FOLLOW THE ROUTE, KNOW WHERE TO STOP: EFFICIENT ATTENTION FOR VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video diffusion transformers denoise all frames of a clip together, and at every step each token attends to every other frame, at a cost quadratic in the number of frames. Efficient attention methods cut this cost for distant frame pairs and decide which pairs to keep, or how to approximate them, by comparing the two frames directly. We show that the value of a distant connection is set in the frames between: by the route the content takes, how far that route carries and where it breaks. When a sparse method keeps fewer connections, point-correspondence accuracy stays level while a quarter of the annotated point tracks can no longer be followed and motion binding drops; when propagation cannot stop, motion grows larger and temporal coherence falls. We propose PathSpan, which builds every distant connection from adjacent-frame matches: each token scores an 11×11 window of the neighboring frame together with a learned no-match entry that ends propagation, two linear-time scans chain these matches, a per-channel retention sets how far each channel reads, and attention within the frame selects from the carried content. On Wan2.2, LTX-Video and Cosmos-Predict2.5, complete generation is 1.52–$4.03\times$ faster than matched dense fine-tuning with VBench totals within 0.05 points, and removing the route, the per-channel reach or the stop lowers motion binding and temporal coherence in all three generators.

1 INTRODUCTION

Video diffusion models generate a clip by denoising all of its latent frames together (Wan-Video, 2025; Lightricks, 2025; NVIDIA, 2025). At every step, attention lets each token read every other frame, which keeps a generated object recognizable as it moves, turns and changes appearance (Nam et al., 2025). The cost grows quadratically with the number of frames and positions: a five-second 720p clip unfolds into more than 100K tokens (Zhang et al., 2025a), and HunyuanVideo needs almost an hour on one A100 GPU for a five-second video (Xi et al., 2025). Acceleration methods therefore compute less for distant pairs (Zhang et al., 2025a; Yang et al., 2025a; Wu et al., 2026; Hu et al., 2026; Zhang et al., 2026; Liang et al., 2026; Ghafoorian et al., 2026), the very connections that consistency relies on, and each must answer one question: which distant connections can be computed less without losing consistency?

Existing methods give two answers. Sparse selection estimates cheaply which distant pairs matter, from coarse attention (Zhang et al., 2025a), block similarity (Wu et al., 2026), token reordering (Yang et al., 2025a) or block scores computed on the fly (Hu et al., 2026), and compares only those directly. Structured aggregation approximates the distant pairs with a learned linear term (Zhang et al., 2026), a Monarch factorization (Liang et al., 2026) or a distilled hybrid attention (Ghafoorian et al., 2026). Neither explicitly chains adjacent-frame matches to weight distant connections.

Take a boxing glove that crosses the image, turns and passes behind another object. The first and last frames show the glove at a different position, orientation and appearance, so the pair scores low and is dropped or coarsely approximated. Between neighboring frames, however, the glove moves only a little, and random-walk correspondence reaches a distant frame by chaining such easy steps (Jabri et al., 2020). Whether a distant connection is worth keeping is therefore decided by the path the content takes: which route it follows, how far that route carries, and where it breaks. A turn weakens the match and shortens the route; an occlusion removes the match and breaks the route (Figure 1). All three lie between the two ends, so a rule that selects at the two ends cannot know what it removes.

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: **PathSpan generations with the events that decide distant connections.** Horse jumping, kiteboarding, a cow passing behind foreground foliage, and a camel approaching water and turning, each at eight ordered moments. Position, pose and visibility change between the first and last frames; the frames between them hold the small steps that connect the two ends.

Correspondence accuracy (Nam et al., 2025), the natural diagnostic, misses the loss as well. When a sparse method (Hu et al., 2026) keeps 10% of its connections instead of 40%, point accuracy on TAP-Vid-DAVIS stays at 15.3–15.6%, while the share of annotated point tracks that can still be followed along the kept connections falls from 99.8% to 73.3%, and motion binding falls with it (Section 3.3). Carrying information farther fails as well: when we remove the option to stop from our operator, generated motion grows larger in all three generators we test, yet temporal coherence and motion binding fall (Section 6.3), because information keeps being carried after its content has gone.

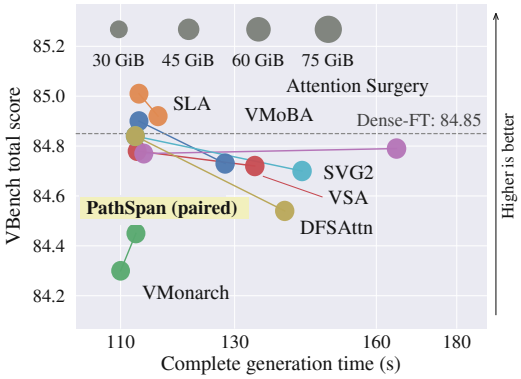


Figure 2: **PathSpan against seven efficient attention methods on Wan2.2-TI2V-5B.** Each segment joins a method to its PathSpan conversion under the same recipe; SVG2 and DFSAttn share default PathSpan. Bubble area is peak memory per GPU; the dashed line is Dense-FT (Table 2).

We therefore propose PathSpan, which hands the route, the reach and the stop to attention itself. Each position compares its token with an 11×11 window of the next frame; the resulting probabilities say where the content has moved, and chaining them frame by frame fetches content along the route it took, as in random-walk correspondence (Jabri et al., 2020) and trajectory attention (Patrick et al., 2021). A no-match entry competes with the real candidates in the same softmax and stops propagation when no match is convincing, a retention coefficient per channel sets how far to read, and attention within the frame picks from the carried content. The native generation loss trains all of them, and the arithmetic grows linearly with the number of frames.

Our contributions are threefold. (i) We identify that the value of a distant connection in a video generator is decided along the route the content takes, which endpoint selection and endpoint accuracy cannot see. (ii) We introduce PathSpan, which chains adjacent-frame probability maps, stops with a learned no-match entry and sets its reach with a per-channel retention. (iii) In three native generators, PathSpan is 1.52–$4.03\times$ faster in complete generation with VBench totals (Huang et al., 2024) within 0.05 points of matched dense fine-tuning, beats seven efficient attention methods in motion binding and temporal coherence (Figure 2), and needs each of route, reach and stop; a perturbation bound explains these effects and predicts the generator's response better than correspondence accuracy.

2 Related Work

Efficient attention in video diffusion transformers. Early video diffusion models factorized attention into spatial and temporal parts (Ho et al., 2022); current generators build on diffusion transformers (Peebles & Xie, 2023) and attend over all latent tokens of a clip (Yang et al., 2025b; Kong et al., 2024; Wan-Video, 2025), and this attention dominates generation time. Acceleration has followed two lines. *Sparse selection* keeps exact scores for a subset of pairs. Some methods constrain sparsity to local 3D tiles (Zhang et al., 2025b), spatial or temporal heads chosen during sampling (Xi et al., 2025) or a density that decays with spatiotemporal distance (Li et al., 2025); other methods choose pairs adaptively, through coarse attention (Zhang et al., 2025a), block similarity (Wu et al., 2026), semantic clustering and permutation (Yang et al., 2025a) or refreshed fine-grained block scores (Hu et al., 2026). *Structured aggregation* covers all pairs at a lower cost per pair, with linear attention (Katharopoulos et al., 2020; Xie et al., 2025), sparse attention plus a learned linear term (Zhang et al., 2026), Monarch factors (Liang et al., 2026) or a hybrid attention distilled from the dense model (Ghafoorian et al., 2026). These methods do not explicitly form distant weights by chaining adjacent-frame correspondence maps. This is the bottleneck: under motion, rotation and occlusion the two ends differ most, while the evidence that connects them lies in the frames between. PathSpan computes each distant weight from the chain of adjacent-frame matches between them.

Following content through time. A second line of work assembles long-range correspondence from short steps. Self-supervised trackers learn to copy colors from a reference frame (Vondrick et al., 2018) or chain adjacent-frame affinities into a random walk whose long-range transitions are products of short ones (Jabri et al., 2020), and trajectory attention pools recognition features along motion paths (Patrick et al., 2021). Generative models hold such matches in their features: image diffusion features match points across images (Tang et al., 2023), video diffusion transformers match points across frames in specific layers, which has been used to guide generated motion (Nam et al., 2025), and trajectory attention adds a branch along supplied trajectories to control motion (Xiao et al., 2025). Gated linear attention (Yang et al., 2024) and selective state spaces (Gu & Dao, 2024) reach linear cost by carrying a decaying state along a sequence and learn how quickly each channel forgets. These works use chained correspondence as a readout or control signal, and carry decaying states along a one-dimensional token order, where the same content in the next frame lies a full frame of tokens away. PathSpan brings the two ideas into the generator's own attention: its state moves along routes found by adjacent-frame matching, each channel learns its own retention as in gated recurrences, a learned no-match entry stops the state where matching fails, and the native generation loss is the only signal that shapes these routes.

3 Distant Connections Are Decided Along the Route

3.1 Cross-Frame Attention and the Endpoint Rule

Let T be the number of latent frames and S the number of spatial tokens per frame. For one attention head of width d, the native projections of frame $t \in \{0, \dots, T-1\}$ are $\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t \in \mathbb{R}^{S\times d}$, whose i-th rows $\boldsymbol{q}_{t,i}^\top, \boldsymbol{k}_{t,i}^\top, \boldsymbol{v}_{t,i}^\top$ belong to position $i \in \{1, \dots, S\}$; queries and keys include the backbone's normalization and positional encoding. Dense attention over all TS tokens, $\mathrm{softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d})\,\mathbf{V}$, costs $O(T^2S^2d)$ per head.

Let $w_{t,i}(u,j)$ be the weight that query (t,i) gives to key (u,j), $u \neq t$. Dense attention sets $w_{t,i}(u,j) \propto \exp(\boldsymbol{q}_{t,i}^\top \boldsymbol{k}_{u,j}/\sqrt{d})$, sparse selection keeps this value when a block score from $\mathbf{Q}_t$ and $\mathbf{K}_u$ is high, and structured aggregation replaces it with a cheaper function of $\boldsymbol{q}_{t,i}$ and a summary of frame u. Up to terms shared by all keys, such as a normalizer or a top-k threshold, all follow one *endpoint rule*,

$$w_{t,i}(u,j) = g\big(\boldsymbol{q}_{t,i},\, \mathbf{K}_u\big), \tag{1}$$

a function of the two frames t and u alone.

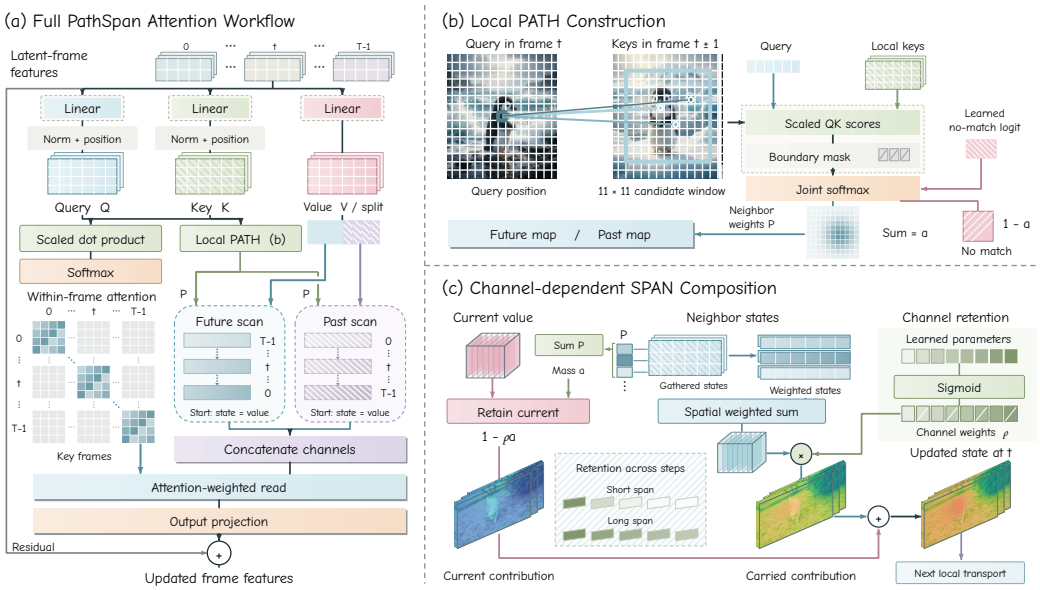


Figure 3: **PathSpan builds distant reads from adjacent-frame steps.** (a) Native queries, keys and values enter a future scan and a past scan; within-frame attention reads the resulting states before the native output projection. (b) Each position scores an 11×11 window of the adjacent frame and a learned no-match logit in one softmax (Equation (3)). (c) A per-channel retention splits each update between the current value and the carried state (Equation (4)); repeated updates chain the steps into routes.

3.2 What Lies Between the Ends

Let $\mathbf{P}_t \in [0,1]^{S\times S}$ give, in row i, the probability of where the content at position i of frame t has moved in frame $t+1$; Section 4 builds it from $\mathbf{Q}_t$ and $\mathbf{K}_{t+1}$ (Figure 3b). Chaining these steps gives

$$\big[\mathbf{P}_t\mathbf{P}_{t+1}\cdots\mathbf{P}_{u-1}\big](i,j) = \sum_{k_{t+1},\ldots,k_{u-1}} P_t(i,k_{t+1})\,P_{t+1}(k_{t+1},k_{t+2})\cdots P_{u-1}(k_{u-1},j), \quad (2)$$

the total probability of all routes from (t,i) to (u,j) through one position k_s in each intermediate frame s. This weight depends on every frame between t and u and carries the three quantities of the glove example: the route is the sequence $k_{t+1},\ldots,k_{u-1}$ that holds the mass; how far it carries is set by how quickly the product shrinks, as when a turn spreads a row over several candidates; and it breaks where a factor vanishes, as when an occlusion leaves no real candidate with mass.

The route holds information that the two ends lack. On all 30 TAP-Vid-DAVIS videos (Doersch et al., 2022), chaining adjacent-frame maps of the densely fine-tuned generator (Dense-FT) as in Equation (2), instead of matching the query directly against the distant frame, raises point accuracy from 15.30% to 16.42% on Wan, from 16.06% to 18.24% on LTX-Video and from 15.19% to 17.14% on Cosmos (Table 4, Appendix A.1).

