Teaching Video Generators to Remember: Eliciting Dynamic Memory for Out-of-Sight State Evolution

Tianshuo Xu[1], **Yichen Xie**[1,2], **Depu Meng**[1§], **Chensheng Peng**[1,2],
Quentin Herau[1], **Bo Jiang**[1], **Yihan Hu**[1], **Wei Zhan**[1,2†]
[1] Applied Intuition [2] University of California, Berkeley

Project Page: `https://remind-applied.github.io/`

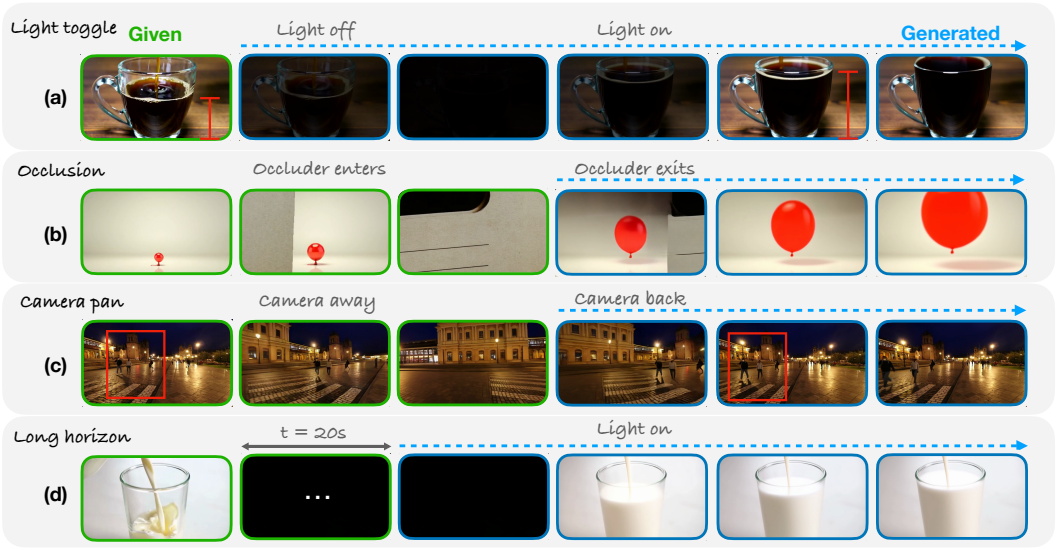

Figure 1: Selected qualitative out-of-sight recovery cases. **(a)** An I2V pouring task with a light-toggle interruption. **(b)** A V2V balloon-inflation process behind a full-frame occluder. **(c)** A pedestrian case with camera return. **(d)** An illustrative 20s dark-interval stress case. These selected examples show visual recovery behavior but do not by themselves establish duration-conditioned state advancement.

Abstract

Video world models should maintain evolving states when evidence is unobserved, yet current generators often freeze hidden states upon interruption. This is not simply a capacity problem: pretrained video diffusion transformers already possess KV-cache mechanisms capable of non-local retrieval, but they are rarely trained to use them as dynamic memory. We introduce ReMind, a framework eliciting dynamic memory behavior via memory-oriented data, event-aware training, and cache adaptation. Organized around a taxonomy of 100+ dynamic events, we build a camera-annotated training mixture combining VLM-filtered real videos, generated hard dynamics, synthetic camera loops, and memory-interruption augmentations. Each clip is converted into a frame graph with protected anchors, degraded intervals, and explicit temporal gaps. A node-structured curriculum—including node-drop, noisy memory, frontier continuation, and reference-cache training—forces the model to retrieve relevant past states across interruptions rather than relying solely on local continuity. PM-RoPE, a camera-phase RoPE extension, provides spatiotemporal addressing through one attention path while preserving pretrained pathways. ReMind achieves the best overall STEVO-Bench score and improves controlled short-horizon recovery.

† Corresponding author: `wei.zhan@applied.co`. § Project lead.

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

1 Introduction

Physical state continues to evolve even when it is temporarily unobserved. A container should keep filling while the lights are off, a balloon should continue inflating behind an occluder, and a pedestrian should keep walking while the camera looks away. Yet current video world models often tie state evolution too closely to immediate visual evidence: when the scene is revealed again, the hidden state may freeze, reset, or resume from an implausible point. This out-of-sight failure, recently formalized in STEVO-Bench [Ma et al., 2026], exposes a gap between generating visually plausible local motion and maintaining persistent world state across interrupted observation.

At first glance, these failures might seem addressable by scaling temporal context or improving autoregressive rollout. Self Forcing and LongLive use self-generated history, streaming tuning, and KV caching for causal generation [Huang et al., 2025, Yang et al., 2026a]; Helios instead uses compressed history injection without a KV cache [Yuan et al., 2026]. Cache compression methods further improve storage efficiency [Ranganath et al., 2026]. In parallel, memory-conditioned generators inject historical views or 3D/4D representations to improve spatial consistency under camera motion [Yu et al., 2025, Xiao et al., 2025, Ren et al., 2025, Yang et al., 2026b, InSpatio Team et al., 2026]. However, these advances do not directly teach a model when an older observation should override recent but corrupted context. We view the missing behavior through the lens of frame graphs: standard training presents frames as a local chain, while interruptions create non-local edges from a recovery frame to the latest reliable state anchor. If training never exposes such edges, attention and KV-cache machinery may store past tokens without learning when to retrieve them as dynamic memory.

We introduce ReMind, a framework for eliciting dynamic memory behavior from a pretrained autoregressive video diffusion transformer [Team Wan, 2025]. ReMind combines memory-oriented data construction, event-aware training, and pretraining-compatible cache adaptation. We build a camera-annotated training mixture organized around a taxonomy of over 100 dynamic events, combining VLM-filtered real videos, generated hard dynamics, rendered camera loops, and memory-interruption augmentations. Each clip is converted into a frame graph with protected state anchors, degraded observation intervals, and explicit temporal gaps, defining which historical nodes should remain useful when local visual evidence becomes unreliable.

Given these memory graphs, ReMind trains the model with a node-structured curriculum, including node-drop, noisy memory, V2V frontier, and reference-cache training. These regimes corrupt, drop, or temporally separate observations so that successful recovery requires retrieving the relevant past state rather than relying only on local continuity. To make such retrieval geometrically meaningful, we add Projective Memory RoPE (PM-RoPE), a camera-phase extension to rotary position embedding (RoPE) [Su et al., 2024]. Unlike dual-attention mechanisms adding a separate spatial branch, PM-RoPE grants cached entries unified spatiotemporal addresses within a single self-attention operation, preserving their original temporal positions and the pretrained attention pathway.

We evaluate ReMind on STEVO-Bench [Ma et al., 2026] and controlled recovery tasks covering occlusion, darkness, and camera lookaway. ReMind achieves the best overall score among compared models, with leading state-progress and competitive physical-plausibility and coherence scores. Matched 5B VBench arms have similar point estimates, without an equivalence or non-inferiority claim. We additionally use KV-importance maps as qualitative diagnostics of whether recovery is associated with attention to intended historical anchors.

Our main contributions are summarized as follows:

- We formulate out-of-sight state evolution as a dynamic-memory problem over frame graphs and introduce ReMind, a memory-elicitation framework constructing camera-annotated training graphs with protected anchors, degraded intervals, and temporal gaps.
- We propose a node-structured training framework with pretraining-compatible cache adaptation: node-drop, noisy memory, V2V frontier, and reference-cache training force retrieval of reliable state anchors, while PM-RoPE provides unified spatiotemporal addressing through one attention path when paired with non-contiguous cache positioning.
- We establish a validation protocol that measures both hidden-state recovery and standard generation quality, combining STEVO-Bench evaluation, controlled V2V recovery cases, reference-cache stress tests, KV-importance diagnostics, and VBench evaluation.

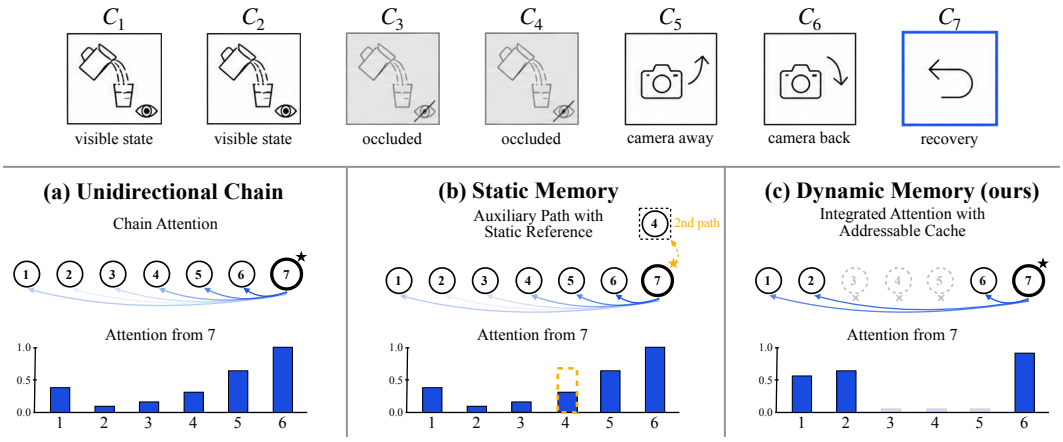

Figure 2: Conceptual comparison of memory use in video generation. **(a) Unidirectional Chain:** KV-cached models primarily retain recent rollout context. **(b) Structured Spatial Memory:** memory- or world-conditioned methods retrieve geometry-aligned historical references for spatial and revisit consistency. **(c) ReMind (Dynamic Memory):** our training exposes the native cache to reliable anchors, degraded observations, and evolving targets so it can use non-local historical evidence for state recovery.

2 Related Work

2.1 Autoregressive Diffusion Video Generation

Autoregressive diffusion adapts pretrained video diffusion backbones such as Wan [Team Wan, 2025] into causal rollout models. Teacher forcing conditions on clean history [Williams and Zipser, 1989], Diffusion Forcing uses independent per-token noise levels [Chen et al., 2024], and Self Forcing reduces the train-test mismatch by rolling out generated context with KV caching [Huang et al., 2025]. LongLive uses streaming-long tuning, short-window attention, and KV recaching [Yang et al., 2026a], whereas Helios uses compressed history injection without a KV cache or causal masking [Yuan et al., 2026]. These systems primarily target long-video quality or throughput rather than selecting an older reliable state witness. Separate cache compression and pruning studies improve efficiency [Ranganath et al., 2026], but do not teach the model when to retrieve an older observation as the latest reliable state witness. ReMind targets this out-of-sight failure [Ma et al., 2026] by training existing attention and cache mechanisms to behave as dynamic memory.

2.2 Memory Mechanisms for Video World Models

There are already some explorations regarding the memory mechanism for video generation models. We conceptually categorize them by their underlying topologies in Fig. 2.

Unidirectional Chain: KV-cached causal models commonly retain recent rollout context as a temporal buffer [Huang et al., 2025, Yang et al., 2026a] rather than an explicitly indexed scene-state store, which may leave them vulnerable when recent observations are corrupted.

