LEARNING TO REMEMBER AND EVOLVE: OBJECT STATE MEMORY FOR VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review

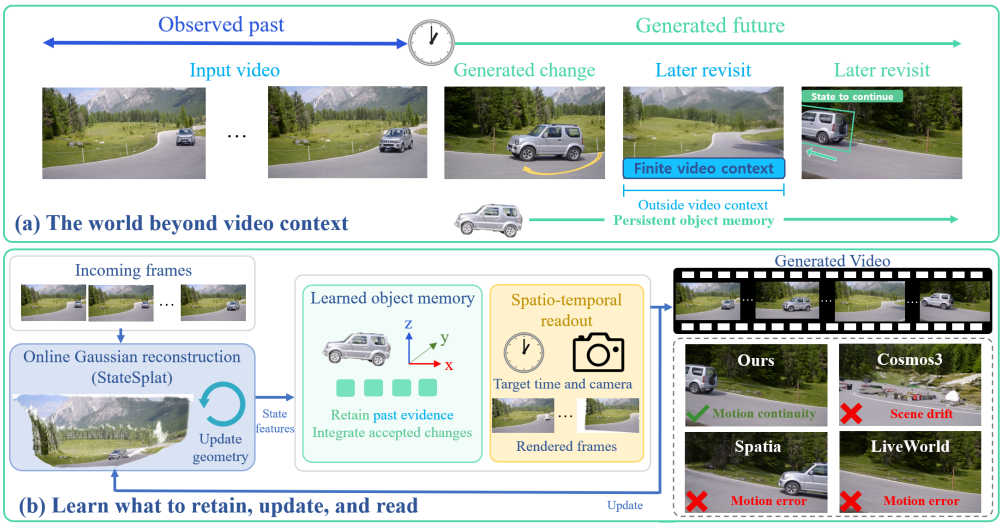

Figure 1: **Persistent world generation beyond the video context. (a)** Generated changes should persist across camera movements and gaps in object visibility. **(b)** Our framework connects online Gaussian reconstruction, learned object memory, and spatiotemporal conditioning. Accepted generated evidence updates the maintained state for subsequent generation. **Right:** Qualitative comparison of object reappearance, motion progression, and scene consistency at revisit.

ABSTRACT

Camera- and text-controlled video world models require persistent scene information to connect generated changes across viewpoints and beyond recent video context. We propose a framework that connects online feed-forward Gaussian reconstruction, persistent state maintenance, and spatiotemporal conditioning in a recurrent generation pipeline. StateSplat provides renderable geometry and object-level visual evidence, which learned memory integrates alongside explicit state records. Observed and accepted generated records remain distinct from future predictions. Time- and camera-dependent readout supplies memory to the generator through temporal tokens and spatial feature injection, while admitted generated frames provide evidence for subsequent updates. Video generation and subsequent-observation reconstruction objectives jointly train feature adaptation, memory updating, and readout. Experiments on CARLA and DAVIS show improved object reappearance and reference-based continuation quality, with ablations supporting the combined memory configuration and spatial conditioning.

1 INTRODUCTION

Camera and text-controlled video world models aim to continuously generate future video from limited observations. For a persistently explorable dynamic world, producing plausible video segments is not sufficient. Previously explored scenes should remain recognizable, and changes depicted during generation should carry forward into subsequent views. When the camera turns away from a moving object and later returns, the model must recover the surrounding scene while accounting

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.

for what has happened to the object. Long-horizon world generation therefore requires a persistent representation that connects spatial structure, evolving content, and future synthesis beyond the generator's recent video context.

A central challenge is converting video content into information that can be retained, updated, and reused. This requires relating incoming observations to previously seen scenes and objects despite changes in viewpoint. Geometric representations provide a common spatial reference for this correspondence. However, reconstruction methods that jointly process a fixed set of input images are not directly suited to generation with continually arriving frames. New evidence should be integrated incrementally while retaining the existing representation, rather than repeatedly processing the accumulated inputs. Even with online integration, spatial correspondence alone does not determine which information to preserve or revise as scenes and objects change over time. Incorporating generated frames further complicates this process because new estimates may reflect both depicted changes and errors in reconstruction or object association. Persistent world generation therefore requires incremental spatial reconstruction, maintenance of evolving information, and conversion of that information into generation conditions at target times and viewpoints.

Recent methods establish important foundations for this process. GEN3C Ren et al. (2025) conditions video generation on renderings of a 3D cache built from observed and generated frames. NeoVerse Yang et al. (2026b) and Vista4D Lin et al. (2026) incorporate dynamic scene representations through 4D reconstruction and point clouds, respectively. SpMem Wu et al. (2025) and Spatia Zhao et al. (2026) extend the available context through explicit spatial memory. Together, these approaches demonstrate how geometric representations can connect video generation to previously explored space. Our focus is on coupling this spatial foundation with the maintenance of evolving scene information and its time-dependent use in generation.

Dynamic persistence has also been addressed through continued simulation and explicit planning. LiveWorld Duan et al. (2026) generates out-of-sight events through virtual monitors and reconstructs their outputs into 4D dynamic memory. WorldDirector Wang et al. (2026a) converts language-model-planned object trajectories into spatial conditions combined with appearance information. These mechanisms highlight the importance of carrying dynamic content beyond the current observer view. However, preserving spatial information or supplying future trajectories does not by itself train reconstruction-derived visual features to retain past evidence useful for subsequent generation. We address this problem by jointly learning geometry-grounded memory updates and target-dependent spatiotemporal readout through generation supervision.

We propose a framework that connects online feed-forward Gaussian reconstruction, persistent state maintenance, and spatiotemporal video conditioning in a recurrent generation pipeline. Incoming observations are reconstructed into a common 3D scene representation and provide visual evidence for memory. The maintained geometry and memory are then queried at target times and camera viewpoints to condition the next video segment. Selected generated frames pass through admission checks before contributing new geometric estimates, state records, and visual evidence. This loop makes reconstruction the interface between video and persistent scene information, while learned memory and readout connect that information to future synthesis.

The pipeline begins with StateSplat, which combines a frozen geometric backbone with a learned reconstruction Transformer and Gaussian head. Backbone geometry provides initial 3D points, and the learned reconstruction pathway refines their positions and predicts Gaussian attributes. Background geometry is accumulated online, while each tracked object is represented by local reference geometry and a time-varying pose. Subsequent observations provide geometry for pose estimation without replacing the retained object reference. This separation supports scene rendering while allowing object placement to change over time. Reconstruction features additionally provide object-level visual evidence, linking the geometric representation to learned memory.

Persistent state maintenance combines explicit records with a learned representation of that evidence. State records describe estimated object placement and its provenance, while object memory integrates geometry-grounded features across observations. An adapted feature branch and a gated memory updater use new evidence together with its source, quality, and timing to construct the memory representation. Observed records and accepted generated records remain distinguishable, and future predictions are kept separate from recorded history. This organization provides a basis

for revising generated estimates while retaining information that is absent from the current video context.

To make this information available to the generator, we read memory according to the target time and camera. Retrieved evidence is combined with explicit state encodings to form time-specific tokens. These tokens enter the video backbone through temporally aligned attention and spatial feature injection guided by object projections, visibility, and uncertainty, alongside scene renderings and recent video history. We jointly train the adapted feature branch, memory updater, and readout pathway through video generation and subsequent-observation reconstruction objectives. The resulting formulation learns how to retain reconstruction-derived evidence and make it available for synthesis, rather than treating reconstruction, memory construction, and conditioning as independent stages.

Our main contributions are as follows:

- **An online reconstruction–memory–generation framework.** We connect feed-forward Gaussian reconstruction with persistent scene information and recurrent video generation. StateSplat provides renderable geometry and reconstruction-derived evidence, while accepted generated frames supply new inputs to the pipeline.
- **Geometry-grounded state and memory maintenance.** We combine explicit object-state records with learned visual memory to retain and update information beyond the recent video context. Evidence integration uses geometric localization and source, quality, and temporal metadata, while recorded observations and generated estimates remain separate from future predictions.
- **Jointly learned spatiotemporal readout and conditioning.** We combine time-specific memory readout with spatial feature injection to condition generation at target times and viewpoints. Video generation and subsequent-observation reconstruction objectives jointly train feature adaptation, memory updating, and readout.

2 Related Work

Geometry-grounded video generation. Feed-forward Gaussian reconstruction recovers scenes from sparse images, while recurrent and streaming methods integrate incoming observations (Charatan et al., 2024; Chen et al., 2024; Wang et al., 2025; Li et al., 2025b; Wu et al., 2026). These representations provide a spatial reference for view-consistent generation. GEN3C conditions video generation on a 3D cache built from input and generated frames (Ren et al., 2025), while NeoVerse and Vista4D use 4D reconstruction and point clouds for synthesis along novel camera trajectories (Yang et al., 2026b; Lin et al., 2026). Our work connects online Gaussian reconstruction to learned object memory, using geometric features as evidence for maintaining and updating object states during generation.

Persistent memory and evolving worlds. Historical-view retrieval and explicit spatial memory extend video generation beyond its recent context (Xiao et al., 2026; Yu et al., 2025; Li et al., 2025a; Wu et al., 2025; Zhao et al., 2026). Other approaches learn context queries, store diffusion features in 3D, or evolve persistent latent scenes (Yu et al., 2026; Wang et al., 2026b; Garcin et al., 2026). Dynamic extensions include motion-relevant retrieval in HyDRA, off-screen event generation and 4D memory in LiveWorld, and trajectory-based object control in WorldDirector (Chen et al., 2026; Duan et al., 2026; Wang et al., 2026a). TeleWorld connects generation with reconstruction, while PermaVid revises memory following edits (Chen et al., 2025; Yang et al., 2026a). Within this broader effort, we study how object memory integrates incomplete observations and potentially erroneous generated evidence, retaining provenance and distinguishing admitted records from future predictions.

Object-centric state and spatiotemporal conditioning. SAVi learns temporally associated object representations, while G-SWM and SlotFormer study object-level temporal prediction (Kipf et al., 2021; Lin et al., 2020; Wu et al., 2022). Complementary work controls video generation through camera motion, boxes, and trajectories (Wang et al., 2024b;a; Zhang et al., 2025; Xiao et al., 2025). These directions motivate connecting persistent object representations to explicit generation conditions. We jointly learn object-memory updates and target-dependent readout, supplying the resulting information through time-aligned attention and spatial feature injection.

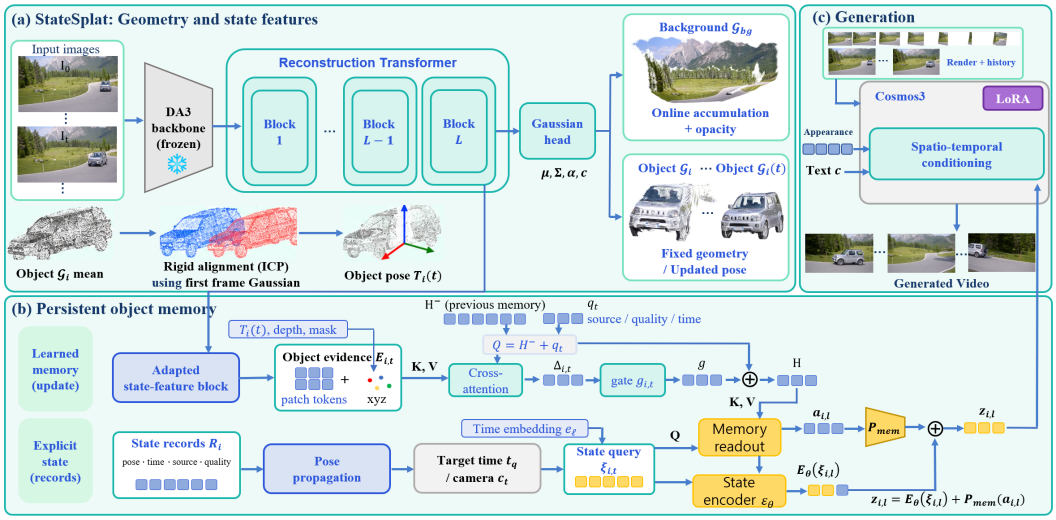

Figure 2: **Overview of our framework. (a) StateSplat** reconstructs background and object Gaussians, estimates object poses, and extracts visual evidence. **(b) Persistent object memory** integrates evidence through gated attention and combines target-dependent memory readout with explicit state encoding. **(c) Generation** uses the resulting state tokens and appearance references to condition LoRA-adapted Cosmos3, together with scene renderings, recent video, and text.