3.3 An Endpoint Diagnostic Misses What Is Lost

We fix the retained fraction of key blocks of DFSAttn (Hu et al., 2026) at 0.4 or 0.1 on Wan and read the retained connections in two ways (Appendix A.3): *point accuracy* asks whether the retained scores place a TAP-Vid query at its annotated position, and *track reachability* asks whether the retained connections link the cells of an annotated track from its first frame to its last. Lowering the budget from 0.4 to 0.1 raises point accuracy from 15.32% to 15.57%, lowers track reachability from 99.8% to 73.3%, and lowers CompBench (Sun et al., 2025) motion binding of the generated videos from 43.4 to 40.8. The endpoint readout improves while a quarter of the routes disappear, and generation follows the routes. Section 4 therefore builds distant connections from Equation (2).

4 PATHSPAN

PathSpan replaces the self-attention call of every video transformer block and keeps its native projections (Figure 3; Algorithm 1 in Appendix B).

4.1 ROUTE AND STOP: ONE SOFTMAX OVER ADJACENT CANDIDATES

Why adjacent frames. Between neighboring latent frames content moves little, so a small window contains its next position and the match is easy; between distant frames it can move anywhere and change appearance. PathSpan therefore scores only adjacent-frame pairs and chains them as in Equation (2). For position i, let $\Omega(i)$ be the valid positions of a $(2r+1) \times (2r+1)$ window centered at i, with $r = 5$ by default, so $m = |\Omega(i)| \leq 121$. The scores reuse the native projections, $\ell_t^\delta(i,j) = \boldsymbol{q}_{t,i}^\top \boldsymbol{k}_{t+\delta,j}/\sqrt{d}$ for $j \in \Omega(i)$, where $\delta = +1$ reads the next frame and $\delta = -1$ the previous one, so each step starts from the correspondence the pretrained generator already carries (Nam et al., 2025).

Why a no-match entry in the same softmax. A route must break where content is occluded or leaves the window, and breaking is a comparison: continue if some candidate matches well, stop if none does. PathSpan makes this comparison inside the softmax with a learned logit β^δ that competes with the real candidates,

$$P_t^\delta(i,j) = \frac{\exp \ell_t^\delta(i,j)}{\exp \beta^\delta + \sum_{j' \in \Omega(i)} \exp \ell_t^\delta(i,j')}, \qquad j \in \Omega(i), \tag{3}$$

and $P_t^\delta(i,j) = 0$ for $j \notin \Omega(i)$. The continuation mass $a_{t,i}^\delta = \sum_j P_t^\delta(i,j) \in [0,1]$ approaches one when a candidate logit dominates β^δ and zero when every candidate is weak; $1 - a_{t,i}^\delta$ is the probability of stopping. Unlike a separate gate, the shared softmax judges every candidate against one learned threshold β^δ per layer, head and direction.

4.2 REACH: PER-CHANNEL RETENTION AND TWO SCANS

Why per-channel retention. Different content persists for different durations: the identity and layout of an object persist across the clip, while its pose, or the petals of an opening flower, change within a few frames. A single reach per head has to trade one against the other. PathSpan gives each value channel c its own retention $\rho_c^\delta = \mathrm{sigmoid}(r_c^\delta) \in (0,1)$, learned per layer, head and direction; all channels share the routes $\mathbf{P}_t^\delta$ and differ only in how strongly they weight long routes.

Why a scan. Evaluating Equation (2) for every pair of frames would cost $O(T^2)$ again; a state that already summarizes everything beyond frame $t+1$ lets frame t reach all later frames in one step. We split the d value channels of each head into halves $\mathbf{V}_t^+$ and $\mathbf{V}_t^-$ of width $d/2$, which read the future and the past. The future state starts at $\mathbf{H}_{T-1}^+ = \mathbf{V}_{T-1}^+$ and runs backward in time, the past state starts at $\mathbf{H}_0^- = \mathbf{V}_0^-$ and runs forward, and at every other frame

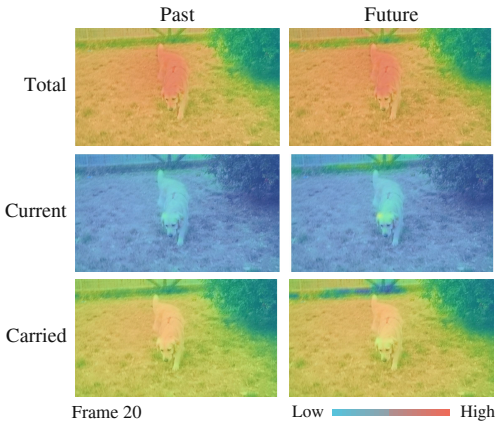


Figure 4: **A state is its current value plus the carried state.** Past and future scans of Equation (4) on DAVIS frame 20 with RGB values: total H, current $(1-\rho a)V$ and carried $\rho P H$, on a shared scale.

$$\begin{aligned} H_{t,i,c}^\delta = &\left(1 - \rho_c^\delta a_{t,i}^\delta\right) V_{t,i,c}^\delta \\ &+ \rho_c^\delta \textstyle\sum_{j \in \Omega(i)} P_t^\delta(i,j)\, H_{t+\delta,j,c}^\delta. \end{aligned} \tag{4}$$

The two weights sum to one, so every state is a convex combination of values, and position i keeps its current value with weight $(1-\rho_c^\delta) + \rho_c^\delta(1 - a_{t,i}^\delta)$: the retention declines to carry more, or the step stopped. Figure 4 shows the two parts on a real clip.

What the scan computes. Unrolling Equation (4) makes the three controls explicit. Fix the future direction and a channel, drop both indices, and write $\boldsymbol{h}_t, \boldsymbol{v}_t \in \mathbb{R}^S$ for the state and value columns. With $\mathbf{D}_u = \mathbf{I} - \rho\,\mathrm{diag}(\boldsymbol{a}_u)$ for $u < T-1$ and $\mathbf{D}_{T-1} = \mathbf{I}$,

$$\boldsymbol{h}_t = \sum_{u=t}^{T-1} \underbrace{\rho^{u-t}}_{\text{reach}} \underbrace{\mathbf{P}_t\mathbf{P}_{t+1}\cdots\mathbf{P}_{u-1}}_{\text{route}} \underbrace{\mathbf{D}_u}_{\text{stop}} \boldsymbol{v}_u, \tag{5}$$

with the empty product equal to $\mathbf{I}$ (Appendix C.1). Frame u reaches frame t through the route weights of Equation (2), discounted by ρ^{u-t} and injected where the route ends: a channel with ρ near one reads far along supported routes, one with small ρ reads nearby frames, and a stop at (s,k), $a_{s,k} \approx 0$, removes every later term that passes through it. When the glove disappears behind the occluder, the step finds no match, the route ends, and earlier frames receive the glove as last seen.

4.3 Select within the Frame, Train, and Cost

Why a within-frame read. The scans deliver to each position the content that arrived there along its routes, and a query often needs content that arrived elsewhere in its frame: a hand needs the state of the ball it holds, and a spatial relation involves two objects. PathSpan concatenates the directional states, $\mathbf{R}_t = [\mathbf{H}_t^+, \mathbf{H}_t^-] \in \mathbb{R}^{S\times d}$, and selects from them with exact softmax attention within the frame,

$$\mathbf{Y}_t = \mathbf{A}_t\mathbf{R}_t, \qquad \mathbf{A}_t = \mathrm{softmax}\big(\mathbf{Q}_t\mathbf{K}_t^\top/\sqrt{d}\big), \tag{6}$$

computed with an exact kernel (Dao, 2024) and passed to the native output projection.

Training. Every operation is differentiable, so each backbone's native denoising or flow-matching loss trains the maps, retentions and no-match logits together. Adaptation starts from the pretrained checkpoint and adds d retention logits and two no-match logits per layer and head; half-lives start log-uniformly between one frame and the clip length, and $\beta = \log 121$ gives the no-match entry half of the mass in a full window at zero logits (Appendix B).

Cost. With n_h heads, within-frame attention costs $O(Tn_hS^2d)$ and adjacent scoring plus the scans cost $O(Tn_hSmd)$, linear in T against $O(T^2n_hS^2d)$ for dense attention. The scans are sequential over frames, and the complete times of Section 6 include this dependency.

5 Which Local Changes Matter: A Response Bound

Section 3.3 showed that generation can change while point accuracy stays level; for PathSpan we can state how a change of the step probabilities reaches the state. Fix a future channel and compare two scans H and $\widetilde{H}$ that share values and temporal boundary and differ in their maps or retentions. At frame s and position k, a scan either stops, with probability $\pi_s(k,\perp) = 1 - \rho a_{s,k}$, taking the candidate value $U_s(k,\perp) = V_{s,k}$, or moves to j, with probability $\pi_s(k,j) = \rho P_s(k,j)$, taking $U_s(k,j) = \widetilde{H}_{s+1,j}$. Let $b_s(k) = \max_z U_s(k,z) - \min_z U_s(k,z)$ be the range of candidate values over stopping and the union of both supports, and let $\gamma_{t,s}(i,k)$ be the probability that the first scan, started at (t,i), reaches (s,k) without stopping, the (i,k) entry of $\rho^{s-t}\mathbf{P}_t\cdots\mathbf{P}_{s-1}$ for $s > t$.

Proposition 1 (Response bound)**.** *With probabilities set to zero outside each scan's support,*

$$\big|H_{t,i} - \widetilde{H}_{t,i}\big| \le \sum_{s=t}^{T-2}\sum_{k=1}^{S} \underbrace{\gamma_{t,s}(i,k)}_{\textit{reachability}} \underbrace{\mathrm{TV}\big(\pi_s(k,\cdot),\widetilde{\pi}_s(k,\cdot)\big)}_{\textit{probability moved}} \underbrace{b_s(k)}_{\textit{value contrast}} =: B_{t,i}, \tag{7}$$

where $\mathrm{TV}(p,q) = \frac{1}{2}\sum_z |p_z - q_z|$ *is the total variation distance.*

Subtracting the two recurrences splits the difference at each step into a local term, the change of the step distribution applied to the candidate values, and the propagated difference of the successors; unrolling gives an exact sum, and bounding each local term by its value range gives Equation (7) (Appendix C.2). A term is large only when all three factors are: a large change that moves probability between nearly equal values, as inside a uniform background, leaves the state unchanged, while a small change on a heavily used route between different contents, as at an object boundary, can change it strongly. Correspondence accuracy has no term for reachability or value contrast, so the bound predicts that the generator's response follows B more closely than accuracy changes (tested in Section 6.4). The past direction and the within-frame read follow in Appendix C.3.

6 EXPERIMENTS

6.1 SETUP

Generators and matched adaptation. We use Wan2.2-TI2V-5B (Wan-Video, 2025), LTX-Video-13B v0.9.8 (Lightricks, 2025) and Cosmos-Predict2.5-2B (NVIDIA, 2025) at native settings (Appendix D.1). PathSpan and the dense baseline, Dense-FT, start from the same checkpoint, share OpenVid-1M samples (Nan et al., 2025), sample order and seeds, and train the self-attention projections and query–key gains for 10,000 updates at batch four; PathSpan also trains its scalars. Numbers average three training roots.

Evaluation. We run the complete prompt sets of VBench (Huang et al., 2024), T2V-CompBench (Sun et al., 2025), EvalCrafter (Liu et al., 2024), VideoPhy-2 (Bansal et al., 2026) and VMBench (Ling et al., 2025) for text-to-video and of AIGCBench (Fan et al., 2023) for image-to-video. Time is complete host-to-host generation on four A100-SXM4-80GB GPUs in BF16 at batch one, including conditioning, guidance, decoding and both LTX passes (Appendix D.4).

6.2 SPEED AND QUALITY

Table 1: Matched adaptation on three video generators and six suites (three-root means). VBench IQ: imaging quality. CompBench MB/DAB: motion/dynamic-attribute binding. EvalCr. MAC: requested-motion accuracy. VideoPhy-2 SA/PC: semantic/physical scores (1–5); Joint: joint success. VMBench TCS/PAS: coherence/amplitude. AIGCBench DOV.: DOVER; I–V/T–V: image–video/text–video CLIP. Other scores use 0–100; s is complete host-to-host time. Bold compares each backbone pair, including ties.

Method	Text-to-video												Image-to-video			
	VBench			CompBench		EvalCr.	VideoPhy-2			VMBench			AIGCBench			
	Total↑	IQ↑	s↓	MB↑	DAB↑	MAC↑	SA↑	PC↑	Joint↑	Total↑	TCS↑	PAS↑	DOV.↑	I–V↑	T–V↑	s↓
Wan2.2-TI2V-5B (5B) (Wan-Video, 2025)																
Dense-FT	**84.85**	**68.51**	205.6	**44.1**	8.53	66.0	**3.61**	**3.20**	**34.4**	70.70	94.17	**37.5**	**63.83**	**90.44**	28.41	199.5
PathSpan	84.84	68.47	**112.4**	43.9	**8.67**	**66.7**	**3.61**	**3.20**	**34.4**	**70.78**	**94.34**	**37.5**	63.74	90.42	**28.42**	**110.7**
LTX-Video-13B v0.9.8 (13B) (Lightricks, 2025)																
Dense-FT	**85.19**	**70.53**	284.9	41.4	**9.44**	**64.7**	**3.68**	**3.17**	**34.2**	70.10	95.14	32.6	**65.75**	**91.02**	**28.85**	289.9
PathSpan	85.14	70.35	**187.3**	**41.6**	9.26	62.7	**3.68**	**3.17**	34.1	**70.18**	**95.32**	**33.1**	65.64	90.98	28.82	**191.4**
Cosmos-Predict2.5-2B (2B) (NVIDIA, 2025)																
Dense-FT	84.42	**67.71**	320.9	46.6	**10.39**	**69.3**	**3.64**	**3.36**	**40.4**	71.02	93.58	**40.2**	62.73	90.92	**28.73**	327.3
PathSpan	**84.46**	67.41	**79.7**	**47.6**	10.25	66.0	**3.64**	**3.36**	**40.4**	**71.10**	**93.74**	**40.2**	**62.76**	**90.96**	28.72	**83.7**

Complete generation is 1.5–4× faster at matched quality. Complete VBench generation falls from 205.6 to 112.4 seconds on Wan, from 284.9 to 187.3 on LTX and from 320.9 to 79.7 on Cosmos, speedups of $1.83\times$, $1.52\times$ and $4.03\times$, and image-to-video speeds up by similar factors (Table 1). The speedup is largest where attention is largest: Cosmos attends over $20 \times 44 \times 80 = 70{,}400$ tokens per call and Wan over $31 \times 22 \times 40 = 27{,}280$, while LTX also runs a second pass and an upscaler. VBench totals differ by at most 0.05 points. PathSpan can also be trained from random initialization (Appendix F).