Structured Spatial Memory: Recent methods inject spatial or 3D representations into memory. WorldPlay [Sun et al., 2025] dynamically reconstitutes a context set and uses an additional PRoPE attention path alongside standard temporal RoPE [Su et al., 2024, Li et al., 2025]. InSpatio-World [InSpatio Team et al., 2026] and NeoVerse [Yang et al., 2026b] use explicit geometric constraints or rendered projections, while MosaicMem [Yu et al., 2026] retrieves spatially aligned 3D patches. These methods primarily target geometric or revisit consistency; our focus is how an evolving state advances while it is unobserved.

Dynamic Memory: Our method, ReMind, moves beyond static memory by treating the KV cache in a dynamic way. Inter-frame relevance is not fixed to pure temporal adjacency or spatial overlapping. Instead, the model learns to adaptively condition the generation on past frames through its own

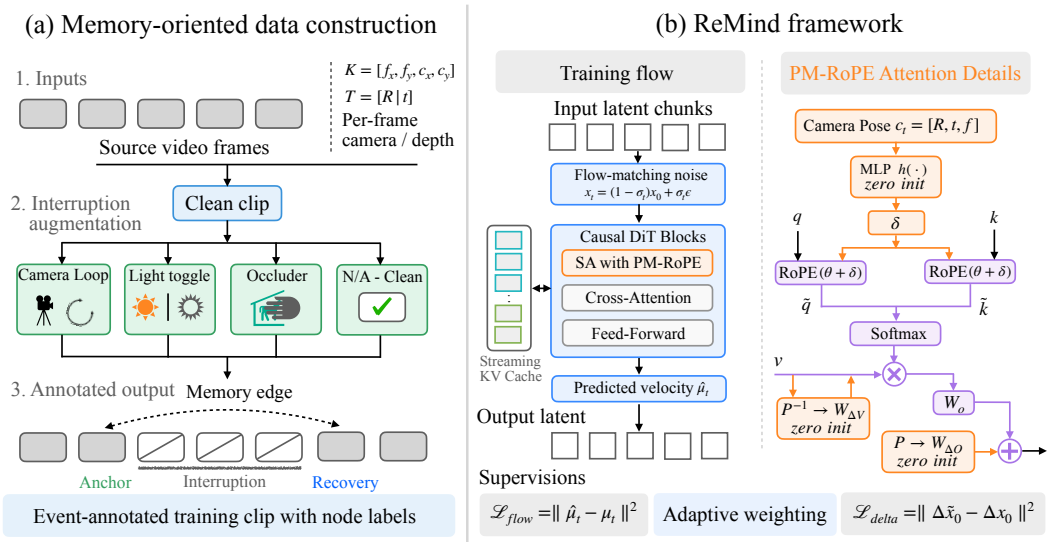

Figure 3: **Overview of ReMind. (a)** Four parallel data streams—general videos, natural dynamics, generated hard dynamics, and rendered camera loops—are normalized independently. Interruption augmentations are then applied within a clip; no cross-clip concatenation or insertion into DL3DV is performed. The resulting frame graphs contain annotated anchor, interruption, and recovery nodes. **(b)** The causal video DiT uses its native KV cache as dynamic memory. Cached chunks retain non-contiguous RoPE positions, while PM-RoPE adds camera-conditioned phase offsets and zero-initialized value/output residuals.

attention module with our proposed PM-RoPE and training regimes, which forces the model to resort to most related historical anchors under various distractors.

2.3 Camera Control for Video Generation

Camera-controlled video generation injects pose information through camera encoders, ControlNet-style modules, geometry-aware conditions, or attention modifications [He et al., 2025, Bahmani et al., 2024, Ren et al., 2025, Yang et al., 2026b, InSpatio Team et al., 2026]. Recent attention-native methods further encode camera geometry through projection-aware transports, parallel camera branches, ray-coordinate RoPE, or relative camera pose embeddings [Li et al., 2025, Sun et al., 2025, Zhang et al., 2026, Xie et al., 2026b, Li et al., 2026, Xie et al., 2026a]. Motion Forcing [Xu et al., 2026] and STANCE [Chen et al., 2025] instead target motion and interaction control rather than camera-trajectory control. PM-RoPE serves a narrower role in ReMind: it provides camera-aware addresses and non-contiguous temporal positions for cached anchors while preserving the pretrained attention pathway, enabling the event-aware curriculum to train non-local state retrieval.

3 ReMind with Dynamic Memory

ReMind builds on a pretrained causal video diffusion transformer and elicits dynamic memory through memory-oriented data, cache-compatible spatial addressing, and node-structured training. Each chunk can attend to cached entries that retain their original temporal positions and camera-aware addresses, allowing the model to retrieve historical state anchors rather than only recent context. Sec. 3.1 describes the memory-oriented data construction, Secs. 3.2 and 3.3 introduce PM-RoPE and the non-contiguous streaming KV cache, and Sec. 3.4 presents the training curriculum that teaches the model to use this memory adaptively. The overall framework of ReMind is illustrated in Fig. 3.

3.1 Memory-Oriented Data Construction

We construct our training data to strengthen the dynamic memory of ReMind from public video sources, OpenVid [Nan et al., 2025] and DL3DV [Ling et al., 2024]. Since these sources are not targeted at dynamic-memory events, we augment them with a taxonomy of over one hundred dynamic event types, scored by visual salience, frequency, and usefulness for hidden-state recovery. The taxonomy drives Pexels retrieval with dynamics-specific keywords and Qwen3-VL filtering [Bai et al., 2025], where the VLM captions events, detects cuts, and estimates the spatial and temporal occupancy of the dynamic region. We further synthesize underrepresented high-occupancy dynamics, such as close-up pouring/filling, and render selected dynamic scenes under loop trajectories with Neoverse [Yang et al., 2026b] for camera-memory supervision. The full dynamic taxonomy and data collection policy are provided in Appendix G.

These sources are four parallel streams, not stages of a single synthesis pipeline: general OpenVid/DL3DV clips are used without injected objects; natural Pexels dynamics remain real clips; hard dynamics are generated independently; and NeoVerse is used only for rendered camera-loop supervision. Orthogonally, light, occluder, zoom, and camera perturbations are applied within one clip. We never concatenate unrelated clips or insert generated content into DL3DV.

We then convert clips into memory-training graphs by applying interruption augmentations inspired by STEVO-Bench [Ma et al., 2026], including camera loops, light toggles, moving occluders, and zoom/camera perturbations. All clips are normalized with RGB frames, captions, event metadata, and, when available, camera intrinsics/extrinsics and depth; for web videos without native geometry, we estimate depth and camera trajectories with Depth Anything 3 [Lin et al., 2025]. Event nodes are derived from loop-return points or known peak/recovery frames, and define the protected anchors and recovery regions used by the memory curriculum.

3.2 Projective Memory Rotary Positional Embedding

Motivation: why couple camera and temporal addressing? PM-RoPE is an effective inductive bias, not a theoretical expressivity necessity: sufficiently deep decoupled spatial and temporal attention can mix both signals. However, WorldPlay-style designs [Sun et al., 2025] instantiate a second attention path and do not expose the conjunction of camera address and pre-degradation time in a single score. Under same-position occlusion, the spatial term ties the clean anchor and recent occluder, leaving finite finetuning to infer the boundary through temporal cache order and ambiguous content (Appendix C). We therefore introduce PM-RoPE, a camera-conditioned extension of rotary position embedding (RoPE) [Su et al., 2024]. It folds camera-dependent phase offsets and zero-initialized value/output residuals into the pretrained self-attention path, providing an explicit joint spatiotemporal address without a second attention operation while preserving the backbone Q/K/V/O projections. The computational advantage is empirical rather than assumed; Table 11 reports the measured latency, memory, cache, and FLOPs.

For each frame i, we build a compact pose descriptor $\mathbf{c}_i$ based on the current camera extrinsic parameters $\mathbf{P}_i = [\mathbf{R}_i|\mathbf{t}_i]$ and focal lengths (f_i^x, f_i^y) as:

$$\mathbf{c}_i = [\mathrm{vec}(\mathbf{R}_i), \mathbf{t}_i, \log f_i^x, \log f_i^y]\,, \tag{1}$$

where translation and intrinsics are normalized. The released implementation is *focal-only*: it retains normalized (f_x, f_y) and sets the principal-point offset (c_x, c_y) to zero rather than consuming the full intrinsic matrix. In each self-attention layer, a zero-initialized MLP $h(\cdot)$ maps $\mathbf{c}_i$ to a per-frame rotary phase offset δ_i, which is added to the standard spatiotemporal phase in RoPE to indicate the relative positions between the camera poses at different frames:

$$\begin{aligned} \tilde{q}_i &= \mathrm{RoPE}_{\theta_i+\delta_i}(q_i), \qquad \delta_i = h(\mathbf{c}_i) \\ \tilde{k}_j &= \mathrm{RoPE}_{\theta_j+\delta_j}(k_j), \qquad \delta_j = h(\mathbf{c}_j) \end{aligned} \tag{2}$$

Inspired by PRoPE [Li et al., 2025], we inject camera-dependent transforms into the value and output paths of the self-attention module with zero-initialized residual projections $W_{\Delta V}$ and $W_{\Delta O}$:

$$\tilde{v}_j = v_j + W_{\Delta V}(\mathcal{P}_j^{-1} v_j), \qquad \tilde{y}_i = \sum_{j\le i} \mathrm{softmax}_j\left(\frac{\tilde{q}_i^\top \tilde{k}_j}{\sqrt{d}}\right)\tilde{v}_j, \qquad o_i = W_O\tilde{y}_i + W_{\Delta O}(\mathcal{P}_i\tilde{y}_i). \tag{3}$$

where $\mathcal{P}_j^{-1} v_j$ denotes the feature-map v_j back-projected from the camera coordinate system of frame j into the world coordinate system (and $\mathcal{P}_i \tilde{y}_i$ represents the subsequent projection into the query frame i). In our implementation, this is achieved by linear projections $W_{\Delta V}, W_{\Delta O}$ operating on the concatenated latent and normalized 6-DoF camera embeddings.

Since $W_{\Delta V}$, $W_{\Delta O}$, and $h(\cdot)$ are initialized as zeros, in the beginning of our finetuning process, PM-RoPE is the same as RoPE in the pretraining with $\delta_i = 0$ and $\tilde{v}_j = v_j$. As the finetuning process progresses, the PM-RoPE gradually takes the camera information into consideration so that the auto-regressive video generation can adaptively exploit historical frames according to the inter-frame relative camera poses in an adaptive manner.

3.3 Dynamic Memory with Streaming KV Cache

We partition the latent video into chunks and train with chunk-causal attention. During inference, dynamic memory is implemented as a streaming KV cache for auto-regressive video generation. Each generated chunk writes its keys and values into a KV cache for later chunks to read.

To ensure temporal plausibility, the key design is to preserve the original frame position of each cached chunk instead of treating them equivalently or compacting all cached entries into consecutive timesteps. This prevents old references from being confused with recent neighbors. For example, a clean reference chunk may be placed at positions $0, \ldots, m-1$, while the target video starts after a sampled gap of G chunks, at $Gm, \ldots, Gm+T-1$. Thus, the cache can store sparse, non-contiguous memories while still telling the model how far each memory item is from the current state.