3 METHOD

3.1 OVERVIEW

Given an observed video, a target camera trajectory, and a text condition, we aim to generate future video consistent with object states established through observations and earlier generations, beyond the generator's recent video context. Our framework follows three stages (Fig. 2): (a) StateSplat constructs an online Gaussian scene and extracts object evidence; (b) explicit pose records and learned visual memory produce target-dependent object tokens; and (c) these tokens condition video generation through temporal attention and spatial injection. Accepted observations from generated frames update the records and evidence logs for subsequent segments.

3.2 STATESPLAT: GEOMETRY AND STATE FEATURES

Learned reconstruction and online state. StateSplat combines a frozen DA3 (Lin et al., 2025) backbone with a reconstruction Transformer and Gaussian head trained on RE10K. DA3 geometry initializes 3D points, which the learned pathway refines as $\boldsymbol{\mu} = \boldsymbol{\mu}^{\text{base}} + \Delta\boldsymbol{\mu}$, while predicting covariance $\boldsymbol{\Sigma}$, opacity α, and appearance $\mathbf{c}$. The resulting Gaussians retain scene geometry beyond the recent video context, providing a common spatial reference for rendering and associating later observations.

We separate background Gaussians $\mathcal{G}_{\text{bg}}$ from object-local references $\{\mathcal{G}_i\}$. Background reconstructions accumulate online, with a learned retention gate modulating their opacities at rendering time. Each object's first admitted observation initializes $\mathcal{G}_i$, which remains fixed thereafter. Subsequent observations still provide new object geometry: their masked depth points are rigidly aligned with the reference means to estimate the object-to-world pose $T_i(t) = [R_i(t), \mathbf{p}_i(t)]$. This separates persistent object shape from changing placement, allowing new observations to revise pose against a consistent geometric reference.

Object evidence. An adapted state-feature block extracts object patch features from the reconstruction Transformer. Each feature is paired with an object-local coordinate, obtained by depth back-projection followed by $T_i(t)^{-1}$:

$$E_{i,t} = \{(\mathbf{f}_{j,t}, \mathbf{x}^o_{j,t}) \mid j \in \Omega_{i,t}\}, \tag{1}$$

where $\Omega_{i,t}$ denotes mask-selected patches. The image-derived features carry visible appearance and structure; local coordinates associate them with the object's reference frame across observations. This evidence complements the explicit pose estimates used by the memory stage.

3.3 Persistent Object Memory

Records and selective memory update. Pose records describe where an object was estimated to be, while visual memory retains what was observed about it. For each object, $\mathcal{R}_i$ stores admitted poses, timestamps, provenance, and quality. A translational velocity estimate from the pose history supports constant-velocity propagation. Records and evidence logs persist across segments; memory tokens H_i are rebuilt by chronologically replaying available evidence.

For each evidence item, patch features and Fourier encodings of their local coordinates form attention keys $K_{i,t}$ and values $V_{i,t}$. With previous memory H_i^- and a metadata embedding $q_{i,t}$ encoding source, quality, and elapsed time, we compute

$$\begin{aligned} \Delta_{i,t} &= \mathrm{Attn}(H_i^- + q_{i,t}, K_{i,t}, V_{i,t}), \\ g_{i,t} &= \sigma\big(\mathrm{MLP}[H_i^-, \Delta_{i,t}, q_{i,t}]\big), \\ H_i^+ &= H_i^- + g_{i,t} \odot \Delta_{i,t}. \end{aligned} \tag{2}$$

Here and below, attention arguments denote queries, keys, and values. The residual path preserves accumulated information, while the gate regulates revisions from partial or potentially inconsistent evidence using its content and metadata. Admission selects evidence for storage; gating controls its contribution to memory. Without admitted evidence, no update is applied.

Target-dependent readout. Memory summarizes past observations, but generation requires object information at a requested future time and viewpoint. We therefore propagate the recorded pose and construct $\xi_{i,\ell}$ from the target-camera-relative state, temporal information, provenance, and validity. Propagation is an extrapolation, not a new observation. The query retrieves visual memory and combines it with an explicit state encoding:

$$\begin{aligned} a_{i,\ell} &= \mathrm{Attn}(e_\ell + W_s \xi_{i,\ell}, H_i, H_i), \\ z_{i,\ell} &= E_\theta(\xi_{i,\ell}) + P_{\mathrm{mem}}(a_{i,\ell}), \end{aligned} \tag{3}$$

where e_ℓ embeds target time. The first term supplies geometric and temporal state; the second supplies query-selected visual information. These tokens are constructed before video generation.

3.4 Spatiotemporal Conditioning and Generation

Generation uses recent video history, scene renderings, and text, together with object-state tokens. Learned queries additionally pool DA3 features from each object's latest real observation into appearance tokens A_i, providing a direct observed reference.

Temporal and spatial conditioning. At selected generator layers, video features X_ℓ query the object conditions $U_\ell = \mathrm{Concat}_i[z_{i,\ell}, A_i]$:

$$X_\ell \leftarrow X_\ell + P_{\mathrm{attn}}\,\mathrm{Attn}(X_\ell, U_\ell, U_\ell). \tag{4}$$

Each latent time accesses its corresponding state tokens, while appearance references are shared across times. This supplies object information without mixing placements predicted for different times.

To additionally specify where that information applies, we place state features on the target latent grid:

$$\begin{aligned} F_\ell(\mathbf{u}) &= \sum_i w_{i,\ell}(\mathbf{u})\phi_\theta(z_{i,\ell}), \\ X_\ell(\mathbf{u}) &\leftarrow X_\ell(\mathbf{u}) + P_{\mathrm{sp}} F_\ell(\mathbf{u}). \end{aligned} \tag{5}$$

The weights use projected object centers and covariances, visibility, and uncertainty. Cross-attention provides content-dependent retrieval; spatial injection provides explicit spatial support. Both contribute learned residuals inside the generator.

Generation feedback. Conditions use only evidence available before the newly generated interval. Reusing generated observations can carry synthesis or association errors into later state estimates. We therefore screen candidates using background, silhouette, and appearance consistency before extending the records and evidence logs, retaining their generated provenance. Rejected or missing observations add no evidence; existing records remain available for propagation. Appearance references continue to use real observations only.

3.5 Training Objectives

Following StateSplat reconstruction pretraining, we jointly train feature adaptation, memory, and conditioning modules with LoRA (Hu et al., 2021) adaptation of the video generator. The objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{fm}} + \lambda_{\mathrm{obj}}\mathcal{L}_{\mathrm{obj}} + \lambda_{\mathrm{aux}}\mathcal{L}_{\mathrm{aux}}. \tag{6}$$

$\mathcal{L}_{\mathrm{fm}}$ is the flow-matching loss over newly generated latent elements, excluding conditioning-history frames. $\mathcal{L}_{\mathrm{obj}}$ evaluates the same target within object-mask regions, increasing the contribution of small objects without changing the generation target. Target masks supervise the loss and are not supplied as future observations at inference.

The auxiliary loss predicts object RGB and depth along sampled rays from memory *before* the corresponding evidence is incorporated. It supervises retention of reconstructable appearance and geometry rather than immediate reconstruction after a memory write. Its decoder is used only during training.

Reappearance windows are selected so that the object is absent from the recent conditioning history but visible in the target, with earlier evidence available to memory. They expose the model to recovery beyond visible continuation and are mixed with other dynamic, parked-object, and background windows. Detailed loss definitions and additional method details are provided in Appendix A.

4 Experiments

4.1 Implementation Details

We adapt pretrained Cosmos3-Nano with LoRA and jointly train object-memory and conditioning modules, initializing the feature branch from StateSplat. Joint training uses a single NVIDIA H100 GPU. Training includes reappearance windows with earlier object evidence but no target in recent history. Implementation and training details appear in Appendix A.4; datasets, baselines, and evaluation protocols are described in Appendix B.

4.2 Comparisons

4.2.1 Quantitative Evaluation

CARLA. Table 1 reports development-set results. On the common interval across three clips, our method detects the target in 203 of 206 eligible frames (0.99), compared with 0.62 for GEN3C Ren et al. (2025), the strongest baseline by detection rate on the same clips. Our conditional localization error is 0.12, compared with 0.43 for GEN3C Ren et al. (2025). On the return interval, our method retains a detection rate of 0.95 (203/214), compared with 0.53 for NeoVerse Yang et al. (2026b)and 0.05 for Vista4D, all with full temporal coverage. The corresponding conditional localization errors are 0.92, 1.06, and 1.00, respectively. These results support improved target reappearance and localization on the evaluated clips, but do not establish identity preservation or causal use of memory. The non-target detection occurrence rate is 0.31 during the absence phases, averaged across clips. Partial-coverage methods and the single-clip, event-disabled LiveWorld Duan et al. (2026) diagnostic are reported separately and are not treated as matched comparisons.

DAVIS. On the ten held-out continuation sequences (Table 2), our method evaluates all 715 target frames and achieves 15.1 PSNR, 0.38 SSIM, 0.46 LPIPS, and 0.47 Object-crop LPIPS. Compared with NeoVerse Yang et al. (2026b), which evaluates the same number of sequences and frames, this improves PSNR by 0.8 dB and reduces LPIPS and Object-crop LPIPS by 0.06 each. Our method also obtains better reported image-quality scores on the development split, which is kept separate from

Table 1: CARLA dev comparison on three shared clips; LiveWorld Duan et al. (2026) is a separate single-clip diagnostic. Common intervals use intersecting generated frames. Return intervals follow full-coverage methods (NeoVerse Yang et al. (2026b), Vista4D, Ours); partial outputs are scored on available subsets with coverage reported. Detection is pooled over frames. Localization error averages per-clip medians, normalized by target size, on detected reference-visible frames.