Where quality moves. Temporal coherence rises in all three generators, and motion binding rises on LTX and Cosmos and falls by 0.2 points on Wan. LTX dynamic-attribute binding falls by 0.17 points (95% CI $[-0.24, -0.11]$) and Cosmos requested-motion accuracy by 3.3 points (95% CI $[-6.2, -0.5]$). On the same videos the EvalCrafter action score and optical-flow warping error stay level (Table 8), so the motion loss concerns which motion category is recognized; Appendix D.5 also shows paired generations (Figure 16).

Against efficient attention methods. Table 2 pairs five trained operators, VSA, VMoBA, SLA, VMonarch and Attention Surgery, each with a PathSpan conversion matched in data, updates, batch and DiT parameter scope, and compares the training-free SVG2 and DFSAttn, run on Dense-FT, with default PathSpan (recipes in Appendix D.3). All seven decide distant weights at the endpoints. PathSpan has higher motion binding and temporal coherence than its partner in all seven comparisons and is faster in six (Figure 2): against VSA it raises requested-motion accuracy from 62.7 to 67.3 at 113.0 instead of 128.2 seconds, and against SLA, the strongest operator, it raises VBench, motion binding and VMBench at a similar time. VMonarch is 2.5 seconds faster at lower VBench and motion binding; Attention Surgery has a 0.02-point higher VBench total at 164.8 instead of 113.8 seconds.

Table 2: External operators on Wan2.2-TI2V-5B (Wan-Video, 2025), evaluated on the six suites (three-root means). Each trained operator is followed by its separately task-matched PathSpan conversion; online operators share default PathSpan. Dense-0 and Dense-FT provide context. Scales follow Table 1. Bold is pairwise for trained conversions and within the online group. Surgery includes its additional teacher and feature-fitting stages (Appendix D.3).

Method / matched recipe	Text-to-video								Image-to-video			
	VBench		CompBench		EvalCr.	VPhy-2	VMBench		AIGCBench			
	Total↑	s↓	MB↑	DAB↑	MAC↑	Joint↑	Total↑	TCS↑	DOV.↑	I–V↑	T–V↑	s↓
Dense-0 (reference)	84.30	203.8	42.8	8.15	65.3	33.8	70.25	93.82	63.43	90.32	28.18	196.3
Dense-FT (reference)	84.85	205.6	44.1	8.53	66.0	34.4	70.70	94.17	63.83	90.44	28.41	199.5
Trained conversions on Wan2.2-TI2V-5B (5B); each adjacent pair uses its own matched task recipe												
VSA (Zhang et al., 2025a)	84.73	128.2	43.2	8.67	62.7	**37.2**	70.80	94.10	**63.87**	**90.53**	28.40	126.8
PathSpan / VSA	**84.90**	**113.0**	**44.0**	**9.15**	**67.3**	36.9	**70.98**	**94.46**	63.82	90.50	**28.48**	**112.2**
VMoBA (Wu et al., 2026)	84.72	133.9	43.7	8.24	**67.3**	35.7	70.82	94.18	**63.83**	90.43	28.38	131.1
PathSpan / VMoBA	**84.78**	**112.7**	**44.5**	**8.60**	66.0	**36.2**	**70.85**	**94.25**	63.77	**90.47**	**28.44**	**112.1**
SLA (Zhang et al., 2026)	84.92	116.2	44.5	9.12	66.0	37.5	71.04	94.37	**64.14**	**90.61**	28.50	114.4
PathSpan / SLA	**85.01**	**113.0**	**44.9**	**9.41**	**68.0**	**37.7**	**71.18**	**94.56**	64.12	90.57	**28.51**	**111.3**
VMonarch (Liang et al., 2026)	84.30	**110.0**	41.4	7.75	61.3	34.7	69.64	93.19	63.06	90.23	28.24	**108.3**
PathSpan / VMonarch	**84.45**	112.5	**43.1**	**8.25**	**62.7**	**35.1**	**70.30**	**93.79**	**63.27**	**90.26**	**28.30**	111.3
Surgery (Ghafoorian et al., 2026)	**84.79**	164.8	43.4	8.80	64.7	**36.9**	70.72	94.26	**63.72**	**90.52**	28.34	160.5
PathSpan / Surgery	84.77	**113.8**	**44.6**	**8.98**	**65.3**	35.8	**70.77**	**94.30**	63.68	90.42	**28.39**	**112.2**
Online replacement on Wan2.2-TI2V-5B (5B) Dense-FT; shared default PathSpan comparator												
SVG2 (Yang et al., 2025a)	84.70	143.5	43.1	8.25	64.7	**36.5**	70.49	94.02	63.63	90.39	28.34	141.0
DFSAttn (Hu et al., 2026)	84.54	139.9	42.6	7.93	62.7	33.5	70.19	93.58	63.62	90.36	28.40	135.5
PathSpan / default	**84.84**	**112.4**	**43.9**	**8.67**	**66.7**	34.4	**70.78**	**94.34**	**63.74**	**90.42**	**28.42**	**110.7**

Table 3: Removing one part of PathSpan at a time (three-root means). Each backbone reports CompBench motion binding (MB), dynamic-attribute binding (DAB) and spatial relationships (SR), and VMBench temporal coherence (TCS) and perceptible amplitude (PAS), all on 0–100 scales. Every control keeps the objective and adaptation protocol. Bold marks the highest mean within each backbone, including ties.

Variant	**Wan2.2-TI2V-5B** (Wan-Video, 2025)					**LTX-Video-13B v0.9.8** (Lightricks, 2025)					**Cosmos-Predict2.5-2B** (NVIDIA, 2025)				
	CompBench			VMBench		CompBench			VMBench		CompBench			VMBench	
	MB↑	DAB↑	SR↑	TCS↑	PAS↑	MB↑	DAB↑	SR↑	TCS↑	PAS↑	MB↑	DAB↑	SR↑	TCS↑	PAS↑
PathSpan	**43.9**	**8.67**	**63.0**	**94.34**	37.5	**41.6**	**9.26**	**60.1**	**95.32**	33.1	**47.6**	**10.25**	60.6	**93.74**	40.2
Fixed address	35.5	7.28	62.2	90.22	32.5	33.5	8.35	59.4	92.03	28.8	36.7	9.14	60.2	88.50	34.7
Shared retention	41.8	6.45	62.0	92.21	36.7	38.3	7.12	59.3	92.88	31.1	44.2	8.18	60.6	91.21	39.2
No-match removed	41.7	8.25	61.1	91.71	**39.4**	39.7	8.86	58.7	92.74	**34.7**	43.9	9.54	59.7	90.41	**43.5**
Single hop	38.8	7.66	62.1	91.69	35.9	37.1	8.11	59.5	93.01	31.8	41.0	9.03	**60.8**	90.39	37.8
No spatial read	41.4	8.14	51.9	92.57	36.6	38.6	8.93	49.5	93.35	31.8	44.1	9.50	50.8	91.64	38.5

6.3 Route, Reach and Stop Are Each Needed

Five controls each remove one part of PathSpan under the same protocol (Table 3, Appendix E.1).

Route. *Fixed address* keeps the continuation mass of each step but reads the same coordinate in the next frame, so content is carried without following motion. *Single hop* reads the neighbor's value V in place of its state H in Equation (4), so every route ends after one frame. The first keeps the chain and loses the motion, the second keeps the motion for one step and loses the chain. Both lower Wan motion binding, from 43.9 to 35.5 and to 38.8, and both lower temporal coherence in all three backbones: a route has to follow the content and continue across frames. A 3×3 window misses the next position more often and lowers Wan motion binding to 37.0 even after retraining (Table 6).

Reach. *Shared retention* gives all channels of a head one coefficient. Dynamic-attribute binding falls to the lowest value of all controls, from 8.67 to 6.45 on Wan, 9.26 to 7.12 on LTX and 10.25 to 8.18 on Cosmos. Figure 5 shows the cause: in the final Wan block, the mass that a first-frame query takes from its current frame ranges from 6.1% to 51.7% across PathSpan's channels, so some channels read far and others stay near, while shared retention sets it to 11.8% for every channel. An attribute that changes over time needs both kinds, far-reading channels to keep the object's identity and near-reading channels to follow its current state.

Stop: carrying farther enlarges motion and lowers coherence. *No-match removed* renormalizes the real candidates to unit mass, so every step continues and a channel keeps its current value only through $1 - \rho_c$. More content then arrives from far away: the mass that first-frame queries receive from the last frame rises from 3.46% to 11.64% (Figure 5). Perceptible amplitude rises in all three generators, from 37.5 to 39.4, 33.1 to 34.7 and 40.2 to 43.5, as expected from carrying more information across the clip, while temporal coherence falls by 2.6, 2.6 and 3.3 points and motion

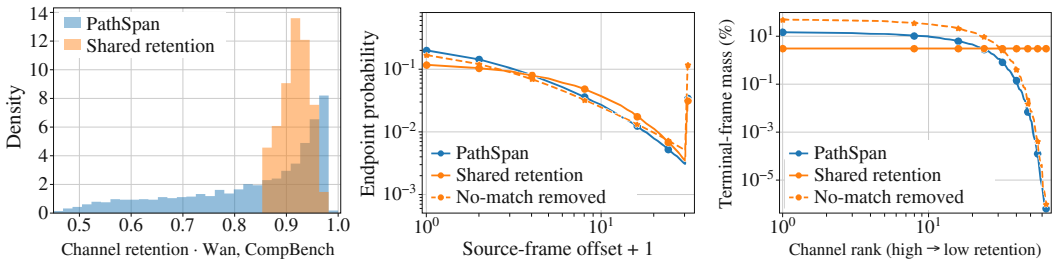


Figure 5: **Retention and stopping shape where the read comes from.** Wan, CompBench, first latent frame, final block, last positive-noise call. Left: learned retention. Middle: probability of each source-frame offset ($+1$ shifts the current frame on the log axis). Right: terminal-frame mass of all 64 channels, ordered by PathSpan retention.

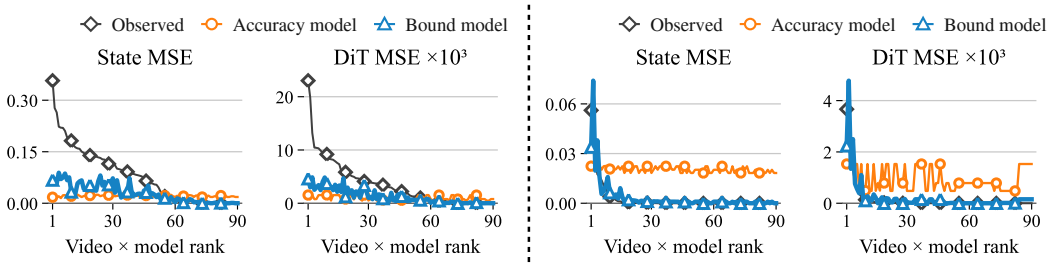


Figure 6: **The bound predicts responses that accuracy misses.** Held-out state and DiT-output responses to replacing radius five by one (left) or nine (right) for all 90 video–model cells (30 DAVIS videos, three backbones), averaged over roots and ordered by observed response. Both predictors are intercept–slope models fitted on PointOdyssey; DiT MSE is scaled by 10^3.

binding falls with it. Without a stop, a state keeps collecting values after the content it followed has been occluded or has left the window, so the videos move more and hold together less.

Within-frame read. *No spatial read* returns $\mathbf{R}_t$ directly. Spatial relationships drop by 9.8 to 11.1 points (63.0 to 51.9 on Wan) while CompBench time falls only from 114.4 to 107.1 seconds: relating two objects needs content that arrived elsewhere in the frame.

6.4 Endpoint Accuracy and the Response Bound

Replacing the learned maps. Recomputing only PathSpan's adjacent-frame logits from the dense checkpoint's queries and keys (Table 5, Appendix A.2) lowers point accuracy by 0.91 to 2.10 points (on Wan, 95% CI $[-4.19, 2.21]$) and motion binding by 4.5 to 6.0 points: a small, uncertain change of endpoint accuracy comes with a large change of generation.

Predicting the response to a local change. At the middle block we replace radius five by one or nine at a common input and measure the change of the state H and of the DiT output. Two capacity-matched affine predictors, built from the mean squared bound B^2 of Equation (7) or from the change of point accuracy, are fitted on 131 PointOdyssey videos (Zheng et al., 2023) and tested on the 30 DAVIS videos. The bound predictor has lower error for both responses in all three generators (for Wan's DiT output, by 3.98×10^{-6} squared MSE; 95% CI $[1.37, 7.41] \times 10^{-6}$). When radius nine leaves the state almost unchanged, the accuracy predictor still forecasts a positive response, while the bound, weighting each change by value contrast, stays near zero (Figure 6).

7 Conclusion

The value of a distant connection in a video diffusion transformer is set by the content between its two frames. PathSpan lets attention learn the route, the per-channel reach and the stop from adjacent-frame matches; this gives 1.52–$4.03\times$ faster complete generation at matched quality, and each of the three controls is needed for motion binding and temporal coherence. Track reachability and the response bound apply to any efficient operator and measure the routes it keeps.

AI USE STATEMENT

We used generative AI to polish writing, find related work, draft sections, and support research ideation and execution, including mathematical exposition and pseudocode preparation. References were verified against primary sources and layout was checked. The authors take responsibility for all content, including AI-assisted text, claims, and artifacts.

REFERENCES

Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. VideoPhy-2: A challenging action-centric physical commonsense evaluation in video generation. In *ICLR*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c02f6a1d5c55e16db50d339dad905b4d-Abstract-Conference.html.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024. URL https://arxiv.org/abs/2307.08691.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, João Carreira, Andrew Zisserman, and Yi Yang. TAP-Vid: A benchmark for tracking any point in a video. In *NeurIPS Datasets and Benchmarks Track*, 2022. URL https://arxiv.org/abs/2211.03726.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *ICCV*, pp. 10061–10072, 2023. URL https://arxiv.org/abs/2306.08637.

Fanda Fan, Chunjie Luo, Wanling Gao, and Jianfeng Zhan. AIGCBench: Comprehensive evaluation of image-to-video content generated by AI. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, 3(4):100152, 2023. doi: 10.1016/j.tbench.2024.100152. URL https://www.tbench.org/single/3-4-4.pdf.

Mohsen Ghafoorian, Denis Korzhenkov, and Amirhossein Habibian. Attention surgery: An efficient recipe to linearize your video diffusion transformer. In *CVPR*, pp. 32915–32925, 2026. URL https://arxiv.org/abs/2509.24899.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *COLM*, 2024. URL https://arxiv.org/abs/2312.00752.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, 2022. URL https://arxiv.org/abs/2204.03458.

