TRACING HOW ROBOT WORLD MODELS HALLUCINATE OBJECTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned video diffusion models now render robot motion convincingly, yet the objects the robot manipulates are often left behind, duplicated or lost. Standard video metrics miss these failures and even reward them, since a clip that repeats its first frame beats the generated video on temporal-difference error in 98% of 540 cases. Rather than adding data or conditioning, we present, to our knowledge, the first mechanistic account of how a robot video world model carries objects, using causal interventions in frozen diffusion transformers (DiTs). The control renders only the arm and steers it through attention routing, since restoring only the queries and keys (Q/K) of the correct control removes 84–88% of the error a corrupted control adds. The object appears only in the first image, so the model must copy it forward. We find that rotary position embedding (RoPE) turns the first image's temporal index into an address for this copy, and shifting that index alone makes the video return to its starting scene inside the predicted frame windows, in 10 of 10 held-out episodes and in two public video models. The copy is already visible in the model's first denoising step and is stored in the partially denoised latent, which redraws the object at its old place after the hand has moved it. Guided by this mechanism, we paint the object into the control, which makes it follow the hand in 6 of 16 new failures, and in our 14B model re-noising a leftover copy's region early removes the copy while keeping the carry. These findings trace a basic limitation of current robot world models to the fact that they are told where objects are only in the first image, and point future models toward conditioning on object state, acting from the first denoising steps and evaluating whether objects go where the actions send them.

1 INTRODUCTION

A robot can plan by predicting the outcome of its actions (Finn & Levine, 2017; Du et al., 2023; Yang et al., 2024). Action-conditioned video diffusion models are now a common way to do this. Given a camera image and a planned motion, they generate a video of what would happen (Alzayer et al., 2026; Gao et al., 2026; NVIDIA et al., 2025; Guo et al., 2026). For planning, what matters most in that video is how the robot's actions change the objects around it (Finn et al., 2016).

These models render the robot well and often get the objects wrong. In Figure 1a, the arm reaches for a red bowl and moves to a blue stack along the recorded path, but the bowl stays where it started.

To understand this failure, we first looked at the model's inputs. The control video renders the robot's arms on a gray background in every frame. The object appears only in the first image, so the model has to carry it forward. Finn et al. (2016) built this step into their predictor by moving the previous frame's pixels. A diffusion transformer (Peebles & Xie, 2023) has no such module, but its attention lets every frame read the first image and every other frame, so it can learn to copy. We therefore hypothesized that objects persist in these models by being copied forward, and that each copy stays tied to where and when its source was drawn. Such copying would keep static objects in place and leave moved objects at their old positions.

We test this hypothesis with causal interventions in frozen models (Meng et al., 2022b; Wang et al., 2023), mainly a 1.3B Wan2.1 control model fine-tuned on a two-arm robot (Wan Team, 2025; Alzayer et al., 2026), and repeat the main tests in a 14B fine-tune, DreamDojo (Gao et al., 2026) and

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

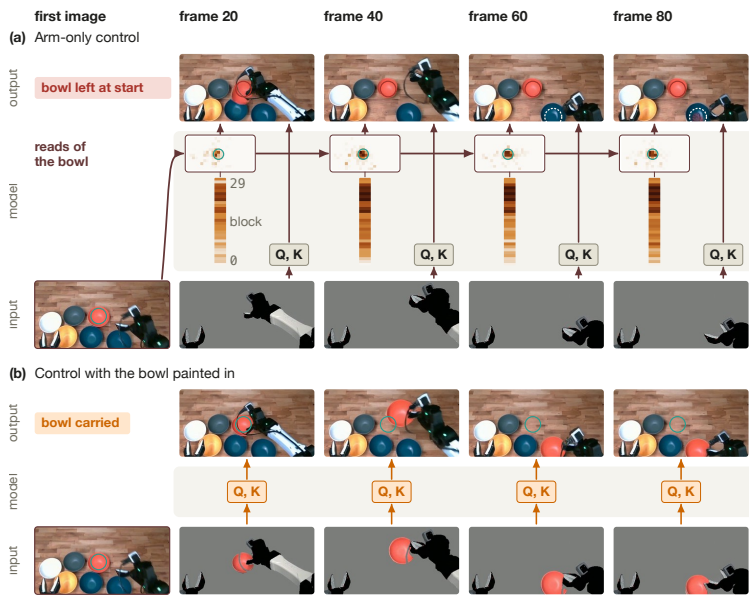

Figure 1: **The bowl stays behind under the arm-only control and is carried when painted into the control.** Overhead frames of one episode. (a) Arm-only control. Each frame reads the bowl's first-image tokens at its old place (maps at update 1, strips by block), and the bowl stays behind. (b) With the bowl painted into the control, the model carries it to the stack.

two public image-to-video models. Our main findings are as follows, and Appendix A summarizes them.

- We show that **restoring the attention routing of the control brings back the arm but not the object**. When the control is corrupted, restoring only the queries and keys it would have produced, which decide where each token reads, removes 84–88% of the added prediction error. Restoring the values alone also moves the arm (Section 4).
- We show that **the object can follow the hand once it is drawn into the control video**. Painting the carried object's recorded pixels into the rendered arm video repairs 6 of 16 new failures, and a stationary object painted the same way repairs none (Section 5).
- We find that **without that input, the model copies the object from earlier content that it finds by time position**. Rotary position encoding places the first image at time 0. Moving it to a later time makes the video return to its starting scene at the frames that share the new time, as the rotary score predicts, in 10 of 10 held-out episodes and in two public video models (Section 6).
- We find that **the outcome is visible from the first denoising update and a leftover copy is stored in the partially denoised video**. The model's first estimate of the finished video, at update 1 of 50, already shows the object only at its start in 14 of 16 failures and in none of 7 carries. In our 14B model, re-noising the copy's region once at update 10 removes it (Sections 7 and 8).
- We show that **standard video metrics reward these failures**. A clip that repeats its first frame beats the generated video on temporal-difference error in 98% of 540 cases, so we judge each video by where its objects end up (Section 3.4).

2 Related work

Robot video world models. Action-conditioned video prediction lets robots plan by predicting how their actions affect objects (Finn et al., 2016; Finn & Levine, 2017; Du et al., 2023; Yang et al., 2024). Current robot world models condition video diffusion on rendered motion, actions or recent frames (Alzayer et al., 2026; Gao et al., 2026; NVIDIA et al., 2025; Guo et al., 2026; Chen et al., 2024). Video generators often violate physics and imitate training cases (Kang et al., 2025; Motamed

et al., 2026; Chefer et al., 2025). Li et al. (2026) trace such violations in Wan to rotary encoding that biases attention toward one image position. Section 8 tests this account on object transport.

Interpreting generative transformers. We build on activation and path patching (Meng et al., 2022b; Wang et al., 2023; Heimersheim & Nanda, 2024; Zhang & Nanda, 2024) and on the separation of attention into routing and content (Elhage et al., 2021). In diffusion models, prior work traces features in diffusion transformers (Mazur et al., 2026), finds temporal correspondence in video attention (Nam et al., 2025), and edits images and videos through attention and diffusion features (Hertz et al., 2023; Tumanyan et al., 2023; Cao et al., 2023; Alaluf et al., 2024; Epstein et al., 2023; Geyer et al., 2024; Li et al., 2024; Yatim et al., 2024; Xiao et al., 2024; Pondaven et al., 2025; Jun et al., 2026). We use such edits to test causally where an object's position and appearance come from.

Conditioning, position and time. ControlNet adds spatial conditions through a trained copy of the encoder (Zhang et al., 2023), and trajectory-conditioned video models build on such designs (Geng et al., 2025; Zhang et al., 2025). Our control is concatenated at the input. Rotary encoding makes attention depend on relative position (Su et al., 2024), and rescaling it changes how far attention reaches (Chen et al., 2023; Zhao et al., 2025). Some video models are trained to accept frames at chosen positions (Wan Team, 2025; HaCohen et al., 2024). We show that models trained with only a first image also copy by time position. Diffusion models commit to coarse structure early (Biroli et al., 2024). Partial re-noising redraws part of an image (Meng et al., 2022a; Lugmayr et al., 2022). We find the object's outcome already visible in this early window. Appendix L reviews more work.

3 Method

3.1 Video diffusion model and sampler

The model generates the whole clip at once, refining all frames together over 50 denoising steps, which we call updates. A VAE compresses each clip of up to 81 frames, with the three cameras tiled into one mosaic, into up to 21 latent frames, each a grid of tokens. The first camera image is encoded as one more latent frame placed at temporal index 0 on the same grid, and generated latent j sits at index $j{+}1$. The control video renders the robot's arms from measured joint and gripper states on a gray background. It is encoded by the same VAE and concatenated with the noisy latent channel by channel before the linear patch embedding, so it enters each token additively, $e_i = W_z z_i + W_c c_i$, at that token's own place and time. Other tokens see the control only through attention, so we look for its effect in the queries, keys and values. Nothing in the input marks where an object is once the hand has moved it.

The sampler follows flow matching (Lipman et al., 2023). The state after update k is $z_k = (1 - \sigma_k)\, x_0 + \sigma_k\, \epsilon$ for a noise level σ_k that falls from 1 to 0, and the network predicts the velocity $\hat{v} = \hat{\epsilon} - \hat{x}_0$. From it we read the model's current estimate of the finished video,

$$\hat{x}_0^{(k)} = z_k - \sigma_k\, \hat{v}, \tag{1}$$

which we call the *clean estimate*. Sampling uses classifier-free guidance (Ho & Salimans, 2022), $\hat{v} = \hat{v}_\varnothing + w(\hat{v}_p - \hat{v}_\varnothing)$ with $w = 5$. Both branches share the control and first image and differ only in the text prompt, so guidance never amplifies the control. Every intervention is applied to both.

3.2 Attention by source and rotary time offsets

Tokens exchange information only through self-attention. A query token i writes $o_i = \sum_j \alpha_{ij} v_j$ with $\alpha_{ij} = \mathrm{softmax}_j(q_i^\top k_j / \sqrt{d})$. The queries and keys set where a token reads, and we call them the *routing*. The values supply what it reads. Because the weights sum to one, the output splits by source. Let $a_i = \sum_{j \in \mathcal{R}} \alpha_{ij}$ be the share of attention on the first image $\mathcal{R}$. Then

$$o_i = a_i\, m_i^{\mathcal{R}} + (1 - a_i)\, m_i^{\mathcal{G}}, \tag{2}$$

where $m_i^{\mathcal{R}}$ and $m_i^{\mathcal{G}}$ are the attention-weighted means of the values read from the first image and from the generated frames (Figure 2a). A small share can still carry a large contribution, so we measure what is read as well as how much. To change only one source, we scale the weights on a key set $\mathcal{S}$ by a gain g and renormalize each row, $\alpha'_{ij} = g^{[j \in \mathcal{S}]} \alpha_{ij} / \sum_l g^{[l \in \mathcal{S}]} \alpha_{il}$.

Queries and keys carry three-dimensional rotary encoding (Su et al., 2024). For a query at temporal index τ and a key at τ' with the same spatial offset, the score is a sum over rotary frequency pairs ω,

$$q^\top R(\tau - \tau')\, k = \textstyle\sum_\omega \rho_\omega \cos\big(\omega(\tau - \tau') + \phi_\omega\big), \tag{3}$$

where ρ_ω and ϕ_ω depend only on the query and key contents. When the contents that dominate a query's reads of the first image have small phases ϕ_ω, every term peaks at $\tau = \tau'$, and the generated frame that shares the first image's index reads it most (Figure 2b). If the image is moved to index c, the peak should move to generated latent $c - 1$, which decodes to video frames $4c - 7$ to $4c - 4$. Section 6 tests this prediction.