Method	Detection ↑	Loc. error ↓	Detected / eligible	Non-target det. ↓	Coverage	Eligible episodes
Common generated interval						
GEN3C Ren et al. (2025)†	0.62	0.43	127 / 206	0.40	1.00	3 / 3
NeoVerse Yang et al. (2026b)†	0.54	0.78	111 / 206	0.45	1.00	3 / 3
Vista4D†	0.52	0.55	107 / 206	0.52	1.00	3 / 3
Spatia Zhao et al. (2026)	0.52	0.57	107 / 206	0.60	1.00	3 / 3
LiveWorld Duan et al. (2026)†	0.30	0.30	25 / 82	0.69	1.00	1 / 1
Ours	**0.99**	**0.12**	**203 / 206**	0.35	1.00	3 / 3
Long-horizon return interval						
GEN3C Ren et al. (2025)†	0.00	N/A	0 / 54	0.23	0.83	1 / 3
NeoVerse Yang et al. (2026b)†	0.53	1.06	114 / 214	0.43	1.00	3 / 3
Vista4D†	0.05	1.00	10 / 214	0.63	1.00	3 / 3
Spatia Zhao et al. (2026)	0.08	1.50	7 / 89	0.84	0.60	2 / 3
LiveWorld Duan et al. (2026)†	0.17	1.54	4 / 24	0.93	0.29	1 / 1
Ours	**0.95**	**0.92**	**203 / 214**	**0.31**	1.00	3 / 3

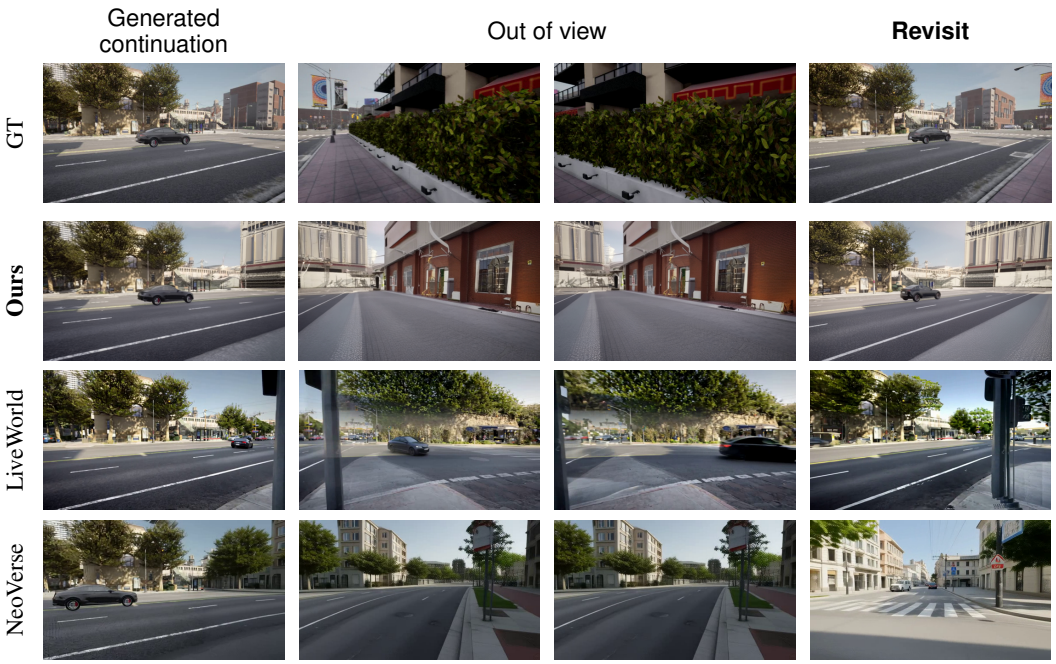

Figure 3: Qualitative comparison across departure and revisit. Columns show matched absolute timestamps, with temporal phases defined by the reference sequence. GT provides the reference scene and object placement; generated rows allow comparison of object presence, placement, and background consistency.

the held-out evaluation. Spatia Zhao et al. (2026) covers nine test sequences and LiveWorld Duan et al. (2026)has incomplete temporal coverage, so their aggregate scores are interpreted with these differences. The continuation results indicate improved agreement with the reference video under the supplied camera trajectory; they do not directly measure consistency along unobserved departure-and-return trajectories. In (Table 3), our method achieves Identity and Continuity scores of 0.76 and 0.56, compared with 0.56 and 0.36 for NeoVerse Yang et al. (2026b), and the lowest Artifacts score of 0.15. These measures assess appearance-based reappearance and consistency with extrapolated observed motion, rather than ground-truth future trajectories.

Table 2: DAVIS continuation from 13 observed frames using a shared DA3 camera trajectory estimated from the full reference video. Future RGB and masks otherwise serve only for scoring. Frames, Cov., and Seq. report evaluated frames, temporal coverage, and sequence counts. Development and held-out test results are separated. Object-crop LPIPS uses reference-mask bounding boxes. Evaluation intervals and aggregation are detailed in Appendix B.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Object-crop LPIPS ↓	Frames	Cov.	Seq.
Unused test sequences (10)							
GEN3C Ren et al. (2025)	12.8	0.25	0.63	0.61	715	1.00	10 / 10
NeoVerse Yang et al. (2026b)	14.3	0.30	0.52	0.53	715	1.00	10 / 10
Vista4D	13.9	0.29	0.55	0.60	715	1.00	10 / 10
Spatia Zhao et al. (2026)	14.9	0.32	0.51	0.53	648	1.00	9 / 10
LiveWorld Duan et al. (2026)	12.9	0.26	0.60	0.63	680	0.95	10 / 10
Ours	15.1	0.38	0.46	0.47	715	1.00	10 / 10
Development sequences (7)							
GEN3C Ren et al. (2025)	11.0	0.24	0.68	0.66	421	1.00	7 / 7
NeoVerse Yang et al. (2026b)	12.2	0.31	0.57	0.58	421	1.00	7 / 7
Vista4D	12.0	0.30	0.60	0.69	117	1.00	2 / 7
Spatia Zhao et al. (2026)	12.7	0.33	0.56	0.57	421	1.00	7 / 7
LiveWorld Duan et al. (2026)	10.9	0.25	0.66	0.68	400	0.95	7 / 7
Ours	13.0	0.34	0.52	0.52	421	1.00	7 / 7

Table 3: DAVIS generation along new departure-and-return trajectories.

Method	Identity ↑	Continuity ↑	Artifacts ↓	Eligible cases
GEN3C Ren et al. (2025)	0.16	0.06	0.35	5 / 5
NeoVerse Yang et al. (2026b)	0.56	0.36	0.20	5 / 5
Vista4D	0.20	0.10	0.30	5 / 5
Spatia Zhao et al. (2026)	0.24	0.10	0.30	5 / 5
LiveWorld Duan et al. (2026)	0.40	0.20	0.25	5 / 5
Ours	0.76	0.56	0.15	5 / 5

4.2.2 Qualitative Evaluation

Figure 3 compares CARLA outputs across departure and revisit at matched timestamps. Our method depicts a vehicle with similar appearance before departure and after return, while retaining the broad road layout. LiveWorld Duan et al. (2026) exhibits different vehicle placement and visibility, whereas NeoVerse Yang et al. (2026b) changes the street layout at revisit and does not recover the reference vehicle in the expected region. Our output nevertheless differs from the reference in intermediate viewpoints and background details, indicating residual scene and camera-path discrepancies.

Figure 4 compares DAVIS continuation with the train remaining visible. Our method retains its overall appearance and the surrounding scene layout, although later motion is less pronounced than in the reference. LiveWorld Duan et al. (2026) exhibits changes in framing and carriage geometry, while NeoVerse Yang et al. (2026b) shows limited motion across the selected frames. Together, these examples distinguish object reappearance from faithful motion progression: recovering an object after absence does not alone establish accurate temporal evolution.

4.3 Analysis and Ablation Study

Memory construction and spatial conditioning. Table 4 shows that the learned memory configuration improves return detection over explicit-state and appearance conditioning, from 0.90 to 0.93 with temporal-only conditioning and from 0.92 to 0.95 with spatial injection. The corresponding conditional localization errors decrease from 1.05 to 0.98 and from 0.96 to 0.92, respectively. These comparisons evaluate the memory updater, feature adaptation, and auxiliary supervision jointly. Spatial injection further reduces localization error under both memory configurations, indicating that spatially localized conditioning complements temporal tokens on the evaluated return intervals.

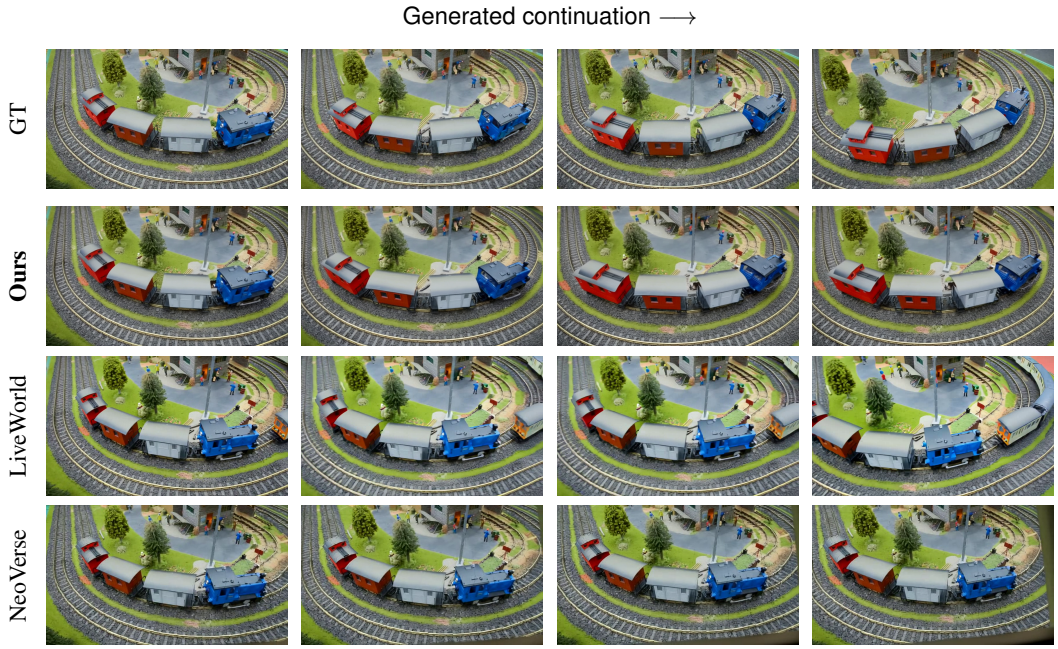

Figure 4: Qualitative comparison on DAVIS continuation. Columns show successive matched timestamps beyond the observed input, with time progressing from left to right. The train remains visible throughout the illustrated interval, allowing comparison of motion progression, object appearance, and background consistency against the reference video.

Table 4: Component ablations under matched training conditions. All variants retain temporal token conditioning. Learned memory jointly includes feature adaptation and auxiliary reconstruction supervision; the final two rows ablate these individually. Return-interval localization error is normalized by target size and conditioned on detection, with detected/eligible frame counts reported.

Variant	Learned update	Spatial injection	Aux. recon. loss	Feature adaptation	Detection ↑	Cond. loc. error ↓	Detected / eligible
Fixed history, temporal-only	No	No	No	No	0.90	1.05	192 / 214
Learned memory, temporal-only	Yes	No	Yes	Yes	0.93	0.98	199 / 214
Fixed history, spatiotemporal	No	Yes	No	No	0.92	0.96	197 / 214
Full model	Yes	Yes	Yes	Yes	**0.95**	**0.92**	**203 / 214**
w/o auxiliary reconstruction loss	Yes	Yes	No	Yes	0.94	0.95	201 / 214
w/o feature adaptation	Yes	Yes	Yes	No	0.94	0.94	201 / 214

Feature adaptation and auxiliary supervision. Removing auxiliary reconstruction supervision increases localization error from 0.92 to 0.95, while removing feature adaptation increases it to 0.94. Both variants reduce detections from 203 to 201 out of 214 eligible frames. These modest differences support retaining both components in the full configuration, although they do not establish that either component alone accounts for the reappearance improvement.

