WHICH LATENT REPRESENTATIONS SUPPORT OBJECT-PERMANENT GENERATION?

Anonymous authors
Paper under double-blind review

ABSTRACT

Video generators often fail to preserve objects that leave the view: occluded objects change identity, fail to return or evolve inconsistently. We ask whether the latent a generator is trained on can natively support object-permanent generation. Our intuition is that a latent encoding the whole scene across cameras and time, rather than one camera's partial view, can represent objects beyond their visibility. We build PERMAKUBE, a synchronized multi-camera dataset with controlled occlusions and object-level annotations, design PERMATOK, a scene tokenizer trained to render unobserved cameras and times, and train generators on frozen video and scene latents. Under every tokenizer supervision we test, scene latents preserve occluded objects more often than video latents despite lower reconstruction fidelity (78.7% versus 47.7% preservation), while accurate motion after occlusion remains challenging for all latents. Cross-view targets add what same-camera targets lack: their encoded scene tokens let a probe trained only on visible objects recover objects hidden from one camera, and their generations reach twice the hide-and-reappear success at similar identity preservation. PERMATOK also adapts to monocular input while retaining strong object-permanent generation, and scene latents are effective diffusion targets built from sparse multi-camera observations. These findings favor scene latents for object-permanent generation and show that identity preservation does not guarantee a readily recoverable hidden-object state, motivating latent probing alongside frame-level evaluation.

1 INTRODUCTION

Today's visual generators can produce strikingly photorealistic footage yet violate basic physics. A telling case is the out-of-sight, out-of-mind artifact: objects change identity, fail to return, or evolve inconsistently after disappearing from view, indicating that the generator failed to preserve a persistent state of the scene (Ma et al., 2026). We decompose the state-preserving property into two aspects: *object permanence*, i.e., preserving an object's identity across occlusion, and *dynamics*, i.e., correct evolution of its motion over time. In this work, we focus on object permanence, as it is a precondition for dynamics preservation. Recent work hypothesizes that the failure lies partly in the latent on which generators are trained: the latent of a monocular video provides no direct observation of the object while it is invisible, leaving its hidden state to be inferred from other times or learned priors (Shin et al., 2026; Chen et al., 2026b; Duan et al., 2026). Existing approaches keep a monocular video latent and use memory mechanisms such as memory retrieval, longer historical context, and hierarchical video representations (Xiao et al., 2025; Xue et al., 2026; Chandratreya et al., 2026). This leads to a complementary question: *Is there a latent representation that natively supports object-permanent generation?*

Our intuition is that to represent objects beyond their visibility in the rendered frame, the latent should encode the state of the whole scene across time rather than one camera's partial observation. Synchronized multi-camera videos make such a scene latent trainable and measurable, since cameras complement each other's views. To study scene latents in controlled experiments, we build PERMAKUBE, a synchronized multi-camera dataset with controlled occlusions and object-level annotations that keep invisible objects measurable. With this dataset, we train a multi-camera scene tokenizer, PERMATOK, and compare generation from its latents and from video latents.

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

Our experiments yield four findings (Sections 5.1 to 5.4). First, scene latents support stronger object-permanent generation than video latents despite lower reconstruction fidelity, with fewer identity changes or deletions, and this preservation advantage holds for every tokenizer supervision we test. Yet accurate motion after occlusion remains challenging for all latents. Second, PERMATOK can be adapted to monocular video input while retaining strong object-permanent generation, which improves with more latent training samples. Third, cross-view targets add what same-camera targets lack: their encoded scene tokens let a probe trained only on fully visible objects recover hidden objects that other cameras see, and their generations reach twice the hide-and-reappear success, although both preserve identity about as often. Fourth, scene latents are effective diffusion targets that use fewer target-encoding frames than video latents and train on fewer but costlier samples, since each requires a full synchronized multi-camera scene; more training scenes further improve their generated object localization.

These results carry two messages for the community. Scene latents require synchronized multi-camera video to train, outperform video latents when generators are trained on the same simulated scenes, and gain higher hide-and-reappear success and hidden-object recovery from cross-view targets, suggesting the value of 4D data for visual generation, which motivates acquiring such data. While out-of-sight, out-of-mind failures are read as evidence of a lost scene state (Ma et al., 2026), identity preservation can also come from a latent whose hidden state is not readily recoverable, so identity-level evaluation can miss flaws that latent probing and occlusion-and-return evaluation reveal; we therefore encourage probing designs that complement generation benchmarks.

Our contributions are (i) PERMAKUBE and PERMAEVAL, a synchronized multi-camera dataset with controlled occlusions and a ground-truth-selected benchmark for object permanence; (ii) PERMATOK, a multi-camera scene tokenizer, with evidence that its scene latents outperform video latents and controls that isolate what cross-view targets add; and (iii) a latent-probing analysis showing that identity preservation does not imply a readily recoverable hidden-object state.

2 RELATED WORK

Object permanence in video generation. Video generators exhibit identity changes and inconsistent motion (Huang et al., 2024), and StEvo-Bench and MemoBench evaluate whether generated frames keep object state across disappearance (Ma et al., 2026; Chen et al., 2026a). Memory methods address these failures in the generator through retrieval, longer context or hierarchical latents (Xiao et al., 2025; Xue et al., 2026; Chandratreya et al., 2026; Duan et al., 2026; Xu et al., 2026), and Shin et al. (2026) argue that pixel-based diffusion targets never supervise the hidden state. These methods model latents of monocular video; we instead learn the latent from synchronized multi-camera video, and evaluate that latent in addition to the generated frames.

Latent representations for generation. Image and video tokenizers such as SD-VAE and WF-VAE reconstruct their own inputs (Rombach et al., 2022; Li et al., 2025), and tokenizer studies show that reconstruction fidelity alone does not determine diffusion quality (Yao et al., 2025; Chen et al., 2025; Skorokhodov et al., 2025; Liu et al., 2025; Zhao et al., 2026; Guo et al., 2026). Scene representation models decode a shared latent at queried viewpoints (Eslami et al., 2018; Sajjadi et al., 2022; Seitzer et al., 2024; Jin et al., 2025), set latents compress 3D content for diffusion (Zhang et al., 2023), and NeuWorld diffuses a renderable scene state learned from sparse posed frames of static scenes (Li et al., 2026). PERMATOK builds on these designs; we compare its scene latents with video latents as frozen generation targets and observe a similar dissociation for object-permanent generation.

Multi-camera and 4D data for generation. Common datasets provide monocular clips (Chen et al., 2024; Carreira & Zisserman, 2017) or posed static scenes (Zhou et al., 2018). Multi-camera and 4D data (Bai et al., 2025b; Zhou et al., 2025; Van Hoorick et al., 2024) and 4D intermediates (Ren et al., 2024; Ma et al., 2025; Zhu et al., 2025) support camera control, novel-view synthesis and 4D reconstruction (Wu et al., 2025; Ren et al., 2025; Bai et al., 2025a). We show a different benefit: latents learned from multi-camera video support object-permanent generation. Kubric's MOVi sets provide dynamic scenes with object annotations (Greff et al., 2022), and Kubric-based datasets add dynamic multi-view renders (Van Hoorick et al., 2024) and occlusion videos with amodal annotations (Van Hoorick et al., 2023); PERMAKUBE combines synchronized fixed and moving cameras with controlled physical occlusions and a ground-truth-selected reappearance protocol.

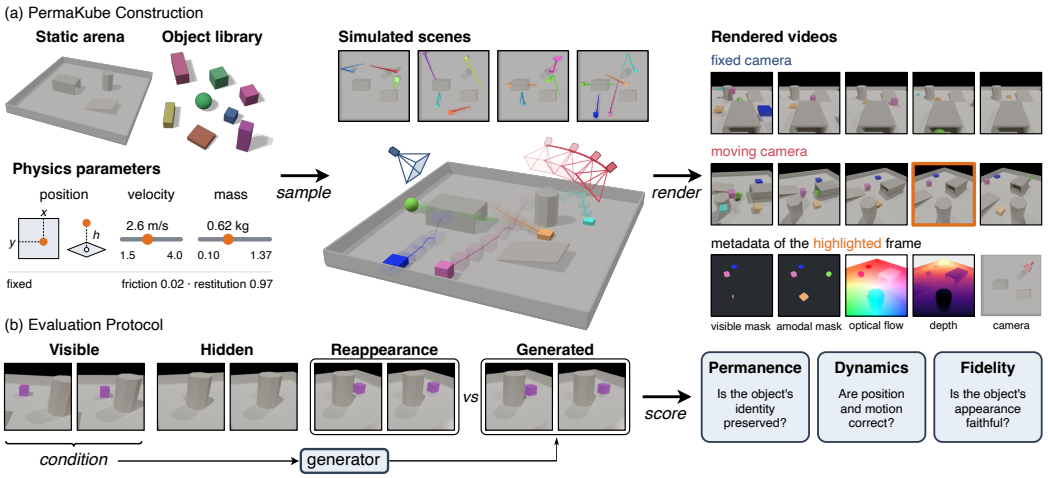


Figure 1: **PERMAKUBE construction (a) and occlusion-and-reappearance evaluation (b).**

Probing hidden state. Probes test what a model encodes beyond its outputs, from intermediate features (Alain & Bengio, 2016) to emergent world states (Li et al., 2023), and intuitive-physics benchmarks test physical understanding in video models (Garrido et al., 2025; Bordes et al., 2025; Motamed et al., 2026). Synthetic benchmarks and object-centric trackers study permanence in perception (Girdhar & Ramanan, 2020; Yi et al., 2020; Riochet et al., 2022; Bear et al., 2021; Shamsian et al., 2020; Tokmakov et al., 2021; Van Hoorick et al., 2023; Elsayed et al., 2022; Wu et al., 2023; Traub et al., 2024), and amodal segmentation predicts occluded extents, including without amodal annotations (Zhu et al., 2017; Follmann et al., 2019). We probe frozen tokenizer latents and find that hidden-object recovery can dissociate from object-permanent generation.

3 PERMAKUBE: MULTI-CAMERA PLAYGROUND FOR OBJECT PERMANENCE

PERMAKUBE makes object visibility controllable and measurable by recording an evolving 3D scene from synchronized cameras. Complementary views provide reconstruction targets, object annotations reveal what is hidden, and the same scenes support multi-camera and monocular comparisons. The conditional generation task observes past RGB frames from synchronized context cameras, ending before a target object is occluded, and continues the video at a query camera through the target's disappearance and reappearance (Figure 1b). Test masks are used only for scoring; training masks weight the tokenizer's reconstruction loss and never condition the generator.

Table 1: **PERMAKUBE statistics.**

Subsets	Scenes	Views	Frames	Objects
Base	2K	20	1.48M	3 to 6
Large	10K	20	7.4M	1 to 5
Extension	20K	8	5.92M	1 to 5
PermaEval	300	20	222K	1 to 5

Construction. Scenes built with Kubric (Greff et al., 2022) use a fixed arena with a pillar, ramp and tunnel, and a small rigid-shape catalogue with sampled sizes, colors, initial positions and velocities (Figure 1a). Train and test share the arena geometry, so held-out scenes test new object and motion configurations, not new occluder placements. Each video contains 37 RGB frames at 12 fps and 256×256 resolution (Table 1). PERMAKUBE-B (Base) and PERMAKUBE-L (Large, five sets with exactly 1 to 5 objects) render ten fixed and ten moving cameras, and Extension renders four of each. Base and each Large set split into 1,600 training, 200 validation and 200 test scenes, giving training pools of 1.6k (PERMAKUBE-B), 9.6k (PERMAKUBE-B/L) and, with 19,800 Extension scenes, 29.4k scenes. All subsets include camera calibration, object metadata and visible masks; Base and Large also include amodal masks, optical flow and saved simulation states (Appendix A.1).

PERMAEVAL. PERMAEVAL selects occlusion-and-reappearance events using ground truth, so the evaluated events do not depend on model outputs. Its 300 new scenes are balanced over five object counts and fixed and moving query cameras, with no overlap with the training pools. Each designated target is visible in the conditioning frames, becomes fully hidden from the query camera behind an occluder from frame 15 or later for 3 to 9 frames, and then reappears clearly for at least

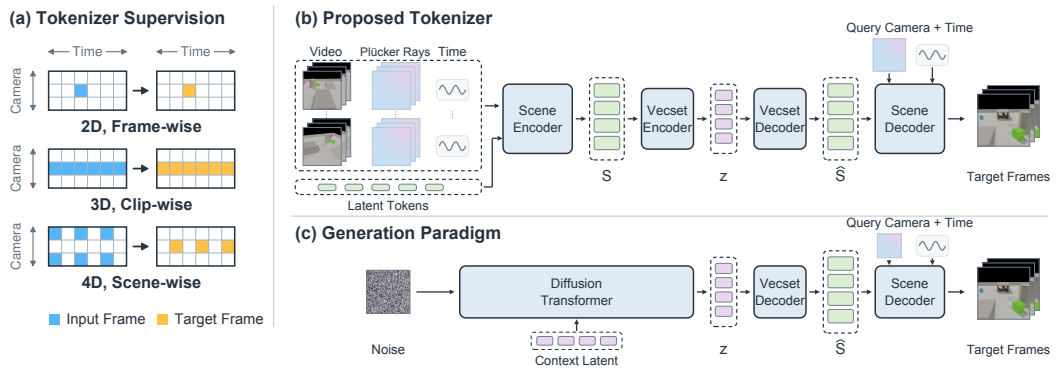


Figure 2: **Schematic of tokenizer supervision (a), PERMATOK (b) and generation (c).**

three frames. We evaluate **object permanence**, **dynamics** and **fidelity** on this conditional task, and separately examine **diffusibility** through unconditional generation (Appendix B.5).