Jie Hu, Zixiang Gao, Yutong He, and Kun Yuan. DFSAttn: Dynamic fine-grained sparse attention for efficient video generation. In *ICML*, 2026. URL https://arxiv.org/abs/2605.23445.

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 *CVPR*, pp. 21807–21818, 2024. URL https://arxiv.org/abs/2311.17982.

Allan Jabri, Andrew Owens, and Alexei A. Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, volume 33, pp. 19545–19560, 2020. URL https://arxiv.org/abs/2006.14613.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, 2020. URL https://arxiv.org/abs/2006.16236.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, et al. HunyuanVideo: A systematic framework for large video generative models, 2024. URL https://arxiv.org/abs/2412.03603.

Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, Maneesh Agrawala, Ion Stoica, Kurt Keutzer, and Song Han. Radial attention: $\mathcal{O}(n \log n)$ sparse attention with energy decay for long video generation. In *NeurIPS*, 2025. URL https://arxiv.org/abs/2506.19852.

Cheng Liang, Haoxian Chen, Liang Hou, Qi Fan, Gangshan Wu, Xin Tao, and Limin Wang. VMonarch: Efficient video diffusion transformers with structured attention. In *CVPR*, pp. 4613–4623, 2026. URL https://arxiv.org/abs/2601.22275.

Lightricks. LTX-Video 13B v0.9.8-dev, 2025. URL https://huggingface.co/Lightricks/LTX-Video. Model release.

Xinran Ling, Chen Zhu, Meiqi Wu, Hangyu Li, Xiaokun Feng, Cundian Yang, Aiming Hao, Jiashu Zhu, Jiahong Wu, and Xiangxiang Chu. VMBench: A benchmark for perception-aligned video motion generation. In *ICCV*, pp. 13087–13098, 2025. URL https://arxiv.org/abs/2503.10076.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. EvalCrafter: Benchmarking and evaluating large video generation models. In *CVPR*, pp. 22139–22149, 2024. URL https://arxiv.org/abs/2310.11440.

Jisu Nam, Soowon Son, Dahyun Chung, Jiyoung Kim, Siyoon Jin, Junhwa Hur, and Seungryong Kim. Emergent temporal correspondences from video diffusion transformers. In *NeurIPS*, volume 38, pp. 123683–123727, 2025. doi: 10.52202/085713-4122. URL https://arxiv.org/abs/2506.17220.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. OpenVid-1M: A large-scale high-quality dataset for text-to-video generation. In *ICLR*, 2025. URL https://arxiv.org/abs/2407.02371.

NVIDIA. Cosmos-Predict2.5-2B (post-trained), 2025. URL https://github.com/nvidia-cosmos/cosmos-predict2.5. Model and inference-code release.

Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *NeurIPS*, volume 34, pp. 12493–12506, 2021. URL https://arxiv.org/abs/2106.05392.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pp. 4195–4205, 2023. URL https://arxiv.org/abs/2212.09748.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. In *Proceedings of the 42nd ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56242–56280, 2025. URL https://arxiv.org/abs/2502.06764.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2V-CompBench: A comprehensive benchmark for compositional text-to-video generation. In *CVPR*, pp. 8406–8416, 2025. URL https://arxiv.org/abs/2407.14505.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *NeurIPS*, 2023. URL https://arxiv.org/abs/2306.03881.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018. URL https://arxiv.org/abs/1806.09594.

Wan-Video. Wan2.2-TI2V-5B, 2025. URL https://github.com/Wan-Video/Wan2.2. Model and inference-code release.

Jianzong Wu, Liang Hou, Haotian Yang, Ye Tian, Pengfei Wan, Di Zhang, and Yunhai Tong. VMoBA: Mixture-of-block attention for video diffusion models. In *ICLR*, pp. 132534–132548, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d6c4014ff8d95025aa35d831c0f81faa-Abstract-Conference.html.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse Video-Gen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *ICML*, 2025. URL https://proceedings.mlr.press/v267/xi25c.html.

Zeqi Xiao, Wenqi Ouyang, Yifan Zhou, Shuai Yang, Lei Yang, Jianlou Si, and Xingang Pan. Trajectory attention for fine-grained video motion control. In *ICLR*, 2025. URL https://arxiv.org/abs/2411.19324.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. SANA: Efficient High-Resolution Image Synthesis with Linear Diffusion Transformers. In *ICLR*, 2025. URL https://arxiv.org/abs/2410.10629.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse VideoGen2: Accelerate video generation with sparse attention via semantic-aware permutation. In *NeurIPS*, volume 38, pp. 96965–96991, 2025a. doi: 10.52202/085713-3243. URL https://doi.org/10.52202/085713-3243.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *ICML*, 2024. URL https://proceedings.mlr.press/v235/yang24ab.html.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025b. URL https://arxiv.org/abs/2408.06072.

Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, Joseph E. Gonzalez, Jianfei Chen, and Jun Zhu. SLA: Beyond sparsity in diffusion transformers via fine-tunable sparse–linear attention. In *ICLR*, pp. 138288–138305, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/df77cca99e2785eafb5b2416d688ae24-Abstract-Conference.html.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. In *NeurIPS*, volume 38, pp. 152509–152534, 2025a. doi: 10.52202/085713-5099. URL https://doi.org/10.52202/085713-5099.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. In *ICML*, 2025b. URL https://arxiv.org/abs/2502.04507.

Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. PointOdyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, pp. 19855–19865, 2023. URL https://arxiv.org/abs/2307.15055.

A Route Diagnostics: Correspondence and Reachability

This appendix specifies the readouts behind Sections 3.2, 3.3 and 6.4: chained and direct correspondence (Appendix A.1), the replacement of learned maps by dense-checkpoint maps (Appendix A.2), and track reachability together with the window-size controls (Appendix A.3).

Table 4: Same-endpoint correspondence on all 30 TAP-Vid-DAVIS videos (Doersch et al., 2022). Direct and chained readouts share features, endpoints and native interpolation. Acc. averages the five official point-distance thresholds; P@2 and P@8 report correctness at two and eight pixels. All entries are percentages averaged over three training roots. Bold compares the two readouts for each metric within the same backbone and feature source.

Readout	Dense-FT features			PathSpan features		
	Acc.↑	P@2↑	P@8↑	Acc.↑	P@2↑	P@8↑
Wan2.2-TI2V-5B (5B) (Wan-Video, 2025)						
Direct distant frame	15.30	6.40	19.46	15.18	6.53	19.15
Composed adjacent maps	**16.42**	**7.10**	**21.77**	**16.76**	**7.32**	**22.47**
LTX-Video-13B v0.9.8 (13B) (Lightricks, 2025)						
Direct distant frame	16.06	6.87	20.05	16.28	7.15	20.46
Composed adjacent maps	**18.24**	**7.11**	**24.06**	**19.37**	**7.79**	**26.09**
Cosmos-Predict2.5-2B (2B) (NVIDIA, 2025)						
Direct distant frame	15.19	6.25	19.37	15.43	6.78	19.31
Composed adjacent maps	**17.14**	**6.78**	**23.21**	**18.28**	**8.20**	**24.52**

A.1 Chained and Direct Correspondence Readouts

Data. We use the complete TAP-Vid-DAVIS set (Doersch et al., 2022) with the first-visible query protocol (Doersch et al., 2023): 30 videos, 650 queries and 28,174 visible evaluation positions after the query frames, grouped by source video. All original frames are encoded in order. The last frame is repeated only to reach $1 + k_{\mathrm{str}}\lceil (F-1)/k_{\mathrm{str}} \rceil$ frames, where F is the original frame count and k_{str} the native temporal stride (4, 8 and 4 for Wan, LTX and Cosmos); evaluation uses only the original frames. Spatial resizing from the official 256×256 input applies the same transform to the coordinates.

Features. Each generator receives the native noisy version of the real video at its last strictly positive noise level, with the empty text condition. Features come from all heads of the middle self-attention block, at zero-based index $\lfloor (L-1)/2 \rfloor$ for L blocks (14, 23 and 13 for the three backbones); LTX uses its final-scale feature interface. This follows the general setting of DiffTrack (Nam et al., 2025).

Chained readout. For a source frame t and a later target frame u, the chained readout conditions the route weights of Equation (2) on reaching the target,

$$C_{t,u}(i,j) = \frac{[\mathbf{P}_t^+ \mathbf{P}_{t+1}^+ \cdots \mathbf{P}_{u-1}^+](i,j)}{\sum_k [\mathbf{P}_t^+ \mathbf{P}_{t+1}^+ \cdots \mathbf{P}_{u-1}^+](i,k)}. \tag{8}$$

Every local row sum is kept until this final normalization, so a stop lowers the mass of a route before the endpoint distribution is formed. For numerical stability, the propagation rescales the accumulated mass and stores the log normalizers, which keeps every query. Earlier targets use the past maps, and both readers use the identity when source and target coincide. For Dense-FT features, the maps use Dense-FT's own queries and keys together with the no-match logits of the paired PathSpan model of the same root, so the two feature sources differ only in their features.

Direct readout and scoring. The direct reader applies a softmax of the source query against every key of the target frame, with the same features and endpoints as the chained reader. Each head gives a distribution; their equal average selects the most probable destination, with ties resolved in raster order. Bilinear interpolation over source coordinates and linear interpolation over the actual VAE temporal anchors map destinations to original frames, identically for both readers and with the native first-frame convention. Strict errors below 1, 2, 4, 8 and 16 pixels at the official resolution define five accuracies, averaged within each video and then equally across videos; Acc. is their mean, and P@2 and P@8 report two of the thresholds. The generator learns its maps only through the generation objective; annotated positions enter only this evaluation.

Results. Table 4 crosses feature source with readout. Chaining improves point accuracy by 1.58–3.08 points with PathSpan features and by 1.11–2.18 points with Dense-FT features, so the gain comes from the readout itself and is present in features trained with dense attention. For source-video uncertainty, 10,000 paired bootstrap draws resample the 30 videos and keep all roots and conditions of a video together (seed 20260911); percentile intervals are computed for the source-equal differences. Wan's gain with PathSpan features has the interval $[-0.81, 4.17]$, and LTX and Cosmos have intervals above zero. Figure 7 shows that the positive mean contains videos and separations with negative gains. In its panel (c), each separation averages the videos available at that separation, so the set of videos changes along the axis.

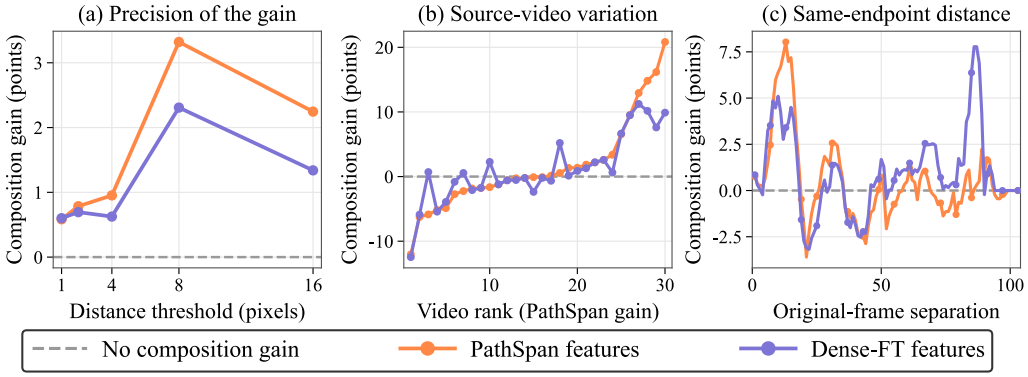


Figure 7: **Chaining gains vary across thresholds, videos and separations.** Wan chained-minus-direct point accuracy with identical features and endpoints, for PathSpan and Dense-FT features. (a) Source-equal gains at the five pixel thresholds. (b) All 30 DAVIS videos ordered by PathSpan's five-threshold mean gain; Dense-FT uses the same order. (c) All 103 observed original-frame separations, averaging the available source videos equally at each separation. Panels (b) and (c) average correctness at 1, 2, 4, 8 and 16 pixels. Curves average three roots; differences are percentage points, and the dashed line is zero gain.

A.2 Replacing the Learned Maps

The test of Section 6.4 recomputes PathSpan's adjacent-frame logits from the query and key projections and normalization gains of the original dense checkpoint, applied to the recipient's current input. The recipient's values, retention and no-match scalars, within-frame read and output functions stay unchanged. Because the logits set both the candidate weights and the no-match mass, the test replaces the learned route and the learned stop together. The correspondence readout uses the same substitution, including the normalization gains. Table 5 places the two outcomes side by side. Source-video bootstrap intervals describe the correspondence variation, and the generation scores aggregate the complete CompBench and VMBench sets; each analysis keeps its own sampling unit.

Table 5: Learned versus dense-donor local projections (three-root means). Acc. is TAP-Vid-DAVIS mean point accuracy (%); MB is CompBench motion binding $\times 100$; MSS/TCS are VMBench smoothness/coherence. Bold compares projections within each backbone.

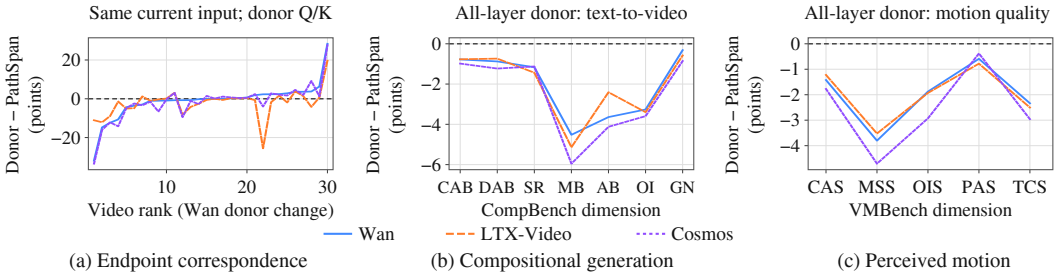
Projections	TAP-Vid	CompB.	VMBench	
	Acc.↑	MB↑	MSS↑	TCS↑
Wan2.2-TI2V-5B (5B) (Wan-Video, 2025)				
Learned	**16.76**	**43.9**	**74.08**	**94.34**
Dense donor	15.85	39.4	70.28	92.00
LTX-Video-13B v0.9.8 (13B) (Lightricks, 2025)				
Learned	**19.37**	**41.6**	**75.26**	**95.32**
Dense donor	17.27	36.4	71.74	92.82
Cosmos-Predict2.5-2B (2B) (NVIDIA, 2025)				
Learned	**18.28**	**47.6**	**72.42**	**93.74**
Dense donor	16.57	41.6	67.72	90.78

Figure 8 resolves the comparison. Donor maps raise point accuracy on some source videos and lower it on others, while the generation scores fall in every CompBench and VMBench dimension, most in motion binding and motion smoothness. The dimensions that depend on following content over time respond the most, which matches the role of the learned route.