5 Conclusion

We presented a framework combining online Gaussian reconstruction, persistent object memory, and spatiotemporal conditioning for video world generation. Geometry-grounded evidence supports learned memory updates, while time- and camera-dependent readout conditions generation. Experiments on CARLA and DAVIS show improved object reappearance and reference-based continuation quality, with ablations supporting the combined memory configuration and spatial conditioning. Motion progression and responsiveness to explicit state changes remain challenges. Future work will better align generated dynamics with retained object states over longer horizons.

AI USE STATEMENT

In this work, we used generative AI tools to assist with Korean–English translation of manuscript text. Additionally, we used these tools for grammar correction, wording refinement, drafting and revising passages, and organizing the manuscript structure.

The authors reviewed and revised the AI-assisted text, checking that the terminology, technical descriptions, and presentation accurately reflected the intended methods and reported results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467. IEEE, 2024.

Kaijin Chen, Dingkang Liang, Xin Zhou, Yikang Ding, Xiaoqiang Liu, Pengfei Wan, and Xiang Bai. Out of sight but not out of mind: Hybrid memory for dynamic video world models. *arXiv preprint arXiv:2603.25716*, 2026.

Yabo Chen, Yuanzhi Liang, Jiepeng Wang, Tingxi Chen, Junfei Cheng, Zixiao Gu, Yuyang Huang, Zicheng Jiang, Wei Li, Tian Li, et al. Teleworld: Towards dynamic multimodal synthesis with a 4d world model. *arXiv preprint arXiv:2601.00051*, 2025.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European conference on computer vision*, pp. 370–386. Springer, 2024.

Zicheng Duan, Jiatong Xia, Zeyu Zhang, Wenbo Zhang, Gengze Zhou, Chenhui Gou, Yefei He, Feng Chen, Xinyu Zhang, and Lingqiao Liu. Liveworld: Simulating out-of-sight dynamics in generative video world models. *arXiv preprint arXiv:2603.07145*, 2026.

Samuel Garcin, Thomas Walker, Steven McDonagh, Tim Pearce, Hakan Bilen, Tianyu He, Kaixin Wang, and Jiang Bian. Beyond pixel histories: World models with persistent 3d state. *arXiv preprint arXiv:2603.03482*, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021.

Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25690–25699. IEEE, 2025a.

Yang Li, Jinglu Wang, Lei Chu, Xiao Li, Shiu-hong Kao, Ying-Cong Chen, and Yan Lu. Streamgs: Online generalizable gaussian splatting reconstruction for unposed image streams. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25841–25850. IEEE, 2025b.

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

Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Yuancheng Xu, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, et al. Vista4d: Video reshooting with 4d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32671–32682, 2026.

Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. In *International conference on machine learning*, pp. 6140–6149. PMLR, 2020.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6121–6132. IEEE, 2025.

Hanlin Wang, Hao Ouyang, Qiuyu Wang, Wen Wang, Qingyan Bai, Ka Leong Cheng, Yue Yu, Yixuan Li, Yihao Meng, Zichen Liu, et al. Worlddirector: Building controllable world simulators with persistent dynamic memory. *arXiv preprint arXiv:2607.02517*, 2026a.

Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522. IEEE, 2025.

Weijie Wang, Haoyu Zhao, Yifan Yang, Feng Chen, Zeyu Zhang, Yefei He, Zicheng Duan, Donny Y Chen, Yuqing Yang, and Bohan Zhuang. Latent spatial memory for video world models. *arXiv preprint arXiv:2606.09828*, 2026b.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024b.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *arXiv preprint arXiv:2506.05284*, 2025.

Zike Wu, Qi Yan, Xuanyu Yi, Lele Wang, and Renjie Liao. Streamsplat: Towards online dynamic 3d reconstruction from uncalibrated video streams. In *International Conference on Learning Representations*, volume 2026, pp. 123114–123137, 2026.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. *arXiv preprint arXiv:2210.05861*, 2022.

Zeqi Xiao, Wenqi Ouyang, Yifan Zhou, Shuai Yang, Lei Yang, Jianlou Si, and Xingang Pan. Trajectory attention for fine-grained video motion control. In *International Conference on Learning Representations*, volume 2025, pp. 102934–102948, 2025.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. *Advances in Neural Information Processing Systems*, 38:49632–49652, 2026.

Shuai Yang, Bingjie Gao, Ziwei Liu, Jiaqi Wang, Dahua Lin, and Tong Wu. Permavid: Consistent video generation across edits via disentangled context memory. *arXiv preprint arXiv:2606.16449*, 2026a.

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. *arXiv preprint arXiv:2601.00393*, 2026b.

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Jiwen Yu, Jianxiong Gao, Jianhong Bai, Yiran Qin, Kaiyi Huang, Quande Liu, Xintao Wang, Pengfei Wan, Kun Gai, and Xihui Liu. Memlearner: Learning to query context memory for video world models. *arXiv preprint arXiv:2606.31734*, 2026.

Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2063–2073. IEEE, 2025.

Jinjing Zhao, Fangyun Wei, Zhening Liu, Hongyang Zhang, Chang Xu, and Yan Lu. Spatia: Video generation with updatable spatial memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4245–4257, 2026.

A Additional Method Details

A.1 Geometric Reconstruction and Object Evidence

Gaussian parameterization. StateSplat uses the geometric estimates of the frozen DA3 backbone to initialize points $\boldsymbol{\mu}^{\text{base}}$. The reconstruction Transformer and Gaussian head, trained on RE10K, predict positional refinements and the remaining Gaussian attributes:

$$\boldsymbol{\mu} = \boldsymbol{\mu}^{\text{base}} + \Delta\boldsymbol{\mu}. \tag{7}$$

This expression represents the initial points and effective offsets in a common coordinate system. Together with covariance $\boldsymbol{\Sigma}$, opacity α, and appearance $\mathbf{c}$, the refined means define the reconstructed primitives.

Online geometry storage and rendering. Background reconstructions extend the accumulated scene representation. A learned retention gate adjusts their contributions by modulating opacity at rendering time, without overwriting the stored geometric parameters. The gate includes an MLP and an attention residual. Uncovered pixels in the rendering condition are filled with black.

For objects, the first admitted observation establishes a local reference geometry; this need not coincide with the first video frame. Subsequent observations provide new visible geometry for pose estimation, but are not fused into or substituted for the fixed reference Gaussians. The scene state therefore evolves through background accumulation and object-pose updates. This configuration models rigid placement changes, rather than time-varying deformation of local shape.

Rigid object placement. Let $T_i(t) = [R_i(t), \mathbf{p}_i(t)] \in SE(3)$ map object-local coordinates to world coordinates. ICP alignment between reference Gaussian means and current object points reconstructed from masked depth estimates determines this transform. Each reference Gaussian is placed as

$$\begin{aligned} \boldsymbol{\mu}^{w}_{i,j}(t) &= R_i(t)\boldsymbol{\mu}^{o}_{i,j} + \mathbf{p}_i(t), \\ \boldsymbol{\Sigma}^{w}_{i,j}(t) &= R_i(t)\boldsymbol{\Sigma}^{o}_{i,j}R_i(t)^{\top}. \end{aligned} \tag{8}$$

Superscripts o and w denote object-local and world coordinates. Pose estimates and their metadata populate $\mathcal{R}_i$ for subsequent propagation and readout.

Adapted feature extraction. The adapted state-feature block is initialized by copying the final reconstruction Transformer block. It receives the features entering that block, providing a separate trainable pathway for memory evidence. The dynR configuration additionally uses a patch residual MLP. Object masks select up to 48 patch features per evidence frame, each with 768 channels. The feature pathway is trained jointly with the memory and generation modules.

Object-local patch coordinates. For selected patch j, let $\mathbf{u}_{j,t}$ denote its center pixel and $d_{j,t}$ the mean depth of object pixels within that patch. Its object-local coordinate is

$$\mathbf{x}^{o}_{j,t} = T_i(t)^{-1}\pi^{-1}(\mathbf{u}_{j,t}, d_{j,t}; C_t), \tag{9}$$

where C_t specifies camera intrinsics and extrinsics, and π^{-1} back-projects into world coordinates. The rigid transformation acts on homogeneous coordinates.

The feature describes visible appearance and structure, while the coordinate identifies its approximate location in the object reference frame. Because the coordinate combines a patch center with aggregated depth, it does not establish an exact surface correspondence across observations.

Evidence encoding for attention. Fourier positional encoding maps the local coordinates to features compatible with the visual tokens. We denote the resulting key and value construction by

$$K_{i,t} = \mathcal{K}_\theta\left(\{\mathbf{f}_{j,t}, \gamma(\mathbf{x}^o_{j,t})\}_{j\in\Omega_{i,t}}\right),$$
$$V_{i,t} = \mathcal{V}_\theta\left(\{\mathbf{f}_{j,t}, \gamma(\mathbf{x}^o_{j,t})\}_{j\in\Omega_{i,t}}\right), \tag{10}$$

where γ denotes Fourier encoding. The operators include the visual projections and the addition of projected positional information. Raw three-dimensional coordinates are not directly added to visual feature vectors.

A.2 Memory Records, Updates, and Readout

Complementary representations. Explicit records encode estimated object placement and its temporal context. Visual memory integrates appearance and structure that these geometric fields do not describe. The records locate the object at a requested time and camera, while memory supplies information retrieved from earlier visual evidence.

Evidence may originate from real observations or admitted generated frames. It is an image-derived estimate, not necessarily a correct observation of the underlying scene. Provenance and quality are therefore retained alongside its visual content.

Persistent storage and evidence replay. The persistent storage comprises object-state records and chronological evidence logs. Memory tokens are not cached across generation chunks. Instead, each chunk reconstructs memory by replaying the available evidence in temporal order, with a maximum replay length of 128 evidence steps. Only evidence available before the new generation interval is used.

The dynR configuration uses four memory tokens with dimension 768. Training uses truncated backpropagation through eight update steps. This distinguishes the persistent evidence storage from the learned memory representation constructed from that storage.

Update metadata and gating. The metadata embedding $q_{i,t}$ includes evidence source, silhouette overlap, background discrepancy, object pixel count, elapsed time, and pose validity. Source categories distinguish observed, generated, and override evidence. The embedding is broadcast across memory tokens and used both in the attention query and in the update gate.

Cross-attention retrieves incoming information according to the previous memory and metadata. The gate depends on the previous memory, retrieved update, and metadata, and modulates the contribution element-wise. Its final bias is initialized to -2.

The residual path preserves the previous representation. Admission and memory gating have distinct roles: admission selects observations for storage, whereas gating controls their contribution during integration. Without admitted evidence, the update is the identity operation.

Motion propagation. Motion information is represented by a translational velocity estimated from the pose history. For an admitted record at time τ, propagation to target time t_ℓ gives

$$\widehat{\mathbf{p}}_i(t_\ell) = \mathbf{p}_i(\tau) + (t_\ell - \tau)\widehat{\mathbf{v}}_i, \tag{11}$$

with time and velocity expressed in consistent units. This is constant-velocity extrapolation from available records, not an additional visual observation or a learned dynamics prediction.

State queries and memory readout. The propagated state is expressed relative to target camera C_ℓ and combined with temporal, provenance, and validity information. The explicit query has 41 components covering camera-relative pose, temporal offsets, source and validity indicators, velocity, historical state information, visibility, and projected support.

The readout query $e_\ell + W_s\xi_{i,\ell}$ depends on target time and estimated camera-relative state. It can therefore retrieve different memory content for different requested times and viewpoints even without additional evidence.