4 LATENT REPRESENTATIONS AND GENERATION

4.1 THREE WAYS TO SUPERVISE A LATENT

We categorize tokenizer supervision into three types (Figure 2(a)). Frame-wise (2D) and clip-wise (3D) tokenizers reconstruct the frame or clip they encode, whereas scene-wise (4D) supervision asks a shared latent to render the same scene at cameras and times absent from its input, which we call cross-view, cross-time supervision. A hidden object has no direct visual target in an occluded view, but another camera can reveal it. We use SD-VAE (Rombach et al., 2022) and causal WF-VAE (Li et al., 2025) as the frame-wise and clip-wise baselines, and PERMATOK (Section 4.2) as the scene-wise tokenizer. We call the SD-VAE and WF-VAE representations *video latents*: SD-VAE encodes each frame independently, whereas WF-VAE jointly encodes space and time.

The latents differ in size: SD-VAE maps each 256×256 frame to 4×32^2 values, WF-VAE maps a 37-frame clip to $10 \times 4 \times 32^2$ values, and PERMATOK maps six cameras at six times to 512×16 values (Appendix Table 7). We report the better-generating four-channel WF-VAE as WF-VAE and the 16-channel variant as WF-VAE (16 ch.), whose complete results are included in the supplementary material (Appendix B.3). Because this comparison changes architecture, latent dimensionality, target structure, temporal sampling and decoding alongside supervision, we add two variants that keep PERMATOK's architecture but change its targets: PERMATOK-SC renders its input cameras at unobserved times (same-camera, cross-time), and PERMATOK-R, fine-tuned from PERMATOK-SC for 13k updates, reconstructs its input frames (Sections 5.1 and 5.3).

4.2 PERMATOK: A MULTI-CAMERA SCENE TOKENIZER

PERMATOK is built on LVSM (Jin et al., 2025): a transformer tokenizer aggregates synchronized multi-camera observations into scene tokens, a compressor reduces them to a compact latent for generation, and a decoder renders the latent at queried cameras and times (Figure 2(b)). We use PERMATOK to denote the combined tokenizer and compressor, which encode observations into a compact scene latent and decode it at a specified camera and time.

Framework. PERMATOK takes calibrated synchronized frames $I_{c,i}$ from camera trajectory c at sampled time i. Subsampled RGB frames and pixel-wise Plücker rays are patchified and augmented with learned time embeddings to form input tokens x. The scene encoder $\mathrm{Enc}_{\mathrm{scene}}$ processes $[x; e]$ with learnable latent tokens e using full self-attention, and its outputs form the scene tokens S. A Vecset encoder (Zhang et al., 2023) compresses S into a 512×16 latent z; $\mathrm{Dec}_{\mathrm{comp}}$ reconstructs the scene tokens, and $\mathrm{Dec}_{\mathrm{scene}}$, also with full self-attention, combines them with camera and time queries to output RGB.

Objectives. PERMATOK and WF-VAE share a GT-mask foreground weight of 10, whereas SD-VAE is used as released; their remaining objectives differ. The scene objective combines separately

normalized foreground/background RGB errors and a 0.5-weighted VGG loss, with target cameras and times sampled outside the input set. The compressor reconstructs whitened scene tokens under a 10^{-6} KL regularizer toward $\mathcal{N}(0, I)$ (Appendix A.2).

4.3 Generation Framework

All main comparisons use a DiT-B (Peebles & Xie, 2023) generator (12 layers, width 768) trained with flow matching (Lipman et al., 2023). For a standardized target latent $y_\star$, encoded past observations h, $u \sim \mathcal{U}[0, 1]$ and $\varepsilon \sim \mathcal{N}(0, I)$,

$$y_u = (1 - u)\varepsilon + u y_\star, \qquad \mathcal{L}_{\mathrm{FM}} = \mathbb{E}\big[\|v_\eta(y_u, u; h) - (y_\star - \varepsilon)\|_2^2\big]. \tag{1}$$

The frozen tokenizer decodes sampled latents at the queried cameras and times.

Conditional generation continues the video of one of six context cameras, the query camera, from their past RGB (Appendix A.2): SD-VAE and PermaTok observe frames $2, 8, 14$ and causal WF-VAE observes 0 to 12. Each family encodes its context with its own frozen tokenizer: the video latents of the six cameras' past frames, or for PermaTok one context scene latent of these 18 frames, form clean context tokens that are concatenated with the noisy target tokens in one sequence. Video-latent targets encode the query-camera clip; scene targets encode six cameras at anchor times $2, 8, 14, 21, 27, 34$, and the decoder renders all 37 frames by querying their times, interpolating in between. Future RGB enters training targets only, never evaluation context, and all models receive the query camera's poses for every target frame. **Unconditional generation** drops h and samples a complete latent from noise. Scene latents are decoded along pose trajectories taken from training scenes, whereas baseline trajectories are implicit in their clips. We sample with EMA weights and 100 Euler steps, or 50 for the 1.6k-scene unconditional SD-VAE and WF-VAE generators (Appendix B.5).

5 Results and Analysis

5.1 Do scene latents support object-permanent generation?

We compare scene and video latents on object-permanent generation and reconstruction fidelity.

Experiment Setup. SD-VAE is the widely used pretrained Stable Diffusion autoencoder and reconstructs objects in our frames most faithfully (Table 2), while WF-VAE and PermaTok are trained from scratch on the same 1,600 PermaKube-B scenes; all three are frozen, and DiT-B generators are then trained on latents of the same 9,600 PermaKube-B/L scenes and conditioned on past frames from the same six context cameras (Section 4.3). The tokenizers' native inputs produce different numbers of latent samples: N scenes with M cameras yield N scene latents or $N \times M$ video latents, so with $M = 20$ the 9,600 scenes give 9.6k scene latents and 192k video latents. The variants PermaTok-SC and PermaTok-R keep the inputs, latent training samples and generator recipe fixed. Budgets follow each pipeline's recipe rather than matched compute: baseline generators train for 200k updates at batch 16, and PermaTok generators for 40k at batch 32, so the latter see fewer target presentations (1.28M versus 3.2M).

Evaluation Metrics. We evaluate generation on PermaEval along three aspects. For **permanence**, we measure identity and location separately: *Preservation* measures whether a vision-language judge recognizes the target after occlusion onset, independent of where and when it reappears, and *P@0.3* measures whether the reappearing object's generated mask overlaps the ground truth with IoU of at least 0.3. For **dynamics**, S_{15} measures whether the reappearing object lies within 15 pixels of its true position, *Pos.* and *Det.* measure its centroid error and detection rate, and motion success measures whether it moves with the correct velocity. For **fidelity**, we report the PSNR of generated frames (gPSNR) and of tokenizer reconstructions (rPSNR). A human audit on 40 stratified scenes reproduces the ranking with a larger PermaTok advantage than the judge's; metric definitions, intervals and audits are included in the supplementary material (Appendix B.2).

Result. Scene latents support stronger object-permanent generation than video latents (Table 2). PermaTok reaches 78.7% preservation and P@0.3 0.378, versus 47.7% and 0.210 for WF-VAE and 39.7% and 0.064 for SD-VAE, although its object rPSNR is lower than both video latents' (23.95 versus 26.55 and 29.51 dB), also on its own input frames (full frame 33.11 versus 38.21

Table 2: **Conditional generation and reconstruction.** Pos.: detected-frame centroid error (px); PSNR in dB; obj/full: object/full frame. Bold: best of SD-VAE, WF-VAE and PERMATOK.

		Permanence		Dynamics			Fidelity			
Tokenizer	Latent Samples	Pres. (%)↑	P@0.3↑	S_{15} ↑	Det.↑	Pos.↓	$\text{gPSNR}_{\text{obj}}$ ↑	$\text{gPSNR}_{\text{full}}$ ↑	$\text{rPSNR}_{\text{obj}}$ ↑	$\text{rPSNR}_{\text{full}}$ ↑
SD-VAE	192k	39.7	0.064	0.152	0.428	20.89	15.36	27.47	**29.51**	37.54
WF-VAE	192k	47.7	0.210	0.364	0.581	15.07	15.70	28.80	26.55	**39.57**
PERMATOK	9.6k	**78.7**	**0.378**	**0.612**	**0.816**	**12.30**	**16.69**	**29.94**	23.95	35.57
Supervision controls										
PERMATOK-SC	9.6k	74.7	0.389	0.590	0.866	13.14	16.68	28.65	20.36	32.34
PERMATOK-R	9.6k	64.9	0.201	0.381	0.718	15.81	15.88	28.08	19.31	31.75

and 37.22 dB; Appendix Table 7); across these families, reconstruction fidelity does not predict generation performance. Hide-and-reappear success, which requires no detected foreground in the reappearance regions at the last fully hidden frame and a P@0.3 reappearance in the target's color, keeps the ranking: PERMATOK succeeds in 28.7% of scenes versus 11.0% and 4.0% for WF-VAE and SD-VAE, and most often passes this hidden-frame check (78.3% versus 70.0% and 71.3%; Appendix B.2). The ranking also holds at equal update counts of 100k and 200k (Figure 7c,d).

PERMATOK-SC shows no detectable difference from PERMATOK in preservation (74.7%) or reappearance location (0.389 P@0.3), and PERMATOK-R is weaker (64.9%), yet both exceed the video latents. Their hide-and-reappear success, however, falls to 14.7% and 10.3%, close to WF-VAE's 11.0%: the hidden-frame check finds foreground in 57% of PERMATOK-SC events versus 22% for PERMATOK, so passing it depends on cross-view targets (Section 5.3). Accurate dynamics remain challenging for all latents: PERMATOK localizes only 61.2% of reappearance frames within 15 pixels and reaches 32.5% motion success at a 2 px/frame tolerance, versus 36.4% and 21.7% for WF-VAE (Figure 3). These errors do not reflect unpredictable motion, because constant-velocity extrapolation of the true past state errs by only 0.4 to 0.7 px, even from WF-VAE's context ending at frame 12; this motion reference is included in the supplementary material (Appendix B.4). For PERMATOK, most reappearance frames fall between its six anchor times and are interpolated, which likely limits velocity accuracy.

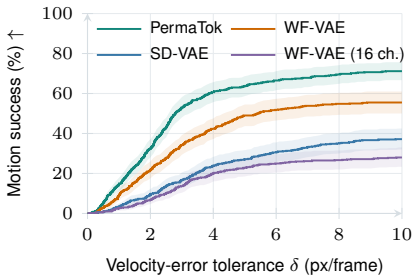


Figure 3: **Motion success versus velocity-error tolerance.** Bands: 95% CIs.

We also identify different failure modes for scene and video latents (Figure 4): PERMATOK distorts the shape of a reappearing object (a), whereas SD-VAE merges objects with the background (b) and WF-VAE merges objects with each other (c) or splits them (d). Across all 300 scenes, the judge labels fewer PERMATOK clips as deletions or identity changes (13.0% and 8.0%) than WF-VAE (18.3% and 32.7%) and SD-VAE (34.7% and 24.7%) clips. In the examples of Figure 5, only PERMATOK regenerates the reappearing object. The complete failure breakdown is included in the supplementary material (Appendix A.5).

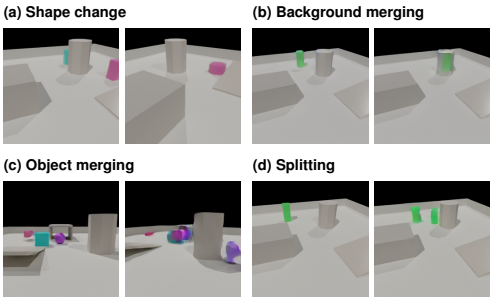


Figure 4: **Failure modes.**

Finding 1. Object-Permanent Generation: Multi-camera scene latents support stronger object-permanent generation than monocular video latents, despite lower reconstruction fidelity, while accurate motion preservation remains challenging for all latents.

5.2 CAN OBJECT PERMANENCE TRANSFER TO MONOCULAR VIDEO INPUT?

Because multi-camera video is not always available, we ask whether a multi-camera tokenizer adapted to monocular input from the same distribution still supports object-permanent generation.