3.4 Training Scheme for Dynamic Memory

To encourage the model to exploit the dynamic memory, we propose a novel strategy. We train with a flow-matching objective. For clean latent x_0, Gaussian noise ϵ, and scheduler value σ_t, the noised input and target are:

$$x_t = (1 - \sigma_t) x_0 + \sigma_t \epsilon, \qquad u_t = \epsilon - x_0. \tag{4}$$

The model predicts u_t and is optimized with a masked mean-squared loss over selected chunks. The training process includes multiple schemes, decided by different masks and timestep schedules:

The curriculum contains four complementary regimes. *All-history training* uses a chunk-causal pass in which each chunk attends to all previous chunks, matching the basic streaming inference path. *Noisy-memory* and *node-drop* training corrupt past chunks with high noise timesteps or replace interruption nodes with pure-noise latents while preserving at least one event anchor, forcing the model to ignore unreliable recent context and recover from surviving historical evidence. *V2V suffix/frontier training* keeps a clean or degraded prefix and supervises only the recovery suffix, matching chunk-wise autoregressive deployment. *Reference-cache training* prepends clean reference chunks from the undegraded clip at old positions and starts the target video after a sampled temporal gap, directly teaching retrieval from non-contiguous memory. Together, these regimes train the model to compare memory candidates with different visual reliability, camera poses, RoPE positions, and state histories, rather than attending only to a continuous prefix or fixed reference frame. Algorithm 1 gives the complete sampling, corruption, masking, and optimization procedure for one training step.

Dynamic Auxiliary Loss and Adaptive Weighting. Localized dynamic events often occupy only a small fraction of the canvas, so a standard flow-matching loss can be dominated by static background reconstruction. We therefore add a dynamic temporal-delta loss with adaptive weighting after warmup, increasing its influence for localized low-variance dynamics and reducing it for large camera or scene changes. Appendix B gives the full objective and hyperparameters.

4 Experiments

4.1 Evaluation Suite

We evaluate ReMind on both targeted hidden-state recovery and general I2V quality. For hidden-state recovery, we use the STEVO-Bench I2V protocol [Ma et al., 2026], where each example provides an initial image and an instruction describing an evolving process under occlusion, lights-off, or camera-lookaway distractors. We report the benchmark's five metrics: *State Progress* measures

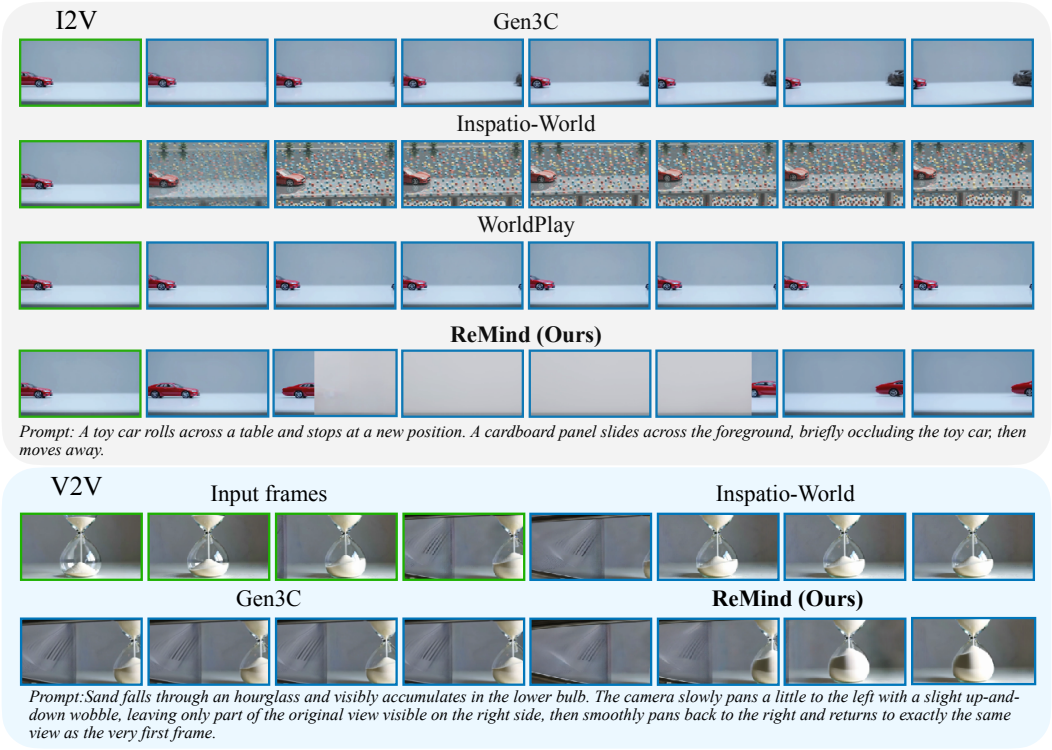

Figure 4: Qualitative comparisons with recent methods. WorldPlay is omitted from the controlled V2V recovery panel because no released result or interface is available for our degraded-prefix recovery protocol.

whether the hidden physical state evolves correctly; *Physical Plausibility* measures artifact-free dynamics; *Coherence* measures temporal scene consistency; *Observation Control* measures whether the requested distractor is applied; and *Action Control* measures whether the instructed action occurs. For general generation quality, we use VBench [Huang et al., 2024] to assess semantic accuracy, temporal quality, and visual quality in standard I2V settings. We separately use controlled V2V recovery, for which a degraded real prefix is supplied and the clean source suffix provides time-aligned ground truth. Appendix E defines both protocols and reports the matched controls and causal memory probes.

4.2 Quantitative Results: Out-of-Sight vs. General Quality

A critical requirement for dynamic memory frameworks is the ability to acquire strong out-of-sight state evolution capabilities without suffering from catastrophic forgetting of general image-to-video (I2V) priors. To demonstrate this, we conduct evaluations on two distinct fronts: STEVO-Bench [Ma et al., 2026] for targeted state evolution under occlusion/interruption, and VBench [Huang et al., 2024] for general generation quality and consistency.

State Evolution (STEVO-Bench [Ma et al., 2026]). We evaluate our method against the base model and recent memory-augmented baselines on the STEVO-Bench I2V protocol. As shown in Table 1, ReMind's dynamic non-local routing allows it to bridge temporal discontinuities, yielding a substantial improvement in state progress and physical coherence compared to other video models.

As shown in Table 1, while ReMind does not achieve the highest score in every individual metric, it achieves the highest official *Total* score by balancing these competing requirements. For instance, the InSpatio-World [InSpatio Team et al., 2026] baseline achieves exceptionally high Coherence (99.6%) but fails Observation Control (0.0%). This suggests that it often maintains visual consistency by avoiding requested degradations such as occlusions or lights-off, and therefore

Table 1: Quantitative I2V evaluation on STEVO-Bench. Metrics are percentages. Baseline scores are the official leaderboard values; our row uses the 1.3B main submission checkpoint. ReMind achieves the highest state progress and class-leading coherence among models without camera-guided image warping.

Model	State Progress ↑	Physical Plausibility ↑	Coherence ↑	Observation Control ↑	Action Control ↑	Total ↑
Video Models						
HunyuanVideo 1.5	4.1	42.1	59.1	31.2	**81.0**	217.5
WAN 2.2	7.7	52.0	58.4	46.2	76.5	240.8
CogVideoX	1.4	68.5	67.1	22.2	75.5	234.7
Camera-Controlled Models						
Gen-3C	0.0	30.6	82.4	**90.6**	57.6	261.2
HY-WorldPlay	0.0	72.2	88.2	55.4	64.9	280.7
Genie 3	2.9	15.2	27.3	84.7	78.6	208.7
Lingbot World	3.4	40.7	76.3	35.6	67.8	223.8
InSpatio-World	4.5	**79.6**	**99.6**	0.0	69.7	253.4
ReMind (Ours)	**14.5**	71.5	96.4	41.6	76.9	**300.9**

does not fully face the recovery challenge. In contrast, ReMind executes these partial observations (41.6% Observation Control) while maintaining strong Coherence (96.4%) and the best State Progress (14.5%). Genie 3 is closed; its row, like the other baseline rows, is an official leaderboard aggregate rather than a local rerun.

General Generation Quality (VBench [Huang et al., 2024]). Table 2 provides a cross-scale reference for standard I2V quality. Because InSpatio-World uses a 1.3B checkpoint and ReMind uses a 5B checkpoint, this comparison is not a component attribution. The matched 5B data-only/full comparison in Appendix E.3 provides matched point estimates, but no prespecified equivalence margin or confidence interval.

Table 2: VBench I2V evaluation under a common runner. InSpatio-World uses its released 1.3B checkpoint; ReMind uses the 5B full-method checkpoint.

Metric Group	InSpatio	**ReMind**
Semantic Accuracy ↑	90.29	**98.16**
Motion & Temporal ↑	76.11	**77.22**
Visual Quality ↑	56.83	**65.18**

4.3 Ablation Study

We conduct ablation studies to analyze three major components in ReMind: **1) The PM-RoPE architecture**, which injects camera geometry into attention; **2) the dynamic loss design**, which improves memory-aware optimization; and **3) the memory-oriented data construction pipeline**, which targets continuous scene evolution and is analyzed in Appendix F.

Failure archetypes. The five columns separate two failures that a single quality score would conflate. InSpatio-World has 99.6 Coherence but 0.0 Observation Control, consistent with *evading the interruption*. Conversely, Gen-3C executes the requested observation control (90.6) but has 0.0 State Progress; the official HY-WorldPlay result likewise has 0.0 State Progress. These are *restore-but-freeze* failures: the scene remains coherent or the control is rendered, but the hidden process does not measurably advance. ReMind raises State Progress to 14.5 while retaining 41.6 Observation Control and 96.4 Coherence. Thus Coherence should be interpreted jointly with both control execution and state advancement, rather than as recovery evidence by itself.

PM-RoPE Architecture. To validate the structural design of PM-RoPE (Sec. 3.2), we ablate the camera geometry injection pathways. We compare our complete architecture (**Full**) with two variants that remove the camera geometry injection in query/key (**VO only**) and value/output (**QK only**) separately, alongside our dual-branch, WorldPlay-style reimplementation [Sun et al., 2025]. This row is not the official HY-WorldPlay model reported in Table 1.

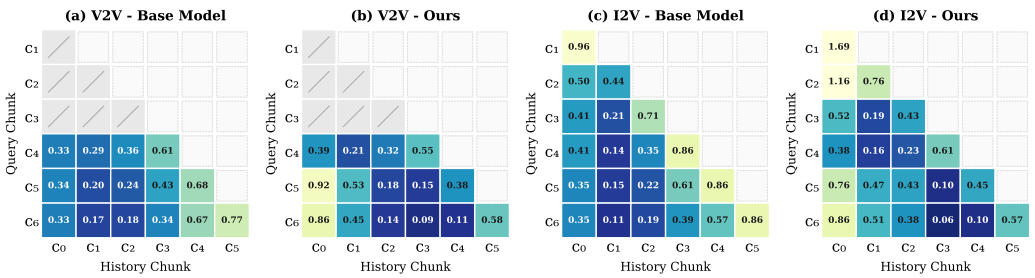

Figure 5: Selected KV-importance heatmaps (maximum over layers 20–24). Rows are query chunks and columns are history chunks; slashed cells in (a, b) are unavailable in V2V. These examples illustrate different attention patterns but do not establish population-level or causal memory use.

