations.

B.2 Few-Step Initialization

In Stage 2, we distill the reference-conditioned model into a four-step causal generator using causal consistency distillation (Song et al., 2023; Zhao et al., 2026). For adjacent noise levels $0 \le \sigma' < \sigma$, we form Z_k^σ using Equation 15. The frozen causal teacher takes a local denoising step with classifier-free guidance to obtain $\widetilde{Z}_k^{\sigma'}$. Writing F_θ for the student's clean-latent prediction and $F_{\bar{\theta}}$ for

its exponential-moving-average target, the consistency loss is

$$\mathcal{L}_{\text{CD}} = \mathbb{E}\left[\left\| F_\theta(Z_k^\sigma, \sigma; c_k) - \text{sg}\Big(F_{\bar{\theta}}(\widetilde{Z}_k^{\sigma'}, \sigma'; c_k)\Big)\right\|_2^2\right], \tag{16}$$

where $c_k = (C_k, H_k^{\text{GT}}, \mathcal{R}_k)$ collects the shared conditions and sg stops gradients.

B.3 Score Models and Stage 3 Updates

For Stage 3, the causal generator starts from the consistency-distilled EMA weights. The bidirectional fake-score model is Wan2.1-T2V-1.3B initialized from a prior fit to the student distribution. The real-score model is Wan2.1-T2V-14B trained on real videos and kept frozen. Both score models first learn the scene and reference conditions on real-video windows. During self-generated-memory DMD, they score each target chunk independently using its scene controls and generation-time memory. The rollout completes each chunk before beginning the next, preserving the causal memory order used at deployment (Huang et al., 2025b; Zhao et al., 2026).

We first freeze the generator and adapt the fake score under self-generated memory. Joint DMD then alternates fake-score and generator updates while retaining the adapted fake model. Deployment uses only the generator's EMA weights with four denoising evaluations per chunk. Table 6 summarizes the generation settings.

The shared records are normalized separately by each role:

$$A_i^{(r)} = \text{LN}_r(F_i), \qquad r \in \{G, \text{real}, \text{fake}\}, \quad i \in B_k. \tag{17}$$

With $Y_k = \widetilde{Z}_{k,s}$, the fake-score flow-matching objective is

$$Y_k^\sigma = (1-\sigma)Y_k + \sigma\xi, \qquad \mathcal{L}_{\text{fake}} = \mathbb{E}_{k,s,\sigma,\xi}\left[\frac{1}{D}\left\|v_\psi(Y_k^\sigma, \sigma; c_k) - (\xi - \text{sg}(Y_k))\right\|_2^2\right]. \tag{18}$$

Here D counts scalar elements in the chunk, ξ is fresh Gaussian noise, and σ is sampled conditionally on the rollout's selected sampling step s. Generator estimates are detached for fake-score updates.

B.4 Training Algorithm

The two procedures below separate causal state construction from parameter updates. Both training and inference use completed chunks to write appearance records. Training additionally retains a selected-step prediction for the loss, but it does not use that intermediate prediction as a memory observation.

Procedure 1: Causal generation and memory writing.

Input: scene conditions $C_{1:K}$, generator G_θ, and sampling schedule.
Initialize: video history $H_1 \leftarrow \emptyset$ and object memory $\mathcal{M}_1 \leftarrow \emptyset$.

1. For each chunk $k = 1, \ldots, K$, select previously stored records B_k for objects in C_k. Keep this snapshot fixed during sampling.
2. Sample Gaussian noise and run all four denoising evaluations under (C_k, H_k, B_k) to obtain the completed latent Z_k.
3. Update the video-history cache from the completed latent, using the same B_k, then decode Z_k into RGB frames X_k with the stateful VAE decoder.
4. For each unregistered eligible object, examine its regions in temporal order. Independently encode the first region satisfying the geometric and token-support conditions in Section 3.
5. Commit its detached features under the object's track ID. Keep all existing records unchanged to obtain $\mathcal{M}_{k+1}$, then output X_k and continue.

Output: the generated chunks $X_{1:K}$. Video-cache management does not remove object records.

Procedure 2: Training under self-generated memory.

Input: the four-step generator, initialized fake-score model, frozen real-score model, and training scene controls.