A.3 Track Reachability and Window Size

Fixed-budget DFSAttn. The test of Section 3.3 fixes the retained fraction of DFSAttn at 0.4 or 0.1 after its native dense prefix and keeps its scoring, Hilbert ordering, pooling, mask construction and cache handling. At the feature-readout noise level, each condition starts from an empty cache and builds its sparse mask explicitly, so the same retained graph serves both the endpoint readout and the reachability test. This fixed-budget comparison is separate from the scheduled DFSAttn run of Table 2.

Reachability. For each annotated track, the graph vertices are the native latent cells that contain the ground-truth positions at the latent temporal anchors, and the edges are the connections retained by one head's mask, including long edges that skip frames. For a head's graph $G_b = (\mathcal{V}, \mathcal{E}_b)$ and track vertices $v_t, \ldots, v_u$, reachability asks whether G_b contains a directed path from v_t to v_u.



(a) Endpoint correspondence (b) Compositional generation (c) Perceived motion

Figure 8: **Learned maps matter more for generation than for correspondence.** Dense-checkpoint query/key projections and normalization gains replace only PathSpan's adjacent logits on the recipient's current input; recipient values, scalars and output functions remain. (a) Donor-minus-PathSpan chained accuracy on all 30 DAVIS videos, averaged over the five thresholds, in a common Wan-based order. (b) Seven CompBench dimensions after substitution at every layer and sampling call (1,400 prompts; scores $\times 100$). (c) Five VMBench dimensions under the same intervention (1,050 prompts; 0–100 scale). Each backbone averages three roots; differences are points.

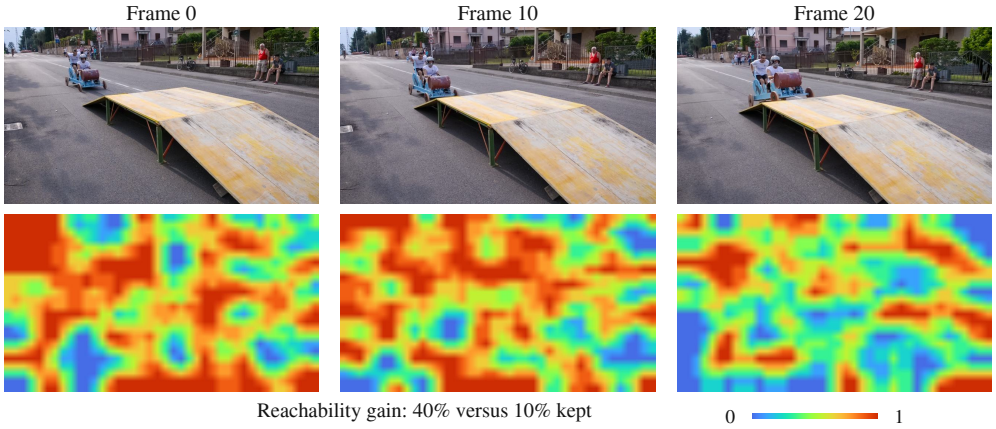


Figure 9: **Keeping fewer key blocks cuts directed routes.** Each DAVIS source frame (0, 10 or 20) has its own field to frame 35. Color is the 40%-minus-10% reachability gain, shown as a 3×3 local proportion on a shared 0–1 scale. Nearest-RGB reference paths are independent of the masks. Nested rankings (seed 20260916) retain 10% or 40% of future 3×4-cell key blocks, allowing adjacent and long edges. These controlled masks isolate path availability; Section 3.3 reports native DFSAttn with annotated tracks.

Adjacent-chain survival. This test asks for the stricter event $\prod_{s=t}^{u-1} \mathbf{1}\{(v_s, v_{s+1}) \in \mathcal{E}_b\} = 1$; a retained long edge can support reachability where an adjacent link is missing. Reachability is evaluated within each head. Intervals with fewer than two genuine anchors, with padding in place of annotations or with an occlusion in between leave the graph tests, while correspondence keeps its official visible-point denominator. Figure 9 shows the same effect with controlled masks on fixed reference paths: where fewer key blocks are kept, directed routes are cut.

Sparse direct reader. The endpoint readout of the sparse model uses only the retained keys at the requested target frame. A head with no retained key contributes zero mass; if every head is empty, or an interpolation endpoint is unavailable, the query counts as incorrect at every threshold. Every eligible query keeps its original sparse mask.

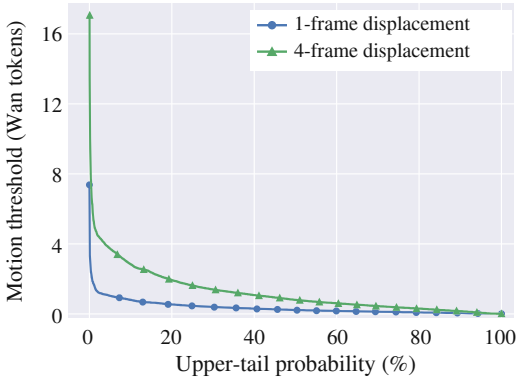


Figure 10: **The latent stride sets the motion a window must cover.** Source-equal upper-tail displacement on 30 DAVIS videos, from 27,703 one-frame and 24,559 overlapping four-frame intervals. Displacement is Chebyshev distance on Wan's 40×22 token grid; every intervening frame must be visible.

Table 6: Window-size controls on Wan2.2-TI2V-5B (Wan-Video, 2025), LTX-Video-13B v0.9.8 (Lightricks, 2025) and Cosmos-Predict2.5-2B (NVIDIA, 2025) (three-root means). MB is CompBench motion binding $\times 100$; TCS is VMBench coherence; s is complete CompBench generation time. Fixed-weight changes and matched retraining are separate comparison groups. Bold marks the best mean within each group and backbone.

Support setting	Wan2.2-TI2V-5B			LTX-Video-13B v0.9.8			Cosmos-Predict2.5-2B		
	MB↑	TCS↑	s↓	MB↑	TCS↑	s↓	MB↑	TCS↑	s↓
Fixed weights: change only the local candidate radius; default radius five is the common reference									
Radius 1 / fixed	34.0	88.80	**109.2**	34.4	90.78	**181.0**	35.1	86.85	**73.3**
Radius 5 / default	**43.9**	**94.34**	114.4	**41.6**	**95.32**	190.0	**47.6**	**93.74**	81.5
Radius 9 / fixed	43.4	93.87	125.1	40.6	94.78	203.8	46.5	92.90	100.5
Matched retraining: use the stated radius throughout the default adaptation schedule									
Radius 1 / trained	37.0	91.45	**110.1**	37.2	93.08	**180.6**	38.3	89.82	**73.0**
Radius 9 / trained	**44.8**	**94.35**	124.5	**41.7**	**95.14**	204.8	**47.9**	**93.77**	100.3

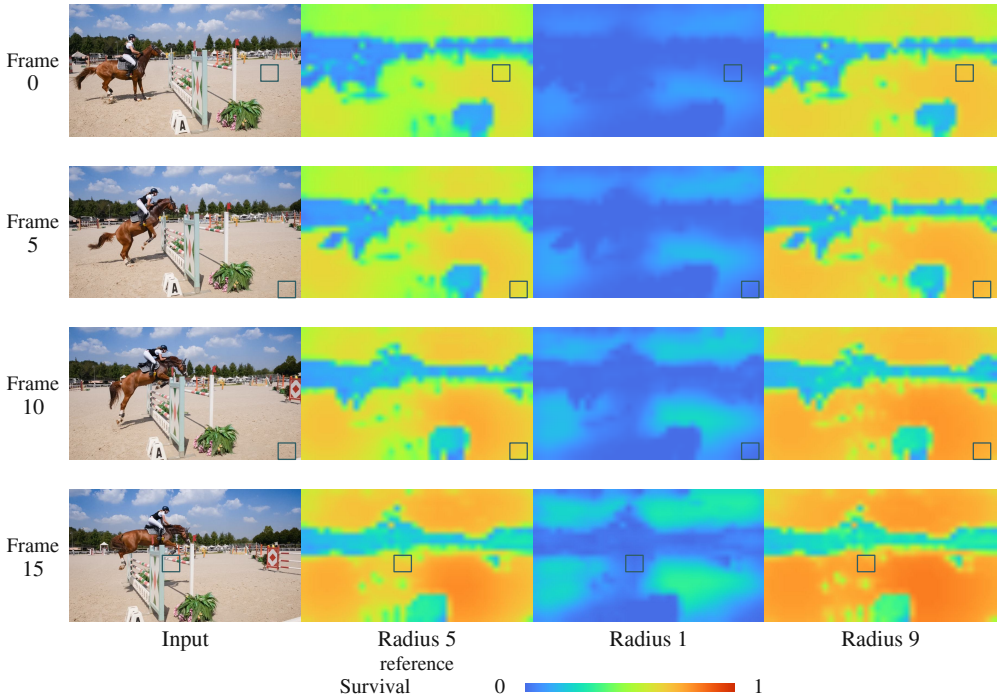


Figure 11: **The window changes the mass that survives along routes.** DAVIS source frames 0, 5, 10 and 15 share target frame 35. Columns show the input and radii five, one and nine. Color is unnormalized survival $\sum_j [\mathbf{P}_t \cdots \mathbf{P}_{u-1}](i,j)$, before the conditioning of Equation (8) and without ρ. Fixed RGB/gradient and no-match logits are renormalized on each window. All twelve fields share the 0–1 scale. Marked regions maximize the local mean absolute radius-nine minus radius-one difference.

Point accuracy thus measures the selected destination, and reachability measures whether the retained edges connect the annotated positions; the two move in opposite directions in Section 3.3. A high conditional probability at an endpoint and the existence of a route through the intermediate annotated positions therefore require separate readouts.

Motion in the input. Figure 10 relates the window to the motion in the input. Wan's four-frame latent stride lengthens the tail of displacements between latent frames, so a radius counted in latent tokens has to cover a longer motion than the same radius between consecutive video frames. Source-equal aggregation gives each video the same weight, and long tracks contribute observations within their source without dominating the curve. The plotted displacement is measured from annotations and is separate from the learned probability of each candidate.

Window size. Table 6 separates two ways of changing the window. Fixed-weight changes keep the learned radius-five parameters and change only the candidate set; matched retraining uses the stated radius throughout the default adaptation (Appendix D.2). At a common input, enlarging a nested window changes both the candidates and the normalization. Let Z_r be the denominator of Equation (3)

at radius r. For $r < R$, $Z_R \geq Z_r$, so every shared candidate has $P_R(i,j) = P_r(i,j)Z_r/Z_R$ and loses weight, although the continuation mass $a = 1 - e^{\beta}/Z$ grows. More continuation can therefore come with less weight on a previously favored route. Radius nine at fixed weights shows this redistribution, lowering motion binding from 43.9 to 43.4, 41.6 to 40.6 and 47.6 to 46.5; retraining adapts the logits and scalars to the larger window and reaches 44.8, 41.7 and 47.9. Figure 11 shows the unnormalized mass that survives along routes under each window, before the conditioning of Equation (8); that conditioning can turn a small surviving mass into a concentrated endpoint distribution, so localization and survival answer different questions.

B Complete Algorithm and Implementation

B.1 Notation

Indices t, u, s denote latent frames, i, j, k spatial positions, c channels and b heads; τ denotes diffusion noise time. Matrices are bold capitals, vectors bold lowercase, and scalar entries plain symbols. Direction $\delta = +1$ reads future values and $\delta = -1$ past values; the head index b, suppressed in the main text, is restored in Algorithm 1.

Symbol	Shape or domain	Meaning
$\mathbf{X}$	$T \times S \times D$	Input to one native self-attention call
$\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t$	$S \times d$ per head	Native projected queries, keys and values of frame t
$\Omega(i)$	subset of $\{1, \ldots, S\}$	Valid adjacent-frame candidates of position i; $\lvert\Omega(i)\rvert \leq m$
$\ell_t^{\delta}(i,j)$, β^{δ}	scalars	Adjacent-frame logit and no-match logit
$\mathbf{P}_t^{\delta}$	$S \times S$, sparse	Real-candidate probabilities (Equation (3)); zero outside $\Omega(i)$
$\boldsymbol{a}_t^{\delta}$	$[0,1]^S$	Continuation mass, the row sums of $\mathbf{P}_t^{\delta}$
$\boldsymbol{\rho}^{\delta}$	$(0,1)^{d/2}$	Per-channel retention
$\mathbf{V}_t^{\delta}, \mathbf{H}_t^{\delta}$	$S \times (d/2)$	Directional current values and states
$\mathbf{R}_t, \mathbf{Y}_t$	$S \times d$	Concatenated states and within-frame read
$\mathbf{A}_t$	$S \times S$	Row-stochastic within-frame attention
$\mathbf{D}_u$	$S \times S$, diagonal	Current-value injection of one channel
$\pi_s(k, \cdot)$	distribution	Stop and move probabilities at (s,k)
$\gamma_{t,s}(i,k)$	$[0,1]$	Probability of reaching (s,k) from (t,i) before stopping
$b_s(k)$, $B_{t,i}$	nonnegative	Candidate-value range and response bound

B.2 Attention Operator

Algorithm 1 specifies the replacement of one video self-attention call. It is invoked separately for every layer, native guidance branch and denoising call that uses video attention, and all maps and states are computed from that call's input. The ruled, numbered format follows Algorithm 1 of SANA (Xie et al., 2025); lines 8–12 implement Equation (3), lines 22–23 Equation (4) and line 29 Equation (6).

Windows, boundaries and memory. For a latent grid of height S_h and width S_w, $S = S_h S_w$. A radius-r window contains the raster-indexed cells within Chebyshev distance r of i, intersected with the grid, so $m \leq (2r+1)^2$. Positions outside the grid have logit $-\infty$; valid logits and β share the stable softmax of lines 8–12, and the real-candidate probabilities keep their original row sums. The temporal boundary uses $\mathbf{H} = \mathbf{V}$, and the scans are empty when $T = 1$. Only the m local entries of each row of $\mathbf{P}$ are stored. Line 29 can use an exact tiled attention kernel with $\mathbf{R}$ as its value argument.