As shown in Table 3, the complete PM-RoPE architecture outperforms the partial variants in both low-level perceptual similarity (LPIPS) and high-level semantic consistency measured by Qwen3-VL [Bai et al., 2025] scoring (Appendix D). PM-RoPE also outperforms the **Dual Attention** baseline despite using a single attention operation, empirically supporting our analysis of decoupled branches under occlusion (Appendix C). These high-level consistency gains align with the stronger State Progress and Physical Plausibility observed in the main STEVO-Bench results. Appendix E.7 further measures this trade-off: PM-RoPE preserves the vanilla KV-cache size, whereas the matched dual-attention reimplementation uses a $2\times$ cache and higher streaming latency.

Table 3: Ablation of 1.3B PM-RoPE structural variants. All rows use the same initialization, data, curriculum, training budget, and evaluation set. "Dual Attention" is our WorldPlay-style `dual_-prope` reimplementation, not the official HY-WorldPlay checkpoint. Metrics are averaged across standard and noisy-memory retrieval conditions.

Architecture Mode	LPIPS ↓	Degradation Visible ↑	Post-Degradation Coherence ↑	Temporal Consistency ↑	Content Preservation ↑
Dual Attention (ours)	0.507	4.0	2.9	2.7	3.3
QK only	0.508	3.5	3.2	2.5	2.9
VO only	0.452	4.0	3.8	3.5	3.7
Full (Ours)	**0.403**	**4.1**	**3.9**	**3.8**	**3.9**

Table 4: Historical 1.3B dynamic-loss comparison. Regime probabilities match, but source sampling weights differ; the result should not be interpreted as a one-factor loss ablation.

Loss Variant	LPIPS ↓	Degradation Visible ↑	Post-Degradation Coherence ↑	Temporal Consistency ↑	Content Preservation ↑
Flow only	**0.395**	2.6	3.1	3.3	3.1
Dynamic + adaptive (Ours)	0.424	**3.4**	**3.5**	**3.6**	**3.6**

Dynamic Loss. We additionally report the historical dynamic-loss comparison used during method development. The **Flow only** run disables adaptive temporal-delta weighting, whereas **Dynamic + adaptive** enables it. Both retain the same regime probabilities, but their stored configurations use different source sampling weights; consequently, Table 4 is descriptive and is not a clean one-factor causal ablation. The adaptive run improves all Qwen3-VL [Bai et al., 2025] semantic dimensions, while the flow-only run obtains lower LPIPS, consistent with LPIPS favoring conservative or frozen predictions. A matched loss-only rerun is required for strict attribution.

4.4 Attention Map Analysis for Dynamic Memory

We use KV-importance maps to illustrate attention routing in selected V2V occlusion and I2V camera-loop cases (Fig. 5). In these examples, the base model concentrates attention near the diagonal. Under occlusion, our model assigns high displayed importance to the pre-occlusion anchor c_0 for queries c_5/c_6 (0.92/0.86); in the camera-loop example, attention shifts toward c_0/c_1 when the

camera returns. Because the figure selects cases and takes a maximum over layers 20–24, it is a qualitative diagnostic consistent with—but not proof of—historical-anchor retrieval.

4.5 Qualitative Visualizations

We present qualitative visualizations in Fig. 4 covering occlusion, camera motion, and re-observation after a positional gap. Together with the KV-importance heatmaps (Sec. 4.4), these selected cases are consistent with retrieval of historical evidence, but neither attention visualization nor selected samples establish general long-gap state evolution. Additional cases and a selected success gallery are provided in Appendix A.

5 Conclusion and Limitation

We present ReMind, empowering autoregressive video transformers to use their KV cache as dynamic memory for out-of-sight state evolution. By training on frame graphs with protected anchors, degraded intervals, and temporal gaps, ReMind learns to recover from reliable historical states instead of merely recent context. PM-RoPE further provides camera-aware spatiotemporal addressing while preserving pretrained attention pathways through one attention path. Experiments support short-horizon recovery under occlusion and darkness; camera-lookaway and long-duration state evolution remain challenging.

Matched controls refine this attribution: memory-oriented data teaches much of the core duration-conditioned recovery behavior, while the complete curriculum/PM-RoPE stack is most beneficial when retrieval must remain stable across viewpoint change. In a 24-case V2V intervention, removing the clean pre-interruption anchor reduces recovery PSNR, showing a measurable anchor contribution in that controlled protocol.

Limitation. The strongest evidence supports short-horizon out-of-sight recovery under occlusion and darkness rather than a general symbolic world state or unlimited-duration physical simulation. Camera lookaway and loop closure are evaluated, but their small subsets and weaker control/progress scores do not support the same claim strength. The duration probe further shows that elapsed-time evidence comes from rendered event content: an otherwise empty RoPE gap preserves an anchor but does not by itself advance state. The accompanying long-gap state sweep is therefore a negative result, not evidence of physical simulation over an unobserved duration. Finally, the pipeline relies on camera and depth quality; pose errors in web videos may weaken PM-RoPE supervision and bias training toward sequences with reliable tracks.

Broader Impact. Like other video generators, ReMind inherits risks around misleading media and dataset bias. We mitigate these through documented provenance, safety filtering, usage guidelines, a non-commercial research license, and release of restricted media as identifiers, manifests, annotations, or derived metadata rather than repackaged raw videos. Positively, ReMind also offers a diagnostic lens for hidden-state evolution, helping identify when visually plausible models fail to maintain persistent world states.

References

Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. VD3D: Taming large video diffusion transformers for 3D camera control. *arXiv preprint arXiv:2407.12781*, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Zhifei Chen, Tianshuo Xu, Leyi Wu, Luozhou Wang, Dongyu Yan, Zihan You, Wenting Luo, Guo Zhang, and Yingcong Chen. STANCE: Motion coherent video generation via sparse-to-dense anchored encoding. *arXiv preprint arXiv:2510.14588*, 2025.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

InSpatio Team, Donghui Shen, Guofeng Zhang, Haomin Liu, Haoyu Ji, Hujun Bao, Hongjia Zhai, Jialin Liu, Jing Guo, Nan Wang, Siji Pan, Weihong Pan, Weijian Xie, Xianbin Liu, Xiaojun Xiang, Xiaoyu Zhang, Xinyu Chen, Yifu Wang, Yipeng Chen, Zhenzhou Fan, Zhewen Le, Zhichao Ye, and Ziqiang Zhao. INSPATIO-WORLD: A real-time 4d world simulator via spatiotemporal autoregressive modeling. *arXiv preprint arXiv:2604.07209*, 2026.

Chunyang Li, Yuanbo Yang, Jiahao Shao, Hongyu Zhou, Katja Schwarz, and Yiyi Liao. ReRoPE: Repurposing RoPE for relative camera control. *arXiv preprint arXiv:2602.08068*, 2026.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3d vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Ziqi Ma, Mengzhan Liufu, and Georgia Gkioxari. Out of sight, out of mind? evaluating state evolution in video world models. *arXiv preprint arXiv:2603.13215*, 2026.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. OpenVid-1M: A large-scale high-quality dataset for text-to-video generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Suraj Ranganath, Vaishak Menon, and Anish Patnaik. KV cache quantization for self-forcing video generation: A 33-method empirical study. *arXiv preprint arXiv:2603.27469*, 2026.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. WorldPlay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

Team Wan. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989. doi: 10.1162/neco.1989.1.2.270.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WorldMem: Long-term consistent world simulation with memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yichen Xie, Depu Meng, Chensheng Peng, Yihan Hu, Quentin Herau, Masayoshi Tomizuka, and Wei Zhan. URoPE: Universal relative position embedding across geometric spaces. *arXiv preprint arXiv:2604.18747*, 2026a.

Yichen Xie, Chensheng Peng, Mazen Abdelfattah, Yihan Hu, Jiezhi Yang, Eric Higgins, Ryan Brigden, Masayoshi Tomizuka, and Wei Zhan. RAYNOVA: Scale-temporal autoregressive world modeling in ray space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25426–25437, 2026b.

Tianshuo Xu, Zhifei Chen, Leyi Wu, Hao Lu, and Ying-cong Chen. Motion Forcing: A decoupled framework for robust video generation in motion dynamics. *arXiv preprint arXiv:2603.10408*, 2026.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Ying-Cong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time interactive long video generation. In *International Conference on Learning Representations (ICLR)*, 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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 *SIGGRAPH Asia Conference Papers*, 2025. doi: 10.1145/3757377.3763833.

Wei Yu, Runjia Qian, Yumeng Li, Liquan Wang, Songheng Yin, Sri Siddarth Chakaravarthy P, Dennis Anthony, Yang Ye, Yidi Li, Weiwei Wan, and Animesh Garg. MosaicMem: Hybrid spatial memory for controllable video world models. *arXiv preprint arXiv:2603.17117*, 2026.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.

Cheng Zhang, Boying Li, Meng Wei, Yan-Pei Cao, Camilo Cruz Gambardella, Dinh Phung, and Jianfei Cai. Unified camera positional encoding for controlled video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38027–38037, 2026.

A Additional Qualitative Cases

We provide additional qualitative examples across various settings. These include Image-to-Video (I2V) recovery cases under occlusions and lighting changes (Fig. 6), Video-to-Video (V2V) generation (Fig. 7), and general I2V content generation (Fig. 8). These coverage-oriented figures are selected to span distinct settings; quantitative claims are computed on the complete evaluation manifests. Figure 9 additionally shows selected joint-success cases. In the coverage-oriented figures, green boxes denote given conditioning frames and blue boxes denote generated frames. This figure is illustrative rather than aggregate evidence. The archival artifact package will include the complete 221-video submission, output map, and per-case evaluation records needed to audit the sample.

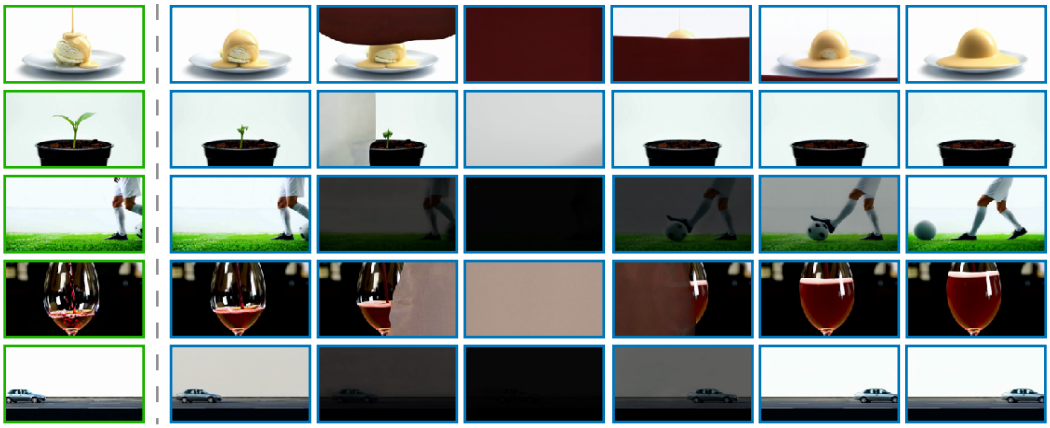

Figure 6: Selected Image-to-Video (I2V) recovery examples spanning occlusion and lighting-change prompts. These examples are illustrative; aggregate behavior is reported in the quantitative evaluation.