1. For each loss evaluation, start a fresh rollout as in Procedure 1. Sample one denoising-step index s uniformly from $\{1, 2, 3, 4\}$ and use it throughout that rollout.
2. In each chunk, retain $Y_k = \widetilde{Z}_{k,s}$ from Equation 8. For a generator update, only this prediction retains a gradient graph. Detach historical caches and subsequent sampling transitions, and complete all four steps before updating history and memory.
3. Pair Y_k with the snapshot B_k used to generate it, even after new records have been written. Add fresh noise at the score noise level σ. Each score model uses its own normalization of the shared records.
4. During fake-score adaptation, freeze the generator and minimize Equation 18 over the fake-score model and its reference normalization. Average the seven chunk losses equally.
5. During joint DMD, retain the same fake-score objective and alternate with generator updates using Equations 9–10. Freeze both score models for this generator loss, then update generator EMA after its optimizer step.
6. Use a fresh rollout for each generator or fake-score loss evaluation and discard its memory afterward. The frozen VAE and real-score model are never optimized in this stage.

Output: the adapted generator and its EMA weights. Only the generator is used at deployment.

Each score call processes one chunk, not the generator's accumulated video history. The current implementation refreshes historical KV entries with a separate forward pass on the completed chunk before decoding and committing memory. On joint iterations that update both models, the fake-score update precedes the generator update. Their independent rollouts each carry their own matched memory snapshots.

B.5 Memory Conditions and Gradient Flow

Procedure 2 specifies the rollout and memory-snapshot construction. Generator and fake-score updates use independent rollouts, and each target retains its own generation-time records for scoring.

The three roles share source records and target geometry and apply their own feature normalization. For generator updates, both score models are fixed. For fake-score updates, the fake model and its feature normalization are trainable. The real model and VAE encoders remain frozen. Both score predictions are conditional-only in this self-generated-memory configuration. The DMD direction divides by the mean absolute real-prediction residual without an epsilon or lower clamp, then applies `torch.nan_to_num`. Equation 9 gives its finite-valued form.

At each sampling transition, the clean prediction is detached and re-noised for the next sampling level. Score models use local position indices for each chunk. The noise level τ_s in Equation 8 is the scheduler's warped level.

B.6 Temporal Alignment

The 81-frame training rollout has 21 latent frames in seven chunks, with an initially empty historical KV cache. Training geometry uses zero-based local RGB indices $(0, 4, 8)$ for the first chunk and $(3, 7, 11)$ for subsequent chunks. Inference uses the same representative indices. Each selected frame supplies the projected hull for its corresponding latent query.

B.7 Score Noise Schedule

The score noise level is sampled from the interval between the selected denoising step and its next scheduler node, then transformed by the same timestep warping used during generation. One level

Table 8: **Re-entry evaluation set.** Counts and construction settings refer to the final evaluation manifest. Interruption duration is the number of consecutive zero-projection frames divided by 10 Hz.

Item	Specification
Distinct source scenes	60
Target classes	50 cars, 5 trucks, 4 buses, 1 construction vehicle
Clip duration and timestamps	81 frames at 10 Hz, timestamps from 0 to 8 s
Target editing rules	Position, speed, route, and optional joint time resampling
Preserved geometry	Target identity and dimensions, road map, rigid camera-to-ego transform
Source and return selection	Fixed geometry-selected pair, pilot frames 0 and 80
Visibility interruption	30–63 fully offscreen frames (3.0–6.3 s)
Sampling seeds per method	One fixed seed, one sequence per case

is shared across the three latent frames of a target chunk. Gaussian noise is freshly sampled, but its scale remains conditional on the selected denoising step s.

C Evaluation Protocols

C.1 Re-entry Set Construction

The re-entry evaluation is derived from nuScenes scenes and their structured annotations. Each case specifies a source scene, a target identity, camera poses, scene controls, and source and return observations. The edits construct exit-and-return events through scene controls, including target motion and, where required by the case recipe, the ego trajectory or timing. Generated appearance is evaluated against the method's own earlier observation of that target.

We edit the target trajectory and, when needed, the ego trajectory to create an exit followed by a return. Case recipes adjust initial positions, speeds, or lane-following paths while retaining the target identity, vehicle dimensions, road geometry, and rigid camera-to-ego transform. Other traffic is retained by default. Actors that obstruct the selected observations or conflict with edited trajectories can be moved or removed, and some early cases use sparse traffic. Where temporal resampling is used, the same monotonic time mapping is applied to the ego, camera, and retained actors before controls are rendered. Other cases use contiguous native-time windows. Figure 4 illustrates the resulting geometry at the source, interruption, and return of one case. Original sensor RGB remains a reference to the source scene, not ground-truth imagery of the edited event.