Initialization. For $C = d/2 > 1$ channels per direction, the half-lives start at $\lambda_c = T^{(c-1)/(C-1)}$, $c = 1, \ldots, C$, with $\rho_c = 2^{-1/\lambda_c}$ and $r_c = \log(\rho_c/(1-\rho_c))$, independently in each head and direction; a single channel uses $\lambda_1 = 1$. Here T is the latent frame count of the adaptation clips. The default radius five uses $\beta = \log 121$, and the retrained radius-one and radius-nine controls use $\log 9$ and $\log 361$, so that the no-match entry starts with half of the mass at zero logits. Fixed-weight

Algorithm 1 PathSpan: adjacent-frame transitions, directional scans and within-frame read

Require: Block input $\mathbf{X} \in \mathbb{R}^{T\times S\times D}$; n_h heads, $d = D/n_h \in 2\mathbb{N}$;
native projections $\Phi_Q, \Phi_K, \Phi_V, \Phi_O$; windows $\Omega(i)$;
retention logits $r^\delta_{b,c}$ and no-match logits β^δ_b, $\delta \in \{-1, +1\}$
Ensure: Projected attention output $\mathbf{O} \in \mathbb{R}^{T\times S\times D}$ for the native residual branch
$(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \leftarrow (\Phi_Q(\mathbf{X}), \Phi_K(\mathbf{X}), \Phi_V(\mathbf{X}))$
Φ_Q, Φ_K include native query–key normalization and positional encoding
for $b = 1, \dots, n_h$ **do**
  $\mathbf{V}^+_b \leftarrow \mathbf{V}_b[:, :, 1 : d/2]$; $\mathbf{V}^-_b \leftarrow \mathbf{V}_b[:, :, d/2 + 1 : d]$
  for $\delta \in \{+1, -1\}$ **do**
    $\rho^\delta_{b,c} \leftarrow \mathrm{sigmoid}(r^\delta_{b,c}), \quad c = 1, \dots, d/2$
    for $t \in \{0, \dots, T-1\}$ such that $0 \le t + \delta < T$ **do**
      for $i = 1, \dots, S$ **do**
        $\ell_j \leftarrow \boldsymbol{q}^\top_{b,t,i} \boldsymbol{k}_{b,t+\delta,j}/\sqrt{d}, \quad j \in \Omega(i)$
        $z \leftarrow \max\big(\{\beta^\delta_b\} \cup \{\ell_j : j \in \Omega(i)\}\big)$
        $Z \leftarrow \exp(\beta^\delta_b - z) + \sum_{j\in\Omega(i)} \exp(\ell_j - z)$
        $P^\delta_{b,t}(i, j) \leftarrow \exp(\ell_j - z)/Z, \quad j \in \Omega(i)$
        $a^\delta_{b,t,i} \leftarrow \sum_{j\in\Omega(i)} P^\delta_{b,t}(i, j)$
      end for
    end for
    if $\delta = +1$ **then**
      $\mathbf{H}^+_{b,T-1} \leftarrow \mathbf{V}^+_{b,T-1}$; $\quad \mathcal{I} \leftarrow (T-2, T-3, \dots, 0)$
    else
      $\mathbf{H}^-_{b,0} \leftarrow \mathbf{V}^-_{b,0}$; $\quad \mathcal{I} \leftarrow (1, 2, \dots, T-1)$
    end if
    for $t \in \mathcal{I}$ in the listed order **do**
      for $i = 1, \dots, S$ and $c = 1, \dots, d/2$ **do**
        $g \leftarrow \sum_{j\in\Omega(i)} P^\delta_{b,t}(i, j)\, H^\delta_{b,t+\delta,j,c}$
        $H^\delta_{b,t,i,c} \leftarrow (1 - \rho^\delta_{b,c} a^\delta_{b,t,i})\, V^\delta_{b,t,i,c} + \rho^\delta_{b,c}\, g$
      end for
    end for
  end for
  for $t = 0, \dots, T-1$ **do**
    $\mathbf{R}_{b,t} \leftarrow [\mathbf{H}^+_{b,t}, \mathbf{H}^-_{b,t}]$
    $\mathbf{Y}_{b,t} \leftarrow \mathrm{ExactAttention}(\mathbf{Q}_{b,t}, \mathbf{K}_{b,t}, \mathbf{R}_{b,t})$
  end for
end for
$\mathbf{O} \leftarrow \Phi_O\big(\mathrm{Concat}^{n_h}_{b=1}\, \mathbf{Y}_b\big)$
return O

window changes keep the learned scalars. Appendix D.2 specifies the shared-retention initialization and the trainable scope.

A three-frame check. For one position and one future channel, let $p_0, p_1 \in [0, 1]$ be the continuation masses of two consecutive steps and V_0, V_1, V_2 the scalar values. The terminal rule gives $H_2 = V_2$, and two updates yield

$$H_0 = (1 - \rho p_0)V_0 + \rho p_0(1 - \rho p_1)V_1 + \rho^2 p_0 p_1 V_2. \tag{9}$$

The coefficients sum to one. Setting $p_0 = 0$ returns V_0, setting $p_1 = 0$ removes the terminal value, and equal values stay equal. These identities test stopping, temporal order and terminal injection together; the same check after reversing the frame order tests the past scan.

B.3 Training and Native Sampling

At each update, sample $(\boldsymbol{x}, \kappa) \sim \mathcal{D}$ and a native noise time τ. The backbone's noise construction gives the noisy latent, regression target and loss weight, $(\boldsymbol{x}_\tau, \boldsymbol{\nu}^\star, w) = \mathcal{N}_\tau(\boldsymbol{x}, \kappa)$; for the flow-

matching backbones $\boldsymbol{\nu}^\star$ is the velocity. With $\widehat{\boldsymbol{\nu}} = f_\theta(\boldsymbol{x}_\tau, \kappa, \tau)$, update the trainable parameters θ_{train} with $\nabla_{\theta_{\text{train}}} w\|\widehat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\star\|^2$. Appendix D.2 fixes θ_{train} and the optimizer.

Generation initializes $\boldsymbol{x}_{\tau_K}$ from the native conditional prior. For $k = K, \dots, 1$, it computes $\widehat{\boldsymbol{\nu}}_k = \mathrm{NativeGuidance}(f_\theta, \boldsymbol{x}_{\tau_k}, \kappa, \tau_k)$ and applies the native update $\boldsymbol{x}_{\tau_{k-1}} = \mathcal{S}_k(\boldsymbol{x}_{\tau_k}, \widehat{\boldsymbol{\nu}}_k, \kappa)$, then decodes $\boldsymbol{x}_{\tau_0}$. The native wrapper supplies normalization, modulation and residual gating around Algorithm 1. LTX applies it in both native passes, and Minecraft prediction passes each generated window into the next history (Appendix F).

In backpropagation, $a^\delta_{t,i} = \sum_j P^\delta_t(i,j)$ stays in the computation graph. A local probability affects both the current-value weight and the carried values, so both derivative paths contribute. Shared maps accumulate gradients from all their channels and downstream states, and each head's retention and no-match parameters accumulate gradients over every position and frame where they act.

C Proofs and Operator Analysis

C.1 Route Expansion and Normalization

Fix the future direction and a channel, and write $\boldsymbol{h}_t, \boldsymbol{v}_t \in \mathbb{R}^S$ for the state and value columns. Let $\mathbf{D}_u = \mathbf{I} - \rho\,\mathrm{diag}(\boldsymbol{a}_u)$ for $u < T-1$ and $\mathbf{D}_{T-1} = \mathbf{I}$. In this notation Equation (4) reads $\boldsymbol{h}_t = \mathbf{D}_t\boldsymbol{v}_t + \rho\mathbf{P}_t\boldsymbol{h}_{t+1}$. The boundary state $\boldsymbol{h}_{T-1} = \boldsymbol{v}_{T-1}$ satisfies Equation (5). If the expansion holds at $t+1$, substituting it into the recurrence gives the expansion at t, and backward induction proves the formula. Each entry of a matrix product is a sum over the intermediate spatial indices, which enumerates the routes of Equation (2) algebraically.

Every coefficient is nonnegative because $0 \le a_{t,i} \le 1$ and $0 < \rho < 1$, and at each step the current and moving coefficients sum to $(1 - \rho a_{t,i}) + \rho a_{t,i} = 1$. Induction from the terminal frame therefore makes each scalar state a convex combination of source values. Constant values are preserved, and each state lies between the smallest and largest reachable source values. Finite logits give every valid candidate positive mass, so a route's contribution is governed by its transition product, the channel retention and the injection at its end. Reversing the frame order proves the past-direction result.

For any ℓ consecutive moves, the substochastic rows give $\|\rho^\ell \mathbf{P}_t \cdots \mathbf{P}_{t+\ell-1}\|_\infty \le \rho^\ell$. Retention therefore sets an upper envelope on the mass that survives ℓ steps, and the observed transition products decide which positions receive that mass. This ties the reach of a channel to the actual local evidence.

Figure 4 in Section 4.2 displays the additive identity of Equation (4) with RGB-valued diagnostic scans. With equal retention, the past and future states still differ, because the two directions meet different observations. The figure makes the current injection and the carried state visible in image coordinates.

C.2 Proof of the Response Bound

Suppress the channel index. The moving matrix is $\mathbf{M}_s = \rho\mathbf{P}_s$, and the probability of reaching an active state is

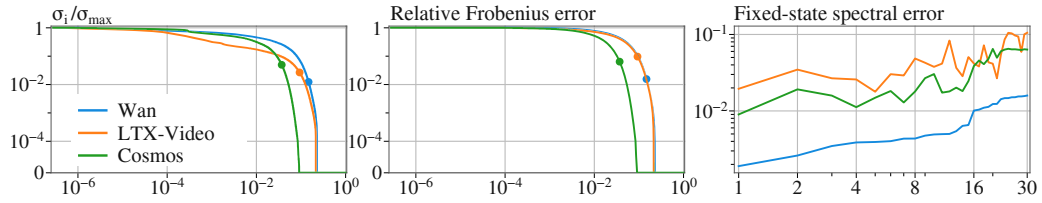
$$\gamma_{t,s}(i,k) = \begin{cases} \mathbf{1}\{i = k\}, & s = t, \\ [\mathbf{M}_t\mathbf{M}_{t+1}\cdots\mathbf{M}_{s-1}](i,k), & s > t. \end{cases} \tag{10}$$

The product contains only moving outcomes, so it includes the probability of surviving every earlier step. Define the state difference $e_s(k) = H_{s,k} - \widetilde{H}_{s,k}$; the common terminal values give $e_{T-1}(k) = 0$. Adding and subtracting the second scan's continuation values yields

$$e_s(k) = \eta_s(k) + \sum_j M_s(k,j)\, e_{s+1}(j), \qquad \eta_s(k) = \sum_z \big[\pi_s(k,z) - \widetilde{\pi}_s(k,z)\big]\, U_s(k,z), \tag{11}$$

where z ranges over stopping and the union of both supports, and stopping uses the shared current value. Repeated substitution gives the exact finite identity

$$H_{t,i} - \widetilde{H}_{t,i} = \sum_{s=t}^{T-2} \sum_{k=1}^{S} \gamma_{t,s}(i,k)\, \eta_s(k). \tag{12}$$



(a) Spectrum: rank / full dimension (b) Error: rank / full dimension (c) Fixed-state error: video rank

Figure 12: **A fixed-size global summary leaves a residual of the value operator.** (a) Full normalized singular spectrum and (b) relative Frobenius truncation error of the future-direction operator on DAVIS goat, averaged over three roots. Rank is divided by the complete output dimension: 1,351,680 (Wan), 2,867,200 (LTX-Video) and 3,604,480 (Cosmos), including structural zeros. Circles mark capacities $q = 199{,}680$, 266,240 and 133,120, which count SLA global preaggregation with its key normalizer and exclude query-dependent correction and bypass. A linear region below 10^{-4} shows exact zero residuals. (c) Fixed-capacity relative spectral error on all 30 DAVIS videos in a common Wan-based order; curves show three-root means.

At a fixed state, set $w_z = \pi_s(k,z) - \widetilde{\pi}_s(k,z)$. Both distributions sum to one, so $\sum_z w_z = 0$, and the positive and negative parts of w each have mass $\mathrm{TV}(\pi_s(k,\cdot), \widetilde{\pi}_s(k,\cdot))$. Subtracting the minimum candidate value leaves $\eta_s(k)$ unchanged and places every shifted value in $[0, b_s(k)]$, hence $|\eta_s(k)| \le b_s(k)\,\mathrm{TV}(\pi_s(k,\cdot), \widetilde{\pi}_s(k,\cdot))$. The triangle inequality applied to Equation (12) proves Proposition 1. Zero extension to the union of finite supports covers window changes as well as probability changes on a fixed window.

C.3 Within-Frame Read

For a fixed row-stochastic $\mathbf{A}_t$, a channel's read difference is $\Delta Y_{t,i,c} = \sum_k A_t(i,k)\,\Delta R_{t,k,c}$, whose absolute value is at most $\sum_k A_t(i,k)\,|\Delta R_{t,k,c}|$. Substituting the directional bounds gives $|\Delta Y_{t,i,c}| \le \sum_k A_t(i,k)\,B_{t,k,c}$. This is the response before the native output projection, for shared values and fixed within-frame probabilities; the convex weights of $\mathbf{A}_t$ carry the direction-specific bounds through the concatenation of the two channel groups. The complete-sampling interventions of Section 6.4 separately measure generated outcomes after inputs and later predictions evolve.

C.4 Fixed-State Spectral Diagnostic

Fixing the maps, retentions and within-frame read makes Equations (5)–(6) linear in the values. For this conditional value operator, let $\sigma_1 \ge \cdots \ge \sigma_n \ge 0$ be its complete singular spectrum. The best rank-q relative Frobenius residual is $(\sum_{j>q} \sigma_j^2 / \sum_j \sigma_j^2)^{1/2}$. Figure 12 compares this residual with the capacity of SLA's global preaggregation (Zhang et al., 2026): a summary of that size leaves a measurable residual of the PathSpan value operator. The diagnostic characterizes the fixed value representation only; Table 2 evaluates SLA's full operator, including its query-dependent correction and bypass.

D Generation Setup and Extended Results

D.1 Native Generation Conditions

Wan. We use the released TI2V-5B checkpoint (Wan-Video, 2025), which produces 121 frames at 1280×704 and 24 frames per second. Sampling uses 50 steps, shift 5 and guidance 5, and text-to-video and image-to-video runs keep their native conditioning procedures. The latent attention grid is $31 \times 22 \times 40$.