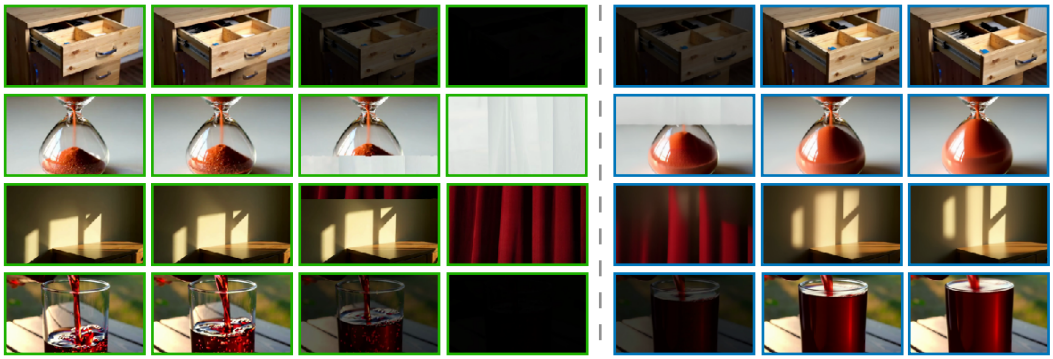

Figure 7: Selected Video-to-Video (V2V) recovery examples conditioned on provided prefixes.



Figure 8: Selected general Image-to-Video (I2V) generations from static input images.

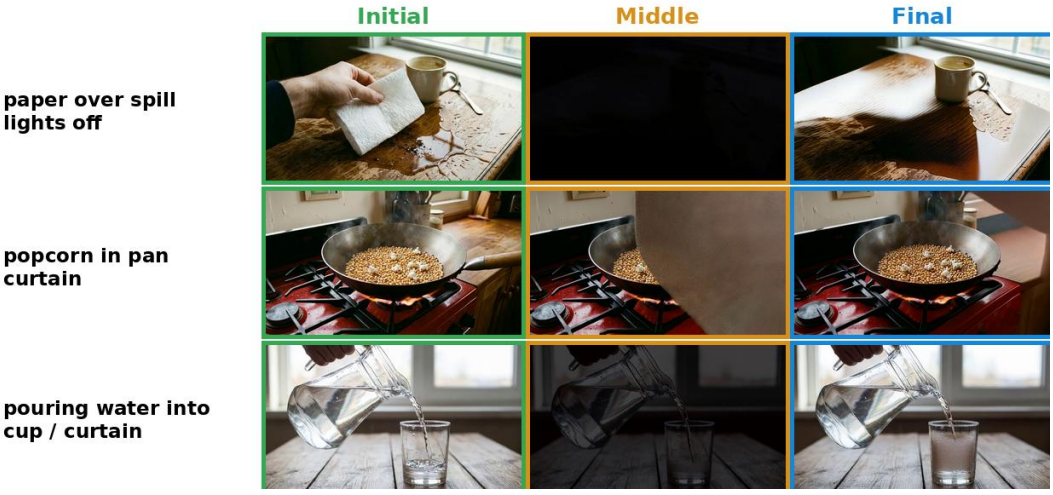

Figure 9: Selected joint-success cases from our 221-video official STEVO-Bench submission. Each displayed case receives positive official binary judgments for both Observation Control and State Progress and is selected for visual legibility. We show frames 0, 40, and 80; green, orange, and blue borders indicate the initial, middle, and final frames, respectively. Frame 40 is not asserted to be the event peak for every task. These selected successes are not representative of the aggregate submission; Table 1 remains the quantitative summary.

B Dynamic Auxiliary Loss Details

A core challenge in training out-of-sight state evolution is the imbalance of motion gradients. In localized dynamic events, such as water pouring into a cup within a largely static room, the state change only accounts for a small fraction of the canvas. Standard flow-matching loss averages errors across all pixels, causing the gradient direction to be dominated by static background reconstruction rather than the critical evolving state. To rectify this, we introduce a *Dynamic Temporal-Delta Loss* $\mathcal{L}_\Delta$ paired with an *Adaptive Weighting* strategy.

Alongside the main flow loss $\mathcal{L}_{\text{flow}}$ predicting the noise vector, we enforce an auxiliary penalty on the reconstructed clean latent $\hat{x}_0$ to match the ground-truth inter-frame changes Δx_0:

$$\mathcal{L}_\Delta = \mathbb{E}_{t,\epsilon}\left[\|(\hat{x}_{0,i} - \hat{x}_{0,i-1}) - (x_{0,i} - x_{0,i-1})\|^2\right], \tag{5}$$

where i denotes the frame index within a chunk. The total objective is $\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{adapt}}\mathcal{L}_\Delta$. Crucially, the weight λ_{adapt} is dynamically adjusted based on the global temporal variance σ_{batch} of the current batch:

$$\lambda_{\text{adapt}} = \alpha \cdot \exp(-\gamma \cdot \sigma_{\text{batch}}), \tag{6}$$

where $\sigma_{\text{batch}} = \text{Var}(\Delta x_0)$ computed over all spatial and temporal dimensions in the batch. We set $\alpha = 0.2$ and $\gamma = 5.0$. When global spatial change is small (e.g., static background with localized evolution), λ_{adapt} boosts the delta loss to provide a strong optimization signal for the hidden state. Conversely, for sequences with massive overall spatial variation (e.g., fast camera pans), the weight automatically decays, returning optimization priority to the standard flow loss to preserve structural stability. This auxiliary loss is enabled after the first 2,000 iterations of finetuning to ensure the model first stabilizes on the primary flow objective.

C Identifiability of Dynamic Memory under Same-Position Occlusion

This section clarifies the mathematical limitation we target. We do not claim that RoPE [Su et al., 2024] forbids long-range retrieval: for a query at time t_i and a key at time t_j, standard RoPE gives:

$$\langle R(t_i)q_i, R(t_j)k_j\rangle = q_i^\top R(t_j - t_i)k_j, \tag{7}$$

so the score depends on the relative temporal offset, but it is not generally a monotone function of distance. The issue is instead *identifiability*: a decoupled spatial-memory branch and a separately reframed temporal branch (like WorldPlay [Sun et al., 2025]) do not provide a single key for "the last clean state at this spatial address before a degradation boundary."

C.1 Problem Setup

Consider a recovery query q after an occlusion is removed. Let p denote the spatial address or camera ray of the dynamic region in the recovered view. There are two relevant cached candidates:

- **Clean anchor** (a): the last visible state before degradation, at time t_a and spatial address p.
- **Corrupted context** (c): a more recent frame during degradation, at time t_c and the same spatial address p, but whose visible content is the occluder or darkness rather than the underlying state.

The times satisfy $t_a < t_c < t_q$. The desired recovery edge is not simply "same spatial address" or "nearest previous time", but the conjunction

$$j^\star = \arg\max_{j<t_q} \mathbf{1}\{s_j = s_q\}\mathbf{1}\{t_j < \tau_{\deg}\}\rho(x_j, x_q), \tag{8}$$

where s_j is the spatial/camera address, $\tau_{\deg}$ is the start of the degradation interval, and ρ measures visual or state compatibility. In words, the key must jointly encode spatial alignment and whether the candidate is the clean pre-degradation state.

C.2 Decoupled Spatial and Temporal Scores

Many memory-augmented generators use separate mechanisms for spatial and temporal routing. Abstractly, a decoupled score has the form:

$$\ell(q, j) = \phi_{\mathrm{sp}}(s_q, s_j) + \phi_{\mathrm{tmp}}(\pi_q, \pi_j) + \phi_{\mathrm{cnt}}(x_q, x_j), \tag{9}$$

where ϕ_{sp} is a spatial or camera-address score, ϕ_{tmp} is a temporal-position score, and ϕ_{cnt} is a content score. In WorldPlay-style reconstituted context memory [Sun et al., 2025], the selected historical frames are written into compact cache slots. Therefore the temporal position π_j used by the RoPE branch is a *cache order* rather than the original absolute time or an explicit degradation-boundary coordinate.

For the clean anchor a and corrupted context c, same-position occlusion gives:

$$s_a = s_c = s_q \quad \Rightarrow \quad \phi_{\mathrm{sp}}(s_q, s_a) = \phi_{\mathrm{sp}}(s_q, s_c). \tag{10}$$

Thus a purely spatial or PRoPE-style branch cannot distinguish whether the same-ray cached entry is a clean state or an occluder. The decision is reduced to temporal cache order and content:

$$\ell(q, a) - \ell(q, c) = \big[\phi_{\mathrm{tmp}}(\pi_q, \pi_a) - \phi_{\mathrm{tmp}}(\pi_q, \pi_c)\big] + \big[\phi_{\mathrm{cnt}}(x_q, x_a) - \phi_{\mathrm{cnt}}(x_q, x_c)\big]. \tag{11}$$

Neither term explicitly contains the predicate $t_j < \tau_{\deg}$ from Eq. 8. If the content term is ambiguous or uninformative–for example, the query region is currently unobserved, blacked out, or covered by an occluder–then the score has no identifiable variable that selects the clean pre-degradation anchor over the more recent corrupted context.

C.3 Implication

The limitation is therefore not a theorem that RoPE must always prefer nearby tokens. Rather, a decoupled spatial-temporal memory lacks the joint address needed for hidden-state recovery: spatial routing can find the same ray, but not the clean state boundary; compact temporal routing can preserve local order, but not the original elapsed-time relation to the degradation event. ReMind addresses this by writing memory nodes with their original RoPE positions and camera metadata, and by training reference-cache and node-drop regimes in which the correct recovery depends on retrieving the clean anchor rather than the recent degraded context.

D VLM Evaluation Details

To comprehensively evaluate the state-recovery capabilities in our ablation studies, we utilize Qwen3-VL [Bai et al., 2025] to provide high-level semantic scoring. For each generated suffix, we prompt the VLM to act as an expert video reviewer and score the video from 1 (worst) to 5 (best) across four dimensions: Overall Quality, Temporal Consistency, Content Preservation, and Artifact Suppression (which correlates to the inverse of Degradation Visible).

Evaluation Protocol. We evaluate 200 clips randomly sampled from the recovery evaluation set across all degradation conditions. To ensure a blind and unbiased assessment, videos from different model variants were shuffled and presented to the LMM without identifying tags. Each video was evaluated three times using a temperature of 0.7 to account for stochasticity in the LMM's reasoning, with the final scores averaged across these trials. We used a fixed random seed of 42 for all sampling and inference steps. Tables 3–4 report point estimates and do not display confidence intervals; the archival artifact package will include per-example scores and the bootstrap script.