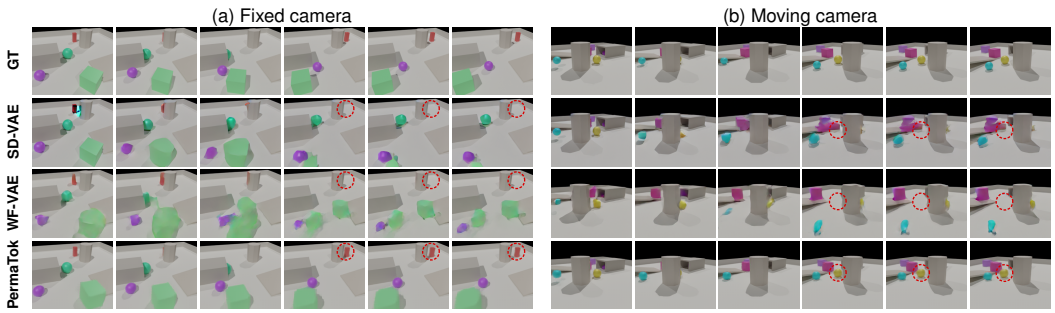


Figure 5: **Object-permanent generation after occlusion.** Circles mark GT positions.

Table 3: **Monocular transfer.** Abbreviations as in Table 2.

Setting	Latent Samples	Pres. (%)↑	P@0.3↑	$\text{gPSNR}_{\text{obj}}$ ↑	$\text{gPSNR}_{\text{full}}$ ↑	$\text{rPSNR}_{\text{obj}}$ ↑	$\text{rPSNR}_{\text{full}}$ ↑
6-cam reference	9,600	78.7	0.378	16.69	**29.94**	**23.95**	**35.57**
1-cam, frozen	57,600	52.3	0.102	15.86	22.58	18.35	24.03
1-cam, fine-tuned	9,600	74.3	0.201	15.85	28.32	21.26	31.38
1-cam, fine-tuned	57,600	**80.0**	**0.499**	**17.36**	28.72	21.26	31.38

Experiment Setup. We fine-tune the pretrained six-camera PERMATOK for 30k updates on PERMAKUBE-B using monocular input and cross-view, cross-time supervision, then retrain the compressor. The generator trains on six separately encoded single-camera clips per scene from the 9,600 PERMAKUBE-B/L training scenes (57,600 in total). We compare against the frozen pretrained tokenizer on the same clips and, as a sample-count control, train a generator on one clip per scene, matching the six-camera reference's 9,600 samples. All generators use DiT-B, 40k updates and batch 32; evaluation follows Section 5.1 with only the query camera's past frames as context.

Result. The fine-tuned scene tokenizer supports strong object-permanent generation from monocular input, reaching 80.0% preservation from the query camera's past alone (Table 3), above both video latents with six context cameras (Table 2). Even with six times fewer latent training samples (9,600 versus 57,600), the fine-tuned tokenizer outperforms the frozen one on preservation (74.3% versus 52.3%) and P@0.3 (0.201 versus 0.102). Scaling the latent training samples to 57,600 further improves P@0.3 from 0.201 to 0.499, although the added samples are other cameras' clips of the same scenes. Fine-tuning particularly benefits moving-camera clips, improving preservation from 26.7% to 79.3% and P@0.3 from 0.016 to 0.389. Complete experimental details, camera-type breakdowns and paired confidence intervals are included in the supplementary material (Appendix A.3).

Finding 2. Monocular Transferability: The multi-camera tokenizer can be adapted to monocular video input, and the resulting latent supports strong object-permanent generation. Generation with these latents can further benefit from scaling the latent training samples.

5.3 DOES OBJECT-PERMANENT GENERATION IMPLY A RECOVERABLE HIDDEN-OBJECT STATE?

We consider the recoverability of objects hidden from the query camera a necessary property of a state-preserving latent. We therefore probe frozen latents with a lightweight segmentation head trained only on fully visible targets and call successful hidden-object segmentation *emergent hidden-object recovery*, since no training target supervises hidden masks; it arises because rendering unobserved cameras requires localizing objects the query camera cannot see.

Experiment Setup. We freeze the pretrained PERMATOK and encode six-camera clips from PERMAKUBE-B, observed at six times spanning the clip, into scene tokens S. Complementary cameras or later frames can reveal a target hidden from the query camera, so the probe tests whether hidden objects are recoverable from the scene tokens of observed clips, not whether compression or a generator retains them over time. A lightweight segmentation head predicts the target's mask from these frozen tokens, a query time and a query camera chosen among the input cameras. We probe S because supervision shapes S while the compressor only reconstructs it; the generator's latent z

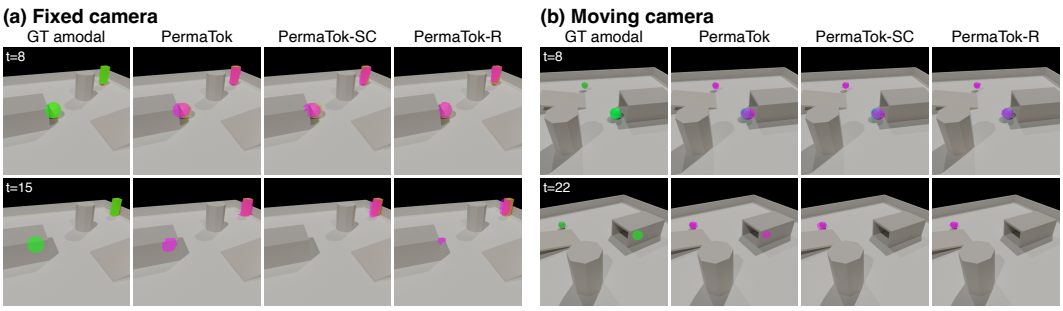


Figure 6: **Latent probing.** Visible-only probe masks for (a) fixed and (b) moving query cameras.

is computed from S, so it cannot hold hidden-object information that S lacks, and for PERMATOK compression costs 1.1/0.8 dB of object/full-frame rPSNR (Appendix B.2). The main *visible-only* probe trains only on frames where the targets are fully visible; as a reference, a *standard* probe also learns amodal masks under other occluders. We evaluate both probes on tunnel-occluded targets in 385 held-out scenes and exclude tunnel-occluded query clips from training, so neither probe learns masks under the test occluder. To test tokenizer supervision, we also encode the same observations with the PERMATOK-SC and PERMATOK-R checkpoints of Table 2.

Evaluation Metrics. We report hidden-region IoU and success at IoU ≥ 0.3 (*H@0.3*, distinct from generation P@0.3) over all eligible tunnel-occluded frames, and *Query-hidden H@0.3* over frames where the target is completely invisible in the query frame (Appendix A.4).

Table 4: **Hidden-object recovery.** Entries: standard/visible-only probe, mean over three seeds.

Tokenizer (supervision)	Hidden IoU↑	H@0.3↑	Query-hidden H@0.3↑
PERMATOK (cross-view, cross-time)	**0.486 / 0.450**	**0.768 / 0.721**	**0.680 / 0.603**
PERMATOK-SC (same-camera, cross-time)	0.138 / 0.094	0.185 / 0.118	0.021 / 0.004
PERMATOK-R (input reconstruction)	0.105 / 0.074	0.131 / 0.080	0.013 / 0.003

Result. PERMATOK's scene tokens show strong emergent hidden-object recovery: visible-only H@0.3 reaches 0.721, approaching the standard probe (0.768), while PERMATOK-SC and PERMATOK-R yield only 0.118 and 0.080 (Table 4). All three tokenizers are trained on the same arena and scenes, so shared occluder geometry alone does not explain this gap. Figure 6 shows PERMATOK recovering hidden masks that PERMATOK-SC and PERMATOK-R miss or replace with false fragments. PERMATOK-SC, however, preserves the target about as often (74.7% versus 78.7%; paired comparison in Appendix A.2) but fails the latent probe on its encoded scene tokens and reaches only 14.7% hide-and-reappear success (Section 5.1), so identity preservation does not guarantee a readily recoverable hidden-object state, and the probe exposes a weakness that preservation misses.

Finding 3. Emergent Hidden-Object Recovery: Cross-view supervision enables a probe trained only on fully visible objects to recover hidden objects from scene tokens. Without cross-view supervision, a same-camera latent still preserves object identity nearly as often, but the probe fails to recover its hidden objects and its hide-and-reappear success is lower, indicating that identity preservation does not imply a readily recoverable hidden-object state.

5.4 ARE SCENE LATENTS EFFECTIVE DIFFUSION TARGETS?

We test scene latents as diffusion targets in unconditional generation and conditional scene scaling.

Experiment Setup. We train unconditional DiT-B generators on frozen tokenizers with 1.6k and 9.6k training scenes. Video targets use ten 37-frame clips per scene; scene targets use six cameras at six times and are rendered along camera trajectories taken from training scenes (Section 4.3).

Evaluation Metrics. FVD (Unterthiner et al., 2018), video KID (Bińkowski et al., 2018) and coverage (Naeem et al., 2020) compare 2,048 generated clips with a shared reference per scene scale.

Table 5: **Unconditional generation.** KID: mean ± subset s.d. At 1.6k scenes, SD-VAE and WF-VAE use 50 sampling steps; other entries use 100.

		1.6k scenes			9.6k scenes		
Tokenizer	Frames/scene	FVD↓	KID↓	Cov.↑	FVD↓	KID↓	Cov.↑
SD-VAE	370	181.83	36.60 ± 8.08	0.70	123.09	23.65 ± 8.54	0.80
WF-VAE	370	78.24	12.24 ± 5.12	0.91	59.08	8.33 ± 4.01	0.90
PERMATOK	36	**55.62**	$\mathbf{8.35} \pm 4.32$	**0.94**	**31.31**	$\mathbf{4.41} \pm 3.87$	**0.95**

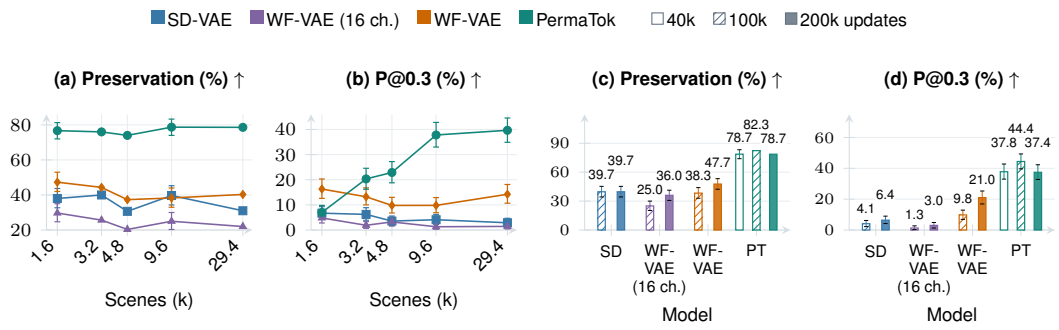


Figure 7: **Conditional-generation ablations.** Curves use 40k updates for PERMATOK and 100k for baselines (80k for WF-VAE (16 ch.) at 29.4k); bars use 9.6k scenes; whiskers: available 95% CIs.

Result. Scene latents support strong unconditional generation under this rendering protocol: at 9.6k scenes, PERMATOK achieves lower FVD than WF-VAE (31.31 versus 59.08), with lower KID and higher coverage, while using 36 rather than 370 input frames per scene to form generator targets (Table 5). A scene latent fuses six synchronized cameras, whereas video latents treat each camera's video as a separate sample, so scene latents train on far fewer but costlier samples (9.6k versus 192k conditional samples; Table 2). The conditional ablations show that with only 1.6k scenes, PERMATOK localizes reappearing objects less often than WF-VAE (0.070 versus 0.163 P@0.3), but 29.4k scenes raise its P@0.3 to 0.397, whereas the video-latent curves do not improve at their fixed update budgets (Figure 7).

Finding 4. Scene Latent Diffusibility: Scene latents are effective diffusion targets that use fewer target-encoding frames than video latents and train on fewer but costlier samples, since each sample requires a full synchronized multi-camera scene; more training scenes further improve their generated object localization.

6 DISCUSSION

Limitations. We focus on object-permanent generation and therefore use one simulated arena with rigid objects, fixed occluders and near-linear motion, which future work can extend to unseen occluder layouts, richer simulation and real multi-camera captures. Rendering by querying cameras and times is native to scene latents, so we evaluate the scene-latent paradigm as a whole, and its design space, such as latent compactness, multi-view targets, temporal anchoring and supervision targets, remains to be explored. Accurate motion after occlusion remains challenging for all latents; instead of downsampling frames, a pretrained video encoder such as VideoMAE (Tong et al., 2022) could compress each camera's full video into motion-preserving tokens that are then fused into a scene latent. Finally, the tokenizer needs scarce synchronized multi-camera data for training, yet it adapts to monocular input, after which generation improves with more monocular training clips (Section 5.2).

Conclusion. Using PERMAKUBE, we show that scene latents learned by PERMATOK support stronger object-permanent generation than video latents, transfer to monocular input and serve as effective diffusion targets, suggesting the value of 4D data for visual generation. Latent probing reveals emergent hidden-object recovery under cross-view, cross-time supervision, and identity preservation alone does not guarantee a readily recoverable hidden-object state, so we encourage probing designs that complement frame-level benchmarks.