A two-layer MLP implements the state encoder E_θ. The token combines its explicit state encoding with $P_{\mathrm{mem}}(a_{i,\ell})$, the projected visual memory readout. The output projection of the memory branch

is initialized to zero. Memory update, readout, and token construction occur before the generator's conditioning operations.

A.3 Generation Conditions and Feedback

Temporal layout and causal conditions. Each generation chunk contains 61 video frames: 13 history frames and 48 newly generated frames. Successive chunks advance by 48 frames, and the generator uses 16 latent temporal positions.

The initial history consists of real observations; later histories use preceding generated frames. Conditions for a new interval are built from evidence already available before that interval. Future reference images are not supplied as generation conditions. The text condition is a fixed domain prompt for the clip.

Scene rendering conditions. Recent history supplies local temporal context. Scene renderings along the target camera trajectory supply geometric context from the accumulated Gaussian representation.

Object RGB is rendered only at the chunk-start frame, which is the first history frame. The remaining rendering conditions contain the background only. Thus, the rendering pathway does not provide a fully rendered future object trajectory: future object information is supplied through the state-conditioning pathways.

Appearance references. For each object's latest real observation, learned queries pool DA3 patch features into appearance tokens A_i. The source features have dimension 3072, and four learned queries produce four appearance tokens per object.

These tokens provide a direct observed reference alongside the state tokens. They are shared across target times and are not refreshed from generated frames. This pathway is distinct from the adapted StateSplat features used as evidence for memory updates.

Time-aligned cross-attention. At latent time ℓ, the condition sequence is

$$U_\ell = \mathrm{Concat}_i[z_{i,\ell}, A_i]. \tag{12}$$

Generator features provide queries, and the object conditions provide keys and values. The temporal restriction lets a video feature access state tokens for its corresponding latent time, while appearance references remain shared. The implementation also includes null tokens.

The added cross-attention modules operate at generator layers 8, 17, 26, and 35. They are distinct from the generator's existing attention projections adapted with LoRA.

Spatial feature placement. Spatial injection assigns object-state features to positions on the latent grid. The weight $w_{i,\ell}(\mathbf{u})$ depends on the projected object center and covariance, visibility, and uncertainty. The mapping ϕ_θ converts state tokens to spatial features, and P_{sp} maps the resulting field to a generator residual.

Spatial injection is applied at the same selected layers as the object cross-attention modules. The spatial residual output projection is initialized to zero, and its layer scale is initialized to one. The notation in the main text groups these operations without requiring geometric projection and residual addition to occur in the same software module.

Cross-attention and spatial injection complement one another: the former retrieves conditions according to video features, while the latter provides explicit projected support. Both modify internal features rather than prescribing output pixels.

Admission and generated provenance. After generation, candidate object observations are reconstructed and associated with tracked objects. Admission assesses whether they are explained by background memory, their agreement with the projected object silhouette, and their appearance consistency.

The checks use background discrepancy, silhouette–mask IoU, and DINOv2 crop-CLS cosine similarity. This appearance check is separate from the DA3-based appearance-token pathway.

Accepted observations extend the pose records and visual evidence logs. Rejected candidates or missing observations add no visual evidence; existing records remain available for propagation and readout. Generated observations retain their source label. The admission mechanism filters candidates but does not establish that every accepted generated estimate is correct.

A.4 Detailed Training Objectives

Training stages. The StateSplat reconstruction pathway is pretrained on RE10K with the DA3 backbone frozen. It supplies geometric reconstruction and initializes the adapted state-feature block. Subsequent joint training optimizes feature adaptation, memory, and conditioning with LoRA adaptation of the pretrained generator.

Region-normalized flow matching. Let x_s be the noisy video latent at flow time s, $v^\star$ the corresponding flow-matching target, and $v_\theta(x_s, s; \mathcal{C})$ the prediction under the generation conditions $\mathcal{C}$. To make channel normalization explicit, $\mathcal{A}$ indexes scalar latent elements, including channel indices. Define

$$\ell(\mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{p \in \mathcal{A}} \left(v_{\theta,p}(x_s, s; \mathcal{C}) - v_p^\star \right)^2 . \tag{13}$$

The generation loss is

$$\mathcal{L}_{\mathrm{fm}} = \mathbb{E}[w(s)\ell(\mathcal{N})], \tag{14}$$

where $\mathcal{N}$ contains newly generated latent elements and excludes conditioning-history frames.

Let $\mathcal{M}_{\mathrm{obj}}$ denote the target object mask mapped to the latent grid and broadcast over channels. The additional object loss is

$$\mathcal{L}_{\mathrm{obj}} = \mathbb{E}[w(s)\ell(\mathcal{N} \cap \mathcal{M}_{\mathrm{obj}})] . \tag{15}$$

It uses the same flow target, channel normalization, and time weighting as the full-region loss. Its contribution is zero when no valid object elements are present.

This additional average increases the contribution of objects occupying a small fraction of the image. Target masks supervise the loss and are not supplied as future observations during inference. The implementation separately records an empty valid region and a missing mask.

Predictive memory reconstruction. An auxiliary decoder predicts object RGB and depth along sampled camera rays using memory before the corresponding evidence is incorporated. For valid object rays $\mathcal{V}$,

$$\mathcal{L}_{\mathrm{aux}} = \frac{1}{4|\mathcal{V}|} \sum_{r \in \mathcal{V}} \left[\|\widehat{\mathbf{c}}_r - \mathbf{c}_r\|_2^2 + (\widehat{d}_r - d_r)^2 \right] . \tag{16}$$

Here $\mathbf{c}_r$ contains three color channels, and d_r uses the auxiliary decoder's depth parameterization. The denominator averages over three RGB channels and one depth channel.

Using pre-update memory supervises information retained from previous evidence rather than immediate reconstruction after writing the target observation into memory. The auxiliary objective complements generation supervision by directly encouraging reconstructable appearance and geometry. The decoder is not used during inference.

Reappearance sampling. Reappearance windows contain no target object in the conditioning history, but include visible target frames in the generation interval and earlier evidence available to memory. They expose training to cases that cannot be handled by simply continuing a currently visible object.

The training pool contains 112 reappearance windows, 356 other dynamic windows, 294 parked-object windows, and 188 background windows. Sampling proceeds by category, then clip, then window. The category probabilities are respectively 0.25, 0.40, 0.20, and 0.15. Reappearance candidates require at least ten visible target frames and at least three earlier valid evidence frames. The selected training reappearance histories were verified to have zero target-mask pixels.

State perturbations. The dynR recipe applies object-conditioning dropout with probability 0.25 and position perturbations with probability 0.5. The position-noise scale corresponds to one latent grid cell and increases by a factor of 1.5 toward the later part of the window.

The dynR2 control removes only the artificial position perturbation for reappearance windows. It retains the other training settings. Removing this perturbation does not make estimated states identical to target object positions: reconstruction and propagation errors remain.

Initialization and optimization. The dynR generator is initialized from the original Cosmos3-Nano checkpoint. LoRA is applied to the generation attention query, key, value, and output projections with rank 16 and scaling parameter 32. The original generator weights are frozen except for the camera/action pathway trained directly; the input and output projections remain frozen.

New object modules are initialized separately, while the adapted feature block is initialized from the pretrained StateSplat reconstruction block. The learning rates are 10^{-4} for LoRA and new object modules, 2×10^{-5} for the adapted StateSplat block, and 3×10^{-5} for the camera/action pathway.

Joint training uses 600 optimizer updates, corresponding to 1200 micro-iterations with gradient accumulation over two micro-iterations. The loss weights are $\lambda_{\mathrm{obj}} = 1.0$ and $\lambda_{\mathrm{aux}} = 0.1$.

B Datasets, Evaluation, and Additional Analyses

This section describes dataset acquisition, evaluation settings, metric definitions, and additional analyses. We distinguish agreement with a reference future from consistency with a model's own generated history. The former evaluates correspondence to a specific reference sequence, whereas the latter evaluates whether generated developments remain internally consistent over time.

B.1 CARLA Acquisition and Dataset Construction

Simulation environment. The collection scripts and environment documentation specify CARLA 0.9.15. All episodes are captured in Town10HD under ClearNoon conditions using synchronous simulation. The simulation step is 0.05 seconds, and frames are recorded every two simulation steps, yielding a capture rate of 10 frames per second. The simulator-version field in the episode metadata was not populated; the version reported here follows the collection scripts and documentation.

Images are captured at 960×540 pixels with a horizontal field of view of 90°. Model inputs and evaluation images use a resolution of 832×480. A common scale factor first resizes the source image to cover the target resolution, followed by a center crop. For source dimensions $W \times H$, the scale is

$$s = \max\left(\frac{832}{W}, \frac{480}{H}\right). \tag{17}$$

Reference masks undergo the same geometric transformation using nearest-neighbor interpolation.

Camera trajectories and target motion. The camera is placed approximately 9 meters laterally from the road. Its position remains fixed, while its orientation changes through in-place rotation. The target vehicle moves along a straight path at constant speed. Two vehicle types, denoted v0 and v1, provide appearance variation.

Three scenario families vary the timing and duration of camera departure and return. The following intervals describe the collection-script phase labels.

- **Scenario A.** The sequence contains 180 recorded frames, and the target speed is $0.6\,\mathrm{m/s}$. The camera holds its observation direction over frames 0–60, rotates away by 120° over frames 71–90, maintains that direction over frames 91–110, and returns over frames 111–120.
- **Scenario B.** The sequence contains 240 recorded frames, and the target speed is $0.8\,\mathrm{m/s}$. The camera tracks the target over frames 0–120, rotates away by 120° over frames 121–140, holds over frames 141–150, and returns over frames 151–160 before resuming tracking.

- **Scenario C.** The sequence contains 240 recorded frames, and the target speed is $0.8\,\mathrm{m/s}$. Camera departure occurs over frames 101–120, followed by a longer hidden interval over frames 121–170 and return over frames 171–180.

Rotations use smoothstep interpolation. For normalized transition time $u \in [0, 1]$, the interpolation weight is $3u^2 - 2u^3$. This gives gradual transitions at the phase boundaries.

These intervals describe the recorded reference sequence rather than the observed RGB prefix supplied to a generator. A return phase also does not necessarily begin at the exact frame in which the target becomes visible. Reference visibility is determined from the target mask after the evaluation image transformation.

Depth-difference target masks. Target masks are constructed using a paired background render. The target vehicle is removed, and the same camera trajectory is rendered again to obtain background depth. Let D_t^{scene} denote the depth image with the target present and D_t^{bg} the paired background-pass depth. The initial foreground mask is

$$M_t(u, v) = \mathbf{1}\left[D_t^{\mathrm{bg}}(u, v) - D_t^{\mathrm{scene}}(u, v) > 0.15\,\mathrm{m}\right]. \tag{18}$$

Connected components smaller than 300 pixels are removed during mask construction.

This procedure identifies visible pixels at which the target introduces a closer surface than the background pass. It does not represent the complete extent of an occluded vehicle. Simulator instance segmentation was not used because the spawned target vehicles were not identified by that sensor in the collection build.

The masks are subsequently transformed to 832×480. The acquisition-stage component filter is distinct from the 200-pixel reference-visibility threshold used for evaluation.

Actor metadata and reference projections. The simulator records actor identifiers, world-space poses, velocities, and bounding-box dimensions. Camera calibration and actor poses provide geometric reference locations even when the target is not directly visible.