The standard prompt template provided to the model follows the structure below, which asks the VLM to observe the transition from the visible history context to the newly recovered state:

> *"You are an expert video reviewer evaluating the physical consistency of generated videos after a period of occlusion or degradation. Observe the initial state of the objects and the background in the first few frames, and then closely examine the frames after the degradation ends.*
> *Please rate the following dimensions on a scale of 1 to 5:*
> *1. Overall Quality: Is the recovered video visually coherent and realistic?*
> *2. Temporal Consistency: Do the objects resume their motion and trajectory naturally without abrupt jumps?*
> *3. Content Preservation: Are the identities, shapes, and textures of the original subjects completely preserved upon recovery?*
> *4. Artifact Suppression: Is the video free from flickering, structural morphing, or lingering textures from the occluder?"*

These individual scores are averaged across the evaluation set to yield the quantitative results reported in the ablation tables.

E Protocol Definitions and Additional Quantitative Results

E.1 I2V and V2V Protocols

We distinguish two evaluation protocols throughout the revision. In *image-to-video* (I2V), the input is a first frame and an instruction; the model must generate both the interruption and the subsequent recovery. STEVO-Bench and VBench use this protocol. In *video-to-video* (V2V), a clean or synthetically degraded prefix through the interruption peak is given, and the model generates the recovery suffix. Because the degradation is applied to a complete source video, V2V provides a time-aligned clean suffix and thus supports direct PSNR, SSIM, and motion-region measurements. Unless stated otherwise, the controlled V2V experiments below use 24 fixed-seed cases and a single generation seed per case. They are single-seed point estimates; no generation-seed variance or confidence intervals are available for these 24-case probes. In both protocols, the submitted video is decoded directly from the generator output: the inference path contains no post-hoc compositor, frame replacement, or benchmark-specific observation-control overlay.

E.2 STEVO-Bench Composition, Metrics, and Aggregation

The evaluation manifest contains 85 underlying projects expanded into 221 task variants: 69 use a cardboard occluder, 69 use a curtain occluder, 67 use a lights-off interval, and 16 use camera lookaway/return control. The cases span seven benchmark annotation levels. The benchmark paper describes 225 tasks; our official submission/evaluator snapshot contains 221 scored variants. All statistics in this paper refer to that exact 221-case snapshot, whose task identifiers and manifest version will accompany the archival artifacts. Each I2V case supplies an initial image and an instruction that specifies both the underlying action or physical state change and the visibility/control intervention; the future video is not provided to the generator. Our submission videos contain 81 frames at 16 fps and 480×832 resolution. Figure 1(a–c) shows representative light-off, full-frame-occlusion, and camera-lookaway cases with their initial, interrupted, and recovered frames.

The official evaluation reports five percentages. *State Progress* asks whether the underlying process reaches a measurably later state despite the unobserved interval. *Physical Plausibility* checks for physically credible, artifact-free dynamics. *Coherence* checks whether objects and the scene remain temporally continuous rather than resetting or cutting. *Observation Control* checks whether the requested occlusion, lighting, or camera intervention occurs, and *Action Control* checks whether the requested action is executed. If $z_{im} \in \{0, 1\}$ is the official case-level judgment for metric m on case i, the displayed metric is

$$S_m = \frac{100}{221} \sum_{i=1}^{221} z_{im}. \tag{12}$$

The Total column in Table 1 is the unweighted sum $\sum_m S_m$ and therefore ranges from 0 to 500. It is not an additional judge. All baseline entries in that table are official leaderboard aggregates; our entry is the official aggregate for the 1.3B full-data `checkpoint-1000` submission. We generate it with 25 denoising steps and seed 1000; each binary judgment is the majority of three Gemini-2.5-Flash evaluations.

E.3 Per-Dimension VBench Results and Dynamic-Degree Attribution

Table 5 expands the three VBench groups in Table 2 into all eight constituent dimensions. The three reported groups are equal-weight means: Semantic Accuracy uses I2V subject and background; Motion & Temporal

uses subject consistency, background consistency, motion smoothness, and dynamic degree; Visual Quality uses aesthetic and imaging quality.

Table 5: Per-dimension VBench I2V results under a common runner, seed set, and resolution. The 5B data-only and full-method arms share initialization, training data, 6,000 optimization steps, and 64 A100 GPUs. Data-only uses standard RoPE and omits the memory curriculum.

Dimension ↑	InSpatio-World	Data-only	ReMind
I2V subject	89.25	97.70	**97.72**
I2V background	91.32	98.54	**98.60**
Subject consistency	76.59	81.26	**81.62**
Background consistency	87.34	89.23	**89.44**
Motion smoothness	98.62	**99.16**	99.15
Dynamic degree	**41.87**	38.61	38.67
Aesthetic quality	54.90	60.12	**60.36**
Imaging quality	58.76	**70.09**	70.00

The 1.3B baseline and 5B full model differ on seven of eight dimensions, but that cross-scale comparison is descriptive rather than causal. Within the matched 5B trained arms, data-only and full differ by at most 0.4 points on every dimension, and both have nearly identical Dynamic Degree (38.61 versus 38.67). Thus the two matched arms have similar point estimates; without paired uncertainty or a prespecified margin, we do not claim equivalence. In the cross-scale comparison, 96 samples change from dynamic to static and 57 from static to dynamic; these counts should not be attributed to PM-RoPE or the curriculum because model size and training data also differ.

E.4 Matched-Compute Data and Generalization Controls

We isolate the contribution of memory-oriented data using matched 5B control runs. Data-only and full ReMind use the same pretrained initialization, complete training mixture, 6,000 steps, 64 A100 GPUs, and 256×448 training/evaluation resolution. The data-only arm uses standard RoPE and no memory curriculum. A leave-one-family-out arm removes all light-toggle examples from training.

Table 6: Matched-compute V2V recovery against the time-aligned clean suffix. Fog is absent from both the training degradation families and STEVO-Bench; the no-light arm tests transfer to a held-out training family. Each cell uses 24 fixed-seed cases.

Model	Fog PSNR/SSIM	Light PSNR/SSIM	Light + camera pan PSNR/SSIM
Raw Wan2.2-5B	8.5 / 0.31	7.6 / 0.24	—
Data-only	19.3 / 0.92	17.6 / 0.90	16.6 / 0.82
No-light training	19.7 / 0.92	**17.8 / 0.91**	—
ReMind	**19.8 / 0.92**	**17.8 / 0.91**	**18.3 / 0.87**

The static light result shows that the data teaches much of the core recovery behavior. Under viewpoint change, however, the complete method improves over data-only by 1.7 dB PSNR and 0.05 SSIM, consistent with the combined curriculum/PM-RoPE stack improving retrieval across camera motion. Removing the light family from training does not reduce light recovery at the reported precision, while the held-out fog condition remains far above the raw model. These controls support degradation-family transfer, while not by themselves separating curriculum from PM-RoPE.

Table 7: Production-checkpoint V2V fog-recovery slice. Both rows use the 5B backbone, 25 denoising steps, the same 24 held-out prefixes, a fixed seed, and a time-aligned clean suffix. Motion-region PSNR is computed inside the ground-truth motion mask.

Checkpoint	Full PSNR ↑	Full SSIM ↑	Motion PSNR ↑
Raw Wan2.2-5B initialization	10.25	0.484	9.789
ReMind teacher	**19.93**	**0.923**	**15.592**

Table 7 is distinct from the matched-compute controls in Table 6: it uses production checkpoints and therefore should not be used for data-versus-method attribution.

E.5 Duration-Conditioned State Advance

We fix the visible prefix and vary only the length K of a rendered dark interval from one to four chunks. Recovery is measured at a fixed offset after the lights return. The objective statistic

$$\Delta(K) = \mathrm{SSIM}(x_K^{\mathrm{rec}}, x_K^{\mathrm{GT}}) - \mathrm{SSIM}(x_K^{\mathrm{rec}}, x_{\mathrm{pre}}^{\mathrm{GT}})$$

is positive when recovery is closer to the time-matched state than to the pre-interruption state.

Table 8: Objective duration-conditioned state advance on 24 V2V cases using the 5B backbone and one fixed generation seed. The percentage is the fraction with $\Delta(4) > 0$ across cases, not uncertainty over seeds.

Model	$K = 1$	$K = 2$	$K = 3$	$K = 4$	Positive at $K = 4$
Raw Wan2.2-5B	-0.001	$+0.007$	$+0.015$	$+0.014$	62.5%
ReMind teacher	$+0.068$	$+0.112$	$+0.141$	$+0.144$	95.8%
ReMind 4-step	$+0.079$	$+0.118$	$+0.150$	$\mathbf{+0.164}$	95.8%

The objective curve increases with hidden duration for both the teacher and four-step student. Independent Gemini and Qwen judges agree in direction, although the objective metric resolves the short-duration differences more reliably. A separate RoPE-offset-only probe retains the anchor but produces approximately flat state advance. Thus, the evidence supports a narrower mechanism: rendered event content supplies elapsed-time evidence, while PM-RoPE and the cache provide robust anchor retrieval. The current checkpoint does not convert an otherwise empty positional gap into extrapolated physical time.

E.6 Causal Anchor Knockout and Long-Gap Retrieval

To quantify the contribution of the pre-interruption image, we remove only the clean anchor from the cache while keeping the black history chunks, latent inputs, positions, prompt, and seed fixed. In this 24-case, one-seed V2V protocol, Table 9 shows that anchor removal reduces recovery PSNR by 2.0 dB. The intervention establishes a contribution within this protocol; it does not by itself rule out other conditioning pathways.

Table 9: Anchor-cache intervention on 24 black-interruption V2V cases using the 5B four-step student.

Cache contents	PSNR ↑	SSIM ↑
Anchor + dark history	**18.32**	**0.909**
Anchor only	17.04	0.863
Dark history only (anchor removed)	16.34	0.881

We additionally place the recovery query at increasingly distant, non-contiguous RoPE positions while keeping the stored cache size fixed.

Table 10: Retrieval and state behavior as the positional gap G increases for the 5B four-step student on 24 cases and one fixed generation seed. Anchor mass is mean recovery attention assigned to the original clean anchor. State advance is the Qwen3-VL process-progress score of the recovery frame minus that of the visible prefix. The 12–25 gap range was configured for reference-cache training but inactive in this checkpoint, making this a retrieval diagnostic rather than evidence of trained gap extrapolation.

G	1	2	4	8	12	16	25	32	40	60
Anchor mass	.834	.818	.812	.858	.849	.839	.745	.879	.834	.694
State advance	+3.1	+4.4	+3.3	-2.3	-0.4	-2.3	+2.1	-1.3	-6.7	-1.0

At $G = 60$, ReMind stores 5,850 tokens, compared with 73,710 tokens for a naive stream retaining every intervening chunk (12.6× more). Across four repeated interruptions, attention mass on the original anchor

is 0.849/0.486/0.387/0.401: newer recoveries absorb some attention, but the original scene witness remains addressable. This probe measures attention routing and cache cost. In contrast, per-case progress versus G has mean Spearman correlation -0.19 (median -0.17), and the aggregate state-advance row remains approximately flat around zero. Thus long-range addressability survives, but an empty positional gap neither supplies elapsed-time evidence nor improves recovery-state advancement.

E.7 Measured PM-RoPE Efficiency

Table 11 reports a controlled single-A100 inference benchmark at 480×832, 81 frames, and bfloat16. Each rollout contains seven chunks, with four denoising calls and one context-write call per chunk; values are medians over five post-warmup runs. Dual attention is our matched WorldPlay-style reimplementation used in Table 3, not the released HY-WorldPlay model.