REPRODUCIBILITY STATEMENT

All data, model checkpoints and code will be released, including PERMAKUBE, PERMAEVAL, split manifests, camera calibrations, object annotations, tokenizer, compressor and generator training code, evaluation scripts, frozen scorer settings and judge prompts. Architectures, optimization schedules, sampling settings and training populations are specified in Appendix A; the evaluation protocol, metric definitions and bootstrap procedure are specified in Appendix B, together with additional final-test results and diagnostics. Unless stated otherwise, each reported model is a single training run; automated-evaluation intervals quantify scene variation on the fixed 300-scene test set (the human audit uses a 40-scene subset; Appendix B.2), and available seed variation is reported in Appendix A.2.

ETHICS STATEMENT

All scene data are rendered with an open-source simulator from procedurally generated scenes, without human depictions, personal data or copyrighted footage. Two human readers assessed synthetic videos; the audit procedure and limitations are disclosed in Appendix B.2. The study concerns the internal consistency of generative video models and does not by itself enable new misuse beyond that of existing video generators; more persistent generative simulators could nonetheless lower the cost of producing convincing synthetic footage, and released models will carry the same usage restrictions as the underlying training code. The automated judge is used only to score synthetic clips.

AI USE STATEMENT

A vision-language model (`gemini-3.1-pro-preview`) is used as an automated judge for the preservation outcome defined in Appendix B.2; it receives only ground-truth and generated frames of synthetic scenes under a fixed, model-blind protocol, and its role, prompts and validation are reported there. Large language model assistants were additionally used to edit and restructure manuscript text, to edit figure and table layout code, and to check consistency between reported numbers and their source tables. All research questions, dataset construction, model design, experiments, analyses and conclusions were conceived, executed and verified by the authors, who take full responsibility for the content.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. URL https://arxiv.org/abs/1610.01644.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. ReCamMaster: Camera-controlled generative rendering from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a. URL https://arxiv.org/abs/2503.11647.

Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. SynCamMaster: Synchronizing multi-camera video generation from diverse viewpoints. In *International Conference on Learning Representations (ICLR)*, 2025b. URL https://arxiv.org/abs/2412.07760.

Daniel M. Bear, Elias Wang, Damian Mrowca, Felix J. Binder, Hsiao-Yu Fish Tung, R. T. Pramod, Cameron Holdaway, Sirui Tao, Kevin Smith, Fan-Yun Sun, Li Fei-Fei, Nancy Kanwisher, Joshua B. Tenenbaum, Daniel L. K. Yamins, and Judith E. Fan. Physion: Evaluating physical prediction from vision in humans and machines. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2106.08261.

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR)*, 2018.

Florian Bordes, Quentin Garrido, Justine T. Kao, Adina Williams, Michael Rabbat, and Emmanuel Dupoux. IntPhys 2: Benchmarking intuitive physics understanding in complex synthetic environments. *arXiv preprint arXiv:2506.09849*, 2025. URL https://arxiv.org/abs/2506.09849.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the Kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Ishaan Preetam Chandratreya, David Charatan, Basile Van Hoorick, Sergey Zakharov, Vitor Guizilini, Phillip Isola, and Vincent Sitzmann. MilliVid: Hierarchical latents for long-range consistency in video generation. *arXiv preprint arXiv:2606.09056*, 2026. URL https://arxiv.org/abs/2606.09056.

Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2502.03444.

Haoyu Chen, Kaichen Zhou, Hang Hua, Kaile Zhang, Jingwen Qian, Wufei Ma, Haonan Chen, Chunjiang Liu, Yizhou Zhao, Xiaoyuan Wang, Weiyue Li, Alan Yuille, Paul Pu Liang, and Yilun Du. MemoBench: Benchmarking world modeling in dynamically changing environments. *arXiv preprint arXiv:2606.27537*, 2026a. URL https://arxiv.org/abs/2606.27537.

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

Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov. Panda-70M: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13320–13331, 2024. URL https://arxiv.org/abs/2402.19479.

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

Gamaleldin F. Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C. Mozer, and Thomas Kipf. SAVi++: Towards end-to-end object-centric learning from real-world videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.07764.

S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. doi: 10.1126/science.aar6170. URL https://doi.org/10.1126/science.aar6170.

Patrick Follmann, Rebecca König, Philipp Härtinger, Michael Klostermann, and Tobias Böttger. Learning to see the invisible: End-to-end trainable amodal instance segmentation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1328–1336, 2019. URL https://arxiv.org/abs/1804.08864.

Quentin Garrido, Nicolas Ballas, Mahmoud Assran, Adrien Bardes, Laurent Najman, Michael Rabbat, Emmanuel Dupoux, and Yann LeCun. Intuitive physics understanding emerges from self-supervised pretraining on natural videos. *arXiv preprint arXiv:2502.11831*, 2025. URL https://arxiv.org/abs/2502.11831.

Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for compositional actions and TEmporal reasoning. In *International Conference on Learning Representations (ICLR)*, 2020.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2203.03570.

Minghui Guo, Shengqiong Wu, and Hao Fei. V-RAE: Rethinking video latent spaces for generation. *arXiv preprint arXiv:2608.13556*, 2026. URL https://arxiv.org/abs/2608.13556.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024. URL https://arxiv.org/abs/2311.17982.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A large view synthesis model with minimal 3D inductive bias. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.17242.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.13382.

Zhiqi Li, Chengrui Dong, Zhenhua Du, Hangning Zhou, Cong Qiu, Hailong Qin, Mu Yang, Dongxu Wei, and Peidong Liu. Walking in the implicit: Interactive world exploration via neural scene representation. In *European Conference on Computer Vision (ECCV)*, 2026. URL https://arxiv.org/abs/2606.30045.

Zongjian Li, Bin Lin, Yang Ye, Liuhan Chen, Xinhua Cheng, Shenghai Yuan, and Li Yuan. WF-VAE: Enhancing video VAE by wavelet-driven energy flow for latent video diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2411.17459.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.02747.

Shizhan Liu, Xinran Deng, Zhuoyi Yang, Jiayan Teng, Xiaotao Gu, and Jie Tang. Delving into latent spectral biasing of video VAEs for superior diffusability. *arXiv preprint arXiv:2512.05394*, 2025. URL https://arxiv.org/abs/2512.05394.

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. URL https://arxiv.org/abs/2603.13215.

Ziqiao Ma, Xuweiyi Chen, Shoubin Yu, Sai Bi, Kai Zhang, Chen Ziwen, Sihan Xu, Jianing Yang, Zexiang Xu, Kalyan Sunkavalli, Mohit Bansal, Joyce Chai, and Hao Tan. 4D-LRM: Large space-time reconstruction model from and to any view at any time. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2506.18890.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026. URL https://arxiv.org/abs/2501.09038.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning (ICML)*, pp. 7176–7185, 2020.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. URL https://arxiv.org/abs/2212.09748.

Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, and Huan Ling. L4GM: Large 4D gaussian reconstruction model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.10324.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2503.03751.

Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. IntPhys 2019: A benchmark for visual intuitive physics understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5016–5025, 2022. URL https://arxiv.org/abs/1803.07616. arXiv version (2018) titled "IntPhys: A Framework and Benchmark for Visual Intuitive Physics Reasoning".

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2112.10752.

Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2111.13152.

Maximilian Seitzer, Sjoerd van Steenkiste, Thomas Kipf, Klaus Greff, and Mehdi S. M. Sajjadi. DyST: Towards dynamic neural scene representations on real-world videos. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.06020.

Aviv Shamsian, Ofri Kleinfeld, Amir Globerson, and Gal Chechik. Learning object permanence from video. In *European Conference on Computer Vision (ECCV)*, 2020.

Joonghyuk Shin, Yicong Hong, Jaesik Park, and Xun Huang. Can video world models track unobserved world states? *arXiv preprint arXiv:2608.30692*, 2026. URL https://arxiv.org/abs/2608.30692.

Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2502.14831.

Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Manuel Traub, Frederic Becker, Sebastian Otte, and Martin V. Butz. Learning object permanence from videos via latent imaginations. In *International Conference on Artificial Neural Networks (ICANN)*, pp. 223–240, 2024. URL https://arxiv.org/abs/2310.10372.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Basile Van Hoorick, Pavel Tokmakov, Simon Stent, Jie Li, and Carl Vondrick. Tracking through containers and occluders in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. URL https://arxiv.org/abs/2305.03052.

Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *European Conference on Computer Vision (ECCV)*, 2024. URL https://arxiv.org/abs/2405.14868.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T. Barron, and Aleksander Holynski. CAT4D: Create anything in 4D with multi-view video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2411.18613.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. SlotFormer: Unsupervised visual dynamics simulation with object-centric models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.05861.

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. URL https://arxiv.org/abs/2504.12369.

Tianshuo Xu, Yichen Xie, Depu Meng, Chensheng Peng, Quentin Herau, Bo Jiang, Yihan Hu, and Wei Zhan. Teaching video generators to remember: Eliciting dynamic memory for out-of-sight state evolution. *arXiv preprint arXiv:2605.25333*, 2026. URL https://arxiv.org/abs/2605.25333.

Bowen Xue, Brandon Y. Feng, Chenguo Lin, Yuchen Lin, Yujia Zeng, Lvmin Zhang, Maneesh Agrawala, Honglei Yan, and Panwang Pan. Ring forcing: Towards precise long-term memory for autoregressive video diffusion. *arXiv preprint arXiv:2608.26794*, 2026. URL https://arxiv.org/abs/2608.26794.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2501.01423.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: CoLlision events for video REpresentation and reasoning. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1910.01442.

Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3DShape2VecSet: A 3D shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4), 2023. URL https://arxiv.org/abs/2301.11445.

Yian Zhao, Feng Wang, Qiushan Guo, Chang Liu, Xiangyang Ji, Jian Zhang, and Jie Chen. Video generation with predictive latents. *arXiv preprint arXiv:2605.02134*, 2026. URL https://arxiv.org/abs/2605.02134.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37(4):65:1–65:12, 2018. URL https://arxiv.org/abs/1805.09817.

Yang Zhou, Yifan Wang, Jianjun Zhou, et al. Omniworld: A multi-domain and multi-modal dataset for 4d world modeling. *arXiv preprint arXiv:2509.12201*, 2025. URL https://arxiv.org/abs/2509.12201.

Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. Aether: Geometric-aware unified world modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8535–8546, 2025. URL https://arxiv.org/abs/2503.18945.

Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1464–1472, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Zhu_Semantic_Amodal_Segmentation_CVPR_2017_paper.html.

A IMPLEMENTATION AND EXPERIMENTAL DETAILS

This part specifies the dataset, models and comparison protocols. Evaluation definitions, additional results and measurement limitations follow in Appendix B.

A.1 DATASET DETAILS

Table 6: **Simulation specification.**

Component	Specification
Playground (fixed, shared by all scenes)	
arena	20×20 m floor, friction 0.08
walls	1.2 m high, restitution 0.9 (elastic bounce-back)
tunnel	$4.5\times2.4\times1.6$ m at $(-4, 0)$: *full occluder*
pillar	r=1.0 m, h=2.8 m at $(3, 3)$: line-of-sight occluder
ramp	4.8×3.2 m, 10° tilt at $(4, -3)$: kinetic element
lighting	3 fixed directional lights (key/fill/rim)
Object pool (fixed catalogue, sampled per object)	
shapes	uniformly sampled: sphere, cube, flat / tall / elongated box
size s	$\mathcal{U}[0.40, 0.70]$ m (radius / half-extent)
color	HSV, $h\sim\mathcal{U}[0, 1)$, $S, V\sim\mathcal{U}[0.7, 1]$
physics	friction 0.02, restitution 0.97; mass 0.5 kg (Base) or density $0.5\,\mathrm{kg\,m^{-3}}$ (Large and Extension)
count n	$\mathcal{U}\{3, \dots, 6\}$ (Base); exactly $k \in \{1, \dots, 5\}$ (each Large set); $\mathcal{U}\{1, \dots, 5\}$ (Extension)
Per-scene variations (initial conditions + cameras)	
spawn	8 zones $+\mathcal{U}[-2.5, 2.5]^2$ m jitter, avoids structures; $z\sim\mathcal{U}[s+0.3, 2.5]$ m
velocity	speed $\mathcal{U}[1.5, 4.0]$ m/s, heading $\mathcal{U}[0, 2\pi)$, v_z=0
tunnel event	p=0.5: sphere rolled through tunnel, $v_x\sim\mathcal{U}[-4, -2]$ m/s
fixed cams	ring r=9 m, $h\in\{4, 6\}$ m, 36° apart; jitter σ_θ=3°, $\sigma_{r,h}$=0.3 m
moving cams	$r\sim\mathcal{U}[6, 13]$ m, $h\sim\mathcal{U}[2, 12]$ m, sweep $\mathcal{U}[40, 90]^\circ$, random direction

Annotation scope. All source pools include RGB, camera calibration, object metadata and visible-object masks. Base and Large also include amodal-mask renders, forward optical flow and saved simulation scenes. Extension omits amodal masks, flow and saved simulation scenes. The final test provides object tracks, depth, flow and validity annotations for evaluation; these labels are not additional generator conditioning. RGB and foreground masks provide the reconstruction targets and object weighting described below.