For a world-space actor center $\mathbf{X}_t^w$, the camera projection is

$$\widetilde{\mathbf{u}}_t = K_t\left(R_t^{cw}\mathbf{X}_t^w + \mathbf{t}_t^{cw}\right), \qquad \mathbf{u}_t = \frac{\widetilde{\mathbf{u}}_{t,1:2}}{\widetilde{\mathbf{u}}_{t,3}}, \tag{19}$$

where K_t is the intrinsic matrix and $(R_t^{cw}, \mathbf{t}_t^{cw})$ maps world coordinates to camera coordinates. Projections are expressed in the evaluation image coordinate system after resizing and cropping.

A projected actor center and the centroid of its visible mask need not coincide. Target localization uses mask centroids, while the non-target diagnostic additionally uses projected actor centers.

Dataset splits. The collected dataset contains 101 training clips, 17 validation clips, and 32 test clips. All splits use Town10HD with different spawn locations. This evaluates generalization across placements within one map rather than across unseen CARLA towns.

The validation collection spans five placement groups with available combinations of vehicle types and scenario families. The test set contains eight placement groups, two vehicle types, and scenarios A and B. The main CARLA comparison uses three validation clips, A v0, B v0, and C v0, covering all three camera-schedule families with the same vehicle variant. These clips form a shared development subset for cross-method comparison; they do not constitute an evaluation over the full validation or test set. The single-clip LiveWorld Duan et al. (2026) diagnostic is reported separately.

Source-clip counts are distinct from the number of training windows sampled from those clips. The reported CARLA results characterize performance on this development subset and should not be interpreted as held-out test performance.

Training-window construction. The training pool contains 112 reappearance windows, 356 other dynamic windows, 294 parked-object windows, and 188 background windows. Their sampling probabilities are 0.25, 0.40, 0.20, and 0.15, respectively. Selection proceeds by category, then clip, then window.

Reappearance windows contain no target-mask pixels in the conditioning history, at least ten visible target frames in the prediction interval, and at least three earlier valid evidence frames. Earlier evidence is available to object memory, while future target frames and annotations provide training supervision. This exposes training to cases in which recent video history alone contains no visible target.

B.2 DAVIS Preparation and Evaluation Tasks

Reference-based continuation. DAVIS continuation uses the first 13 frames, indexed 0–12, as observed RGB input. Subsequent generated frames are compared with the corresponding reference frames. Observed input and repeated handoff frames are excluded from scoring.

The original 854×480 images are transformed to 832×480 using the same resize-and-center-crop rule as above. Model outputs at a different resolution are resized to 832×480 using OpenCV `INTER_AREA`. Reference masks undergo the corresponding geometric transformation using nearest-neighbor interpolation.

Provided camera trajectory. DA3 estimates the target camera trajectory from the original video, including frames beyond the observed RGB prefix. The estimated trajectory is supplied identically to the compared methods.

The task therefore assumes a provided future camera path and does not evaluate camera-path prediction. Future reference RGB contributes to this trajectory estimation but is otherwise used for scoring rather than future visual conditioning. Reference masks are used for evaluation.

Development and held-out sequences. The evaluation separates seven development sequences from ten sequences unused during development. The held-out set contains `bus`, `soapbox`, `boat`, `stroller`, `bmx-trees`, `dog`, `camel`, `goat`, `elephant`, and `parkour`.

For multi-object sequences, the union of all annotated DAVIS object identifiers defines the reference foreground mask. Object-region evaluation can therefore cover several objects rather than a single tracked instance.

Novel departure-and-return trajectories. A second task uses a new camera trajectory that departs from and revisits the scene. No corresponding reference video exists along this trajectory, so reference-based PSNR, SSIM, and LPIPS are not applied.

Instead, the evaluation reports Identity, Continuity, and Artifacts over five eligible cases per method. These measurements assess appearance-based object reappearance, positional agreement with extrapolated observed motion, and identity-qualified ghost or duplicate detections. They do not measure accuracy against a ground-truth future trajectory. Metric definitions are provided in Sec. B.7.

B.3 Frame Selection and Evaluation Settings

Absolute frame correspondence. Outputs are aligned using the `frame_index` mapping stored in `run.json`. Correspondence is not inferred solely from the position of a frame in an exported video. Observed source frames and repeated chunk-handoff frames are excluded from newly generated outputs.

Let $\mathcal{G}_{m,e}$ denote the set of newly generated absolute frame indices for method m and episode e. This set describes output availability independently of detection success.

Common generated frames. For a directly compared set of methods $\mathcal{M}$, the common frame set for episode e is

$$\mathcal{C}_e = \bigcap_{m \in \mathcal{M}} \mathcal{G}_{m,e}. \tag{20}$$

Matched comparisons use shared episode and frame sets. The separate single-clip LiveWorld Duan et al. (2026) diagnostic does not define the shared three-clip CARLA comparison.

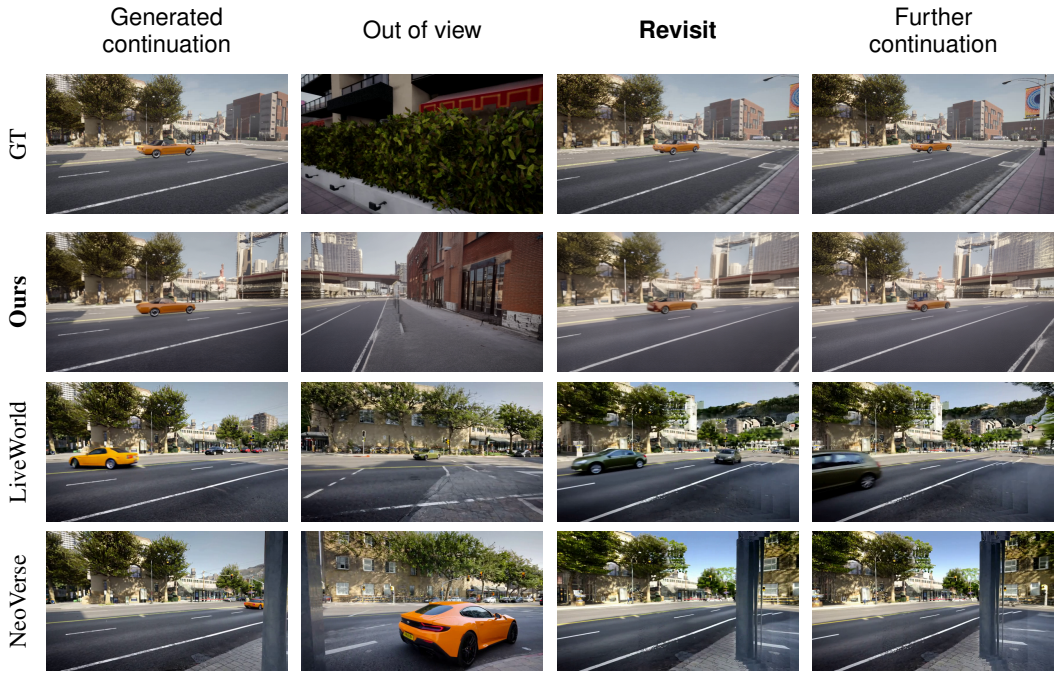

Figure 5: Additional qualitative results across departure, revisit, and subsequent continuation. Columns show matched absolute timestamps beyond the observed input, with temporal phases defined by the reference sequence. The final column extends the comparison beyond reappearance to show subsequent object motion and scene evolution. GT provides the reference for comparing object appearance, placement, and background consistency.

Return frames and eligibility. The reference CARLA return set contains target-visible return frames generated by all full-coverage methods: NeoVerse Yang et al. (2026b), Vista4D, and Ours. Partially covering methods are scored on the subset they generated, with coverage reported separately.

The return-eligibility rule requires a method to generate all of the first ten reference-visible return frames in an episode. An eligible episode can still have missing later return frames. Outputs are not repeated or stretched to extend their reported duration.

Frame and sequence coverage. Temporal coverage describes the generated portion of a requested interval. Episode or sequence counts describe how many members of the declared evaluation set contribute results. These quantities are reported separately.

In the DAVIS held-out comparison, Spatia Zhao et al. (2026) covers nine of ten sequences, whereas LiveWorld Duan et al. (2026) has incomplete temporal coverage. Their aggregate scores therefore do not use exactly the same evidence as methods covering all ten sequences and all requested frames. The reported frame and sequence counts expose these differences.

Baseline executions. We compare with GEN3C Ren et al. (2025), NeoVerse Yang et al. (2026b), Vista4D, Spatia Zhao et al. (2026), and LiveWorld Duan et al. (2026). Adapted executions are distinguished from official configurations.

The single-clip LiveWorld Duan et al. (2026) result in the main CARLA table uses the official few-step 14B pipeline without DINOv3, with the event branch inactive. It is reported as a diagnostic rather than as a complete evaluation of the official event-enabled method.

A separate category-name-matching execution activates the event branch through a substitute association mechanism. Its outputs are not pooled with the event-disabled execution.

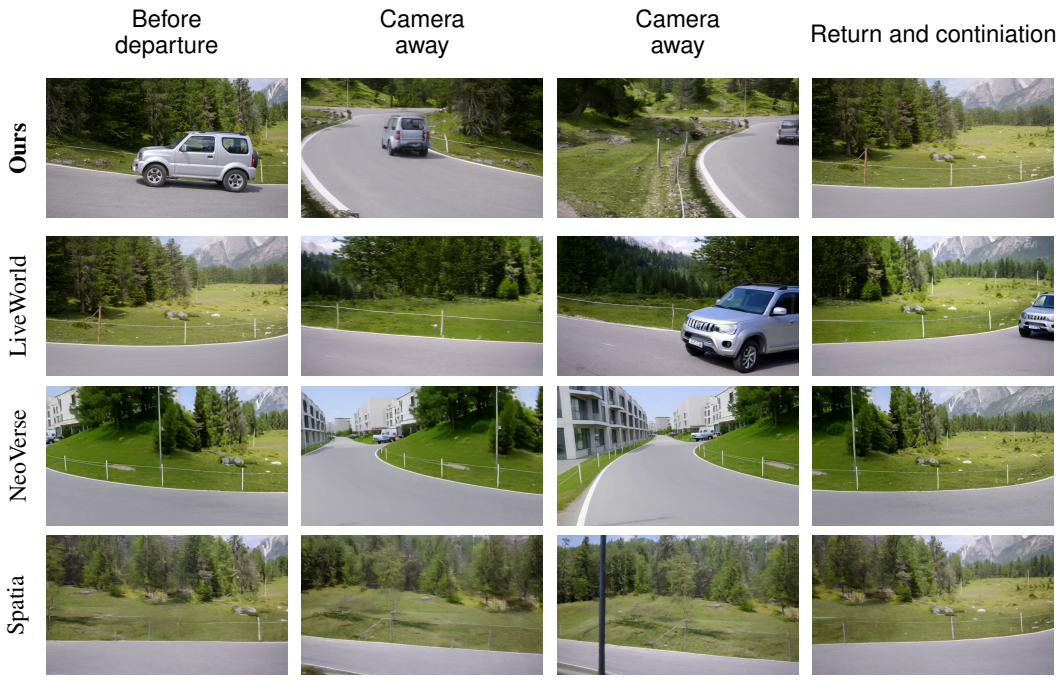

Figure 6: Additional qualitative comparisons on DAVIS under a novel departure-and-return camera trajectory. Columns show matched timestamps during generation: before camera departure, while the camera is away, upon return, and during further continuation. All displayed frames are generated after the observed input. No ground-truth video is available along this trajectory; the comparison illustrates object reappearance, appearance consistency, and subsequent motion, alongside changes in scene layout.