Table 11: Measured streaming efficiency. KV cache is measured at full 21-latent- frame history; FLOPs are analytic self-attention FLOPs per block at the final chunk.

Backbone	Mode	Time (s)	Peak GB	KV GB	GFLOPs/block
1.3B	Vanilla RoPE	7.48	9.25	6.04	1030
1.3B	PM-RoPE	9.00	9.69	**6.04**	1075
1.3B	Dual attention	12.67	15.77	12.08	1973
5B	Vanilla RoPE	3.95	13.22	3.02	206
5B	PM-RoPE	4.88	14.42	**3.02**	250
5B	Dual attention	6.05	16.97	6.04	324

PM-RoPE leaves KV-cache size unchanged relative to vanilla RoPE, whereas the dual branch duplicates it. Dual attention is $1.41\times$/$1.24\times$ slower than PM-RoPE on the 1.3B/5B backbones, respectively. Its latency also grows faster with retained history (54 versus 30 ms per added chunk at 1.3B), because the second attention pass rereads the growing duplicated cache. These are inference-only measurements on one A100; they do not estimate distributed training cost. PM-RoPE itself is $1.20\times$/$1.24\times$ slower than vanilla RoPE in the 1.3B/5B implementations, so "one attention path" does not mean zero overhead. Absolute values should only be compared within a backbone because the two implementations use different token layouts and cache dimensions.

Table 12: Main training data components.

Component	Source	Role
General videos	OpenVid, DL3DV-10K	preserve appearance and camera motion
Natural dynamics	Pexels + VLM filtering	visible state evolution and interactions
Hard dynamics	generative video models	high-occupancy state changes
Camera loops	Pexels + Neoverse rendering	loop-closure memory supervision

Table 13: Concurrent source-sampling distribution for the 1.3B full-data run. Weights sum to one; they are sampling probabilities, not sequential stages.

Source	Weight	Source	Weight
OpenVid DA3	0.10	DL3DV DA3	0.20
Pexels DA3	0.16	Helios pouring	0.04
Helios curriculum	0.10	Helios broad dynamics	0.10
Helios loop/pan	0.05	Helios STEVO-gap	0.12
Veo3 occlusion	0.03	Pexels pan/evolution	0.10

Scale and exposure accounting. The weights in Table 13 are configured draw probabilities, not realized exposure counts. The current artifact package does not contain a frozen step-level exposure log for the historical 1.3B run, so we cannot audit exact accepted unique clips/hours, per-source draw counts, or cross-source deduplication for that run. Candidate-pool and public-release row counts must therefore not be interpreted as the realized training size. The archival artifact ledger must include the pinned input manifests, their hashes and accepted-row counts, the deduplication and split unit, and a realized per-source draw census before making a stronger scale claim.

F Ablation of Data Construction

We study memory-oriented data construction through three *separate* trainings from the same initialization and for the same number of steps. Within each run, all listed sources are sampled concurrently by a weighted sampler; the rows are cumulative data mixtures, not sequential training stages. The *Base* variant uses general videos (OpenVid [Nan et al., 2025] and DL3DV [Ling et al., 2024]). *Real Dynamics & Loops* adds Pexels and NeoVerse-rendered camera loops, and **Full** additionally adds generated hard dynamics.

Table 14 is a descriptive comparison of cumulative mixtures. The real-dynamics-and-loops row has the lowest LPIPS, while the full mixture has the highest reported post-degradation coherence, temporal consistency, and content-preservation scores but worse LPIPS. Because sources are bundled and uncertainty is unavailable, the table does not isolate NeoVerse or establish statistical significance for an individual source.

Table 14: Cumulative 1.3B data-mixture ablation. Each row is a separate, matched-budget run whose listed sources are mixed concurrently; rows are not stages of one curriculum.

Data Mixture	LPIPS ↓	Degradation Visible ↑	Post-Degradation Coherence ↑	Temporal Consistency ↑	Content Preservation ↑
Base (General Videos)	0.422	2.0	1.9	1.8	1.9
+ Real Dynamics & Loops	**0.363**	**2.2**	2.2	2.0	2.3
+ Synthetic Hard Dynamics	0.515	**2.2**	**2.4**	**2.2**	**2.6**

G Data Construction Details

The main components of our training data are listed in Table 12.

G.1 Dynamic Taxonomy and Retrieval Policy

We build a broad dynamic taxonomy covering over one hundred scene-event types. Each type receives an importance score that reflects three factors: whether the state change is visually salient, whether the event is common enough to collect at scale, and whether it is useful for hidden-state recovery after interruption. The high-priority groups include monotonic state changes such as pouring/filling, burning, melting, growing, and diffusion, as well as large-occupancy motion, deformation, lighting/weather changes, and human-object interactions.

The taxonomy serves two roles. For Pexels retrieval, it provides keyword queries and negative filters. For generated or VLM-annotated clips, it acts as caption guidance: captions must describe the object, the state variable, and the temporal direction of change. After retrieval, Qwen3-VL [Bai et al., 2025] is used for secondary filtering. It estimates whether the clip contains a scene cut, which dynamic event is present, how much image area the dynamic region occupies, and how long the event remains visible. We keep high-occupancy, temporally sustained dynamic clips and discard clips dominated by static scenery, montage cuts, or tiny localized motion.

Table 15: Dynamic retrieval taxonomy. The full internal taxonomy covers over one hundred event types; this table groups them by retrieval and captioning role. Counts are from the tagged curation table and are not mutually exclusive.

Tag group	Candidates	Retrieval keywords and caption cues
Translational	39,205	walk/run, vehicle motion, fall/drop, fly/float; caption moving object, direction, and speed
Monotonic state	20,076	pouring/filling, burn, melt, grow/bloom, diffuse/spread; caption state variable and monotonic change
Deformation	13,793	dance/yoga, body gesture, fabric motion, bubbles; caption shape change and actor/object
Lighting/weather	11,745	sun, rain/snow, wind/fog, cloud shadow, light switch; caption illumination or weather transition
Interaction	10,218	cooking, mixing, assembly, ball sports, tool use; caption actor-object interaction and outcome
Periodic	3,210	mechanical rotation, biological cycles, water periodic motion; caption periodic source and phase
Discrete	1,147	doors, lids, page flips, switches; caption trigger, before-state, and after-state

G.2 Event Nodes and Degradations

For each selected clip, we construct a clean video, an optional degraded video, a caption, and a set of latent chunk nodes used by the memory curriculum. The main event families are camera loops, light toggles, moving occluders, and zoom or camera perturbations. Camera-loop nodes are derived from the trajectory: the memory anchor is the initial view, and the recovery node is the later frame whose camera pose returns nearest to that view. Light-toggle nodes are derived from the dimming schedule, with anchors before the dark interval and recovery nodes after relighting. Occluder nodes are derived from the enter, full-cover, and exit frames. Zoom nodes are derived from peak crop scale and recovery frames, with intrinsics updated to match the crop.

Targeted return-pose sanity check. For the 12 fixed `pan_yaw_8dir` trajectories used in our project-page and evaluation examples, we compute rotation relative to frame 0 and define a return as re-entering $\max(2^\circ, 0.1\, r_{\max})$ after the interior rotation peak. All 12 trajectories peak at frames 38–42 with $r_{\max} = 14.05^\circ$–35.56°, re-enter the return threshold at frames 63–65, and end with only 0.04°–1.20° residual rotation. Nonzero camera-center residuals remain, so this verifies orientation return for these 12 trajectories rather than exact pixel closure or corpus-wide pose quality. We therefore define return supervision using pose overlap and protect the initial witness instead of requiring the final camera center to equal the initial center. This selected-example sanity check is not a substitute for a corpus-wide retention statistic or a real-versus-rendered loop ablation.

All sources are normalized into the same training format. Clips are sampled as 81 frames at 16 fps and resized to the training resolution (480×832 in our main runs). After VAE temporal compression, this gives 21 latent frames, which we group into seven chunks of three latent frames. Intrinsics are scaled to the training resolution; extrinsics are stored as OpenCV camera-to-world matrices with the first frame anchored as the origin. During model forward passes, intrinsics are normalized by image width and camera translations are centered and scaled per clip.

H Reproducibility, Compute, and Release Details

H.1 Training Configuration

The main submission experiments fine-tune a 1.3B autoregressive video diffusion transformer initialized from the InSpatio-World/Wan2.1 1.3B checkpoint family (Team Wan, 2025, InSpatio Team et al., 2026). Its full-data run uses AdamW at 10^{-5}, 100 warmup steps, 30,000 optimization steps, bfloat16, FSDP, gradient checkpointing, seed 42, and 64 A100-80GB GPUs. The matched recovery, duration, anchor, and detailed VBench controls in Appendix E instead use the 5B Wan2.2-TI2V backbone and are labeled separately; their matched training runs use AdamW at 5×10^{-6} with $\beta = (0.9, 0.95)$, weight decay 10^{-4}, gradient clipping at 1.0, and 6,000 low-resolution steps on 64 A100 GPUs. In both model families the VAE and text encoder are frozen, all DiT and PM-RoPE parameters are trainable, and only newly added PM-RoPE modules are zero-initialized.

For the 1.3B full-data run, 43% of steps use the V2V family. The remaining steps sample all-history/node-drop/noisy-memory with relative probabilities 0.56/0.30/0.14; within V2V, 22% use reference-cache and 25% of the remainder use a two-chunk frontier. The resulting total mixture is approximately 31.9% all-history, 17.1% node-drop, 8.0% noisy-memory, 25.2% plain V2V, 8.4% frontier, and 9.5% reference-cache. Reference-cache uses gaps of 12–25 chunks. In contrast, the 5B checkpoint used by the gap-attention probe in Table 10 had reference-cache probability zero; that table is therefore explicitly a routing diagnostic, not evidence that this 5B checkpoint learned elapsed-time extrapolation from reference-cache training.

Algorithm 1 specifies one complete training step. Our scheduler is indexed so that $k = 0$ has $\sigma \approx 1$ (pure noise) and $k = 999$ has $\sigma \approx 0$ (clean). An 81-frame video becomes 21 VAE latent frames and $N = 7$ chunks of $m = 3$ latent frames. The per-frame loss mask M may contain zero or soft weights: it changes supervision only. All chunks participate in the same chunk-causal forward pass, and history K/V tensors are never detached. Therefore, loss on a recovery frontier backpropagates through the history representations it reads, even when those history frames have zero direct loss.

The main data mixture is the normalized 81-frame format described in Appendix G. Training sources include OpenVid [Nan et al., 2025], DL3DV [Ling et al., 2024], Pexels-based dynamic clips, Pexels pan/evolution camera-loop data, Veo3 full-frame occlusion clips, and Helios-generated hard dynamics [Yuan et al., 2026]. In the production full-data configurations, the sampler mixes real DA3-packed [Lin et al., 2025] sources with generated dynamics and loop sources. V2V-heavy runs use degradation-aware recovery training with frontier continuation and reference-cache reads, while I2V-aligned runs use clean first-frame seeding and node-drop training to align train/test cache structure.

Algorithm 1 One ReMind training step.