Training and evaluation splits. Base and each of the five object-count groups in Large contain 2,000 scenes: 1,600 training, 200 validation and 200 test. The 9,600-scene training pool combines the Base and Large training splits. Each scale study trains on a fixed list of physical scenes. Extension contains 20,000 scenes, of which 19,800 join the 9,600-scene pool to form the 29,400-scene training set; 200 are reserved as held out. It renders eight cameras per scene with RGB, modal masks and metadata. It has no amodal masks, flow or depth, which the occlusion evaluation requires. Table 1 summarizes these populations. The 300-scene occlusion test is separately constructed and does not reuse the source-pool test splits.

Dataset construction and selection. Table 6 gives the sampling distributions of the scene simulator. A tunnel-directed trajectory does not guarantee an eligible evaluation event: clear return, readable area and camera visibility must also hold. PERMAEVAL aims candidate trajectories and keeps only events that meet the ground-truth criteria. Its conditional distribution therefore differs from unfiltered training data. We report this selection explicitly rather than describing it as a random sample of all simulated motion.

A.2 TOKENIZER AND GENERATOR DETAILS

Tokenizers and normalization. SD-VAE is the released pretrained Stable Diffusion 1.5 autoencoder, kept frozen and encoded with its posterior mean. WF-VAE and PERMATOK are trained from

scratch on the same 1,600 PERMAKUBE-B training scenes, then frozen for generator training. Both use the same GT-mask foreground weight of 10; perceptual, adversarial and representation-specific terms retain their respective recipes. WF-VAE is trained on 25-frame camera clips with its reconstruction, perceptual, wavelet and adversarial objectives. At generation time its latent represents a padded temporal encoding of a 37-frame target clip. Per-channel whitening uses training data. Conditional and unconditional generators use separate normalization statistics where their recipes require it.

Scene tokenizer. The scene tokenizer combines sparse camera and time observations into one representation. An image patch has side length 8, giving 1,024 image tokens per 256×256 frame. The tokenizer uses learned embeddings for the 37 times. It is trained in two stages: first with two input cameras at six times (12 input frames), then fine-tuned with all six input cameras (36 input frames). Training camera identities are sampled, whereas the generator training latents use the fixed camera anchors.

Architecture and optimization. The scene tokenizer has six encoder and six decoder layers, width 768, and 88.23M trainable parameters. The two-camera stage uses 100k updates at global batch 64 and the six-camera stage 30k updates at global batch 48; the peak learning rate is 4×10^{-4}. The compressor has eight layers in each encoder and decoder, width 768, and 124.5M parameters. Both stages use the same 1,600 training scenes. Compression training uses 40k updates, global batch 8 and learning rate 2×10^{-4}. All tokenizer and compressor weights are then frozen for generator training.

Object-weighted supervision. Foreground weighting gives object reconstruction an explicit role despite the larger background area. Let O and B denote foreground and background pixels in a target image. For the scene tokenizer, the reconstruction objective normalizes their errors separately:

$$\mathcal{L} = \frac{1}{|B|} \sum_{p \in B} \ell_2(p) + \alpha \frac{1}{|O|} \sum_{p \in O} \ell_2(p) + \lambda \mathcal{L}_{\text{perc}}, \qquad \alpha = 10, \quad \lambda = 0.5. \tag{2}$$

This sets the relative loss weights of the two regions; it does not guarantee a fixed ratio of parameter gradients. No tokenizer or generator in this paper uses an amodal-mask prediction head.

Table 7 compares tokenizer inputs, latent sizes, compression and reconstruction PSNR. Its rPSNR is measured on 50 validation scenes, separate from the final test, so it differs from the rPSNR on PERMAEVAL in Table 2. Compression ratios count the stated input frames rather than temporal padding; PERMATOK reconstructions include compression (28.0 dB object PSNR before it).

Table 7: **Tokenizer configurations.** Compr.: input RGB values per latent value; PSNR: object-region/full-frame peak signal-to-noise ratio (dB).

Tokenizer	Input per latent	Latent	Compr.	rPSNR↑	Novel-view PSNR↑
SD-VAE (2D)	one frame	4×32^2	$48\times$	29.98 / 37.22	N/A
WF-VAE (16 ch.)	one camera, 37 frames	$10\times16\times32^2$	$44\times$	28.40 / 39.52	N/A
WF-VAE	one camera, 37 frames	$10\times4\times32^2$	$178\times$	26.72 / 38.21	N/A
PERMATOK (4D)	6 cameras, 6 timesteps	512×16	$864\times$	26.24 / 33.11	24.9 / 31.2

Compression. The compressor reduces scene tokens to the compact 512×16 stochastic latent used by the generator, while learning to reconstruct their features. It reconstructs whitened scene features $\bar{S}$ (training-set statistics) with channel weights $w_j = 1 + 10 v_j / \operatorname{mean}_k v_k$, where v_j is the across-scene variance of the token-mean feature, under a variational regularizer averaged over latent elements:

$$\mathcal{L}_{\text{comp}} = \mathbb{E}\Big[\operatorname{mean}_{i,j} w_j (\widehat{\bar{S}}_{ij} - \bar{S}_{ij})^2\Big] + 10^{-6}\, \overline{D}_{\text{KL}}\big(q_\psi(z \mid \bar{S}) \,\|\, \mathcal{N}(0, I)\big)\,. \tag{3}$$

The tokenizer objective is equation 2 with input sets and target queries sampled across cameras and times. The generator uses the compressor trained on the 1,600-scene reference pool.

Conditional generator. The flow-matching model interpolates between Gaussian noise ε and a normalized target latent $y_\star$ as $y_u = (1-u)\varepsilon + uy_\star$ and predicts $y_\star - \varepsilon$. Flow time u is distinct from video time t. Sampling integrates the predicted velocity from noise and reverses the training-set normalization before the frozen tokenizer decodes the latent. DiT-B uses 12 layers and width 768. Image/video latents are patchified and augmented with camera, time, position and context information. Scene latents use a shared coordinate normalization for target and context. Context dropout removes the entire context with probability 0.1. The six context cameras are fix00, fix03 and fix07 (fixed) and mov00, mov03 and mov07 (moving); every PERMAEVAL query camera is one of them, namely fix07 for the 150 fixed-camera clips and mov00, mov03 or mov07 for 45, 58 and 47 moving-camera clips. The PERMATOK conditional generator uses learning rate 10^{-4}, 40,000 updates, global batch 32 and EMA 0.999.

Primary conditional sampling. The main conditional evaluation uses EMA weights, 100 Euler steps and classifier-free guidance with weight $w = 3$ in $v = v_\varnothing + (1+w)(v_h - v_\varnothing)$, where $v_\varnothing$ drops the context; the same setting applies to every model and was not selected on test scores. On validation, weight 3 is best or tied for object PSNR in 9 of 10 baseline configurations.

Training variation and scene uncertainty. A three-run validation study of PERMATOK gives object PSNR of 18.42 ± 0.12 and 21.40 ± 0.35 dB at 1.6k and 9.6k scenes, respectively (mean $\pm$ s.d. over training runs). These validation scenes are distinct from the final 300-scene test. They therefore do not establish final-test training-seed robustness; the main results use one trained model per configuration and scene-bootstrap intervals.

PERMATOK-SC generation control. This comparison tests whether complementary target cameras improve generation when prediction across time is retained. Both configurations use six-camera input, the same latent size, 9,600 generator examples and the same generator recipe. PERMATOK-SC predicts its input cameras at unobserved times, rather than reconstructing only observed frames; the same checkpoint is probed in Table 4. Table 8 reports paired results, with differences defined as PERMATOK minus PERMATOK-SC. Neither the P@0.3 nor the Preservation interval excludes zero; this is not an equivalence test. The Preservation comparison has McNemar $p = 0.26$.

Table 8: **PERMATOK versus PERMATOK-SC generation.** Δ: PERMATOK$-$PERMATOK-SC; FP: false-positive rate; CI: confidence interval.

Metric	PERMATOK	PERMATOK-SC	Δ	95% CI
P@0.3	0.3778	0.3889	-0.0111	$[-0.0767, +0.0556]$
Return-mask IoU	0.2443	0.2456	-0.0013	$[-0.0337, +0.0312]$
Object gPSNR	16.6898	16.6823	$+0.0075$	$[-0.3232, +0.3442]$
Position	12.20	13.09	-0.886	$[-2.074, +0.326]$
Preservation (%)	78.67	74.67	$+4.00$	$[-2.33, +10.33]$
Detection rate	0.8156	0.8656	-0.0500	$[-0.0911, -0.0089]$
Visible-control IoU	0.6568	0.5627	$+0.0941$	$[+0.0804, +0.1076]$
Empty-control FP	0.2067	0.5500	-0.3433	$[-0.4111, -0.2744]$

The higher PERMATOK-SC detection rate coexists with a higher empty-control rate, so it does not indicate better object recovery. The paired centroid means (12.20 and 13.09 px) differ slightly from the marginal means in Table 2 (12.30 and 13.14 px) because paired contrasts use only scenes in which both models detect the target. This control retains temporal prediction and does not isolate the effect of cross-time supervision itself.

PERMATOK-R generation. PERMATOK-R fine-tunes PERMATOK-SC for 13k updates to reconstruct its input frames, so its contrast with PERMATOK-SC replaces cross-time targets with input reconstruction during fine-tuning. It yields preservation of 64.9% and P@0.3 of 0.201, and all generation and reconstruction metrics are reported in Table 2. Its paired preservation difference from PERMATOK is -13.7 percentage points (95% CI $[-20.7, -6.7]$).

A.3 MONOCULAR TRANSFER DETAILS

Monocular protocol. Each of the 9,600 training scenes supplies six separate camera videos, giving 57,600 latent context/target pairs. Context and target share a camera: their respective anchors are $[2, 8, 14]$ and $[2, 8, 14, 21, 27, 34]$. At evaluation, context uses only past frames from the query camera, and decoding uses that clip's single-camera pose basis. Both generators trained on all 57,600 samples use DiT-B, 40k updates, batch 32 and training seed 0, with EMA and 100-step Euler sampling under the evaluation protocol of Section 5.1. Without fine-tuning, the pipeline uses the six-camera tokenizer and compressor.

Adaptation recipe and reconstruction. The tokenizer is fine-tuned on 1,600 scenes with one-camera input and cross-view, cross-time targets for 30k updates. A new compressor is trained from scratch on 9,600 latents (1,600 scenes $\times$ six cameras), using posterior-mean decoding on half the examples and the recipe in Appendix A.2 otherwise. Table 9 reports tokenizer reconstruction on 20 validation scenes, before compression. These scenes are disjoint from PERMAEVAL; these scores measure reconstruction, not recovery after occlusion. Novel-view full-frame PSNR from fixed-camera input, which the table does not show, falls by 0.9 to 1.3 dB while object PSNR improves.

Table 9: **Monocular tokenizer fine-tuning.** PSNR: peak signal-to-noise ratio (dB).

	Object PSNR↑		Full-frame PSNR↑	
Input camera	Frozen	Adapted	Frozen	Adapted
Moving (mov00)	17.65	**27.07**	19.07	**30.15**
Fixed (fix00)	22.79	**27.12**	26.64	**30.42**

Table 3's rPSNR values include compression. The adapted generator trained on all 57,600 samples has P@0.3 0.4989 [0.449, 0.550], full-frame gPSNR 28.72 [28.39, 29.04] dB and 240/300 positive preservation judgments. The human judge audit covers the main comparison rows, not transfer (Appendix B.2).

Additional transfer statistics. For the six-camera reference, the monocular pipeline without fine-tuning and the adapted pipeline trained on all 57,600 samples, respectively, S_{15} is 0.6122, 0.3989 and 0.6722, and centroid error is 12.30, 17.13 and 11.80 px. Table 10 reports paired adaptation effects and camera-type results. Moving-camera preservation improves substantially, while spatial recovery remains below the fixed-camera result. Intervals resample scenes for fixed trained models; they do not estimate training-seed variability.

Table 10: **Monocular fine-tuning.** Left: fine-tuned minus frozen; right: frozen → fine-tuned. Pres.: preservation; gPSNR: generation PSNR.