3.3 Causal interventions

Following activation patching (Meng et al., 2022b; Wang et al., 2023), two runs differ in one input, and one internal signal is carried from the donor into the recipient and recomputed at every update from the recipient's own state. All else runs as usual. *Routing restoration* replaces the recipient's queries and keys with those of a shadow pass under another control, $o_i = \sum_j \mathrm{softmax}_j(\tilde{q}_i^\top \tilde{k}_j/\sqrt{d})\, v_j$, with the values, feed-forward layers and normalization computed from the recipient. *State edits* replace the tokens of a region of z_k once, after update k. The replacement can be another sample's tokens, the model's own estimate re-noised with new noise, $(1 - \sigma_k)\hat{x}_0^{(k)} + \sigma_k \epsilon'$, or new noise alone. *Position edits* change the first image's temporal index or add a second clean image at a chosen index. *Input edits* paint objects into the control under SAM 2.1 masks (Ravi et al., 2025). *Path patching* (Wang et al., 2023) holds a downstream signal at its unedited value to test whether an effect needs it, and replays only its changed value to test whether it suffices. We compare every edit with an exact replay of the unedited run, which we call the native run, and with the same edit applied elsewhere in the scene (Appendix C).

3.4 Models and outcome labels

Our primary model is the 30-block Wan2.1 1.3B control backbone (Wan Team, 2025) adapted with Masked Visual Actions (Alzayer et al., 2026) to a two-arm tabletop robot with an overhead camera and two wrist cameras, at steps 2,000, 3,500 and 9,000 of one fine-tuning run. We test scale with a 14B two-expert fine-tune, another action input with DreamDojo (Gao et al., 2026), and generality with Wan2.2-TI2V-5B (Wan Team, 2025) and LTX-Video (HaCohen et al., 2024), which never saw our robot and receive no actions.

Standard video metrics reward the failure. A sample that deleted a cup stack had 24% lower pixel error over the object region than a sample that kept it (Appendix D.1). We therefore label each video from every frame in all three cameras as carried, left at its start, duplicated, vanished, wrong object or unreadable (Appendix D).

4 Restoring attention routing restores the arm

The control enters every token at its own place and time, so it can change where each token reads. If it steers the arm mainly through routing, then after we corrupt the control, restoring only the queries and keys that the correct control would produce should repair the prediction.

Prediction error. We corrupt the control by freezing it or playing it backwards and measure the error of the predicted update at a fixed noisy input. On eight held-out episodes, restoring queries and keys at all 30 blocks removes 84–88% of the added error at high noise, for both corruptions and two checkpoints (steps 2,000 and 9,000), and it helps in all 32 comparisons (Figure 3a). The recovery grows steadily with the fraction of routing restored, to about 60% at update 26 of 50, but it does not help at the last update (Figure 3b). In our 14B fine-tune it removes 89% of the added error. Values carry part of the control as well, and restoring them alone also brings the arm back (Appendix E).

Complete videos. When we recompute the restoration at every update and let generation finish, whole-scene error falls in all 12 comparisons, and the arm moves again and closes where the recorded gripper closes, but the objects do not follow. Bowls keep the wrong color or stay in place

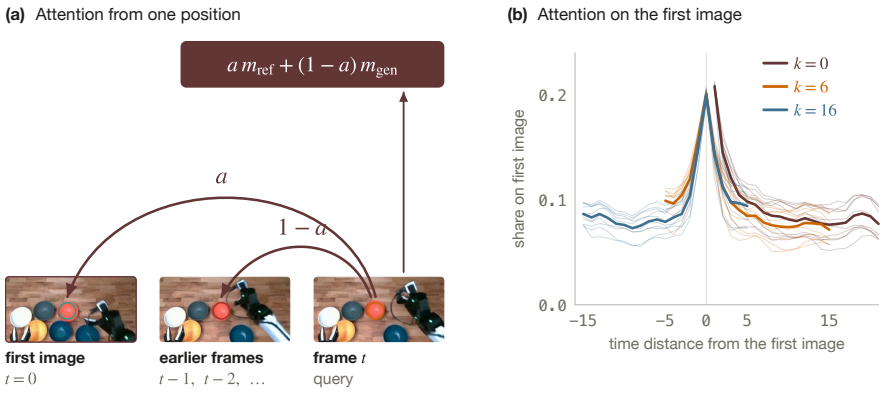

Figure 2: **Each place copies from the first image and from earlier frames.** (a) A query at the bowl's old place reads the first image with share a and earlier frames with $1 - a$ (Equation 2). (b) Share on the first image against time distance, for temporal indices 0, 6 and 16 in ten held-out episodes, peaking at zero distance (Equation 3).

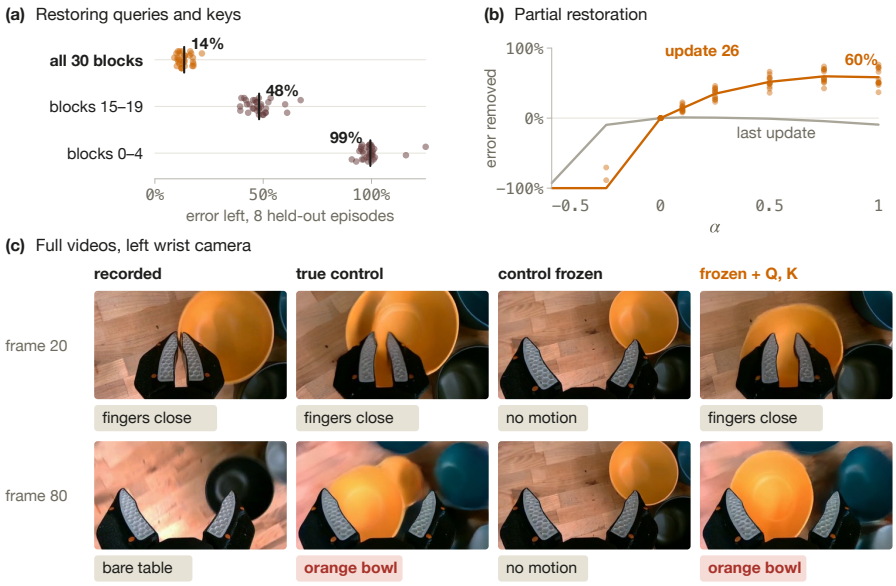

Figure 3: **Restoring queries and keys repairs only the arm.** (a) Error left after restoring blocks 0–4, 15–19 or all 30 (8 held-out episodes, 2 checkpoints, 2 corruptions). (b) Error removed against the restored fraction α at update 26 and at the last update. (c) Left wrist camera, frames 20 and 80.

after the recording has moved them (Figure 3c). This dissociation could mean that the arm and the object use separate routes, or simply that the control does not show the object. Section 5 tests the second explanation.

5 With the object in the control, the object can follow the arm

If the object fails because it is missing from the input, adding it to the input should make it behave like the arm, and restoring routing should then repair it too. We paint the carried object's recorded pixels into the control at every frame, under SAM 2.1 masks (Ravi et al., 2025).

Painted object. Figure 4b shows the first eight failures we tried. With the carried object in the control, the object follows the hand and its start clears in the three cases where we first saw the effect and in 2 of the 5 others, with a partial carry in a third. A stationary object painted the same

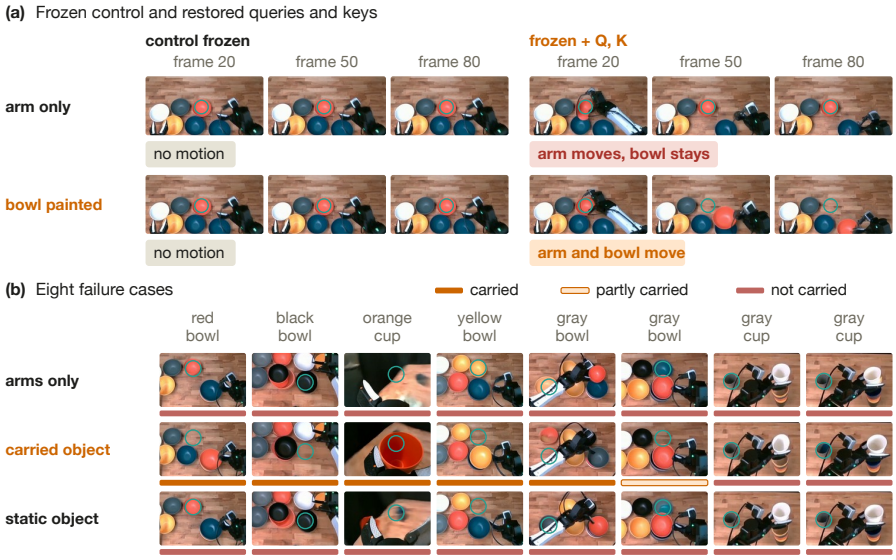

Figure 4: **With the object in the control, restoring routing can bring the object back with the arm.** (a) Red-bowl episode with a frozen control, alone or with the correct control's queries and keys restored, for the arm-only control and with the bowl painted in. (b) All eight failures at a late overhead frame under three controls, with the outcome as a bar.

way fixes none of the eight. We then repeated the test on 16 failures from 44 new episodes. The painted object gave a carry in 6 and the stationary object in none, and SAM 2.1 tracks of the object agree (Appendix F). The repair comes from information about the object that moves, although most failures remain.

Routing restoration with the object in the control. We then repeated the routing experiment with the object in the control. In the red-bowl episode (Figure 4a), freezing the control stops all motion, and restoring the queries and keys of the moving control brings the arm back with either control. The bowl comes back only when it is in the control, and it then lifts, clears its start and follows the hand to the stack. Across the 16 new failures, restoring queries and keys brought the arm back in 15 with either control, and brought the object along in 4 when it was painted in against 1 when it was not. Restoring the values alone also moved the arm but carried no object. The dissociation of Section 4 therefore comes mostly from what the control contains.

Painting the object in only until the lift fixes one of three cases, so the model also needs to see where the object goes after the lift. Our 14B fine-tune ignores painted content, and a bowl painted blue in its control stays orange.

6 The model selects what to copy by time position

We next asked how the model chooses which earlier content to copy into a given frame. Equation 3 makes every attention score depend on the time offset between query and key, so the first image's temporal index should act as an address. If it does, moving the image to a later time should make its content reappear at the frames that share the new time.

Shifting the first image's temporal index. We change only the first image's temporal rotary index, in every query and key, from 0 to 11. Near frame 39 the arms and bowls move back toward the starting arrangement for a few frames and then continue (Figure 5c). Frames 37–40 are the frames of the latent at time 11, and shifting as many background tokens gives no return. On ten held-out episodes, the frame closest to the start falls inside the predicted window for times 6 and 16 in every episode (Figure 5b), and six of the ten also pass a stricter check in every camera. The same holds in Wan2.2 and LTX-Video (Figure 5b,c) and partly at 14B (Table 2).