LTX-Video. We use the 13B v0.9.8 development checkpoint (Lightricks, 2025), which produces 121 frames at 1216×704 and 30 frames per second. Its multistage pipeline uses 27 and then 13 noise updates and keeps the upscaler and spatiotemporal guidance. The first pass uses an 800×448 video grid with $16 \times 14 \times 25$ tokens, and the second uses 1600×896 with $16 \times 28 \times 50$ tokens before decoded resizing. Both passes and all guidance calls are included in complete time.

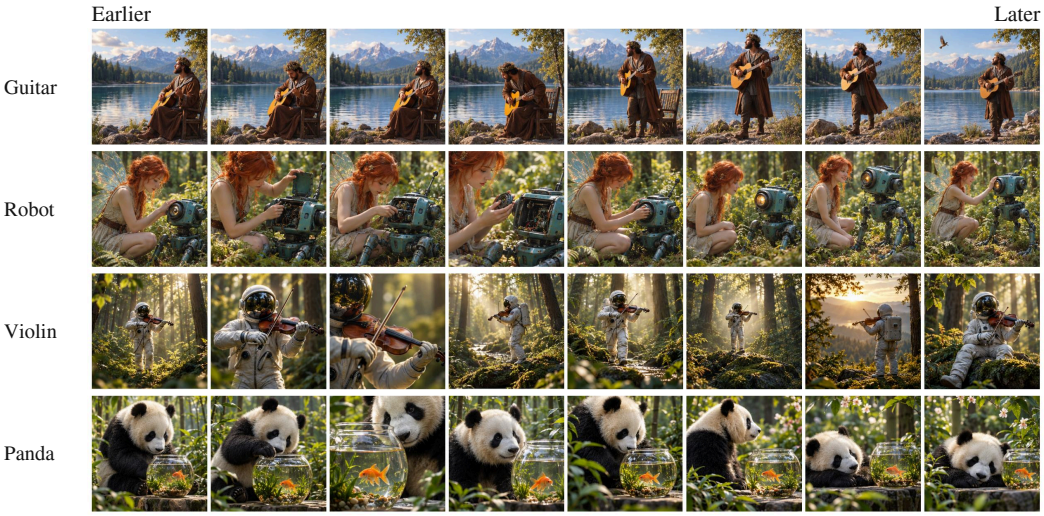


Figure 13: **Four native task contexts.** Guitar playing, robot interaction, violin playing and a panda scene at eight ordered moments. Position, pose and subject–object relations change between distant moments, and the frames between them hold the steps that connect them.



Figure 14: **Dance motion across four samples.** Eight ordered moments per row show changes of pose, orientation and position together with the surrounding scene; each row follows its own dance from earlier to later poses.

Cosmos. We use the post-trained Predict2.5-2B checkpoint (NVIDIA, 2025), which produces 77 frames at 1280×704 and 16 frames per second with 35 steps and guidance 7. Its token grid is $20 \times 44 \times 80$. The bidirectional single-clip mode uses zero or one conditioned latent frame for the two tasks. Its independently trained transformer gives a separate backbone comparison, although its tokenizer shares a model lineage with Wan.

Figure 13 shows four native task contexts in which position, pose and subject–object relations change between distant moments; these are the changes that routes must carry. Figure 14, together with Figure 1 in Section 1, shows generated sequences with displacement, turning, fast pose change and foreground occlusion, the cases in which a route must follow content across frames and stop when it disappears; each row keeps the temporal order of one sample. Benchmark scores always use each complete evaluation set.



Figure 15: **Flower opening across four samples.** Each row is a separate sample for the prompt *A timelapse of a flower bud blooming into a full flower.* Eight ordered moments show petal geometry that changes within each sample while the flower's identity persists, the combination that dynamic-attribute binding scores.

D.2 Data and Optimization

Adaptation uses accessible OpenVid-1M records (Nan et al., 2025). Duplicate sources and confirmed overlapping content are grouped before splitting, and benchmark videos, conditioning images, references and exact normalized test prompts are excluded from the training pool. Source-group hashing assigns 90% of the groups to training, 5% to validation and 5% to an unused reserve, keeping all clips of a source together. Independent hashing assigns each training source to text-to-video or image-to-video, and every batch holds equal numbers of both tasks. Training exposure is reported as presented samples, with source repeats permitted.

The default presents 40,000 samples per root. LTX alternates its two native training grids equally. AdamW uses learning rate 2×10^{-5}, coefficients $(0.9, 0.999)$, weight decay 0.01 and gradient clipping at 1, with 500 warmup steps followed by cosine decay over 10,000 updates. The trainable parameters are the original self-attention query, key, value and output projections, their query–key normalization gains and the PathSpan scalars; other transformer parameters, text encoders and VAEs stay frozen. The three roots are 17, 29 and 43, with paired sample order and input noise, and the final scheduled checkpoint defines each reported condition.

Retention half-lives are initialized logarithmically between one and the latent clip length with $\rho_c = 2^{-1/\lambda_c}$, and default no-match logits start at $\log 121$ (Appendix B.2). Shared retention uses one coefficient per head and direction, initialized from the geometric mean of these half-lives. The retrained radius-one and radius-nine controls use their windows throughout training, with initial no-match logits $\log 9$ and $\log 361$. All controls of Table 3 share the default optimization and data.

D.3 Comparison with Efficient Attention Methods

Table 2 in Section 6.2 reports the six generation suites for every compared operator. Each trained conversion keeps the operator of its method and receives its own study-specific adaptation, and each is matched by a separate PathSpan conversion with the same data, updates, batch and native DiT parameter scope.

Trained conversions. VSA (Zhang et al., 2025a) uses 4^3 cubes, final Top32 selection, batch 64, 4,000 updates and learning rate 10^{-5}. Only attention pads the grid to $32 \times 24 \times 40$, masks invalid tokens and pools by valid counts; after 50 dense updates, the candidate count decreases by ten every 50 updates until Top32. VMoBA (Wu et al., 2026) uses 1D/2D/3D blocks of 8/48/72 tokens, global and cumulative-similarity threshold selection with threshold 0.25, and dense attention for the first 13 of 50 sampling steps, with batch 8, 2,000 updates and learning rate 10^{-5}. SLA (Zhang et al.,

Table 7: Complete category profiles under default matched adaptation (means over three roots). CompBench (Sun et al., 2025): consistent/dynamic attribute binding (CAB/DAB), spatial relationships (SR), motion/action binding (MB/AB), object interactions (OI) and generative numeracy (GN), all multiplied by 100. VMBench (Ling et al., 2025): commonsense adherence (CAS), motion smoothness (MSS), object integrity (OIS), perceptible amplitude (PAS) and temporal coherence (TCS), on 0–100 scales. Bold compares the two methods within each backbone; displayed ties are both bold.

Method	CompBench: all seven compositional categories							VMBench: all five motion dimensions				
	CAB↑	DAB↑	SR↑	MB↑	AB↑	OI↑	GN↑	CAS↑	MSS↑	OIS↑	PAS↑	TCS↑
Wan2.2-TI2V-5B (5B) (Wan-Video, 2025)												
Dense-FT	**73.06**	8.53	62.6	**44.1**	**66.38**	69.87	**52.63**	**64.57**	73.55	**83.68**	**37.5**	94.17
PathSpan	72.77	**8.67**	**63.0**	43.9	66.18	**70.38**	52.39	64.47	**74.08**	83.48	**37.5**	**94.34**
LTX-Video-13B v0.9.8 (13B) (Lightricks, 2025)												
Dense-FT	**74.61**	**9.44**	59.9	41.4	**63.83**	68.14	**53.93**	**62.43**	75.16	**85.16**	32.6	95.14
PathSpan	74.12	9.26	**60.1**	**41.6**	63.60	**68.61**	53.75	62.13	**75.26**	85.07	**33.1**	**95.32**
Cosmos-Predict2.5-2B (2B) (NVIDIA, 2025)												
Dense-FT	71.48	**10.39**	**61.3**	46.6	69.54	**73.18**	**51.25**	**67.59**	72.12	81.60	**40.2**	93.58
PathSpan	**71.78**	10.25	60.6	**47.6**	**69.60**	73.11	50.99	67.26	**72.42**	**81.86**	**40.2**	**93.74**

2026) uses block size 64, selection fractions 0.05 and 0.10 and its softmax feature map, with batch 64, 2,000 updates and learning rate 10^{-5}. VMonarch (Liang et al., 2026) uses two factorization iterations, frame-based factors, a correction clamp of 0.1 and the native first-frame rule, with batch 8, 800 updates and learning rate 10^{-6}; the 121-frame Wan pipeline differs from its public 93-frame setting.

Attention Surgery (Ghafoorian et al., 2026) first generates teacher states for 1,000 training prompts at all 50 sampling steps and both guidance branches. Each of 30 candidate layers receives 4,000 isolated feature-map updates with AdamW at learning rate 10^{-3}, minimizing the L1 error of the value output. Separate biased query and key networks map $128 \to 256 \to 256$ and are partitioned into degree-one and degree-two features. Validation every 500 updates uses 256 disjoint sources, and the 15 layers with the lowest validation L1 error are selected, with ties resolved by block index. Teacher states are reused across candidate layers. Task adaptation then trains the full DiT and feature maps for 1,000 updates at batch 16 and learning rate 10^{-5}, and the matched PathSpan conversion shares the scope and exposure of this task stage.

Training-free operators. SVG2 (Yang et al., 2025a) and DFSAttn (Hu et al., 2026) start from default Dense-FT. SVG2 uses 100 query and 500 key clusters, keeps dense attention for the first 30% of sampling steps and preserves its centroid, cache and permutation operations. DFSAttn uses blocks of 128 with 16-token subblocks; sampling steps 0–11 and the first transformer block stay dense, and masks refresh at steps 12, 24 and 36 with fractions 0.3, 0.2 and 0.1. Step 48 reuses the last valid mask because a further decrement would be nonpositive. This scheduled run is distinct from the fixed-budget test of Appendix A.3. Complete runtime includes each method's selection and reordering costs.

D.4 Evaluation and System Measurement

VBench (Huang et al., 2024) uses all 946 official entries, representing 944 distinct prompts, with its official aggregation. VMBench (Ling et al., 2025) uses all 1,050 prompts and the official mean of its five dimensions. CompBench (Sun et al., 2025) uses 1,400 prompts across seven 200-prompt categories and reports the native category scores. VideoPhy-2 (Bansal et al., 2026) uses 591 prompts and the official 7B automatic evaluator; its joint rate requires both the semantic and the physical score to be at least four on their five-point scales. EvalCrafter (Liu et al., 2024) uses all 700 prompts, with motion classification on the official applicable subset. AIGCBench (Fan et al., 2023) uses all 3,928 image–text pairs; image–video CLIP uses ViT-L/14 and averages over generated frames, DOVER follows the official video-quality evaluation, and DOVER and both CLIP scores are multiplied by 100. Wan restores the conditioned first latent at every sampling step and uses a frozen causal VAE (Wan-Video, 2025), so both attention variants share the same first-frame reconstruction. Figures 15 and 16 show samples of two settings that these suites score: an attribute that changes within a clip, and image-conditioned generation.

Uncertainty. Every generation score is first aggregated within a root by its benchmark's official rule, and tables display the mean over the three roots. Sample standard deviations use denominator two, and paired differences are computed before rounding. Root confidence intervals use the paired t distribution with two degrees of freedom and quantify training-root variability; aggregate-score

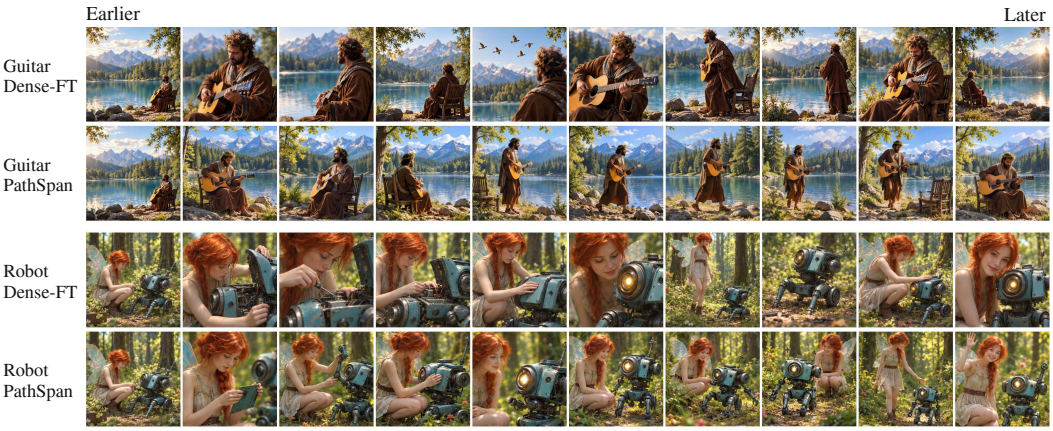


Figure 16: **Matched image-conditioned generation.** Each pair of rows shares the image and text condition, including the first displayed frame, and shows ten corresponding moments of Dense-FT and PathSpan. Table 1 reports complete-set scores.

differences are reported descriptively, and the correspondence analysis separately resamples source videos. Table 7 expands the default models into all seven CompBench and all five VMBench categories.

Timing. Timing uses four A100-SXM4-80GB devices connected by NVLink, with rank-zero text and VAE execution and the native distributed transformer, at batch and concurrency one. Three warmup runs precede complete host-to-host measurements. Synchronization brackets the whole pipeline, including conditioning encoders, guidance branches, attention overhead, communication, upscaling and VAE decoding, and the clock stops when the decoded tensor returns to host memory. Model loading, compilation and compressed-video writing are excluded. Reported maximum footprints take the maximum per-device process memory across the six suites within each root and then average across roots; the bubbles of Figure 2 show the VBench-specific process memory of each operating point.

D.5 Attribute and Action Outcomes

Table 8 separates four questions: whether attributes evolve as requested, whether the action matches the prompt, whether the requested motion category is recognized, and whether optical-flow warping stays consistent. These readouts locate the two losses of Section 6.2. On LTX, requested-motion accuracy falls from 64.7 to 62.7 while the action score moves from 73.61 to 73.64 and the warping error stays at 0.0224; on Cosmos, requested-motion accuracy falls from 69.3 to 66.0 while the action score and warping error are level or better. The loss therefore concerns which motion category the generated video shows, and the quality and consistency of the motion are kept. Figure 15 shows the evolving geometry that dynamic-attribute binding measures, and Figure 16 compares paired Dense-FT and PathSpan generations.