Metric	Difference [95% CI]
P@0.3	$+0.397$ [$+0.340$, $+0.452$]
S_{15}	$+0.273$ [$+0.207$, $+0.340$]
Position (px)	-5.3 [-6.5, -4.0]
$\text{gPSNR}_{\text{obj}}$ (dB)	$+1.50$ [$+1.20$, $+1.78$]
$\text{gPSNR}_{\text{full}}$ (dB)	$+6.13$ [$+5.65$, $+6.64$]
Preservation (pts)	$+27.7$ [$+20.7$, $+35.0$]

Metric	Moving	Fixed
P@0.3↑	$0.016 \rightarrow \mathbf{0.389}$	$0.189 \rightarrow \mathbf{0.609}$
S_{15} ↑	$0.422 \rightarrow \mathbf{0.564}$	$0.376 \rightarrow \mathbf{0.780}$
$\text{gPSNR}_{\text{full}}$ ↑	$18.65 \rightarrow \mathbf{28.38}$	$26.52 \rightarrow \mathbf{29.05}$
Pres. (%)	$26.7 \rightarrow \mathbf{79.3}$	$78.0 \rightarrow \mathbf{80.7}$

Latent training samples. The 9,600-sample setting keeps one camera per scene from the adapted setting's 57,600 samples, with the same tokenizer, compressor and 40k-update generator budget. The six-camera reference and both 57,600-sample settings draw on the same 9,600 scenes and 345,600 anchor frames; splitting views creates more samples, not independent scenes. The six-camera comparison also differs in observation/target structure and tokenizer training: it has no matched 30k-update tokenizer continuation. The two adapted settings share the same reconstruction (21.26/31.38 dB object/full rPSNR). Preservation with 9,600 samples is 223/300, with no uncertain or failed judgments. Fixed-camera preservation is 70.7% versus 80.7% with all 57,600 samples; moving-camera preservation is 78.0% versus 79.3%. We do not test the preservation difference against the six-camera reference.

A.4 HIDDEN-OBJECT PROBE DETAILS

Protocol. With tokenizers frozen, we train a probe comprising two cross-attention blocks of width 384, camera-ray/time queries, and binary cross-entropy plus soft-Dice loss. The probe reads each tokenizer's scene tokens S before compression, because tokenizer supervision acts on S while the compressor is trained only to reconstruct S. It therefore measures what is readily recoverable from S, not what the compressed latent z or generated latents retain. Both input regimes include the query camera and encode six times across the clip; complementary views or later frames may reveal the target. Standard probes learn amodal masks under pillar, wall and ramp occlusions; visible-only probes restrict training query frames to fully visible targets. Both exclude tunnel-occluded query clips, although other input views may contain tunnel occlusions. Matched probes use identical batches, 6k updates and three seeds.

Evaluation scores hidden target pixels inside the projected tunnel, requiring at least 50 such pixels per frame and excluding visible pixels. H@0.3 measures hidden-region IoU success; fully query-hidden means no target pixel is visible in the query frame. The six-camera test contains 31,857 frames from 385 held-out scenes; the monocular test contains 15,140 frames from 398 scenes. Scene-bootstrap 95% intervals quantify test-scene uncertainty, separately from probe-seed variation, and do not establish tokenizer-seed robustness. We compare supervision within each probe variant, since visible-only filtering changes the training population.

Six-camera comparisons. Table 11 reports three-seed means and 95% scene-bootstrap intervals for the three tokenizer objectives in Table 4. PERMATOK versus PERMATOK-SC gives paired H@0.3 gains of 0.583 $[0.567, 0.599]$ (standard) and 0.603 $[0.584, 0.621]$ (visible-only); the visible-only hidden-IoU gain is 0.356 $[0.345, 0.366]$.

Table 11: **Six-camera probe uncertainty.** QH: fully query-hidden; Perm.: scene-permuted H@0.3.

Tokenizer	Hidden IoU↑	H@0.3↑	QH H@0.3↑	Perm.
		Standard probe		
PERMATOK	0.486 [0.474, 0.497]	0.768 [0.751, 0.785]	0.680 [0.653, 0.706]	0.024
PERMATOK-SC	0.138 [0.130, 0.147]	0.185 [0.172, 0.199]	0.021 [0.016, 0.026]	0.008
PERMATOK-R	0.105 [0.099, 0.112]	0.131 [0.120, 0.143]	0.013 [0.009, 0.017]	0.005
		Visible-only probe		
PERMATOK	0.450 [0.438, 0.462]	0.721 [0.701, 0.740]	0.603 [0.574, 0.635]	0.020
PERMATOK-SC	0.094 [0.088, 0.101]	0.118 [0.107, 0.128]	0.004 [0.003, 0.006]	0.007
PERMATOK-R	0.074 [0.070, 0.079]	0.080 [0.074, 0.087]	0.003 [0.002, 0.005]	0.004

PERMATOK-SC improves H@0.3 over PERMATOK-R by 0.054 $[0.046, 0.063]$ (standard) and 0.038 $[0.032, 0.044]$ (visible-only), but the visible-only fully query-hidden difference is not distinguishable from zero (0.001 $[-0.001, 0.002]$). Across probe seeds, PERMATOK-R H@0.3 is 0.131 ± 0.011 (standard) and 0.080 ± 0.007 (visible-only). For moving cameras, different times also change pose, so this comparison does not isolate time alone.

Probe controls. Scene-permuted latents test dependence on the correctly paired representation. For fully query-hidden targets, visible-only probes give H@0.3 0.0034 versus 0.0036 after permutation on PERMATOK-R, and 0.0041 versus 0.0071 on PERMATOK-SC. Near-floor recovery does not prove that the latent lacks hidden-object information. Overall amodal IoU remains similar between PERMATOK-SC and PERMATOK-R ($0.515/0.510$ standard; $0.482/0.487$ visible-only); these are full-mask scores, not visible-pixel-only measurements.

Monocular input. Both tokenizers fine-tune the six-camera PERMATOK on monocular input, with cross-view, cross-time targets (PERMATOK (1-cam), the adapted tokenizer of Section 5.2) or input-reconstruction targets (PERMATOK-R (1-cam), a different checkpoint from the six-camera PERMATOK-R) (Table 12). This tests retention during fine-tuning and changes camera and time targets jointly. Paired H@0.3 gains are 0.288 $[0.266, 0.309]$ (standard) and 0.250 $[0.230, 0.269]$ (visible-only). The visible-only hidden-IoU gain is 0.141 $[0.132, 0.151]$, despite similar overall amodal IoU ($0.509/0.510$); fully query-hidden success remains low ($0.039/0.002$).

Table 12: **Monocular hidden-object recovery.** Both tokenizers fine-tune the six-camera PERMATOK on monocular input. Entries: standard/visible-only. H@0.3: hidden-region IoU success at 0.3; query-hidden: fully invisible in the query frame.

Tokenizer (targets)	Hidden IoU↑	H@0.3↑	Query-hidden H@0.3↑
PERMATOK (1-cam; cross-view, cross-time)	**0.257 / 0.189**	**0.404 / 0.283**	**0.130 / 0.039**
PERMATOK-R (1-cam; input reconstruction)	0.104 / 0.048	0.116 / 0.033	0.015 / 0.002

Interpretation limits. The six-camera PERMATOK-SC and PERMATOK-R probed in Table 4 are the checkpoints evaluated for generation in Table 2. Before compression, which is what the probe reads, PERMATOK-SC reconstructs the scored frames at 38.63 dB full-frame PSNR, above PERMATOK's 36.38 dB, so weak reconstruction is unlikely to explain its near-floor recovery. Near-floor recovery still does not prove that the latent lacks hidden-object information.

Input visibility. A post-hoc audit checks all 36 encoder frames and stratifies 9,868 fully query-hidden test frames from 313 scenes (Table 13). Frame groups are disjoint; scene counts overlap. On 5,008 frames with no query-camera input observation, H@0.3 is 0.565 versus 0.004 despite identical input evidence.

Table 13: **Recovery by input visibility.** Entries: PERMATOK/PERMATOK-SC; ΔH: difference in H@0.3; CI: 95% scene-bootstrap confidence interval.

Input visibility	Frames	Scenes	IoU↑	H@0.3↑	ΔH@0.3 CI
Never	5,008	257	0.345 / 0.005	0.565 / 0.004	[0.528, 0.596]
Earlier only	4,015	201	0.371 / 0.004	0.610 / 0.003	[0.564, 0.648]
Later only	260	40	0.483 / 0.012	0.814 / 0.009	[0.721, 0.876]
Both sides	585	61	0.485 / 0.008	0.798 / 0.007	[0.716, 0.863]
All	9,868	313	0.367 / 0.005	0.603 / 0.004	[0.571, 0.628]

A separate inventory of 6,010 (frame, object) instances with no query-camera anchor observation, based on 13,860 instance masks, finds complementary-camera visibility in every instance: 99.5% earlier, 94.1% later and 16.1% concurrently (overlapping categories). This inventory has a different denominator from the scored frames; none of its targets is absent from all inputs. "Earlier only" in the table concerns query-camera anchors, while other cameras may reveal the target later; concurrent visibility requires an encoded query time. The experiment therefore tests whole-clip recovery, not past-only forecasting or temporal state maintenance. Permutation controls are not matched to individual visibility strata and are not their respective floors.

A.5 ADDITIONAL CONTROLS AND DIAGNOSTICS

Data scale and training updates. Figure 7 varies scenes with frozen tokenizers and tests longer training at 9.6k scenes. PERMATOK P@0.3 rises from 0.070 at 1.6k scenes to 0.397 at 29.4k, while preservation remains between 74.0% and 78.7%. Baseline degradation from 1.6k to 9.6k at 100k updates is not evidence of a representation ceiling: fixed updates reduce exposure per scene. Extending WF-VAE to 200k raises P@0.3 from 0.098 to 0.210, exceeding its 1.6k score of 0.163. SD-VAE improves from 0.041 to 0.064 and 16-channel WF-VAE from 0.013 to 0.030. Scene curves use 40k updates for PERMATOK and 100k for the baselines; the 4.8k-scene training sets differ between PERMATOK and the baselines. The 200k 16-channel WF-VAE and 100k PERMATOK continuations are unannealed. The 29.4k 16-channel WF-VAE result uses its 80k checkpoint.

Semantic scores and failure accounting. Tables 14 and 15 give the category rates and the semantic rates with outcome-specific denominators. Preservation for the remaining configurations uses the same judge, with 4,499 definite responses and one failed call among 4,500 judgments. The 29.4k and transfer results are in Tables 21 and 3. The examples in Figure 4 illustrate the identified failure types and do not estimate their frequencies.

Table 14: **Semantic outcomes (% of all scenes).**

Model / scenes / updates	Preserved↑	Deletion only↓	Identity only↓	Combined↓	Unresolved↓
PERMATOK 9.6k/40k	78.7	13.0	8.0	0.3	0.0
PERMATOK 1.6k/40k	76.7	13.7	9.3	0.3	0.0
SD-VAE 1.6k/100k	38.0	44.0	16.7	1.3	0.0
SD-VAE 9.6k/100k	39.7	38.3	19.7	2.3	0.0
SD-VAE 9.6k/200k	39.7	34.7	24.7	1.0	0.0
WF-VAE (16 ch.) 1.6k/100k	29.7	28.3	37.7	4.3	0.0
WF-VAE (16 ch.) 9.6k/100k	25.0	34.0	39.0	2.0	0.0
WF-VAE (16 ch.) 9.6k/200k[†]	36.0	36.7	24.7	2.7	0.0
WF-VAE 1.6k/100k	47.3	23.7	26.0	3.0	0.0
WF-VAE 9.6k/100k	38.3	20.0	37.7	4.0	0.0
WF-VAE 9.6k/200k	47.7	18.3	32.7	1.3	0.0
GT as generation	94.7	4.7	0.7	0.0	0.0

Rates use exact category counts, and every row contains 300 scenes. These rates include combination cases in the denominator and therefore differ from the rates in Table 15. Failure subcategories are not human-validated: the human audit assessed preservation only. [†]Unannealed intermediate.

Table 15: **Semantic rates with outcome-specific denominators (%).** Brackets: 95% scene-bootstrap CIs.

Model / scenes / updates	Preservation↑	Deletion↓	Identity change↓
PERMATOK 9.6k/40k	78.7 [74.0, 83.3]	13.0	8.0
PERMATOK 1.6k/40k	76.7 [72.0, 81.3]	13.7	9.4
SD-VAE 1.6k/100k	38.0 [32.7, 43.7]	44.6	16.9
SD-VAE 9.6k/100k	39.7 [34.3, 45.3]	39.2	20.1
SD-VAE 9.6k/200k	39.7 [34.3, 45.3]	35.0	24.9
WF-VAE (16 ch.) 1.6k/100k	29.7 [24.7, 35.0]	29.6	39.4
WF-VAE (16 ch.) 9.6k/100k	25.0 [20.3, 30.0]	34.7	39.8
WF-VAE (16 ch.) 9.6k/200k[†]	36.0 [30.7, 41.3]	37.7	25.3
WF-VAE 1.6k/100k	47.3 [42.0, 53.0]	24.4	26.8
WF-VAE 9.6k/100k	38.3 [33.0, 44.0]	20.8	39.2
WF-VAE 9.6k/200k	47.7 [42.3, 53.3]	18.6	33.1
GT as generation	94.7 [92.0, 97.0]	4.7	0.7

Definite-response rate is 100% for preservation. These deletion-only and identity-change-only rates exclude combination and unresolved causes; Table 14 reports the corresponding category counts divided by all 300 scenes. [†]Unannealed intermediate.

The judge runs at default temperature with one call per clip.

Table 16 reports fixed/moving results for 150 scenes each, using the 200k baseline and 40k PERMATOK checkpoints.