Observation times are fixed from the edited geometry before model inference and shared across methods. For the 59 non-pilot cases, we identify the sole interval in which the target cuboid projects to zero pixels in the front view. On either side, we select the frame with the largest visible target area under depth-tested cuboid rendering, breaking ties in favor of the earlier frame. The pilot retains its predefined observations at frames 0 and 80. Visibility-interruption statistics for the selected cases are summarized in Table 8.

C.2 Re-entry Appearance Consistency

RAC adapts object-level appearance measurement to a controlled re-entry. DrivingGen measures DINOv3 similarity on tracked agent crops (Zhou et al., 2026), while MemoBench evaluates object consistency after disappearance and reappearance (Chen et al., 2026). We use the known target geometry to associate observations at predefined source and return times, comparing each method's own generated appearance. Equation 11 averages DINOv3 feature similarity over valid source–return pairs. Each case contributes one pair, and both matched detections must yield usable normalized embeddings. We extract features from the matched detections rather than the entire projected region. We use GroundingDINO-base with box and text thresholds of 0.25 and non-maximum suppression at IoU 0.60. A detection is eligible for a projected target box if their IoU is at least 0.20 or their center distance, divided by the projected box diagonal, is at most 0.50. Eligible matches are assigned greedily in descending order of $\text{IoU} + \max(0, 0.50 - d)$, where d is this normalized center distance,

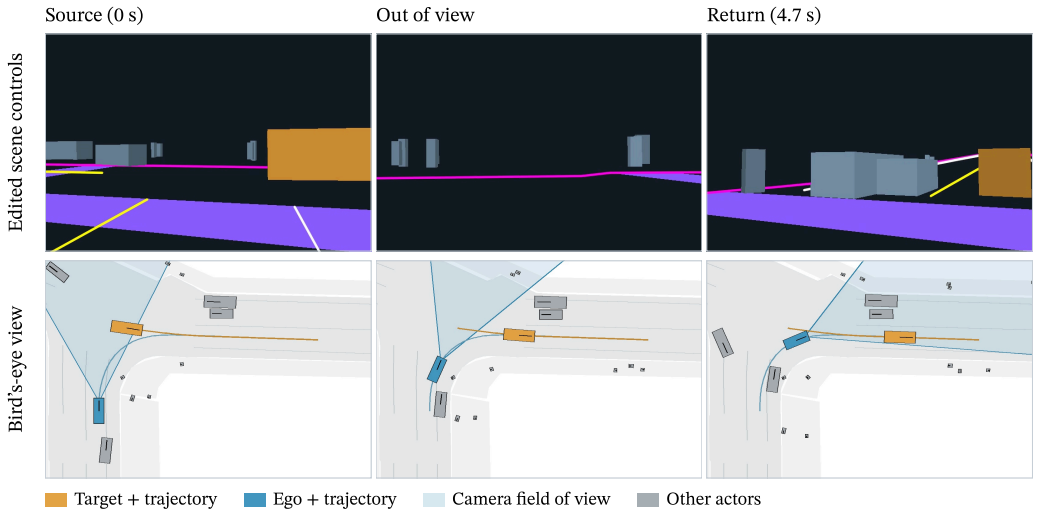


Figure 4: **Constructing a controlled re-entry case.** Edited front-view geometry (top) and the corresponding bird's-eye view (bottom) show the target at the source observation (0 s), outside the camera view (2.6 s), and after its return (4.7 s). In the bird's-eye view, orange marks the target and its trajectory, while blue marks the ego vehicle and horizontal camera sector. These are rendered scene controls, not generated video or edited RGB ground truth.

Table 9: **Valid observations for re-entry evaluation.** Counts are out of 60 cases. RAC uses valid source–return pairs. Presence counts matched target detections at return.

Method	Memory reads	Valid pairs	Return presence
MagicDrive-V2	—	31	33
UniMLVG	—	48	48
DreamForge-DiT	—	48	49
X-Scene	—	53	58
GT-reference DMD	Off	45	53
GT-reference DMD	On	49	56
ReDreamer	Off	42	50
ReDreamer	On	42	52

enforcing one-to-one assignments. For feature extraction, we expand each matched detection by 12% of its original width and height on each side, enforce a minimum crop width and height of 12 pixels, and clip to the image boundary. The minimum size enlarges the crop, rather than serving as a detection rejection threshold.

Table 9 reports valid-pair and return-presence counts. RAC averages valid pairs only. Presence records a matched target detection at return. A valid pair additionally requires a usable source observation.