Require: video V, caption y, cameras C
Require: event boundaries E, degradation peaks P, optional reference V^{ref}
$x_0 \leftarrow \mathrm{VAE}(V)$ ▷ VAE frozen
$e_y \leftarrow \mathrm{TextEnc}(y)$ ▷ text encoder frozen
If present, $x_0^{\mathrm{ref}} \leftarrow \mathrm{VAE}(V^{\mathrm{ref}})$
Split x_0 into N chunks of $m = 3$ latent frames; sample $\epsilon \sim \mathcal{N}(0, I)$
Sample $k_c \sim \mathcal{U}\{0, \ldots, 999\}$ for every chunk c; set $M \leftarrow \mathbf{1}$ and positions $p_i \leftarrow i$
Sample regime r from the configured curriculum distribution
if $r = \mathrm{NoisyMemory}$ **then**
    $k_{1:N-1} \sim \mathcal{U}\{500, \ldots, 984\}$; $M_{1:N-1} \leftarrow 0$ ▷ last chunk is the target
else if $r = \mathrm{NodeDrop}$ **then**
    for each segment $S = [E_j, E_{j+1})$ **do**
        Drop each $c \in S$ independently with probability 0.35; if all are dropped, retain one uniformly
        $k_c \leftarrow 0$ for each dropped c ▷ pure-noise history; full-clip loss remains
    end for
else if $r \in \{\mathrm{V2VSuffix}, \mathrm{V2VFrontier}\}$ **then**
    Choose split s from P and pin $k_{c<s} \leftarrow 999$ ▷ clean teacher-forced history
    For suffix, set $M_c = \mathbf{1}[c \geq s]$; for frontier $[f, f + w)$, set $M_c = \mathbf{1}[f \leq c < f + w]$
else if $r = \mathrm{ReferenceCache}$ **then**
    Prepend clean chunks from x_0^{ref} and retain the target near its degradation peak
    Sample $G \sim \mathcal{U}\{12, \ldots, 25\}$; assign reference positions from 0 and target positions from Gm
    Pin reference and target seed clean; sample frontier $[f, f + w)$ and set $M_c = \mathbf{1}[f \leq c < f + w]$
end if
if the batch uses I2V-aligned seeding **then**
    Pin the configured first latent frame(s) clean and set their direct loss weights to zero
end if
Apply configured pre-peak and recovery-region soft weights to M when applicable
$\sigma \leftarrow \Sigma(k)$; $x_k \leftarrow (1 - \sigma)x_0 + \sigma\epsilon$; $u \leftarrow \epsilon - x_0$
$\hat{u} \leftarrow f_\theta(x_k, k, p, C, e_y$; chunk-causal PM-RoPE attention$)$
$\mathcal{L}_{\mathrm{flow}} \leftarrow \frac{\sum_i M_i \| \hat{u}_i - u_i \|_2^2}{\sum_i M_i \, d}$; $\hat{x}_0 \leftarrow x_k - \sigma\hat{u}$
$\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{flow}} + \lambda_{\mathrm{adapt}} \mathcal{L}_\Delta(\hat{x}_0, x_0, M)$ ▷ Appendix B
Backpropagate $\mathcal{L}$ through the single full-sequence graph; clip gradient norm and take one AdamW step

H.2 Evaluation Protocols

STEVO-Bench [Ma et al., 2026] and VBench [Huang et al., 2024] I2V generation follow their respective evaluator snapshots. Our STEVO-Bench snapshot uses 81 frames at 16 fps and 480×832; each VBench comparison uses a common format across its reported arms. We use fixed generation seeds and per-sample manifests so aggregate scores can be traced back to generated videos, prompts, checkpoint identifiers, and metric outputs. For ablation tables scored by Qwen3-VL [Bai et al., 2025], we report means over the fixed evaluation set and release the per-example scores and bootstrap resampling script used to estimate confidence intervals over examples.

Table 16: Protocol and model-size ledger for the quantitative result families. "Official" denotes the STEVO-Bench leaderboard evaluator; the other rows use the controlled protocols defined in this paper.

Result family	Backbone	Evaluation provenance
STEVO-Bench main	1.3B	Full-data `checkpoint-1000`; official task list and three-vote aggregates; seed 1000; 25 steps; 81 frames at 16 fps and 480×832
VBench I2V	1.3B baseline / 5B trained arms	Released InSpatio-World baseline and matched 5B data-only/full checkpoints under one runner, seed set, and resolution
PM-RoPE, loss, and data ablations	1.3B	Fixed 200-clip recovery set; seed 42; each Qwen3-VL score averaged over three trials
Matched recovery controls	5B	Matched initialization and 6,000-step budget; 24 fixed-seed cases at 256×448
Production fog recovery	5B	Production checkpoints; 25 denoising steps; 24 held-out, time-aligned cases
Efficiency	1.3B / 5B	DiT-only streaming benchmark on one A100-80GB; median of five runs

H.3 Artifact Map and Causal Probe Entry Points

Our evaluation codebase keeps probe entry points separate from training. The held-out fog-transfer experiment in Table 7 is implemented by `tools/run_remind_fog_holdout.py`. It applies a ramp-in/hold/ramp-out fog event to latent-history chunks 2–4, pins the first five chunks, and scores the two generated recovery chunks against the time-aligned clean suffix. The anchor intervention in Table 9 is implemented by `tools/run_remind_anchor_knockout.py`; it holds the input latents, RoPE positions, prompt, denoising schedule, and seed fixed while selecting `full`, `no_anchor`, or `anchor_only` cache content. Both scripts expose checkpoint paths, denoising steps, shard index/count, and noise seed as command-line arguments, consume the state-v2 task and source manifests, and write per-case metrics together with generated videos. Their default evaluation format is 81 frames at 16 fps and 480×832, with seed 20260711. A minimal invocation requires only `-output-dir`; raw-base and teacher controls are selected with `-skip-generator-checkpoint-load` and `-no-lora -steps 25`, respectively.

The remaining controlled analyses use the same interface and manifest family: `tools/run_remind_dark_duration.py` produces the duration-conditioned state-advance measurements in Table 8; `tools/run_remind_gap_probe.py` sweeps the hidden RoPE gap and records per-layer anchor-attention mass for Table 10; and `tools/bench_pmrope_efficiency.py` benchmarks the standard, PM-RoPE, and dual-attention modes in Table 11. The gap runner additionally writes one generated recovery video and one metrics record per case and gap, so the archival output package is designed to permit re-judging without rerunning generation. Each causal runner writes generated videos and a `metrics.json` under each case directory, plus `summary_shard0.json`; shard outputs can be merged by case identifier without averaging shard means. The efficiency runner writes one JSON record per attention mode, including per-chunk time, peak allocated memory, KV-cache size, and analytic final-chunk attention FLOPs. These probe scripts and their pinned manifests are provenance identifiers here, not a claim that every entry point is already in the current public repository; they will be packaged with the archival artifacts. Given the official per-case STEVO summary and a directory containing the submitted videos, `supplementary/build_stevo_selected_gallery.py` regenerates Fig. 9; the script validates the joint official judgments and records the legibility-selected task identifiers.

H.4 Compute Resources

All reported training and evaluation jobs were run on Lilypad using A100 80GB GPU workers. Debug and evaluation jobs used 8 GPUs, the main I2V-aligned and V2V ablation runs used 32 GPUs, and the full-data production runs used 64 GPUs. Helios data generation jobs used 8-GPU shards for the curriculum sources used in the training mix. We will release the corresponding run configuration records and per-run GPU-hour accounting so the compute budget can be audited for each reported table.

H.5 Public Research Release and License

The current project page provides released 5B models, DMD-EMA weights, inference code, and examples. A separate manifest-first data release contains 1,419,331 released rows (1,027,329 Helios generated rows, 265,945 pair-fixed InSpatio camera-augmentation rows, and 126,057 Kubric camera-augmentation rows). These are release-manifest rows, not unique-video counts or the realized exposure census of the historical 1.3B run. The archival research package will additionally include the training, data-construction, and evaluation code; exact run configurations; evaluation manifests; prompts and seeds; generated-video metadata; per-example metric records; and the scripts used for STEVO-Bench [Ma et al., 2026], VBench [Huang et al., 2024], causal probes, and ablation tables. We will include checksums and a table-to-artifact index so every reported aggregate can be traced to its cases and checkpoint. The package will use a non-commercial research license rather than an OSI-approved open-source license. Existing assets retain their upstream licenses and terms. Public or third-party media that cannot be redistributed directly will be represented by source identifiers, manifests, derived annotations, camera/depth metadata, and preprocessing instructions rather than repackaged raw videos.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's intended contributions: PM-RoPE, streaming KV memory, graph/node-structured training, STEVO-Bench evaluation, and the reported overall performance. These claims are scoped to the experiments and diagnostics reported in the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Sec. 5 discusses limited event coverage, dependence on estimated geometry, and the diagnostic rather than causal status of KV-importance.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper proposes a model and training/evaluation protocol, but does not claim formal theoretical theorems requiring proofs.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [No]

   Justification: The manuscript documents the architecture and major protocols, and a 5B inference release is available. However, the 1.3B headline checkpoint, complete training/evaluation code, pinned manifests, realized training-exposure census, and per-example records needed to reproduce every main result are not yet publicly available.

   Guidelines:

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

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

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

   Answer: [No]

   Justification: The current public research release contains 5B models, DMD-EMA weights, inference code, examples, and manifest-first data. The 1.3B benchmark checkpoint, training and evaluation code, pinned result manifests, and table-to-artifact index remain planned for the archival package; therefore the current release does not yet reproduce all main results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.

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

6. **Experimental setting/details**

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

   Answer: [No]

   Justification: The paper gives the main optimizer, architecture, source weights, memory regimes, and evaluation formats, but does not yet provide the historical run's exact accepted-clip census, realized per-source exposure, cross-source deduplication, complete split hashes, or every table's pinned configuration.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The taxonomy and conditional STEVO analyses include bootstrap intervals, but the main architecture/loss/data ablations and 24-case controls are single-seed point estimates without confidence intervals or training-seed replication.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: Appendix H reports A100-80GB worker counts by job family, but per-run wall-clock time, GPU-hours, failed/preliminary-run compute, and total project compute are not yet available.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research is a generative-video modeling study using cited public, synthetic, and internally generated assets, and Sec. 5 discusses responsible-release considerations.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Sec. 5 discusses positive impacts for diagnosing world-model memory failures and negative impacts related to synthetic video misuse, privacy, and biased web-video data.

    Guidelines:

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

11. **Safeguards**

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

Answer: [No]

Justification: The release uses a non-commercial research license, provenance records, and identifier-only handling for restricted media, but the paper does not yet document enforceable misuse controls, detailed unsafe-content filtering thresholds/audits, or a reporting and takedown process.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [No]

Justification: The paper cites the major assets and states the redistribution policy, but it does not yet provide an asset-by-asset ledger with exact versions, licenses/terms, source URLs, redistribution basis, and treatment of identifiable people.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [No]

Justification: The current release documents and provides several 5B/model/data assets, but the headline 1.3B checkpoint, complete evaluation artifacts, unified manifest-loader documentation, and table-level provenance package are not yet available.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not describe crowdsourcing experiments or new human-subject studies.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve new human-subject studies or direct interaction with study participants.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: Sec. 3.1 states that VLM-generated captions are used in dataset construction and event-node conditioning, which is part of the methodology rather than only writing assistance.

Guidelines:

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