Table 16: **Results by camera type.** Entries: fixed/moving. Pres.: preservation; Det.: detection rate; Pos.: detected-frame centroid error.

Model	Pres. (%)↑	P@0.3↑	S_{15} ↑	Det.↑	Pos. (px)↓
SD-VAE	31.3 / 48.0	0.051 / 0.078	0.153 / 0.151	0.413 / 0.442	21.73 / 20.12
WF-VAE	47.3 / 48.0	0.238 / 0.182	0.402 / 0.327	0.547 / 0.616	13.32 / 16.58
PERMATOK	78.7 / 78.7	0.384 / 0.371	0.684 / 0.540	0.840 / 0.791	10.85 / 13.85

Preservation uses the same judge as Table 2; the human audit is not stratified by camera type, and no paired fixed-versus-moving test is computed.

B EVALUATION PROTOCOL

The conditional test set and mask-based scoring rules were fixed before any generator was evaluated on it, and one preservation-judge protocol was applied unchanged to all models. Unconditional reference sets are fixed within each training-scale comparison. We distinguish automatic spatial diagnostics from judged semantic outcomes and report their measurement limitations.

B.1 EVALUATION POPULATIONS AND THEIR RELATIONSHIPS

Table 17 distinguishes the scene populations, query subsets and reference samples used throughout the paper. Queries and generated samples are not independent physical scenes. The final test uses scene means and resamples physical scenes for uncertainty.

Table 17: **Evaluation populations and subsets.**

Population	Count and subsets	Definition and use
Final conditional test	300 new scenes; 301 eligible objects	Strict three-stage protocol; selected target and eligible additional objects.
Conditional fidelity	50 validation scenes	PSNR and guidance selection, separate from final-test results.
Base unconditional reference	2,000 clips / 200 validation scenes	Fixed reference for 1.6k training.
Base/Large unconditional reference	2,048 validation clips	Fixed Base/Large reference for 9.6k training.
Unconditional samples	2,048 clips per quality row	Used for FVD, KID and coverage.

B.2 CONDITIONAL EVALUATION PROTOCOL

Population and independence. The final test contains 300 newly generated physical scenes, balanced over five object counts and two camera types, with 30 scenes in each group. A further 30 scenes are retained as reserves and do not enter the primary estimates. No development or previously evaluated scene is reused in the final set. Selection uses simulator annotations, never model success. The primary target is selected before evaluating any generator. One physical scene is the unit of resampling even if it yields several objects, windows or samples.

Camera and duration controls. The fixed condition uses the supported fix07 camera with its native jitter. Moving-camera selection accounts for height, radius and sweep. The initial duration target per group is 12 short (3 to 4 fully hidden frames), 12 medium (5 to 6), and 6 long (7 to 9), with recorded reallocation where a cell is infeasible. The realized short, medium and long counts are 115, 119 and 66.

Stage definitions. The target must be visible in supported conditioning frames. The fully hidden stage starts at $t \geq 15$ and requires zero modal area, nonzero in-frame amodal support, and a physical occluder. At return, three consecutive frames must each have at least 350 modal pixels, modal-to-amodal area at least 0.9, and the specified eight-pixel image-border margin. At most six partial emergence frames precede the scored window, and the final scored frame is no later than frame 36.

Additional objects. Only one non-primary object passes the independent visibility and clear-return criteria. Additional objects therefore do not establish a population-level multi-object result. All main endpoints use the preselected primary target, with one scene as the resampling unit.

Scoring windows and generated masks. For scene s, let $W_s = \{r_s, r_s + 1, r_s + 2\}$ be the three GT-selected clear-return frames. The primary population has $N = 300$ scenes, hence 900 scored frames and 600 consecutive pairs. Identity evaluation instead uses the generated interval from first full occlusion τ_s through frame 36. The two windows answer different questions and are fixed independently of generated return times.

The mask extractor subtracts a rendered background plate from generated RGB inside a GT-defined region: the target's amodal bounding box, with a one-pixel silhouette tolerance and 20-pixel padding. Dilated masks of other GT objects restrict this region while protecting the target. Shadow rejection, morphological closing and small-component filtering produce the generated foreground mask $\widehat{M}_{s,t}$. The comparison mask $M_{s,t}$ is the GT visible mask restricted to the same region. Detection $d_{s,t}$ requires more than 80 generated foreground pixels. The extractor uses an RGB difference threshold of 0.06 after Gaussian blur $\sigma = 0.5$, one closing iteration, minimum component area 12, and other-object exclusion dilation 12 pixels; shadow filtering uses cosine similarity above 0.995 and brightness ratio between 0.2 and 0.97. These settings are frozen, not fitted to test outcomes. This mask extractor measures spatial agreement and can select an unrelated foreground region; it does not establish object identity or whole-frame absence.

Overlap and fidelity. With $I_{s,t} = |\widehat{M}_{s,t} \cap M_{s,t}| / |\widehat{M}_{s,t} \cup M_{s,t}|$, an empty generated mask has zero overlap. We report

$$\mathrm{P@0.3} = \frac{1}{N} \sum_{s} \frac{1}{3} \sum_{t \in W_s} \mathbf{1}[I_{s,t} \geq 0.3]. \tag{4}$$

This is a frame-level overlap-success rate, not identity precision.

For RGB images scaled to $[0, 1]$ and pixel region A, define $\mathrm{MSE}_A = (3|A|)^{-1} \sum_{p \in A} \|\widehat{x}(p) - x(p)\|_2^2$ and $\mathrm{PSNR}_A = -10 \log_{10}(\mathrm{MSE}_A)$. Object PSNR uses the full-image GT visible mask dilated by five pixels; full-frame PSNR uses every image pixel. We average frame PSNRs within each scene and then across scenes, rather than taking the logarithm of a pooled MSE. Both include failed object generations because the region is defined by GT, not detections.

In Table 2, gPSNR applies this calculation to generated frames; rPSNR applies it to tokenizer reconstructions of GT at the same 900 scored queries, with the same regions and scene aggregation. Reconstruction follows each tokenizer's target encoding and includes compression for PERMATOK. It receives GT target observations unavailable to the conditional generator, so rPSNR is a privileged reconstruction reference rather than a mathematical upper bound on gPSNR. The baseline reconstructions encode and decode the scored GT frames. The PERMATOK control encodes 36 GT frames (six cameras at times $2, 8, 14, 21, 27, 34$), passes them through the compressor encoder and decoder, and decodes the query camera at the three return times. These times are generally absent from the inputs, so its rPSNR also measures temporal interpolation. On all 300 scenes, $\mathrm{rPSNR}_{\mathrm{obj}}$ is 23.95 [23.57, 24.34] dB and $\mathrm{rPSNR}_{\mathrm{full}}$ is 35.57 [35.12, 36.03] dB. Without compression the same reconstruction gives 25.04/36.38 dB, so compression costs 1.08/0.81 dB (computed from unrounded scores).

Semantic preservation and failure types. The judge (`gemini-3.1-pro-preview`) receives all 37 GT and generated frames, in separate ordered sequences at 256×256, with up to three enlarged target crops from GT. One call asks whether the target's combined shape, color and distinguishing attributes are recognizable in any generated frame of $[\tau_s, 36]$. Late or misplaced returns count; color-only blobs, shadows and observations solely before onset do not. By design, preservation isolates identity: timing and position are measured by P@0.3 and S_{15}, and disappearance by hide-and-reappear success below.

The judge protocol divides non-preservation into deletion only (no recovered target or identifiable replacement by clip end), identity change only (a different object takes the target's role), a separate combination category, or unresolved cause. Deletion is neither the complement of preservation nor a claim that the object would never return. Combination cases are reported separately, not included in either exclusive failure rate.

For each outcome, let Y, N, U denote yes, no and unresolved counts. Preservation is $100Y/(Y+N)$; definite-response rate is $100(Y + N)/300$. For the failure breakdown in Table 14, each scene instead receives exactly one label: preserved, deletion only, identity change only, combination, or unresolved. Unresolved includes an ambiguous failure cause, an uncertain preservation response or a technical failure. Each category rate is $100n_k/300$, so the complete breakdown sums to 100% before rounding. Table 14 adopts this common denominator; its category rates use exact counts. Table 15 uses outcome-specific denominators, so its deletion and identity-change rates are not comparable with this breakdown.

Audits of the main comparison. Two non-author readers reproduce the preservation ranking on 40 shared scenes, and the judge agrees with reader 1 on 80.0% of clips ($\kappa = 0.603$; Table 19). The mask scores are spatial by design: the extractor segments foreground inside a ground-truth region, and its empty-control rate, the fraction of last-hidden-frame regions with detected foreground, is 20.7% to 29.0% for SD-VAE, WF-VAE and PERMATOK and 55.0% for PERMATOK-SC. The ground-truth-as-generation control reaches 94.7% preservation.

Hide-and-reappear success. To measure disappearance and reappearance together, we score hide-and-reappear success for SD-VAE, WF-VAE, PERMATOK, PERMATOK-SC and PERMATOK-R from their saved mask-extractor outputs. A scene succeeds if the mask extractor detects no foreground at the last fully hidden frame inside the regions of the three reappearance frames, and at least one reappearance frame reaches IoU ≥ 0.3 with a generated mask whose mean color is closest to the target's base color among all scene objects. The hidden-frame check targets the regions where the object will reappear, so it tests disappearance exactly where a leaked or prematurely returning target would appear, rather than over the whole hidden interval or image. Because a detection in any of the three regions fails the check, its scene-level foreground rates (21.7% for PERMATOK and 57.3% for PERMATOK-SC) slightly exceed the region-level empty-control rates above (20.7% and 55.0%). PERMATOK exceeds WF-VAE by 17.7 points (95% scene-bootstrap CI 12.0 to 23.3) with at least one reappearance frame and by 15.3 points (9.7 to 21.0) with at least two, and exceeds SD-VAE by 24.7 (19.3 to 30.0) and 22.0 (17.0 to 27.0) points (Table 18). Its invisibility rate at the last hidden frame is also higher, by 8.3 points (1.7 to 15.0) over WF-VAE and 7.0 (0.3 to 13.7) over SD-VAE. Among the supervision controls, PERMATOK exceeds PERMATOK-SC by 14.0 points (8.3 to 19.7) and PERMATOK-R by 18.3 points (12.3 to 24.0) with at least one reappearance frame, and its invisibility rate exceeds theirs by 35.7 (28.7 to 42.7) and 22.7 (15.7 to 29.7) points.

Table 18: **Hide-and-reappear success.** Rates (%) over the 300 scenes with 95% scene-bootstrap CIs.

Model	Hidden-frame check passed↑	Success, ≥ 1 frame↑	Success, ≥ 2 frames↑
SD-VAE	71.3 [66.3, 76.3]	4.0 [2.0, 6.3]	2.3 [0.7, 4.3]
WF-VAE	70.0 [64.7, 75.0]	11.0 [7.7, 14.7]	9.0 [6.0, 12.3]
PERMATOK	**78.3** [73.7, 83.0]	**28.7** [23.7, 33.7]	**24.3** [19.7, 29.0]
PERMATOK-SC	42.7 [37.0, 48.3]	14.7 [10.7, 18.7]	9.7 [6.3, 13.3]
PERMATOK-R	55.7 [50.0, 61.3]	10.3 [7.0, 14.0]	8.3 [5.3, 11.7]

Human preservation audit. We audit whether human judgments reproduce the judge's method ordering and how closely individual labels agree. Two non-author readers each assessed 120 generated clips: 40 shared test scenes, four per object-count and camera-type group, from each of the PERMATOK, WF-VAE and SD-VAE rows of Table 2. Both were blind to method identities, judge answers and the other reader's labels, and neither saw any judge label during labelling. They used the preservation question and visual evidence given to the judge. Reader 1's analysis was fixed before labelling and run once, without an agreement threshold. The readers' rates are reported separately; no consensus labels are constructed.

Agreement between the judge and reader 1 is 80.0% ($\kappa = 0.603$), and the judge is conservative: it rejects 20 clips that reader 1 accepts and accepts only four that reader 1 rejects (Table 19). Both readers rank PERMATOK above WF-VAE and SD-VAE. Reader 1's paired advantages are $+25.0$ points $[12.5, 37.5]$ and $+37.5$ $[22.5, 52.5]$, respectively (95% CIs); the judge's corresponding gaps are $+20.0$ $[2.5, 37.5]$ and $+30.0$ $[10.0, 47.6]$. Reader 2's per-method rates give advantages of $+25.0$ and $+47.5$ points, so both readers see a larger PERMATOK advantage than the judge.

Table 19: **Human preservation audit.** Left: agreement between the judge and reader 1; right: preserved/total (%). CI: confidence interval.