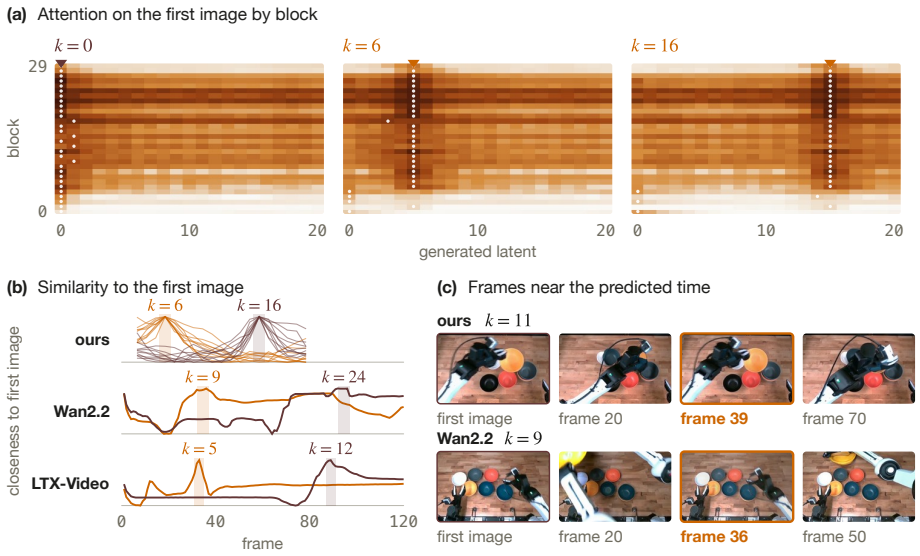

Figure 5: **The first image's temporal index sets where it is read and redrawn.** (a) Share of attention on the first image by block and generated latent at update 26, ten held-out episodes, temporal indices 0, 6 and 16. (b) Similarity of each frame group to the first image, with predicted frames shaded. (c) Frames at the return in our model (index 11) and Wan2.2 (index 9).

In the attention maps of each block, the generated latent that reads the first image most is the one that shares its time. With the image at time 6 or 16, that peak moves to latent 5 or 15 in 24 and 25, respectively, of the 25 blocks from block 5 onward (Figure 5a), as Equation 3 predicts.

A future frame at its own temporal index. The time address should also let a clean future frame be copied into its matching frames. We add the recorded frame 40, in which the arm has lifted the red bowl, as a second clean image at time 11, without retraining. Near frame 39 the model redraws the bowl in the gripper, and from there the bowl travels with the arm to the stack, although no input shows it after frame 40 (Figure 6a). The carry also survives when frames 37–40 cannot see the added frame (Appendix G). Early generated frames act as copy sources too. In a case where a cup vanishes at the grasp, exchanging only the values that the first generated latent sends to later frames exchanges the outcome, and path patching shows that the effect runs through the later queries (Appendix H).

7 The outcome is visible in the first update

Diffusion models settle coarse structure early (Biroli et al., 2024), so we asked whether the object's outcome is settled early too.

Update schedules. We switched the control partway through denoising (Figure 6b). With the object in the control for only the first 5 of 50 updates, the red bowl is still carried to the stack and the black bowl is placed with its start cleared, while adding the object only from update 11 fails. The future frame of Section 6 shows the same timing. Given only during updates 1–5 it produces the carry, introduced from update 11 it leaves the bowl beside the stack, and introduced from update 26 it leaves the bowl at its start. Across the 16 new failures this timing does not hold in general. Object input only in updates 1–5, or only from update 11, gave 2 carries each, against 6 with input throughout (Appendix F), and early-only input mostly turned failures into grasps of the wrong object.

First estimate. At update 1, the clean estimate (Equation 1) of the red-bowl run already has the arm at the stack and the bowl at its start (Figure 6c). On 44 new episodes, each first estimate was labeled before its final video was viewed. It shows the object only at its start in 14 of 16 failures and in none of 7 successful carries (Table 3). Appearance settles later, by update 25 (Appendix I).

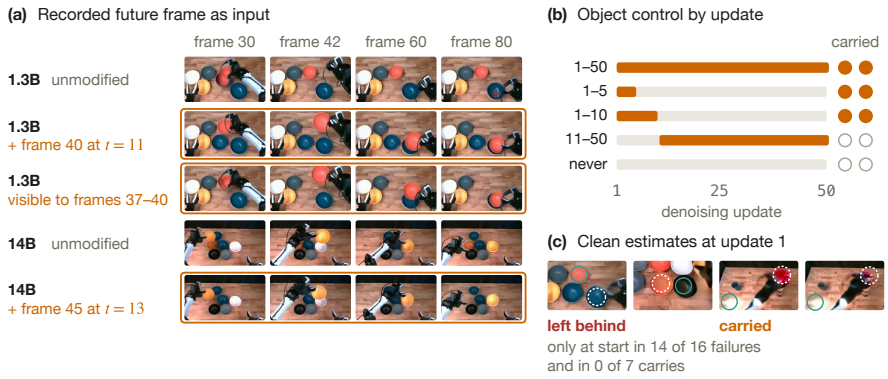

Figure 6: **A future frame is copied at its own time, and the first estimate shows the outcome.** (a) Recorded future frame added at its own time in the 1.3B and 14B models, outlines marking a carry. (b) Object in the control during chosen update windows, two cases. (c) First estimates at update 1 for runs that leave the object behind and runs that carry it. Examples come from an earlier set of runs, and counts from the 44 new episodes.

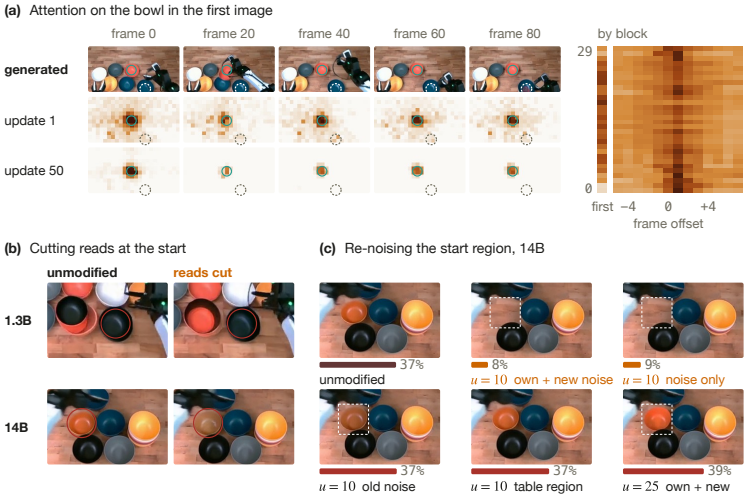

Figure 7: **A leftover copy is held in the partially denoised video.** (a) Attention to the red bowl's first-image tokens at updates 1 and 50 (start ringed, destination dashed), and by block and frame offset at the old place. (b) Cutting the old place's reads leaves the copy at 1.3B and 14B. (c) One 14B sample. New noise in the start region at update 10 removes the copy. Baseline edits leave it. Bars give the orange share in the region over frames 60–80.

8 A LEFTOVER COPY IS HELD IN THE PARTIALLY DENOISED VIDEO

In the simplest version of our account, a place that keeps copying the first image redraws the object after the hand has taken it, and the new place, empty in the first image, is redrawn empty. Li et al. (2026) attribute physics violations in the same base model to rotary encoding that biases attention toward the same position, and our read maps show this bias (Figure 7a).

Stale reads at the destination. In DreamDojo, a plate carried to a black stack collapses when the hand lets go. The conditioning frame shows the stack empty. Turning down the stack's reads of that frame keeps the plate, more so at lower gain. The same cut on background does not keep it (Figure 8a, Appendix Figure 23).

Read cuts at the source. If a leftover copy were held by its reads of the first image, cutting them after the hand has left should clear it. We applied this cut in three models (Figure 7b shows two).

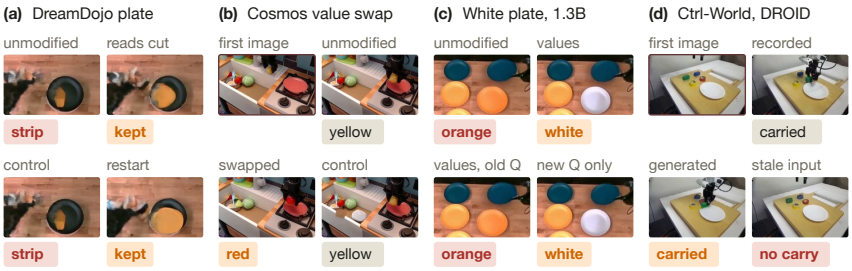

Figure 8: **The account in other cases and models.** (a) DreamDojo after release, with stale stack reads cut or a restart, against a background cut. (b) Cosmos-Predict2, first-image values of two objects swapped. (c) A white plate in our model kept white by the values of the first generated latent. (d) Ctrl-World, native and with its image condition fixed to the first frame.

Halving the reads leaves a black bowl's copy in our 1.3B model, and cutting them to zero leaves the object at its start in our 14B model and in LTX-Video. Cutting the old place's reads of every other generated frame as well only changes the copy's color, so these reads set its appearance.

Local re-noising. In our 14B fine-tune, replacing the state tokens over the bowl's start with those of a sample that clears it removes the copy in the overhead view and keeps the arm, carry and placement. The same patch over empty table has no effect. No clean sample is needed (Figure 7c). Re-noising the region once at update 10, with the model's own clean estimate and new noise or with new noise alone, lowers the orange share at the start from 37% to 8–9% and keeps the carry. New content under the old noise, the same reset over empty table and a reset at update 25 all leave the copy. At 1.3B the same reset is not specific to the copy's region (Appendix J).

9 The account in other cases and models

In another scene of our model, the values of the first generated latent (Section 6) keep an untouched white plate from turning orange (Figure 8c). In Cosmos-Predict2, which embeds actions globally, exchanging either the first-image values or the keys of a yellow potato and a red strawberry makes the gripper carry a mostly red object (Figure 8b). Ctrl-World re-feeds its latest frame for each chunk and never left an object behind. Fixing its image condition to the first frame stops both the carry and the arm (Figure 8d).

10 Discussion and conclusion

Robot video models see the arm in every frame and the object only in the first image, and this asymmetry helps explain their most characteristic failure. Restoring attention routing moves the arm, and once the object is painted into the control it can carry the object too. Without that input, the model copies the object from the first image and from frames it has already drawn, and rotary time position selects the source. The outcome is visible in the first denoising update, and a leftover copy is held in the partially denoised video.

World models should take object state, such as a mask or track along the planned gripper path, as input (Geng et al., 2025; Zhang et al., 2025) and be trained to read it. Guidance should act from the first denoising updates, where the outcome first appears. Since pixel and temporal scores reward this failure, world models should be scored on whether objects go where actions send them. More broadly, world models should keep an internal state for each object that actions update. Learning such state from unlabeled robot video over long horizons, with many objects and on new robots, is a large open problem.

AI USE STATEMENT

In this work, we used generative AI tools to design and give feedback on the experiments and their controls, and to support the qualitative analysis of generated videos. We have not used generative AI tools to generate synthetic data sets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims, to provide ingredients for or write proofs, to propose or refine hypotheses, to implement methods, to clean or reformat data sets, or to interpret results, and translation is not applicable to this work. Additionally, we used generative AI tools to create and modify scientific figures, to write and edit software code, to draft parts of the paper and edit it for readability, to brainstorm, to discover research topics and identify gaps, to search for information, to identify, summarize and analyze relevant literature, and to propose the title. We reviewed all AI-assisted work. Intervention code was checked against exact native replays and numerical tests, outcome labels were made from every frame of all three camera views before any score was computed, and every reference was checked against its source. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All experiments intervene in frozen model weights at inference time, and no model is trained. Section 3 and Appendices B to D describe the models, inputs, sampler settings, intervention sites and controls, and every experiment includes an exact native replay that reproduces the unedited output bit for bit. For each result we keep the protocol, the rule used to select cases, the inputs, the saved outputs and the labels, and we will release the intervention code and the generated videos with the paper.

ETHICS STATEMENT

This work analyzes generated videos and the internal computation of frozen video models, and no robot was driven by a generated plan. Better diagnostics of robot world models can make them safer to use for planning, and we see no direct negative application of the analysis.