The evaluated checkpoint and appearance encoder were selected during evaluation development. The reported results therefore characterize the selected system on this evaluation set rather than an independent confirmation set.

C.3 Data, Generation, and Scoring

For quantitative re-entry evaluation, we generate 81-frame sequences at 320×480 using one fixed seed. Both training variants use their generator EMA weights, four denoising steps, and CFG 0. Table 6 summarizes the generation configuration.

Normal reads generated appearance memory. Memory off uses the same weights, observation encoding, and writing procedure, with memory reads disabled.

Table 10: **External baseline generation settings on 150 nuScenes validation scenes.** Dimensions are width $\times$ height per view. All five implementations generate six views without ground-truth RGB initialization. Scoring uses the first 16 front-view frames. Steps refer to video sampling, with image bootstrapping described below.

Method	Evaluated weights	Native output	Steps / CFG
MagicDrive v1	Released 60-frame, 80k checkpoint	400×224, 12 Hz, 61 frames	20 / 2
MagicDrive-V2	Stage-3 EMA, CogVideoX VAE and T5	1600×900, 12 Hz, 17 frames	30 / 2
UniMLVG	OpenDWM public 60k checkpoint	448×256, 10 Hz, 35 frames	50 / 3
DreamForge-DiT	Released DiT-s and DiT-t EMA	800×448, 12 Hz, 17 slots	30 / 2
X-Scene	Released non-occluded image/video models	400×224, 12 Hz, 19 frames	20 / 1.2

Table 11: **Video-quality scoring protocol.** The final protocol fixes both the generated cohort and the transformation used to extract features.

Item	Specification
Scene / window selection	150 nuScenes validation scenes, FRONT, one generated draw per scene
Native outputs	Table 10, ReDreamer: 480×320, 10 Hz
Temporal alignment	First 16 native frames, native 10/12 Hz retained
Spatial alignment	Full front-view frames, method-specific FOV and GT processing retained, metric-specific resizing below
FID	`pytorch-fid` 0.3.0, full frame, Inception pool3, 2048 dimensions, 150 scenes $\times$ 16 frames per side, FP64 statistics
FVD	Original LVDM FP32, full-frame bilinear resize to 224×224, 400-dimensional I3D features, 150 clips $\times$ 16 frames per side
VBench	Official IQ (MUSIQ) and MS (AMT-S), the same 150 front-view clips, 16 native frames per clip, equal weight per clip, displayed as percentages

RAC and its aggregation rules are specified in Appendix C.2.

C.4 External Baselines and Video Quality

Table 10 lists generation settings for the 150 nuScenes validation scenes. We retain each implementation's text, map, object-box, and camera inputs and evaluate its front view. UniMLVG uses the OpenDWM implementation. X-Scene uses the author's nearest-keyframe VLM descriptions for video-quality evaluation and case-local edited captions for re-entry evaluation, with the original CLIP/T5 templates. Its re-entry outputs are mapped to the 336×224 evaluation plane by a fixed case-specific homography and sampled at the nearest timestamps to the common 10 Hz timeline.

MagicDrive v1 uses UniPC with double CFG. MagicDrive-V2 retains the official 1600×900 padded canvas around 1600×848 content. DreamForge uses a generated DiT-s image to initialize DiT-t. Its official 17-slot ordering repeats the initial token, which we retain. UniMLVG starts without image latents and subsequently reuses its own three-frame overlap. X-Scene bootstraps from its image model.

Table 11 summarizes scoring. FID uses full-frame `pytorch-fid` 0.3.0 with 2048-dimensional Inception features and internal bilinear resizing to 299×299 (Heusel et al., 2017). We extract features in FP32 with TF32 disabled and compute means, unbiased covariances, and distances in FP64, without an additional crop. FVD follows the original LVDM implementation: complete frames are bilinearly resized to 224×224, normalized to $[-1, 1]$, and encoded into 400-dimensional I3D features. Extraction and distance computation use FP32 with TF32 disabled (He et al., 2023; Unterthiner et al.,

2019). Native frame rates and ground-truth preprocessing are retained, so equal frame counts do not guarantee equal durations or fields of view.

VBench evaluates losslessly encoded versions of the same front-view clips (Huang et al., 2024). Imaging quality uses MUSIQ with the official longer-side preprocessing: images larger than 512 pixels on that side are resized proportionally. Scores are averaged over frames, then equally over clips. Motion smoothness uses AMT-S interpolation between even-indexed frames and evaluates odd-indexed frames 1 through 13 as $1 - \mathrm{MAE}/255$, followed by equal averaging over clips. Both dimensions are displayed as percentages without leaderboard min–max normalization.