Measure	Value	95% CI
Agreement	0.800	[0.733, 0.867]
Cohen's κ	0.603	[0.466, 0.734]
	Judge yes	Judge no
Reader 1 yes	54	20
Reader 1 no	4	42

Method	Reader 1	Reader 2	Judge
PERMATOK	33/40 (82.5)	35/40 (87.5)	26/40 (65.0)
WF-VAE	23/40 (57.5)	25/40 (62.5)	18/40 (45.0)
SD-VAE	18/40 (45.0)	16/40 (40.0)	14/40 (35.0)

Procedure. An author supervised reader 1's session, and labels from a discarded calibration pass are excluded. Each case is a single yes/no recognizability question, answered in a median of 5 seconds. Agreement between the judge and reader 1 ranges from 0.775 to 0.825 across methods, and both readers reproduce the ranking.

Aggregation, uncertainty and missing data. Frame and pair measurements are averaged within scenes before cross-scene aggregation; the primary set has one target per scene. Conditional errors use only scenes with eligible detections or pairs, and report their contributing counts. The completed primary rows have all 300 scenes with no missing frames.

For the automated evaluations, reported 95% intervals use 10,000 scene-bootstrap resamples. Paired contrasts use shared scene identities and one common resample, restricted to jointly resolved scenes where needed; this denominator can differ from that of a displayed marginal rate. Semantic ratio rates are bootstrapped as ratios of per-scene numerator and denominator sums. These intervals quantify scene sampling for the evaluated outputs, not training-seed variability or judge accuracy.

Controls and measurement limits. GT-as-generation and tokenizer reconstructions check different parts of the pipeline. Reconstructions receive true future frames and are not forecasting competitors. WF-VAE's visible control uses $t = 12$ rather than 14, because its causal context ends at frame 12.

B.3 PAIRED EFFECTS AND SECONDARY OUTCOMES

Table 20 reports the 11 primary generator configurations, and Table 21 adds the 29.4k rows and the monocular row without fine-tuning. All rows are scored on the same 300 scenes, and these generators use DiT-B. Their intervals quantify scene variation for the evaluated outputs, not replication over independently trained models.

Table 20: **Conditional generation.** P@0.3: mask-IoU success at 0.3; S_{15}: 15-pixel localization success; †unannealed checkpoint.

Representation	Scenes	Updates	P@0.3 ↑ [95% CI]	S_{15} ↑
SD-VAE	1,600	100k	0.067 [0.043, 0.093]	0.157
WF-VAE (16 ch.)	1,600	100k	0.048 [0.028, 0.071]	0.170
WF-VAE	1,600	100k	**0.163** [0.126, 0.203]	**0.272**
PERMATOK	1,600	40k	0.070 [0.044, 0.098]	0.160
SD-VAE	9,600	100k	0.041 [0.022, 0.062]	0.109
WF-VAE (16 ch.)	9,600	100k	0.013 [0.003, 0.027]	0.098
WF-VAE	9,600	100k	0.098 [0.069, 0.129]	0.210
PERMATOK	9,600	40k	**0.378** [0.330, 0.428]	**0.612**
SD-VAE	9,600	200k	0.064 [0.042, 0.090]	0.152
WF-VAE (16 ch.)†	9,600	200k	0.030 [0.013, 0.049]	0.132
WF-VAE	9,600	200k	0.210 [0.169, 0.253]	0.364

Table 21: **Additional conditional results.** Pres.: preservation; P@0.3: mask-IoU success at 0.3; S_{15}: 15-pixel localization success.

Model	Scenes	Updates	Pres. (%)↑	P@0.3↑	S_{15} ↑	Position (px)↓
PERMATOK	29,400	40k	**78.6**	**0.3967** [0.3489, 0.4456]	**0.5911**	**12.14**
WF-VAE	29,400	100k	40.3	0.1422 [0.1067, 0.1811]	0.2422	17.06
SD-VAE	29,400	100k	31.0	0.0289 [0.0133, 0.0467]	0.1078	22.81
WF-VAE (16 ch.)*	29,400	80k	22.0	0.0144 [0.0044, 0.0267]	0.1678	19.61
PERMATOK, 1-cam, frozen	9,600	40k	52.3	0.1022 [0.0733, 0.1333]	0.3989	17.13

*80k checkpoint. Preservation uses the same judge protocol and excludes unresolved responses; the PERMATOK 29.4k rate uses 299 definite responses. Preservation intervals are not reported for these rows. Position is detection-conditioned, whereas missing detections fail the success metrics. The 1-cam row without fine-tuning freezes the six-camera-trained tokenizer and encodes context and target from the same single camera; it is distinct from the same-camera supervision control.

Relative to WF-VAE-9.6k at 200k updates, PERMATOK-9.6k has a paired P@0.3 difference of 0.168 [0.107, 0.227] and an S_{15} difference of 0.248 [0.183, 0.311]. Table 16 reports results by camera type.

Mask-extractor controls. The empty control applies the mask extractor to the generated frame at the last fully hidden time, inside the regions used for the three return frames, where any detection is counted. The visible control applies it at conditioning times where the target is visible. We keep one fixed extractor for all models rather than tuning it per model, so its empty-control rate, 19.3% for PERMATOK-1.6k, 20.7% for PERMATOK-9.6k and 21.6% to 39.4% for the baseline configurations, is a shared property of the measurement. The mask scores therefore measure spatial agreement, and identity is assessed by preservation and by the color check of hide-and-reappear success (Appendix B.2).

B.4 POSITION AND DYNAMICS EVALUATION

Position and detection availability. Let $c_{s,t}$ and $\widehat{c}_{s,t}$ be the centroids of the GT and generated masks in the frozen search region, and $e_{s,t} = \|\widehat{c}_{s,t} - c_{s,t}\|_2$ in pixels. The reported Position metric first averages $e_{s,t}$ over detected frames within each scene, then over scenes with at least one detection. It is conditional on detection; a model can avoid difficult errors by failing to produce a

detectable object. We therefore also report

$$S_{15} = \frac{1}{N}\sum_s \frac{1}{3}\sum_{t\in W_s} \mathbf{1}[d_{s,t}=1 \ \wedge \ e_{s,t}\le 15], \qquad D = \frac{1}{3N}\sum_s \sum_{t\in W_s} d_{s,t}. \tag{5}$$

S_{15} uses all return frames and treats a missing detection as failure; D is detection availability. Neither proximity nor detection guarantees identity.

Motion success and conditional velocity error. For $t \in \{r_s, r_s+1\}$, define $v_{s,t} = c_{s,t+1} - c_{s,t}$ and $\widehat{v}_{s,t} = \widehat{c}_{s,t+1} - \widehat{c}_{s,t}$. The velocity error is $E_{s,t} = \|\widehat{v}_{s,t} - v_{s,t}\|_2$ in pixels/frame. The all-pair endpoint is

$$\mathrm{Motion}_\delta = \frac{1}{N}\sum_s \frac{1}{2}\sum_{t=r_s}^{r_s+1} \mathbf{1}[d_{s,t}=d_{s,t+1}=1 \ \wedge \ E_{s,t}\le\delta]. \tag{6}$$

All 600 GT pairs remain in its denominator, with missing endpoint detections counted as failures. We report the curve over $0 \le \delta \le 10$ pixels/frame as an exploratory, post-hoc sensitivity analysis of the saved final-test outputs (Figure 3). No single tolerance is selected as a calibrated primary endpoint. A consistently offset trajectory can have low velocity error, so S_{15} and preservation are reported alongside it.

Conditional velocity error averages $E_{s,t}$ over available pairs within each scene and then over contributing scenes. Pair availability is $(2N)^{-1}\sum_{s,t} d_{s,t}d_{s,t+1}$. Speed error is $|\|\widehat{v}\|_2 - \|v\|_2|$. All velocities are in the query image plane and include camera motion; multiplying by 12 converts pixels/frame to pixels/second.

Scope of dynamics evaluation. These metrics assess motion during visible return, not an observed trajectory inside the fully hidden interval. Figure 3 plots the motion-tolerance curves in the main text. At the evaluated tolerances, PERMATOK has higher all-pair motion success than the three baselines. For example, success is 32.5% versus 21.7% for WF-VAE at two pixels/frame, and 53.0% versus 34.2% at three. These points illustrate the curve rather than define a chosen threshold. With no error restriction, detection-pair availability caps success at 76.7%, 56.2%, 39.8% and 30.5% for PERMATOK, WF-VAE, SD-VAE and 16-channel WF-VAE, respectively. GT-as-generation reaches 100% from two pixels/frame. The curves retain the frozen detector's identity limitations; the intervals are pointwise, not simultaneous bands.

Table 22: **Privileged motion references.** Error: projected-center distance (px); S_{15}: localization success within 15 px.

Protocol	Reference	Times	Error (px) [CI] ↓	S_{15} ↑	Frames
PERMATOK / SD-VAE	Constant velocity	8,14	0.4 [0.4, 0.5]	1.000	900
PERMATOK / SD-VAE	Stationary	8,14	67.8 [66.3, 69.4]	0.000	900
WF-VAE	Constant velocity	6,12	0.7 [0.6, 0.8]	1.000	900
WF-VAE	Stationary	6,12	76.1 [74.4, 77.9]	0.000	900

Mean ground-truth speed in the scored window is 3.7 [3.6, 3.9] pixels/frame. Detected-mask centroids elsewhere use a different center definition, so these references bound the task rather than the scorer.

Motion references. The stationary reference keeps the last observed target position; constant velocity extrapolates from two visible past states. Simulator-state references receive privileged 3D positions and project predictions through the query camera. These past states lie inside each model's context, at times 8 and 14 for PERMATOK and SD-VAE and at 6 and 12 for WF-VAE. No future object state is provided to the predictor; projection uses the query-camera trajectory. The constant-velocity reference has low error, which shows that the selected return windows largely admit simple extrapolation given true past state. It does not test inference of that state from pixels or establish generalization to nonlinear, contact-rich or unseen motion regimes.

B.5 UNCONDITIONAL EVALUATION AND RESOURCE ACCOUNTING

References and metrics. Each quality row uses 2,048 generated videos. At 1.6k scenes, the SD-VAE and WF-VAE rows use 50 sampling steps and all other rows use 100; for the 16-channel WF-VAE at 1.6k scenes, FVD is 154.83 with 50 steps and 153.28 with 100. The Base reference contains

2,000 real clips from 200 validation scenes. The Base/Large reference is a fixed set of 2,048 clips from Base and Large validation scenes; it is not matched to the training mixture. FVD (Unterthiner et al., 2018), video KID (Bińkowski et al., 2018) and coverage (Naeem et al., 2020) use their declared feature pipelines. We compare FVD only within a shared reference and preprocessing setup.

Generation quality. Let R and G be real and generated video features under one fixed extractor and preprocessing pipeline. With means μ_R, μ_G and covariances Σ_R, Σ_G, FVD (Unterthiner et al., 2018) is

$$\mathrm{FVD} = \|\mu_G - \mu_R\|_2^2 + \mathrm{tr}\Big(\Sigma_G + \Sigma_R - 2(\Sigma_G\Sigma_R)^{1/2}\Big). \tag{7}$$

KID (Bińkowski et al., 2018) in the main text denotes video KID, which estimates kernel discrepancy between video-feature distributions. Lower FVD and KID indicate closer distributions. Reported KID error bars are subset standard deviations, not confidence intervals.

Precision, recall, density and coverage. The PRDC analysis (Naeem et al., 2020) uses I3D features (Carreira & Zisserman, 2017) with $k = 5$. For each real feature r_i, let ρ_i be its distance to the kth other real neighbor; define generated radii analogously. Precision is the fraction of generated features inside any real neighborhood ball. Recall is the fraction of real features inside any generated ball. Density is the average number of real balls containing a generated feature, divided by k, and can exceed one. Coverage is

$$\mathrm{Coverage} = \frac{1}{|R|}\sum_i \mathbf{1}\left[\min_{g\in G}\|r_i - g\|_2 \le \rho_i\right]. \tag{8}$$

Coverage measures how much of the reference distribution has a nearby generated example.

Latent size. Table 23 reports the 9.6k latent-size comparison for WF-VAE. The four-channel WF-VAE reaches lower FVD than the 16-channel variant (59.08 versus 104.01). Generator update budgets differ across families.

Table 23: **Latent size.** FVD: Fréchet Video Distance; Cov.: coverage.

Tokenizer	FVD↓	Cov.↑
WF-VAE (16 ch.)	104.01	0.77
WF-VAE	59.08	0.90
PERMATOK	**31.31**	**0.95**

9.6k scenes; 100 sampling steps; WF-VAE: 160k updates; PERMATOK: 80k. Bold: best quality.

Table 5 reports the primary generation-quality comparison.

Unconditional training data. Counts of independent scenes, encoded targets and input frames differ between pipelines and exclude tokenizer supervision.

Video-latent training sets use ten cameras and 37 frames per scene; scene-latent training sets use six cameras and six times. The 1.6k SD-VAE generator uses its 90k-update checkpoint with batch 16. The 1.6k WF-VAE checkpoints use 100k updates with batch 16; PERMATOK uses 40k with batch 32. At 9.6k, SD-VAE and WF-VAE use final 160k-update checkpoints with batch 16 and PERMATOK uses 80k with batch 32. Both nominal totals are 2.56M target presentations, but targets, sequence lengths and compute differ.