Run identity and random seeds. Each method row corresponds to one execution condition. Estimated generated-state updates define the main setting for Ours. Update-disabled and evidence-history interventions remain separate analysis conditions. Each episode/frame pair is counted once within a declared condition.

Evaluation uses seed 0, except for LiveWorld Duan et al. (2026), which uses the official example seed 42. The reported comparisons are not multiple-seed evaluation averages. The training callback configuration specifies seed 0, but the effective global training seed has not yet been verified from the training log.

B.4 CARLA Detection and Localization Metrics

Detector and candidate filtering. Generated images are evaluated at 832×480 using torchvision Mask R-CNN ResNet50-FPN v2 with the default COCO-pretrained weights. The retained classes are *car*, *bus*, and *truck*, with confidence at least 0.5. Instance masks are binarized at 0.5. Candidates with fewer than 200 mask pixels are discarded for target detection and localization.

Reference visibility. Let $M_{e,t}$ be the transformed reference target mask. A frame is reference-visible when

$$|M_{e,t}| \geq 200. \tag{21}$$

This criterion is applied after the reference mask has been transformed to the evaluation resolution.

The reference location is the mask centroid:

$$\mathbf{p}^{\star}_{e,t} = \frac{1}{|M_{e,t}|} \sum_{\mathbf{u} \in M_{e,t}} \mathbf{u}. \tag{22}$$

Predicted vehicle locations are likewise the centroids of their binary instance masks. Bounding-box centers are not used for localization.

Target association. Among retained vehicle detections, the evaluator selects the candidate nearest to the reference target centroid. For candidate centers $\mathcal{P}_{m,e,t}$, define

$$d_{m,e,t} = \min_{\mathbf{p} \in \mathcal{P}_{m,e,t}} \|\mathbf{p} - \mathbf{p}^{\star}_{e,t}\|_2, \tag{23}$$

with $d_{m,e,t} = +\infty$ when no candidate exists. The detection indicator is

$$b_{m,e,t} = \mathbf{1}[d_{m,e,t} \leq 150]. \tag{24}$$

Detection therefore requires a vehicle candidate within 150 pixels of the reference location. It does not count every vehicle anywhere in the image as a successful target recovery. However, it remains a location-based association: a different vehicle near the reference location can satisfy the rule.

Pooled detection rate. Let $\mathcal{F}_{m,e}$ contain the scored, reference-visible frames. Detection is pooled across these frames:

$$D_m = \frac{\sum_e \sum_{t \in \mathcal{F}_{m,e}} b_{m,e,t}}{\sum_e |\mathcal{F}_{m,e}|}. \tag{25}$$

The numerator and denominator are reported as "Detected / eligible." Missed detections remain in the denominator.

This metric measures spatially associated vehicle recovery. It does not independently establish identity preservation or accurate motion.

Conditional normalized localization error. For a successful detection, the framewise localization error is

$$\epsilon_{m,e,t} = \frac{d_{m,e,t}}{\sqrt{|M_{e,t}|}}. \tag{26}$$

The normalizer is the square root of the visible reference-mask area, measured in pixels. The error is dimensionless. A value of one corresponds to a centroid displacement of $\sqrt{|M_{e,t}|}$ pixels.

This normalization accounts for apparent object size, but it also depends on the visible mask extent. For example, partial occlusion can reduce mask area and increase normalized error for the same pixel displacement.

Within each episode, localization is the median over successful detections:

$$L_{m,e} = \operatorname*{median}_{t \in \mathcal{F}_{m,e}:\, b_{m,e,t}=1} \epsilon_{m,e,t}. \tag{27}$$

The table averages valid episode medians:

$$L_m = \frac{1}{|\mathcal{E}^+_m|} \sum_{e \in \mathcal{E}^+_m} L_{m,e}, \tag{28}$$

where $\mathcal{E}^+_m$ contains episodes with defined localization measurements. Without a successful detection, conditional localization error is undefined rather than zero.

Detection and localization therefore use different aggregation rules: pooled frame counts for detection and a macro-average of episode medians for localization. A method detecting only easy frames may have a low conditional error despite poor overall recovery. Both metrics and their counts must be read together.

Accuracy within 60 pixels. Additional diagnostics report the fraction of eligible frames with a detection within 60 pixels:

$$A^{60}_m = \frac{\sum_e \sum_{t \in \mathcal{F}_{m,e}} \mathbf{1}[d_{m,e,t} \leq 60]}{\sum_e |\mathcal{F}_{m,e}|}. \tag{29}$$

This retains missed detections in the denominator.

Diagnostic categories distinguish: *near*, with distance at most 60 pixels; *far*, with distance greater than 60 and at most 150 pixels; *other*, with retained detections but none within 150 pixels; and *none*, with no retained vehicle detection. These categories describe spatial correspondence, not object identity.

B.5 NON-TARGET DETECTION DIAGNOSTIC

Spatial exclusion and candidate size. The non-target diagnostic considers vehicle detections with instance-mask area at least 1500 pixels. This is distinct from the 200-pixel threshold used for target detection.

A candidate is counted only when its centroid is more than 150 pixels from every available reference location in the following set: the target-mask centroid, the projected target 3D center, and projected 3D centers of other reference vehicles.

Let $\mathcal{P}_{e,t}^{\mathrm{ref}}$ denote these reference locations. The non-target detection count is

$$n_{m,e,t} = \sum_{k:\,|\widehat{M}_k|\geq 1500} \mathbf{1}\left[\|\widehat{\mathbf{p}}_k - \mathbf{p}\|_2 > 150 \quad \forall \mathbf{p} \in \mathcal{P}_{e,t}^{\mathrm{ref}}\right]. \tag{30}$$

Excluding other reference vehicles prevents their correct generation from automatically being counted as a non-target error.

The diagnostic remains dependent on detector outputs and geometric reference locations. It is an auxiliary measure of detections away from known vehicles rather than an exhaustive measure of visual artifacts.

Occurrence rate in the main table. The main table reports the fraction of scored frames containing at least one non-target detection:

$$N_{m,e} = \frac{1}{|\mathcal{A}_{m,e}|} \sum_{t\in\mathcal{A}_{m,e}} \mathbf{1}[n_{m,e,t} > 0], \tag{31}$$

where $\mathcal{A}_{m,e}$ is the frame set for this diagnostic. The reported value is the equal-weight average of these rates across evaluated clips.

For common-interval rows, the diagnostic uses the corresponding common frames. For return-comparison rows, it instead uses frames in the `look_away` and `hidden` phases. Thus, the non-target column in the return comparison measures behavior during absence rather than within the return frames themselves.

Count-based auxiliary export. The separate 23-frame return export, `return_157_179_dev.csv`, reports the mean number of non-target detections per frame:

$$\overline{n}_{m,e} = \frac{1}{|\mathcal{A}_{m,e}|} \sum_{t\in\mathcal{A}_{m,e}} n_{m,e,t}. \tag{32}$$

Unlike an occurrence rate, this quantity can exceed one. It is not interchangeable with the main-table non-target metric and is labeled separately.

B.6 DAVIS IMAGE-QUALITY METRICS

PSNR. PSNR is computed on the full RGB frame using the 8-bit intensity range. For images of size $H \times W$,

$$\mathrm{MSE}(\widehat{I}, I) = \frac{1}{3HW} \sum_{u,v,c} \left(\widehat{I}_{u,v,c} - I_{u,v,c}\right)^2, \tag{33}$$

and

$$\mathrm{PSNR}(\widehat{I}, I) = 10 \log_{10} \frac{255^2}{\mathrm{MSE}(\widehat{I}, I)}. \tag{34}$$

Higher values indicate lower pixelwise disagreement. The metric is sensitive to spatial misalignment and reflects both foreground and background differences.

SSIM. SSIM is computed using `skimage.metrics.structural_similarity`, with `channel_axis=2`, `data_range=255`, and other parameters at their defaults. This uses a 7×7 uniformly weighted window, rather than Gaussian window weighting.

The local comparison is

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{35}$$

where μ, σ^2, and σ_{xy} are local means, variances, and covariance. The default constants correspond to $C_1 = (0.01 \cdot 255)^2$ and $C_2 = (0.03 \cdot 255)^2$. Spatial and channel comparisons are aggregated by the implementation into an image-level score. Higher values indicate greater structural similarity.

LPIPS. Full-frame LPIPS uses the AlexNet backbone with LPIPS version 0.1. RGB inputs are normalized to $[-1, 1]$. The metric compares normalized deep features using learned channel weights:

$$\mathrm{LPIPS}(\widehat{I}, I) = \sum_k \frac{1}{H_k W_k} \sum_{u,v} \left\| \mathbf{w}_k \odot \left(\phi_k(\widehat{I})_{u,v} - \phi_k(I)_{u,v} \right) \right\|_2^2. \tag{36}$$

Here ϕ_k denotes normalized features and $\mathbf{w}_k$ the learned channel weights at layer k. Lower values indicate greater perceptual similarity. Full-frame LPIPS includes both foreground and background differences.

Object-crop construction. The reference foreground mask is the union of all annotated DAVIS object identifiers. Frames with fewer than 20 foreground pixels are excluded from Object-crop LPIPS, but remain included in full-frame PSNR, SSIM, and LPIPS.

For a valid foreground mask, its axis-aligned bounding box is expanded on each side by 10% of the corresponding box dimension. If either resulting dimension is smaller than 64 pixels, that dimension is expanded about the box center to 64 pixels. The box is then clipped to the image boundary. Boundary clipping can reduce the final crop size.

The same box B is applied to the reference and generated images:

$$\mathrm{LPIPS}_{\mathrm{obj}} = \mathrm{LPIPS}\left(\mathrm{crop}(\widehat{I}, B), \mathrm{crop}(I, B) \right). \tag{37}$$

The crops are evaluated without resizing.

Using the same reference-defined crop preserves sensitivity to target displacement or disappearance. The generated object is not separately recentered before comparison. The crop contains nearby background and can span multiple objects. Object-crop LPIPS is therefore distinct from foreground-masked LPIPS and from an identity-specific similarity score.

Sequence-level aggregation. Scores are computed per frame, averaged over the common generated frames within each sequence, and then averaged equally over evaluated sequences:

$$Q_m = \frac{1}{|\mathcal{E}_m|} \sum_{e \in \mathcal{E}_m} \frac{1}{|\mathcal{F}_{m,e}|} \sum_{t \in \mathcal{F}_{m,e}} q(\widehat{I}_{m,e,t}, I_{e,t}). \tag{38}$$

Object-crop LPIPS uses the subset of frames with valid foreground crops. Observed frames 0–12 and repeated handoff frames are excluded.

This aggregation gives sequences equal weight, rather than allowing longer sequences to dominate. PSNR is averaged as a per-frame score rather than computed from a dataset-wide pooled MSE.

All four metrics evaluate agreement with a particular reference future. They do not establish physical plausibility or consistency of an alternative generated continuation.

B.7 Assessment of Novel-Trajectory Generation

Observation-based reference. Unlike reference-based DAVIS continuation, this departure-and-return evaluation uses observed frames 0–60. There is no ground-truth target video along the novel

camera trajectory. We therefore construct a common reference by extrapolating the target motion estimated from the observed frames at constant velocity.

Let $\widehat{\mathbf{X}}_t$ denote the extrapolated target position and C_t the evaluation camera. The expected image location is

$$\widehat{\mathbf{p}}_t = \pi(\widehat{\mathbf{X}}_t; C_t). \tag{39}$$

The trajectory is computed from observations alone and is shared across methods. It represents an extension of observed motion, not a ground-truth future trajectory. Consequently, agreement with this reference measures consistency with the observed-motion extrapolation.