C.5 Memory-Conditioned Training Comparison

Both training procedures start from the same consistency-distilled four-step generator. GT-reference DMD keeps dataset references available during student rollout. ReDreamer instead starts each rollout with an empty bank and makes a record available only after its source chunk has been fully generated. The GT-reference baseline uses interleaved rolling denoising, whereas ReDreamer completes each chunk before writing its memory, as detailed in Appendix B.4.

ReDreamer adapts its fake score under self-generated references before joint DMD. Both scores condition on the generated records without classifier-free guidance. The comparison evaluates the complete training recipes, including their rollout and score-conditioning procedures. Both generators read self-generated memory at evaluation.

Qualitative examples. Each method supplies its own generated history. The selected examples in Figure 3 retain the original ReDreamer inference runs, whereas the aggregate memory-enabled scores use corrected temporal routing. Selected examples illustrate individual outcomes rather than aggregate success rates.

C.6 Runtime Measurement Details

The external times in Table 2 are averages from existing inference logs. MagicDrive-V2, DreamForge, and X-Scene each use five runs producing six views: respectively 17 frames (16 output frames), 17 slots (16 output frames), and 19 frames. Their recorded intervals cover sampling through latent saving, resident image/video generation through output saving, and image bootstrapping through video saving and readback, respectively. UniMLVG uses 79 six-view, 81-frame runs and includes sampling and final decoding. Each external run uses one PPU. ReDreamer's time uses 81 front-view frames on one H20. These entries describe the observed generation configurations. The controlled reading-overhead comparison uses the common setup below.

Measurements use the selected generator EMA weights. Reads-off retains observation encoding and memory writes. Each setting uses six fixed cases after one excluded warm-up on a single NVIDIA H20. The session and random state are reset for each measured case. Measurements use BF16 precision, batch size 1, four denoising steps, and CFG 0. Each sequence contains 81 frames: 9 in the first chunk, followed by six chunks of 12.

FPS is total generated frames divided by total pipeline time. First-chunk latency is the mean time until decoded chunk-0 RGB is available. The interval includes control encoding, generation, video decoding, historical-cache updates, and object-memory encoding, reading, and writing. It excludes checkpoint loading, disk input, host transfer, and MP4 encoding. Peak memory denotes PyTorch allocated memory.

D Additional Results

D.1 Additional Qualitative Results

Figure 5 presents additional comparisons of appearance retention with external driving generators. Each panel includes the source observation needed to interpret its return appearance.