REFERENCES

Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *ACM SIGGRAPH Conference Papers*, 2024. URL https://arxiv.org/abs/2311.03335.

Hadi Alzayer, Wenlong Huang, Haonan Chen, Christopher Luey, Lvmin Zhang, Maneesh Agrawala, Gordon Wetzstein, Li Fei-Fei, Yilun Du, Jiajun Wu, and Jia-Bin Huang. Masked visual actions for unified world modeling. *arXiv preprint arXiv:2607.19343*, 2026. URL https://arxiv.org/abs/2607.19343.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15, 2024.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. MasaCtrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2304.08465.

Hila Chefer, Uriel Singer, Amit Zohar, Yuval Kirstain, Adam Polyak, Yaniv Taigman, Lior Wolf, and Shelly Sheynin. VideoJAM: Joint appearance-motion representations for enhanced motion generation in video models. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.02492.

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

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, 2023.

Nelson Elhage, Neel Nanda, Catherine Olsson, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Dave Epstein, Allan Jabri, Ben Poole, Alexei A. Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.00986.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, 2016.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Naveed Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi Fan. DreamDojo: A generalist robot world model from large-scale human videos. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.06949.

Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, Oliver Wang, Andrew Owens, and Deqing Sun. Motion prompting: Controlling video generation with motion trajectories. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. TokenFlow: Consistent diffusion features for consistent video editing. In *International Conference on Learning Representations*, 2024.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, 2026.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. LTX-Video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024. URL https://arxiv.org/abs/2404.15255.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *International Conference on Learning Representations*, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Youngjun Jun, Seil Kang, Woojung Han, and Seong Jae Hwang. I'm a map! Interpretable motion-attentive maps: Spatio-temporally localizing concepts in video diffusion transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2603.02919.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. In *International Conference on Machine Learning*, 2025.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.07724.

Xirui Li, Chao Ma, Xiaokang Yang, and Ming-Hsuan Yang. VidToMe: Video token merging for zero-shot video editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yueyan Li, Haibo Wang, Caixia Yuan, and Xiaojie Wang. Why do video diffusion models violate physics? Unveiling the flaws in attention mechanisms. *arXiv preprint arXiv:2609.23658*, 2026.

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*, 2023.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Artyom Mazur, Nina Konovalova, and Aibek Alanov. DifFRACT: Diffusion feature reconstruction and attribution for circuit tracing. *arXiv preprint arXiv:2606.15796*, 2026. URL https://arxiv.org/abs/2606.15796.

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

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022b. URL https://arxiv.org/abs/2202.05262.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 948–958, 2026. URL https://arxiv.org/abs/2501.09038.

Jisu Nam, Soowon Son, Dahyun Chung, Jiyoung Kim, Siyoon Jin, Junhwa Hur, and Seungryong Kim. Emergent temporal correspondences from video diffusion transformers. In *Advances in Neural Information Processing Systems*, 2025.

NVIDIA, Niket Agarwal, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Alexander Pondaven, Aliaksandr Siarohin, Sergey Tulyakov, Philip Torr, and Fabio Pizzati. Video motion transfer with diffusion transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22911–22921, 2025. URL https://arxiv.org/abs/2412.07776.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568, 2024.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1921–1930, 2023. URL https://pnp-diffusion.github.io/.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, 2023.

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

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.00593.

Zeqi Xiao, Yifan Zhou, Shuai Yang, and Xingang Pan. Video diffusion models are training-free motion interpreter and controller. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.14864.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024.

Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://diffusion-motion-transfer.github.io/.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16042.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision*, 2023.

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

Min Zhao, Guande He, Yixiao Chen, Hongzhou Zhu, Chongxuan Li, and Jun Zhu. RIFLEx: A free lunch for length extrapolation in video diffusion transformers. In *International Conference on Machine Learning*, 2025.

A SUMMARY OF RESULTS

Table 1 summarizes the main causal tests.

Table 1: **Main causal tests.** Results are for our 1.3B model unless noted.

Finding	Intervention	Result
Routing restores the arm (Sec. 4)	Restore the queries and keys of the correct control at all blocks	84–88% of the added error removed, 89% at 14B, and the object stays wrong
The object is missing from the control (Sec. 5)	Paint the carried object into the control	Carried in 6 of 16 new failures, against 0 of 16 for a stationary object
The temporal index is a copy address (Sec. 6)	Move the first image to time 6 or 16	Return inside the predicted window in 10 of 10 held-out episodes, Wan2.2 and LTX-Video
The outcome is visible early (Sec. 7)	Decode the clean estimate at update 1	Object only at its start in 14 of 16 failures and 0 of 7 carries
A leftover copy is held in the state (Sec. 8)	Re-noise the start region once at update 10 (14B)	Orange share at the start falls from 37% to 8–9%, and the carry is kept
Metrics reward the failure (Sec. 3.4)	Score a clip that repeats the first frame	It beats the generation on temporal-difference error in 98% of 540 cases

B MODELS AND INPUTS

Primary model. Our main model is the Wan2.1 1.3B control backbone (Wan Team, 2025) adapted with Masked Visual Actions (Alzayer et al., 2026) to a two-arm tabletop robot. The adaptation is a LoRA update (Hu et al., 2022) on every self-attention, cross-attention and feed-forward projection of the 30 transformer blocks. The VAE and text encoder are frozen. We use checkpoints at steps 2,000, 3,500 and 9,000 of one fine-tuning run. Each clip has 81 frames (49 or 65 in some cup scenes) in which the overhead camera and the two wrist cameras are tiled into one mosaic. The VAE compresses it to 21 latent frames. The first recorded image is encoded as an extra latent frame at temporal rotary index 0, and generated latent j sits at index $j{+}1$. The same image also enters as a CLIP image embedding (Radford et al., 2021) read through cross-attention, together with the text prompt. The control video is produced from measured follower joint and gripper states. The robot is rendered in a simulator at the camera timestamps, and every pixel that does not belong to the robot is set to gray. It is VAE-encoded and concatenated with the noisy latent along the channel dimension before patchification. Sampling uses 50 updates of the model's flow-matching scheduler with classifier-free guidance, and both guidance branches are edited wherever an intervention applies. The clean estimate at update k is the scheduler's prediction of the final latent from the current state and velocity, decoded with the same VAE.

The 14B fine-tune. A two-expert Wan model (40 blocks per expert, a high-noise and a low-noise expert with their own adapters) fine-tuned on the same robot data, used at step 13,117 with the same kind of control and first image.

DreamDojo. A 2B robot world model (Gao et al., 2026) that generates chunks of 25 frames conditioned on a first frame and on actions. We use two configurations, one driven by latent actions extracted from the recorded video, which carry information about the future, and one driven by planned arm commands.

Public models. Wan2.2-TI2V-5B (Wan Team, 2025) generates 121 frames from 31 latent frames and LTX-Video 0.9.6 (HaCohen et al., 2024) generates up to 121 frames with a different VAE. Both put the first frame into the same token sequence as the video and receive only a text prompt. Cosmos-Predict2-2B (NVIDIA et al., 2025), in its action-conditioned Bridge configuration (Walke et al., 2023), predicts 12 frames from one image and 12 actions, and Ctrl-World (Guo et al., 2026) generates DROID (Khazatsky et al., 2024) rollouts in chunks of five frames and re-feeds its latest frame as the condition for the next chunk. All were run at their documented settings.

C INTERVENTIONS AND THEIR CONTROLS

Routing restoration. A shadow pass under the donor control computes queries and keys after normalization and rotary encoding at every block. These replace the recipient's queries and keys, while values, feed-forward layers, cross-attention and normalization are computed from the recipient's own activations. In complete generations the shadow pass runs at the recipient's current state at every update and in both guidance branches, and only the recipient advances the sampler. Graded versions replace Q, K by $Q_{\text{live}} + \alpha(Q_{\text{donor}} - Q_{\text{live}})$.

State edits. A handover takes the whole state z_k after update k from one run and continues it with another checkpoint, keeping noise, first image, control and scheduler fixed. A regional edit replaces the latent tokens of a spatial box in chosen latent frames once, after a chosen update, with the tokens of another sample at the same update, with an encoded image noised to that update's level, with the model's own clean estimate mixed with new noise, $(1 - \sigma_k)\hat{x}_0 + \sigma_k\epsilon$, or with new noise alone. The sampler keeps no memory of earlier predictions, so nothing else carries the other sample's state.

Attention edits. To scale the attention from a set of queries to a set of keys by a gain g, we multiply the corresponding unnormalized weights by g and renormalize each query's row, exactly, at every block and update. The size of the change in attention output is logged. Value exchanges replace the value vectors at chosen tokens with those of another token or another run. Direction-only exchanges keep each vector's norm from the recipient.

Position edits. The temporal rotary index of the first image's queries and keys is changed at every block, update and guidance branch, with spatial indices and values unchanged. A second clean image is added as a further latent frame with its own temporal index, and visibility masks restrict which generated latents may attend to it.

Input edits. Objects are segmented in the recorded video with SAM 2.1 (Ravi et al., 2025) from prompts on the first frame, and their pixels are drawn into the rendered control under the masks. The stationary-object control paints a different object that is not carried. The gray-mask control fills the carried object's mask with flat gray. Schedules supply the edited control only during chosen updates. The post-lift cutoff is applied either in pixels or exactly at a latent-frame boundary of the encoded control.

Native replay. Every experiment first reproduces the unedited output exactly, with the intervention code active but supplying the recipient's own signals, before any edit is applied.

D EVALUATION

Outcome labels. Each output is labeled from every frame in all three cameras as carried with its start cleared, left at its start, duplicated (placed and still at its start), vanished, wrong object (another object taken or drawn instead) or unreadable, with notes on collateral changes to the arm and to other objects. Labels are made before any score is computed. The first estimates in Table 3 were labeled without seeing the final videos, and the outputs for the 16 new failures (Figure 13) were shuffled and labeled without knowing which input produced them. The cases for each experiment were chosen by a rule written before any edit was run. The routing predictions were tested on eight episodes set aside beforehand, and the checkpoint handover on eight held-out episodes, of which six had usable inputs and five a visible difference between checkpoints.

D.1 STANDARD METRICS PREFER THE FAILURE

For each of 540 saved generations, 300 from our model and 240 from DreamDojo, we built a clip that repeats its first frame and scored both against the recording. The repeated-frame clip has lower temporal-difference error in 529 of 540 cases, higher SSIM in 217 of 300 and 76 of 240, and lower LPIPS in 97 and 34. Plain pixel error is the least affected. Figure 9 shows the cup-stack case, in which a sample that removes the stack has 24% lower pixel error over the object region than the sample that keeps it, while passing every arm check.



Figure 9: **Metrics reward the static outcome.** Top, each dot compares a generation with a clip that repeats its first frame, both scored against the recording, for four metrics in our model and DreamDojo. Bottom, the object region at frame 20 of one clip, where the sample that removes the cup stack has the lower pooled pixel error.

Continuous measures. Prediction error is the mean squared error of the predicted update on future latents at a fixed noisy input, at high, middle and low noise. The crossover uses proximity R of the cup region to the newer native endpoint and its projection F along the native difference. The rewind uses the whole-image mean absolute difference to the first image per frame. The DreamDojo plate uses the fraction of orange pixels over the stack after release, and the local reset the orange share in the source box over frames 60–80.