Expected target size. Let $\mathcal{O}_{10}$ contain the last ten observed frames and M_τ their target masks. The reference image-space size is

$$s_{\text{obs}} = \sqrt{\frac{1}{|\mathcal{O}_{10}|} \sum_{\tau \in \mathcal{O}_{10}} |M_\tau|}. \tag{40}$$

The expected size at evaluation time t is

$$s_t = s_{\text{obs}} \frac{z_{\text{obs}}}{\widehat{z}_t}, \tag{41}$$

where z_{obs} is the observation-time reference depth and $\widehat{z}_t$ is the depth of the extrapolated target under camera C_t.

This inverse-depth scaling approximates the change in apparent linear size. It does not model changes in projected shape caused by object rotation or deformation.

Vehicle candidates and appearance features. We use the same COCO-pretrained Mask R-CNN ResNet50-FPN v2 detector as in the CARLA evaluation. Candidates belong to the car, bus, or truck classes, have confidence at least 0.5, and contain at least 200 mask pixels. Instance masks are thresholded at 0.5.

For each candidate, its mask bounding box is cropped with 10% padding. DINOv2 ViT-B/14 provides a CLS embedding $\mathbf{e}_{d,t}$ for candidate d at time t. All methods use the same feature extractor and checkpoint.

The reference embedding is the mean of target-crop embeddings from the last ten observed frames:

$$\mathbf{e}_{\text{ref}} = \frac{1}{|\mathcal{O}_{10}|} \sum_{\tau \in \mathcal{O}_{10}} \mathbf{e}_\tau^{\text{obs}}. \tag{42}$$

Similarity is measured by cosine similarity,

$$\text{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}. \tag{43}$$

Observation-calibrated identity threshold. Each sequence uses a threshold calibrated from its observed target appearance:

$$\tau_{\text{id}} = \min_{\tau \in \mathcal{O}} \text{sim}\left(\mathbf{e}_\tau^{\text{obs}}, \mathbf{e}_{\text{ref}}\right), \qquad \mathcal{O} = \{0, \ldots, 60\}. \tag{44}$$

Thus, the threshold is the lowest similarity to the reference embedding attained by the observed target.

Let $\mathcal{D}_t$ be the retained vehicle candidates. The identity-qualified set is

$$\mathcal{D}_t^{\text{id}} = \left\{d \in \mathcal{D}_t : \text{sim}(\mathbf{e}_{d,t}, \mathbf{e}_{\text{ref}}) \geq \tau_{\text{id}}\right\}. \tag{45}$$

This is an appearance-based identity proxy. It can confuse visually similar vehicles and does not provide a ground-truth identity assignment.

Eligible cases and return window. Let v_t indicate that the extrapolated target projects inside the image. Within the return interval 180–240, a case must contain at least ten such frames. The return window $\mathcal{R}$ comprises the first ten of these frames and is determined independently of generated detections.

A method is eligible only if it generates all frames in $\mathcal{R}$. A missing output is therefore distinct from a generated frame on which no target is detected. The number of eligible cases is reported in the table.

Identity. The case-level Identity score is

$$\text{Identity} = \frac{1}{|\mathcal{R}|} \sum_{t \in \mathcal{R}} \mathbf{1}\left[\mathcal{D}_t^{\text{id}} \neq \varnothing\right]. \tag{46}$$

It measures the fraction of return frames containing at least one vehicle whose appearance passes the observation-calibrated threshold. A frame without a qualifying candidate contributes zero. The score does not require the candidate to appear near the extrapolated position.

Continuity. When identity-qualified candidates exist, we select the one closest to the expected target location:

$$d_t^* = \arg\min_{d \in \mathcal{D}_t^{\text{id}}} \|\mathbf{c}_{d,t} - \widehat{\mathbf{p}}_t\|_2, \tag{47}$$

where $\mathbf{c}_{d,t}$ is the candidate mask centroid. The frame succeeds when

$$\frac{\|\mathbf{c}_{d_t^*,t} - \widehat{\mathbf{p}}_t\|_2}{s_t} \leq 1. \tag{48}$$

A frame without an identity-qualified candidate fails. The case-level score is

$$\text{Continuity} = \frac{1}{|\mathcal{R}|} \sum_{t \in \mathcal{R}} \mathbf{1}\left[\mathcal{D}_t^{\text{id}} \neq \varnothing \text{ and } \|\mathbf{c}_{d_t^*,t} - \widehat{\mathbf{p}}_t\|_2 \leq s_t\right]. \tag{49}$$

This measures reappearance of an appearance-consistent target near the continuation of observed motion. It is not ground-truth trajectory accuracy or a direct measure of frame-to-frame motion smoothness. By construction, Continuity cannot exceed Identity for the same evaluated case and return window.

Artifacts. The artifact diagnostic considers generated frames in the interval 100–240. It flags two types of events:

1. **Unexpected presence:** the extrapolated target is outside the image, but at least one identity-qualified vehicle is detected;
2. **Duplication:** the extrapolated target is inside the image, and at least two identity-qualified vehicles are detected.

Let $n_t^{\text{id}} = |\mathcal{D}_t^{\text{id}}|$. The event indicator is

$$a_t = \mathbf{1}\left[(v_t = 0 \text{ and } n_t^{\text{id}} \geq 1) \text{ or } (v_t = 1 \text{ and } n_t^{\text{id}} \geq 2)\right]. \tag{50}$$

For the generated frame set $\mathcal{A} = \mathcal{G} \cap [100, 240]$, the score is

$$\text{Artifacts} = \frac{1}{|\mathcal{A}|} \sum_{t \in \mathcal{A}} a_t. \tag{51}$$

Each frame contributes at most one event, regardless of the number of qualifying detections.

Identity filtering reduces the contribution of unrelated vehicles but cannot eliminate false matches. Moreover, unexpected presence is defined relative to an extrapolated trajectory, so a legitimate departure from constant-velocity motion can also trigger this diagnostic. The score does not measure all visual defects, such as background distortions or texture artifacts.

Case-level aggregation. Each metric is computed within a case and then averaged equally over eligible cases:

$$\overline{Q}_m = \frac{1}{|\mathcal{E}_m|} \sum_{e \in \mathcal{E}_m} Q_{m,e}, \tag{52}$$

where Q denotes Identity, Continuity, or Artifacts. The Eligible cases column reports the number of contributing cases. Identity and Continuity use the same return window within each case, while Artifacts uses the generated frames from the longer 100–240 interval.

B.8 Controlled State Interventions

Intervention construction. The state-position intervention holds generated history and sampling seed fixed while modifying the supplied object position. Two base executions establish regeneration variability. Alternative positions are selected to project inside the camera view.

Subsequent estimated-state updates are disabled after the intervention. This prevents newly generated observations from immediately overwriting the manipulated condition and distinguishes condition sensitivity from the effect of later evidence admission.

Directional agreement. Let $\Delta\mathbf{p}_{\mathrm{proj}}$ denote the projected displacement of the supplied state and $\Delta\mathbf{p}_{\mathrm{car}}$ the displacement of the corresponding generated vehicle relative to the base. For a nonzero projected displacement,

$$A = \frac{\Delta\mathbf{p}_{\mathrm{car}}^{\top} \Delta\mathbf{p}_{\mathrm{proj}}}{\|\Delta\mathbf{p}_{\mathrm{proj}}\|_2^2}. \tag{53}$$

A value of one indicates matching displacement along the intervention direction. Zero indicates no displacement component along that direction, and a negative value indicates movement in the opposite direction. The statistic is not bounded to $[0, 1]$.

For $\mathbf{u} = \Delta\mathbf{p}_{\mathrm{proj}}/\|\Delta\mathbf{p}_{\mathrm{proj}}\|_2$, parallel and perpendicular displacement are

$$d_{\parallel} = \Delta\mathbf{p}_{\mathrm{car}}^{\top}\mathbf{u}, \qquad d_{\perp} = \left\|\Delta\mathbf{p}_{\mathrm{car}} - d_{\parallel}\mathbf{u}\right\|_2. \tag{54}$$

Detection failures and unresolved vehicle correspondence are reported separately rather than assigned zero displacement.

Observed state sensitivity. In the completed dynR intervention on B v0 and B v1, the condition projections move by 30–145 pixels, whereas the generated vehicle moves by less than one pixel. Repeated base executions show zero detected vehicle displacement. The evaluated checkpoint therefore exhibits limited responsiveness to these position interventions, despite successful target reappearance.

This result concerns sensitivity of a fixed trained model to explicit state changes. It does not measure the effect of retraining with or without an architectural component.

B.9 Component Ablations and Training Controls

Matched-training comparisons. Table 4 reports six trained configurations under matched training conditions. The first four vary learned visual memory and spatial feature injection. The final two remove auxiliary reconstruction supervision and feature adaptation individually from the full configuration. Temporal token conditioning remains enabled in every variant.

The full-model row repeats the main comparison result to provide a local reference for the ablations. It represents the same model and evaluation setting, not an additional independent experiment.

Explicit state and appearance without learned visual memory. Setting `obj_memory=False` removes the learned visual memory tokens H. Time-dependent conditions are instead constructed from the explicit state vector and appearance features from the latest real observation.

The explicit state includes the three most recent pose-history entries, represented through displacement and elapsed time, together with velocity and propagation confidence. These historical pose values remain available even though recurrent visual-memory updates are disabled.

The rows labeled "Fixed history" therefore retain explicit pose history, not fixed aggregation of the same historical visual evidence used by learned memory. Comparison with the learned configuration evaluates the combined contribution of visual memory, feature adaptation, and auxiliary supervision.

Temporal-only conditioning. Setting `obj_spatial_inject=False` removes spatial feature injection while retaining token cross-attention. For learned memory, the temporal-only configuration retains the updater, feature adaptation, and auxiliary reconstruction supervision. Its difference from the full configuration is the spatial injection pathway.

Comparisons within each memory configuration measure the effect of spatial injection on return detection and localization. Improved scores under ordinary evaluation do not, by themselves, establish that output positions track explicit inference-time state displacements.

Auxiliary reconstruction supervision. The variant without auxiliary reconstruction sets `obj_mem_aux='none'`. The full configuration reconstructs object RGB and depth through ray queries with auxiliary weight $\lambda_{\text{aux}} = 0.1$.

Removing this objective leaves the memory and conditioning pathways trained through the generation objectives. This comparison evaluates the contribution of direct reconstruction supervision to the retained memory representation within the full architecture.

Feature adaptation. The variant without feature adaptation sets `obj_mem_adapt=False` and `obj_mem_open_blocks=0`. The patch residual MLP is disabled, the final StateSplat block remains frozen, and cached reconstruction features are used.

This comparison tests whether adapting visual evidence for memory-conditioned generation improves over fixed reconstruction features. It does not remove the object evidence pathway or the learned memory updater.

Evaluation and component effects. All six variants use the same 214 eligible return frames and the detection and localization definitions given above. Detection pools frame counts, while localization averages per-clip medians over successful detections.

Learned memory improves detection from 0.90 to 0.93 under temporal-only conditioning and from 0.92 to 0.95 with spatial injection. The corresponding localization errors decrease from 1.05 to 0.98 and from 0.96 to 0.92.

Removing auxiliary reconstruction supervision or feature adaptation reduces detections from 203 to 201 and increases localization error from 0.92 to 0.95 or 0.94, respectively. These results describe the evaluated configurations; the small differences are not estimates of statistical significance or variability across repeated training runs.