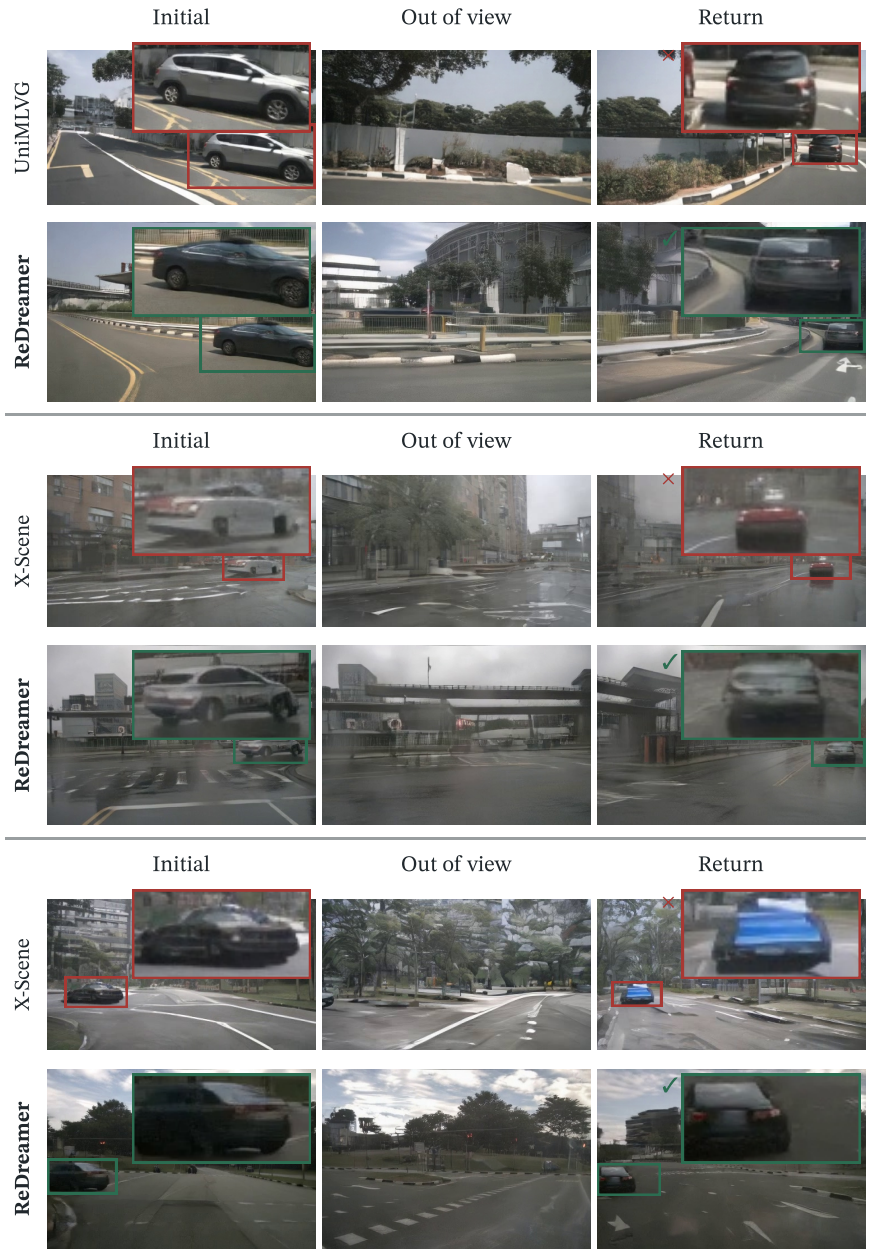


Figure 5: **Additional re-entry comparisons.** Three selected cases compare ReDreamer with UniMLVG (top) and X-Scene (middle and bottom). Each row shows that method's own generated source, an out-of-view observation, and the returned target. Insets enlarge the marked vehicle regions. Green checks and red crosses indicate coarse appearance retention and visible changes, respectively. ReDreamer retains the broad vehicle appearance in these examples, while the baselines change color or body shape.



Figure 6: **Failure cases in controlled re-entry.** Source, out-of-view, and scheduled return observations are shown with memory reads enabled (EMA checkpoints). (a) At step 600, the car changes from reddish-brown at 1.8 s to gray at 8.0 s, with an intermediate observation at 4.5 s. Orange boxes mark the enlarged regions. (b) The same night-scene controls are generated at steps 600 and 400 and shown at 0, 3.5, and 6.9 s. Step 600 develops severe water-like distortions, whereas step 400 retains recognizable scene structure. Each sequence contains 81 frames at 10 Hz.

All methods use the same seed and separate generated histories. ReDreamer uses the corrected temporal routing used for the quantitative evaluation. X-Scene panels retain the native image plane and use the nearest native timestamps. Inset magnification varies with target size.

D.2 Failure Cases and Analysis

Figure 6 illustrates appearance changes and image degradation in ReDreamer. In the first case, a reddish-brown car returns with a gray appearance despite memory reading being enabled. In the night-scene example, water-like distortions spread across the scene and obscure its structure by the return observation.

To examine the checkpoint dependence of this degradation, we repeat the night-scene generation with the earlier step-400 checkpoint, keeping the scene controls, random seed, and inference settings fixed. The step-400 sequence does not exhibit the pronounced water-like distortions seen at step 600. This single-case comparison supports checkpoint sensitivity of the image degradation, but it does not establish a general failure rate or isolate memory reading as its cause.

E Limitations and Future Directions

Real-score generalization. The frozen real-score model is trained on real videos paired with real-image references, but during DMD it evaluates student samples conditioned on generated memory. Training the generator and fake-score model on these records does not adapt the real-score model itself. The distillation signal therefore depends on the real-score model generalizing to this changed reference distribution. Improving the teacher's reliability under generated-memory conditions, while preserving its role as a model of real-video quality, is an important direction for future work.

Evolving appearance and memory capacity. Our memory retains a fixed appearance record for each registered object. It does not explicitly update that record to represent subsequent changes in the object's appearance. Incorporating new observations while preserving identity would extend the current design to evolving object states. Storage also grows as previously unseen objects are registered. Sustained operation under a fixed memory budget would require retention, compression, or eviction policies that balance resource use against future re-entry needs.