E ROUTING

Blocks and heads that carry the control. The change in attention under a corrupted control is spread over most heads of blocks 12–26. In a one-block screen (Figure 10a), restoring a block's queries alone improves all twelve comparisons at blocks 10, 15 and 22–27. Before the held-out test we also made two narrower predictions, one for queries and keys together at block 24 and one for queries alone at head 8 of block 26. Both held in all eight episodes, both checkpoints and both corruptions. A third, at head 7 of block 24, failed in two of four groups. Values carry part of the control too. At blocks 24 and 26, restoring values alone beats restoring queries and keys in all twelve high- and middle-noise comparisons of the screen. Queries and keys and values therefore each carry enough of the control to move the arm.

Partial restoration and denoising stage. On three development episodes, restoring a fraction α of the donor queries and keys removes a share of the added error that rises from zero to about 60% as α goes from 0 to 1 at update 26. At the last update full restoration makes the error worse (Figure 24a). On the held-out episodes, full restoration lowers the added error at the first and middle updates in every case and raises it at the last, in 32 of 32 comparisons in the 1.3B model and in 96 and 92 of 96 in the 14B model (Figure 11). The low-noise errors are 100 to 1000 times smaller, so the late harm is small in absolute terms.

Routing restoration in DreamDojo. Driven by commands that hold its arm still, DreamDojo leaves its hand at a box. With commands that move the arm, the hand withdraws. Giving the held run the moving run's queries and keys at all 28 blocks and 35 updates makes its hand withdraw. In two further scenes the moving run's queries did not bring the movement back.

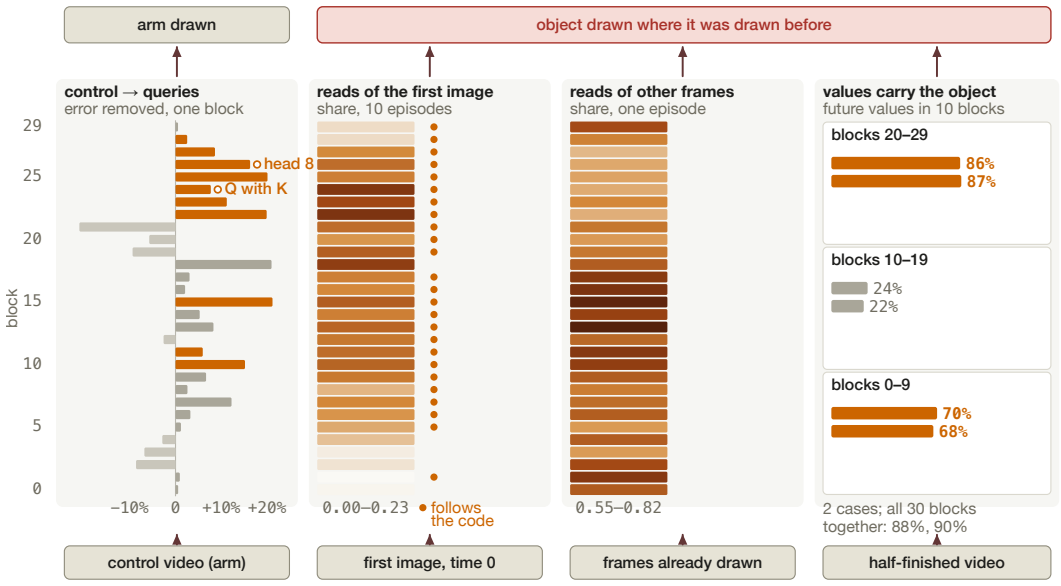

Figure 10: **Block anatomy.** (a) Share of the added error removed by restoring one block's queries in twelve comparisons. Orange blocks improve in all twelve, and the heads confirmed on held-out episodes are marked. (b) Share of attention on the first image by block, in native runs of ten held-out episodes at update 26. Dots mark blocks whose peak follows the temporal index. (c) Share of attention on other generated frames by block at the red bowl's start, update 1. (d) Reduction in object-latent error in two cases when values from a run that saw the recorded future are copied into three block ranges at update 26. Panel (d) uses privileged future information and is a diagnostic.

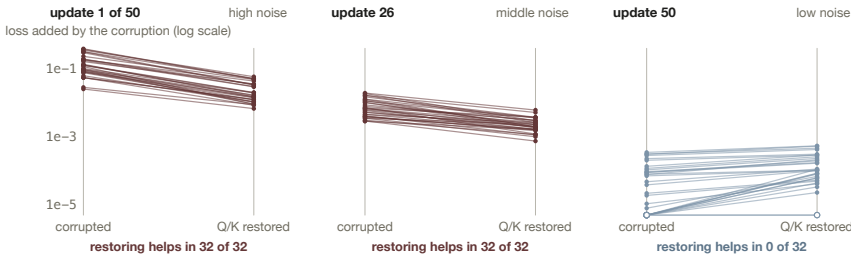

Figure 11: **Routing restoration helps early and hurts late.** Loss added by the corruption before and after restoring queries and keys at all 30 blocks, one line per episode, checkpoint and corruption, at three updates.

F OBJECT IN THE CONTROL

The painted object in the first eight failures. Figure 12 shows the red-bowl case under each control. The three original failures were the red bowl left at its start, the black bowl placed with a copy at its source, and the orange cup lost from the left wrist camera at the grasp (a 49-frame clip). All three were fixed with the carried object in the control, and none with the stationary object. We chose five more left-behind generations from an earlier set of first estimates (Appendix I) before painting any of them. A yellow bowl and a gray bowl were carried with their starts cleared in the overhead view, although the moving wrist cameras stayed inconsistent. A third cleared its start, but

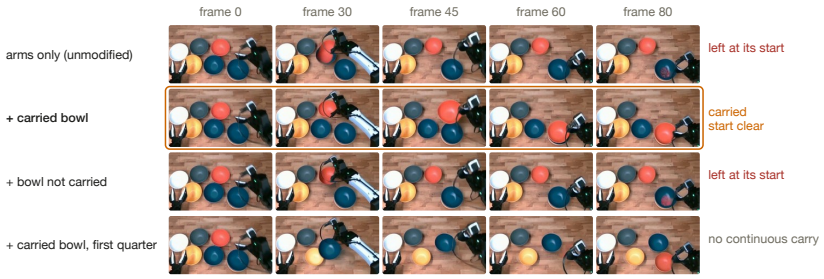

Figure 12: **Only the carried object, shown at every frame, gets the bowl to the stack.** Red-bowl episode, overhead crops. Each row changes only what the control video shows.

the placed bowl was hard to see. Two samples of one cup-stacking scene were not fixed, because the hand carried a different cup with or without the painted one. In the red-bowl episode the wrist cameras show a second, partial bowl behind the carried one for about ten frames.

Partial and gray-filled objects. With the object shown only up to frame 20 (frame 12 in the cup clip), before the lift, the red bowl vanishes from its start and reappears at the stack much later without a continuous carry, the cup still vanishes, and only the black bowl is carried. Filling the object's mask with flat gray removes the cup's dropout and moves the red bowl, but leaves the black bowl's copy. With the object shown until just after the lift, the cup and the black bowl are carried on, also when the cut is placed exactly at a latent-frame boundary of the encoded control. The red bowl turns gray at the frames after the cutoff, and keeping only its outline, filled with gray, keeps it much longer.

Routing restoration with the object painted in. In all three cases, freezing the control stops everything, and restoring the moving control's queries and keys at all blocks brings the arm back. With the object in the control it also brings back readable object behavior, and with the arm-only control it leaves major object errors (Figure 4a).

The painted object in 16 new failures. To test the repair beyond the cases where we found it, we took the 44 new episodes of Appendix I. The 16 whose native video left the object at its start, duplicated it or lost it received the ten inputs of Figure 13, and each output was labeled without knowing which input produced it. With the carried object in the control, 6 of 16 were carried with the start cleared, against none with a stationary object painted the same way, and all six failures where the two inputs differ favor the carried object. Painted pixels of either object also pulled the gripper toward other objects, which were taken in 4 and 6 cases against 3 in the exact replay. With the object only in updates 1–5, or only from update 11, 2 of 16 were carried in each case. Under a control frozen at frame 0, restoring the queries and keys of a moving pass with the same control brought the arm back in 15 of 16 for both controls, and the object came along in 4 of 16 with the object in the control against 1 without it. Restoring only the values of the moving arm-only pass also brought the arm back in 15 and carried nothing. The native video and its exact replay are identical to the pixel, yet their object labels agree in 11 of 16. None of the disagreements involves a carry, so the carry counts are more reliable than the counts of failure types. SAM 2.1 tracks of the object give the same picture. The object is gone from its start at the end in 13 of 16 with the carried object painted in, against 2 of 16 for the stationary object, and in 5 and 7 for the early and late windows. The tracks agree with the labels on whether the object stayed at its start (Cohen's $\kappa = 0.91$), but a single track cannot follow a second copy, so start clearance alone overstates the repairs. The painted pixels come from the recording, so a deployed model would not have them.

The painted object in the 14B model. In the one 14B case with a copy at the source, painting the carried object into the control, a stationary object, or the carried object shown only through the lift, all leave the copy. Painting the carried bowl blue in the control leaves the generated bowl orange, and an orange patch on empty table produces nothing there. Restoring queries and keys under a frozen control brings back the arm's approach, lift and release with either control, and the copy stays.

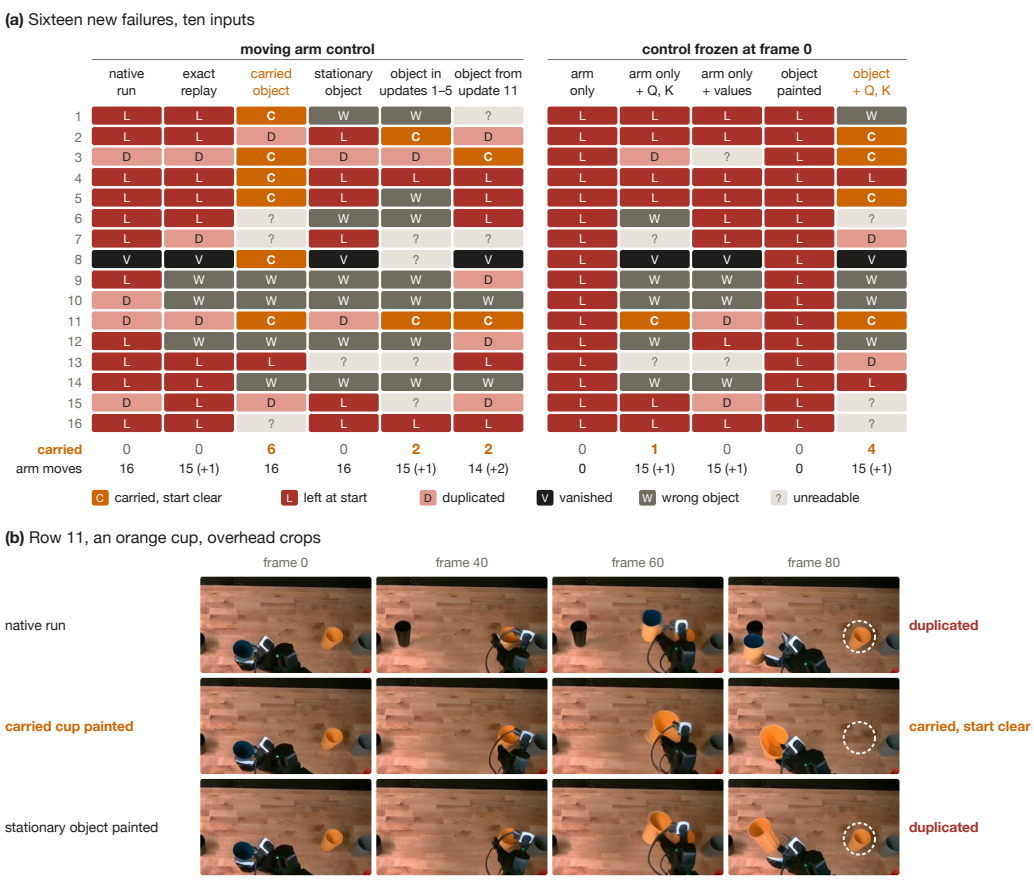

Figure 13: **Painting the carried object into the control clears the start in 6 of 16 new failures, and a stationary object clears none.** (a) Outcome of each failure (row) under each input (column). Left, the moving arm control with nothing painted, an exact replay, or the carried or a stationary object painted in, the carried object also during only part of denoising. Right, a control frozen at frame 0, alone or with the queries and keys or the values of a moving pass restored at every block. The bottom rows count carries and videos in which the arm moves, with partial motion in parentheses. (b) Row 11. The native run and the stationary object leave a copy of the cup at its start (ringed), and the painted cup clears it.

G TIME AS AN ADDRESS

Predicted return frames in the original episode. With the first image moved to time 11, the frame closest to the starting scene is frame 39 (difference 0.021 against 0.098 unmodified). For times 6 and 16 we predicted the windows, frames 17–20 and 57–60, before running, and the closest frames fell inside both. The return is easy to see at times 11 and 16. At time 6 the difference dips at the right frame while the gripper's jaws are doubled (Figure 14).

Return frames in ten held-out episodes. We set aside sixteen cup- and plate-stacking episodes. Six were dropped before any edit because their unmodified videos failed basic checks. In the remaining ten, the closest frame falls at 18–19 for time 6 and at 57–59 for time 16, and the same shift applied to 1,800 background positions produces no such minimum (Figure 15). Six of the ten also pass a stricter automatic check that the returned scene is close to the first image in every camera and that the background control is unambiguous.

Why attention peaks at zero time offset. With rotary encoding, a query at time τ and a key at τ' have score

$$s(\tau, \tau') \approx b + \sum_{\omega} \left[A_\omega \cos \omega(\tau - \tau') + B_\omega \sin \omega(\tau - \tau') \right], \tag{4}$$

where b, A_ω and B_ω depend on the query and key contents. If the contents that dominate the first image's share have $A_\omega > 0$ and small B_ω, every term peaks at $\tau = \tau'$, so the frame that shares the image's time reads it most, and moving the image moves the peak. Early in denoising the generated frames hold little and the clean image supplies much of what each place draws. Later, frames read mostly themselves and their neighbors (Figure 10c), so each place keeps drawing what it drew before. The sketch holds contents fixed, so it explains the tendency but does not predict the measured values.

Signals the return depends on. Holding the first image's own internal states on their unedited trajectory while its position is moved removes the return, and holding as many background states does not. Placing the moved run's first-image states into a run whose image sits at time 0 produces no return. The return needs both the moved position and the image's own states.

The return in other models. In Wan2.2, moving the first image to time 9 or 24 brings the starting arrangement and arm pose back around frames 33–36 and 92–96, where the predicted spans are 33–36 and 93–96. Moving only a small object or background patch gives no rewind. In LTX-Video, times 5 and 12 give their closest frames at the predicted frames 33 and 89, and at time 12 two bowls that had vanished return at frame 89 (Figure 16). In our 14B model, time 16 brings back a start-like layout around frames 54–66 while damaging the arm, and time 6 gives no localized return. DreamDojo and Cosmos-Predict2 show no rewind.

Future frames in the eight failures. Figure 17 shows the original cases in the 1.3B and 14B models. On the eight object-in-control failures, a recorded frame just after release, placed at its own time, repaired 1 case. A recorded mid-carry frame repaired 2 of the 4 cases in which our gripper-aperture rule found a mid-carry frame. A keyframe synthesized from the first image and the planned control, with the object pasted at the planned gripper position and the start inpainted, repaired none, with or without an arm mask that preserves the object. The copied frame therefore has to show the object connected to the hand, and a two-dimensional paste does not yet provide that.

Future-frame controls and recolored images. Placing the future frame at time 0 alongside the first image creates conflicting forms early and is not a clean control. Showing the future frame to every generated frame except 37–40 also keeps the lift and the carry. The recolored first image at time 11 turns the bowl blue by frame 28. Exchanging the two images makes the bowl start blue and turn red at frame 20. The blue image at time 3 gives a blue bowl for frames 2–19, and the first image again at time 11 changes nothing. None of these carries the bowl.

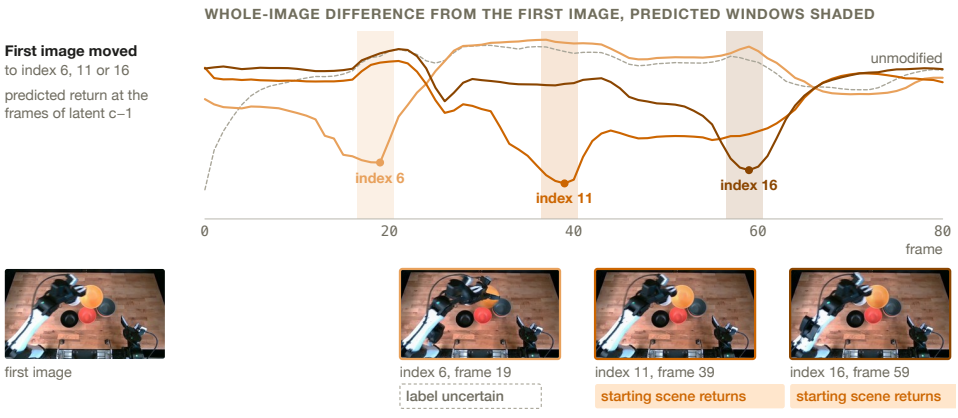

Figure 14: **Three predicted positions in one episode.** Whole-image difference from the first image with the image at time 6, 11 or 16. Shaded bands are the frames predicted before the runs, with the closest frame of each below.

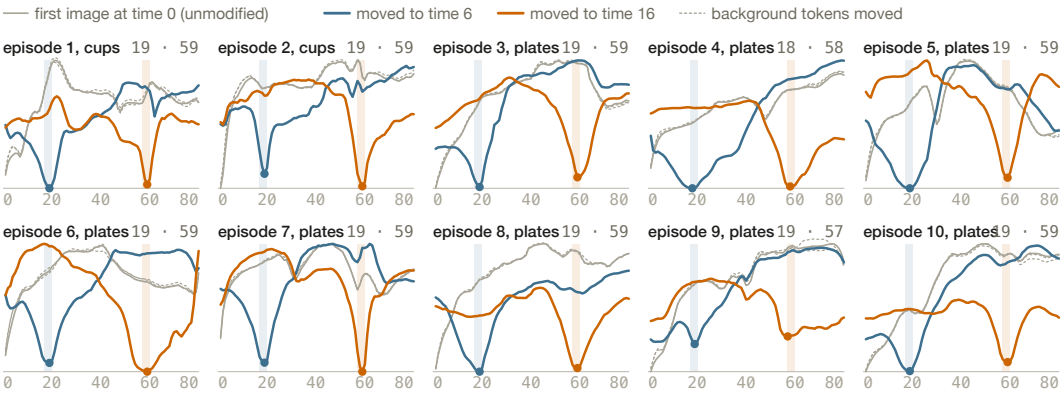

Figure 15: **The rewind in ten held-out episodes.** Blue shows the first image at time 6 and orange at time 16. Gray is unmodified, and dotted curves show background tokens shifted by the same amount. Shaded bands mark where the latents at times 6 and 16 decode, and the numbers are the frames of the two minima.

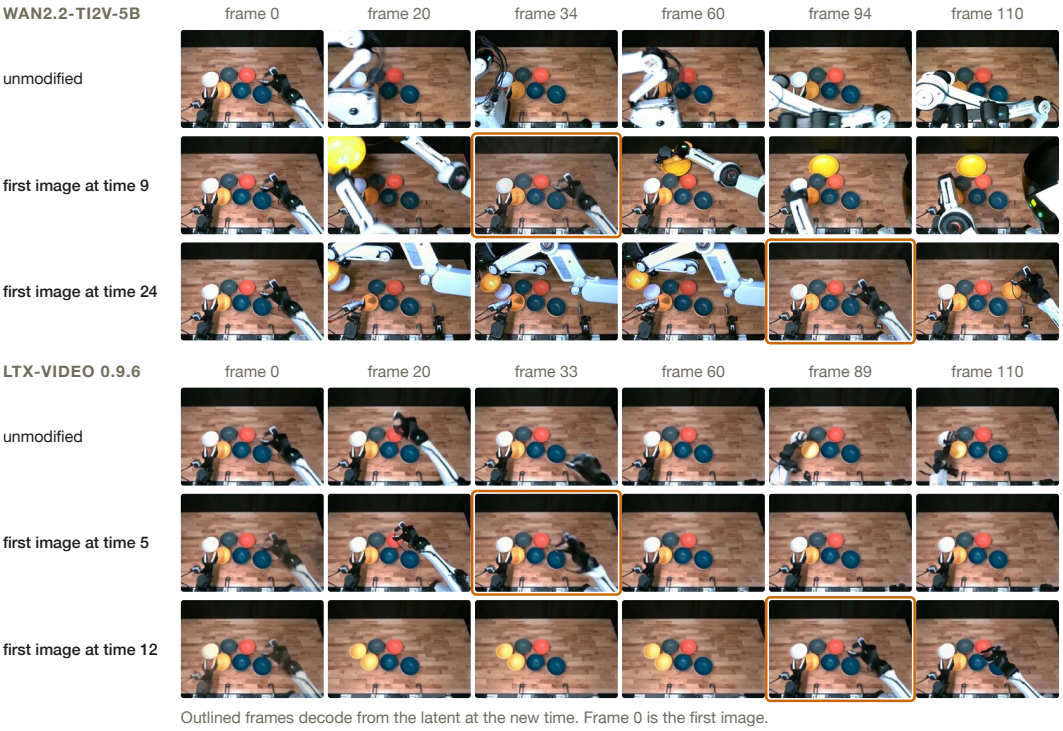

Figure 16: **The rewind in Wan2.2 and LTX-Video.** Frames from the shifted runs. At the outlined frames the table, bowls and arm return toward the first image.

H MESSAGES FROM THE FIRST GENERATED FRAME

In one episode the left wrist camera loses an orange cup for frames 19–23 as the gripper closes, and replacing only the first encoded segment of the control with the current frame held still keeps it. The difference is already present in the generated latents, and it needs the later moving control. For one generated latent at a time, at every block and update, we exchanged only the keys and values it offers to later frames with those the other run would produce from the same state. The first generated latent is the one that matters. Given the improved run's first-frame keys and values, the unmodified run keeps the cup, and the reverse swap makes the improved run lose it. Exchanging the first image's

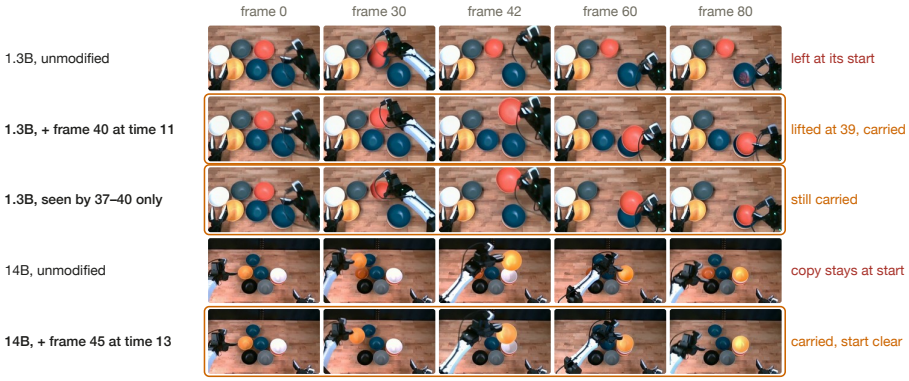

Figure 17: **A future frame at its own time is redrawn there, and the object is then carried with the hand.** The top three rows, from the 1.3B model, show the unmodified run, the run with the recorded frame 40 as a second image at time 11, and the run with that image visible only to frames 37–40. The bottom rows, from the 14B model, show that the recorded frame 45 at its own time removes the copy left at the start.

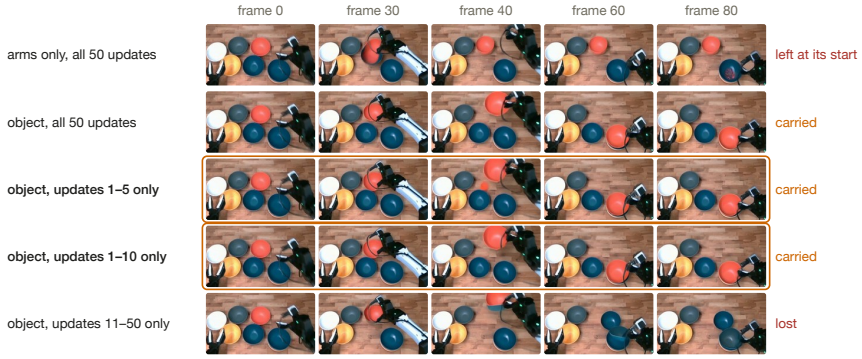

Figure 18: **In the red-bowl episode, object information in the first five updates is enough for the carry.** Overhead frames. Only which control the model receives at each of the 50 updates changes. Outlined rows receive the object in the control during the first 5 or 10 updates only.

Table 2: **Two predictions of the account across six models.** ✓ means the prediction holds. ∼ means it holds partly or a control misbehaves. ✗ means it does not hold. – means it cannot be tested, since the model never carries the object.

Model	Rewind at the predicted frames	Future frame redrawn and carried
Wan2.1 1.3B robot fine-tune (ours)	✓ 10 of 10 episodes	✓
Wan 14B two-expert fine-tune (ours)	∼ start-like layout, arm damaged	✓ removes the copy
DreamDojo 2B	✗	✓ keeps the plate
Cosmos-Predict2 2B	✗	∼ also carries at the wrong time
Wan2.2-TI2V 5B (public)	✓	–
LTX-Video 2B (public)	✓	–

keys and values does neither. Values alone transfer the effect and keys alone do not, and borrowing only the values' direction, with each vector's norm kept, transfers it both ways while borrowing only the norm does nothing. With the value direction swapped, holding the later frames' queries at their unedited values removes the effect in both directions, holding their keys does not, and replaying only the changed queries into an untouched run transfers it (Figure 19). On two further samples of the scene, the swap does no harm where the cup is already kept and does not fix a sample that carries the wrong cup, both as predicted. A swap at the second generated latent also helps in one direction there. On a new case, chosen by a rule set before the test, in which a white plate turns orange, the

repair direction replicates. The first-frame values or their direction keep the plate white, holding later queries blocks this, and replaying the queries suffices.

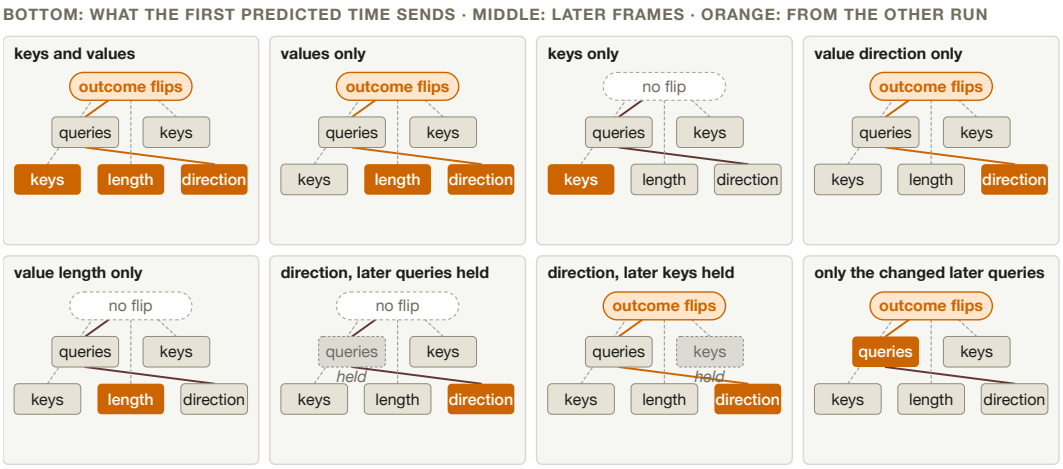

Figure 19: **Eight interventions on the cup case.** Each panel is one intervention at every block and update. It shows which part of the first generated frame's message comes from the other run (orange), which later-frame signal is held at its unedited value, and whether the run takes on the other run's outcome.

The same path in ordinary carries. We also asked whether the same path sustains ordinary transport. On the two new episodes with clean native carries that passed our region checks, replacing the object-region value directions of the first generated latent with those of a frozen-control pass, or replaying the later queries that this replacement produced, left both carries intact. On these cases the path is not necessary for transport. We could not test more cases because clean native carries are rare (3 of 104 episodes we screened).

I Early placement

Handing the state between checkpoints. Figure 20 shows the original crossover. Rows fix the checkpoint that produced the state at update 25, columns the one that finished it, and the diagonal cells replay the native outputs exactly. In the held-out episodes, all five cases in which the two checkpoints differ followed the state in both directions, within the bounds on R and F that we set beforehand. In one direction the arm region's error against the recording rose by 8.5%. In a separate episode, exchanging the states of a carrying run and a still run at update 25, and finishing each under the other's control, moves the whole event with the state. The carrying run still carries under the still control. Replaying the other run's remaining updates as well reverses this, so both the state and the later updates can decide the outcome.

Values in late updates. Copying self-attention values at the future left-camera rows in blocks 20–29, at every update from 26 on, from another sample moves the cup region 89% and 87% of the way to the donor, while shuffled and relocated edits of matched norm transfer nothing. The arm region also changes.

First estimates of 44 new episodes. Figure 18 shows the update schedule on the red-bowl episode. For 44 episodes not used in any earlier experiment, we decoded the update-1 estimate and labeled it from all frames and views before viewing the final video. The first estimate showed the object only at its start in 14 of 16 failures (10 left at start, 4 duplicated) and in 0 of 7 successful carries, and at the destination or in the gripper in 4 of the 7 carries. Among the other 21 episodes, 11 carried a wrong object and 10 had outcomes too unclear to count.

Table 3: **The first estimate predicts whether the object is left behind.** Update-1 estimates of 44 new episodes, labeled before viewing the final videos, against the final outcome.

First estimate shows the object	Carried	Left	Duplicated	Vanished	Wrong object	Unclear
only at its start	0	10	4	0	7	4
at the destination or in the gripper	4	0	0	1	0	2
in both places	3	0	0	0	0	1
unclear	0	1	0	0	4	3

Choosing among samples by the early estimate. Searching over initial noise improves image diffusion samples (Ma et al., 2025), so we asked whether the early estimate can pick a good sample. On 24 held-out scenes with 8 samples each, only 11 of 192 samples were clean carries, and 16 scenes had none, so picking the best of eight could succeed in at most 8 scenes. A score of the update-5 estimate, based on whether the source is clear and the object is at the destination, did no better than random choice (a gain of 8 points in scenes where the samples differ, 95% interval -19 to $+41$). Samples of one scene share the same inputs and mostly fail together, as expected if the inputs set the copy.

An earlier set of first estimates. An earlier set used twenty generations from about a dozen scenes, several of them different samples of one scene. Their final outcomes had been labeled in earlier experiments, and their update-1 estimates were labeled without seeing those outcomes.

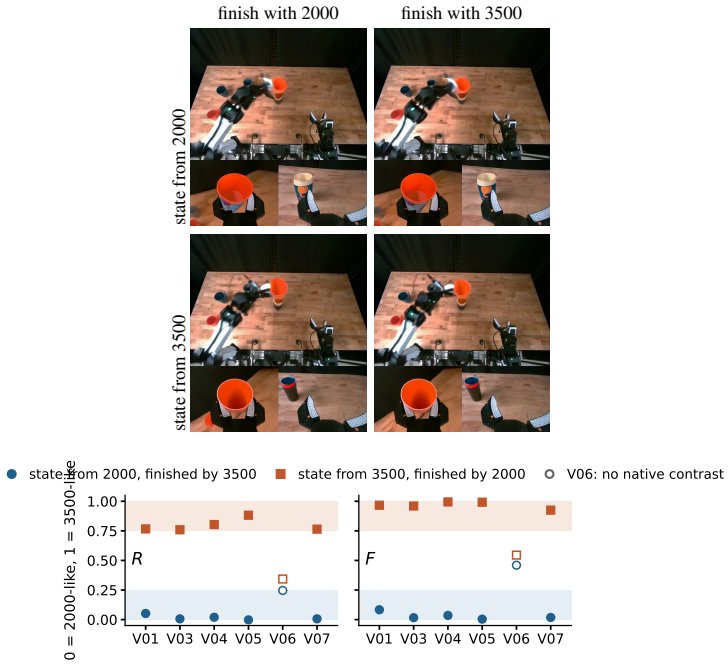

Figure 20: **Appearance follows the state at update 25.** Top, final frames of the original crossover. Rows give the checkpoint that produced the state, columns the one that finished it, and the diagonal reproduces the native outputs. Bottom, the six usable held-out cases, with proximity R to the newer native endpoint and projection F along the native difference, and bands at the bounds set beforehand.

J WHERE A COPY IS HELD

Read maps and the rotary account. In the red-bowl episode, future frames read the bowl's first-image tokens at its starting position from the first update, mostly in blocks 18–26, with up to a third of attention at update 1 and a few percent by the last (Figure 21). In our 14B model, the same

reads sit at the start only for generated latents 1–8, and by latents 15–18 they are strongest at the destination. Shrinking the spatial rotary angles over the first five updates, the correction of Li et al. (2026), left the object at its start in two cases.

Cutting reads of the first image. In the 1.3B black-bowl case, every gain condition (reads at the start or at a background patch, of the bowl or of background, at half strength, and all first-image reads at half strength) keeps both bowls. On an episode where the carry works, removing only the action-dependent part of what future frames read from the bowl's starting tokens makes the hand leave without the bowl, so these reads matter for whether an object leaves with the hand, although they do not hold a copy in place (Figure 22).

Replacing the copy's region with a clean sample. The 14B patch from a clean sample covers the start box from the lift to the end. After update 40 it clears the overhead view while one wrist camera keeps the copy. The inpainted empty-table patch re-noised to update 10 removes the copy with arm, carry, placement and retreat intact. At update 25 it also clears the copy but smears the forearm.

Re-noising the copy's region at 1.3B. We applied the 14B reset to eleven 1.3B cases that leave a copy at the start, with the region and timing set by a rule from the first image and the control. Clean outcomes were 1 of 11 natively, 2 with the model's own estimate re-noised at update 10, 3 with new noise alone at update 10 (which also damaged the scene in five cases), and 2 each for the same reset over empty table and at update 25. In this model the reset depends on the case and is not specific to the copy's region. We also drew new noise for the overhead copy region at update 10 eight times per case. At least one draw gave a clean outcome in 3 of the 11 cases, while eight new samples of the whole video did so in 6, and picking a draw by its update-15 estimate was no better than the native run. Many draws redrew the copy, which often stayed visible in the wrist views that this edit leaves unchanged.

Stale destination reads in DreamDojo. Figure 23 gives the dose curve. We had also predicted that the failing forecast reads its conditioning frame more than the restart, and the opposite holds (13.3% against 19.1%). The restart's conditioning frame shows the plate in the gripper, so what matters is what the stale picture says at the destination. On a second interaction in the scene, a bin being set down, the same edit left bin-like shapes in two compartments where the recording has one bin.

K Other models

Table 2 summarizes the two transferable predictions. The native rollouts differ in the failure itself. Ctrl-World kept an apparently correct carry in three of eight released pick-and-place episodes, and in none of them did it leave the object behind or leave a copy. Its most common error is to place an object even when the recorded grasp fails. Fixing its condition to the first frame for every chunk keeps objects near their starts but also stops the arm. Cosmos-Predict2 carried the object in all eight of its three-second rollouts. Wan2.2 and LTX-Video, prompted to put one bowl on another from our first frame, did not complete the task in eight tries each. Most often the arm moved and the bowl stayed.

L Extended related work

Video models for robot planning. Action-conditioned video prediction lets robots plan by imagining how their actions affect objects (Finn et al., 2016; Finn & Levine, 2017; Du et al., 2023; Yang et al., 2024). Current robot world models condition video diffusion on rendered motion, latent or commanded actions, or recent frames (Alzayer et al., 2026; Gao et al., 2026; NVIDIA et al., 2025; Guo et al., 2026; Chen et al., 2024).

Physical fidelity and evaluation. Video generators trained on simple physics generalize by retrieving similar training cases and weight color above size, velocity and shape when they do (Kang et al., 2025). Physics-IQ and VideoJAM show that realism and pixel reconstruction miss physical and motion errors (Motamed et al., 2026; Chefer et al., 2025). Li et al. (2026) attribute physics

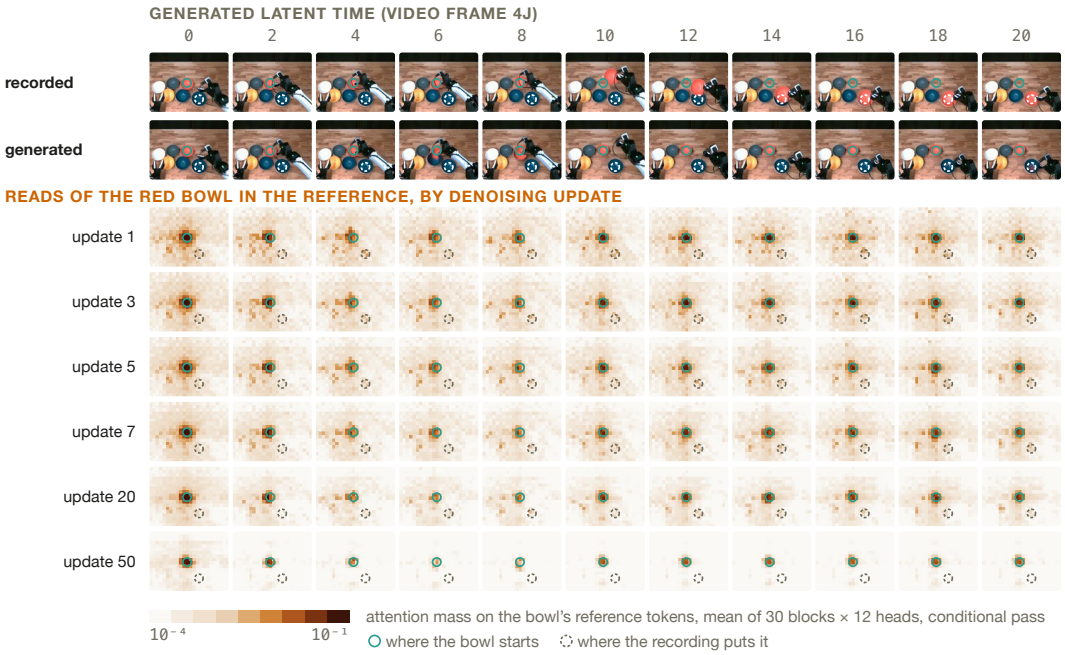

Figure 21: **Future frames read the red bowl only where it started.** Attention from each future position to the bowl's first-image tokens at six of the 50 updates.

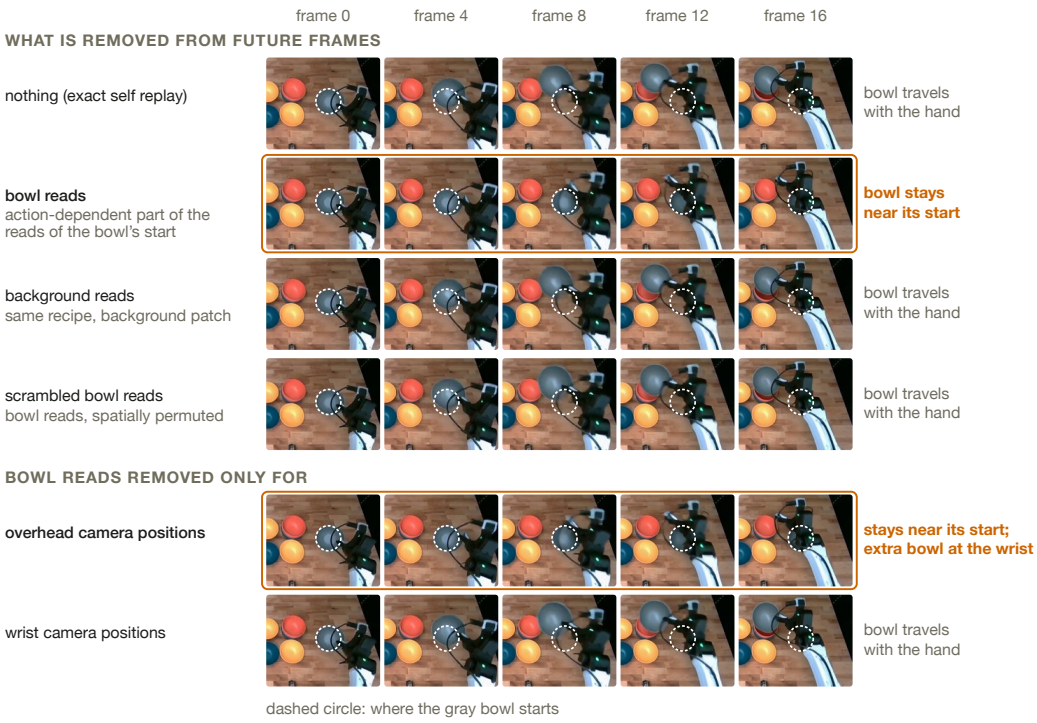

Figure 22: **Removing the action-dependent reads of the bowl's start stops the carry.** Overhead crops around the hand, frames 0–16. Outlined rows stop the carry.

violations in Wan to rotary encoding that pins attention to the same image position, and we test that account on object transport in Section 8.

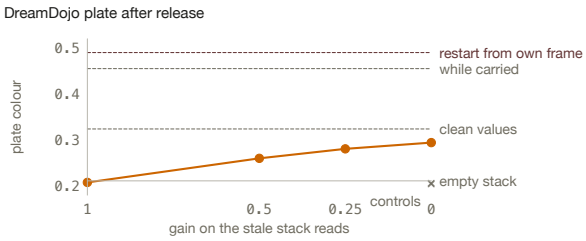

Figure 23: **The less the stack copies its stale picture, the more plate survives.** Fraction of orange plate pixels over the stack in DreamDojo at frames 21–24, against the gain on the stack tokens' attention to the conditioning frame's stack tokens. Squares are the two controls at gain 0, and dashed lines mark the restart, the clean-value transplant and the plate before release.

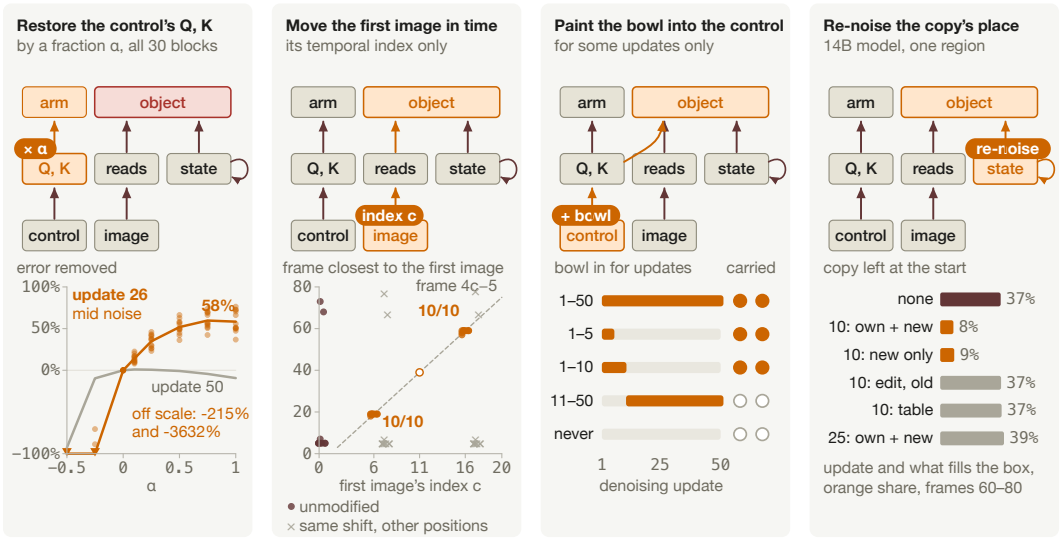

Figure 24: **Four interventions and their measured responses.** (a) Share of the corruption penalty removed at update 26 and the last update by restoring a fraction α of the true control's queries and keys. (b) The first image's temporal index against the frame closest to it, for ten held-out episodes and the original case. Crosses are the background control, and the dashed line is the prediction. (c) Object in the control during chosen update windows, two cases. (d) Orange share at the start after a local reset of the 14B copy's region.

Interpreting transformers and diffusion models. Activation patching tests which internal signals an output depends on (Meng et al., 2022b; Wang et al., 2023; Heimersheim & Nanda, 2024; Zhang & Nanda, 2024), and we separate attention into routing and content (Elhage et al., 2021). Related work traces features in diffusion transformers (Mazur et al., 2026), finds temporal correspondence in video diffusion attention (Nam et al., 2025), and edits or propagates content through attention and diffusion features (Hertz et al., 2023; Tumanyan et al., 2023; Cao et al., 2023; Alaluf et al., 2024; Epstein et al., 2023; Geyer et al., 2024; Li et al., 2024; Yatim et al., 2024; Xiao et al., 2024; Pondaven et al., 2025; Jun et al., 2026).

Position, time and conditioning. Rotary encoding makes attention depend on relative position (Su et al., 2024), and rescaling it changes how far attention reaches (Chen et al., 2023; Zhao et al., 2025). Some video models are trained to accept frames at chosen positions (Wan Team, 2025; HaCohen et al., 2024), and we find that models trained only with a first image already treat its temporal index as an address. Trajectory-conditioned generators supply object motion explicitly (Geng et al., 2025; Zhang et al., 2025). Diffusion models commit to coarse structure early (Biroli et al., 2024), guidance acts differently across noise levels (Kynkäänniemi et al., 2024), and partial re-noising lets a sampler redraw part of an image (Meng et al., 2022a; Lugmayr et al., 2022). Our work adds a causal account

of the time-addressed copy, of its early appearance in the clean estimate and of its storage in the partially denoised video.