Table 8: Fine-grained attribute/action outcomes (three-root means). CompBench DAB/AB are dynamic-attribute/action binding, multiplied by 100. EvalCrafter MAC/Act. are requested-motion accuracy/action score; Warp is native warping error. Bold compares each backbone pair, including ties.

Method	CompBench		EvalCrafter		
	DAB↑	AB↑	MAC↑	Act.↑	Warp↓
Wan2.2-TI2V-5B (5B) (Wan-Video, 2025)					
Dense-FT	8.53	**66.38**	66.0	**74.66**	0.0241
PathSpan	**8.67**	66.18	**66.7**	74.46	**0.0239**
LTX-Video-13B v0.9.8 (13B) (Lightricks, 2025)					
Dense-FT	**9.44**	**63.83**	**64.7**	73.61	**0.0224**
PathSpan	9.26	63.60	62.7	**73.64**	**0.0224**
Cosmos-Predict2.5-2B (2B) (NVIDIA, 2025)					
Dense-FT	**10.39**	69.54	**69.3**	76.07	0.0270
PathSpan	10.25	**69.60**	66.0	**76.10**	**0.0267**

E Analysis of the Controls and the Response Bound

E.1 Component Task Signatures

Figure 17 expands Table 3 into complete Wan task profiles. Each control has its own signature: fixed address and single hop lower motion binding most, shared retention lowers dynamic-attribute binding,

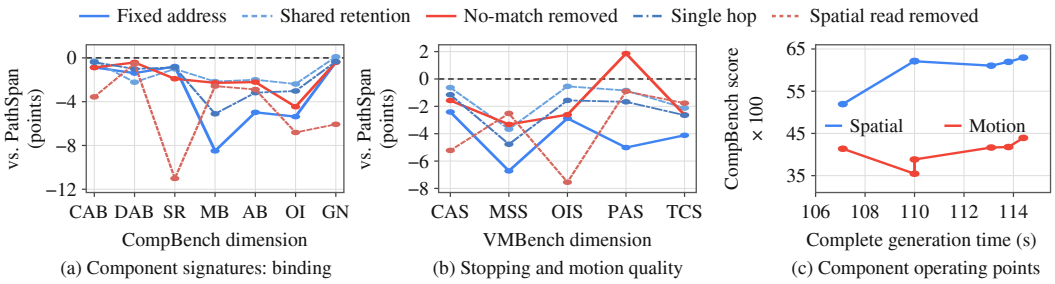


Figure 17: **Each part of PathSpan protects a different aspect of generated behavior.** Five trained Wan controls against matched PathSpan under the default adaptation. (a) Paired score changes on all seven CompBench dimensions (1,400 prompts; native scores $\times 100$). (b) Changes on all five VMBench dimensions (1,050 prompts; 0–100 scale). Zero is PathSpan in both panels. (c) Spatial-relation and motion-binding scores at the six measured complete-generation operating points, including PathSpan, joined in time order. All curves average three paired roots; Table 3 reports the absolute means of the five selected dimensions.



Figure 18: **The within-frame read changes the carried state.** Panels show the observed DAVIS frame, the state before reading, the state after $\mathbf{Y} = \mathbf{AR}$, and the RMS read change. Two RGB-valued scans form the six-coordinate $\mathbf{R} = [\mathbf{H}^+, \mathbf{H}^-]$; triplets are averaged only for RGB display, while the response uses all six coordinates. A row-stochastic RGB/gradient affinity $\mathbf{A}$ reads the fixed $\mathbf{R}$, which is also the no-read control. Color spans 0–0.15 normalized value units.

removing no-match raises amplitude and lowers coherence, and removing the within-frame read lowers spatial relationships by about 11 points while saving time. The time panel pairs each saving with the task dimension that the removed part protects.

Figure 18 shows what the within-frame read does to an already carried state. The largest read changes appear at object boundaries, along the back of the cow and at the fence and water trough, where a query draws on content that arrived at neighboring positions carrying different content.

E.2 Where the Read Comes From

For a future channel c, let $d_{u,j}$ be the diagonal entry of $\mathbf{D}_u$ in Equation (5), with value one at the terminal frame. The contribution of value coordinate (u, j, c) to the read output $Y_{t,i,c}$ is

$$\mu_{t,i,c}(u, j) = \sum_k A_t(i, k)\, \rho_c^{u-t}\, [\mathbf{P}_t \mathbf{P}_{t+1} \cdots \mathbf{P}_{u-1}](k, j)\, d_{u,j}. \tag{13}$$

Summing over j gives the probability $p_{t,i,c}(u)$ that the read comes from frame u, including the current frame; the past direction reverses the product. These probabilities include the within-frame read, the survival of routes and the injection of values, so they sum to one over the actual source frames, with the temporal boundary absorbing the remaining mass.

The readout of Figure 5 uses the last strictly positive-noise call and the final video-attention block, keeps every head, channel, direction and guidance branch, and averages spatial queries. The controls share prompts and initial noise with the full model. Retention uses 30 density bins over $[0.45, 1]$ pooled over heads and roots; terminal curves keep all 64 channels in the PathSpan-retention order.

Figure 19 shows the mechanism behind the stop result of Section 6.3 at the level of a single scan. The upper group shows where the no-match entry takes mass and how the state keeps local values there; the lower group forces every step to continue and shows how far the state then departs from the stopping version.

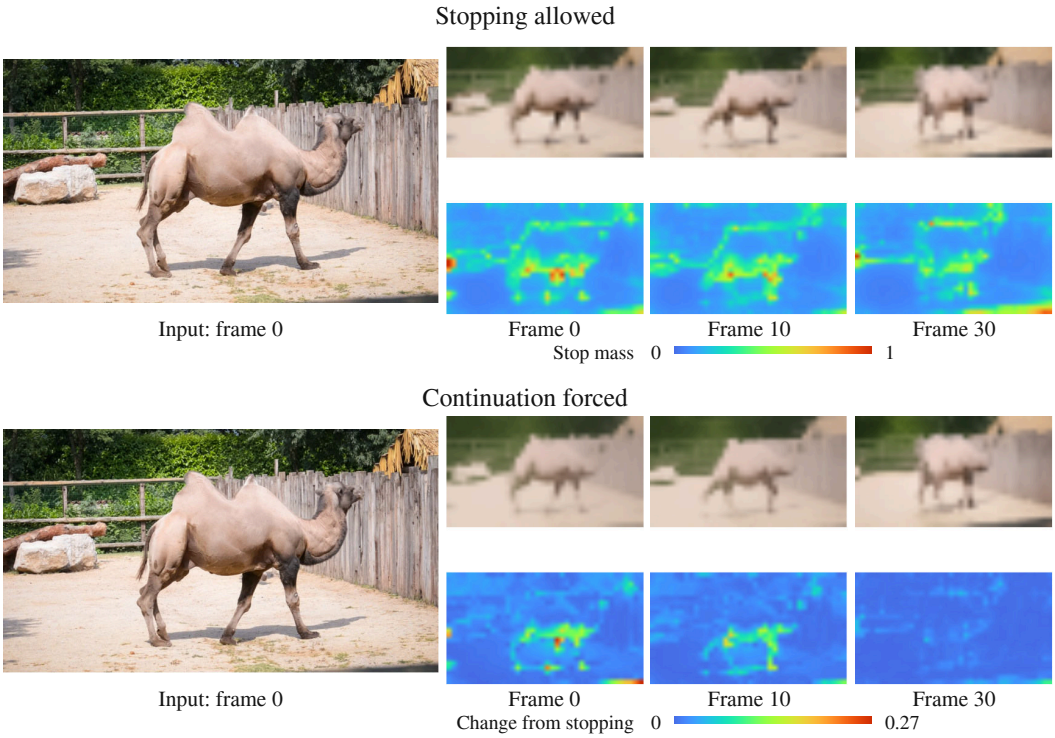


Figure 19: **Stopping changes the value mixture within the scan.** The upper group allows stopping; the lower group forces continuation. Both use the same DAVIS RGB clip, with frame 0 as context. In each group, the upper fields at frames 0, 10 and 30 show RGB-valued states at fixed $\rho = 0.92$, and the lower fields show the stop/current-value mass (0–1) or the magnitude of the state difference (0–0.27). The control divides each map row by its continuation mass before scanning. The scans follow Equation (4) with RGB values and descriptor-based transition weights.

Figure 20 separates how widely the read spreads over time from how concentrated it is in space. Joint time–space coverage can be broad while the spatial mass conditioned on one source frame stays concentrated, so a channel can read far in time and still read a small spatial region of each frame.

E.3 Response Prediction

Responses. At the correspondence feature block, radius-one and radius-nine operators replace the radius-five operator at a common input. The state response is the MSE over all coordinates of $\mathbf{H}$. The DiT response propagates this single-block change through the unchanged remainder of the DiT and measures the MSE of the final DiT output at the same input; it differs from changing every block throughout complete sampling.

Predictors. The bound predictor uses $U = N_H^{-1} \sum_{k=1}^{N_H} B_k^2$, where B_k is the directional bound of Equation (7) for scalar state coordinate k. The competing predictor uses the absolute change of mean point accuracy for the same source. Each predictor fits an intercept and a slope by source-normalized least squares on all 131 PointOdyssey v1.2 training videos (Zheng et al., 2023), separately for each backbone and response, with native 121- or 77-frame windows, equal weight per source and negative predictions clipped to zero; no DAVIS source enters the fit.

Test. The frozen predictors are evaluated on the 30 DAVIS videos, each with three roots and two window changes, giving six paired observations per video. Prediction MSE averages these observations within a video and then across videos. Paired bootstrap draws resample videos with all their observations. U bounds the common-input state MSE by Proposition 1; for the downstream DiT response it is an explanatory variable of an empirical model. Prediction-error differences have units of squared MSE, and Wan's DiT reduction of 3.98×10^{-6} has the source-bootstrap interval $[1.37 \times 10^{-6}, 7.41 \times 10^{-6}]$. Figure 21 shows the bound and the observed change for one example: the bound is valid everywhere and loose by a spatially varying factor, so its value as a predictor comes from ranking responses correctly, which Figure 6 tests.

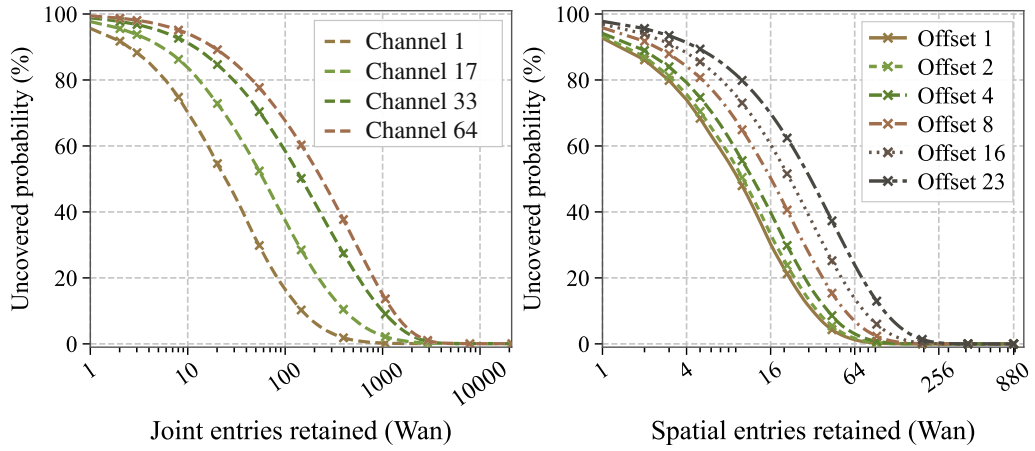


Figure 20: **Temporal spread and spatial concentration are different properties.** Wan future-direction endpoint probabilities on DAVIS goat, source row zero, averaged over 24 heads and three roots. Left: uncovered joint probability $100(1 - \sum_{j=1}^{k} p_j^{\downarrow})$ for channels 1, 17, 33 and 64, for every $k = 1, \ldots, 21{,}120$. Right: the same complement for channel 33 conditioned on offsets 1, 2, 4, 8, 16 and 23, for every spatial count $k = 1, \ldots, 880$. Sorting is performed within each probability domain before aggregation; logarithmic horizontal axes omit only $k = 0$.

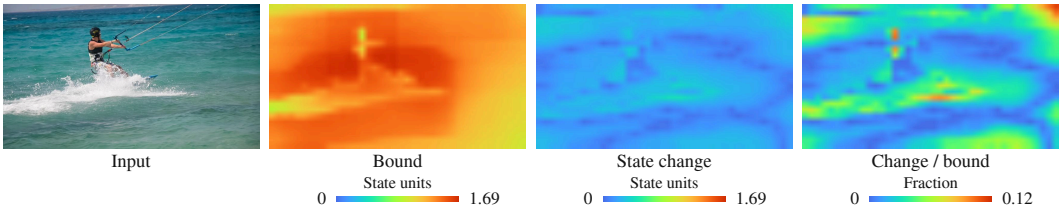


Figure 21: **Common-input change and its response bound.** Panels show the DAVIS input, the bound B, the absolute change $|H - \widetilde{H}|$ and their ratio. Radius five is restricted to radius one with shared RGB-luminance values and fixed descriptor and scalar parameters. In Equation (7), total variation includes stopping and the value range includes the current value. Bound and change share a square-root display with raw 0–1.69 endpoints; the ratio spans 0–0.12 and reaches about 0.113.

From one block to complete sampling. Figure 22 follows fixed-weight interventions through three stages on CompBench: the state of the changed block, the DiT output, and the final latent after all denoising updates and guidance. Each error is divided by the donor error of the same readout, so equal normalized heights express equal relative effects within a readout while the absolute scales stay separate.

E.4 Remaining Distance and Unit Response

The remaining frame count L_{rem} measures how far a perturbation can still propagate, while the local values decide the contrasts it meets on the way. Figure 23 separates the observed unit response, the adjusted frame effect and the independent prediction. The profiles are nonmonotone and the adjusted terms take both signs, so distance alone does not determine sensitivity: a longer distance adds possible transitions, and their effect still depends on survival and on the value contrasts along the route.

F Learning PathSpan from Random Initialization

Action-conditioned Minecraft prediction with DFoT's DiT-B (Song et al., 2025) (12 blocks, width 768, 12 heads) tests whether the native objective alone learns PathSpan. Random and released-checkpoint initializations share the frozen image VAE and action conditioning and train all DiT parameters and PathSpan scalars.

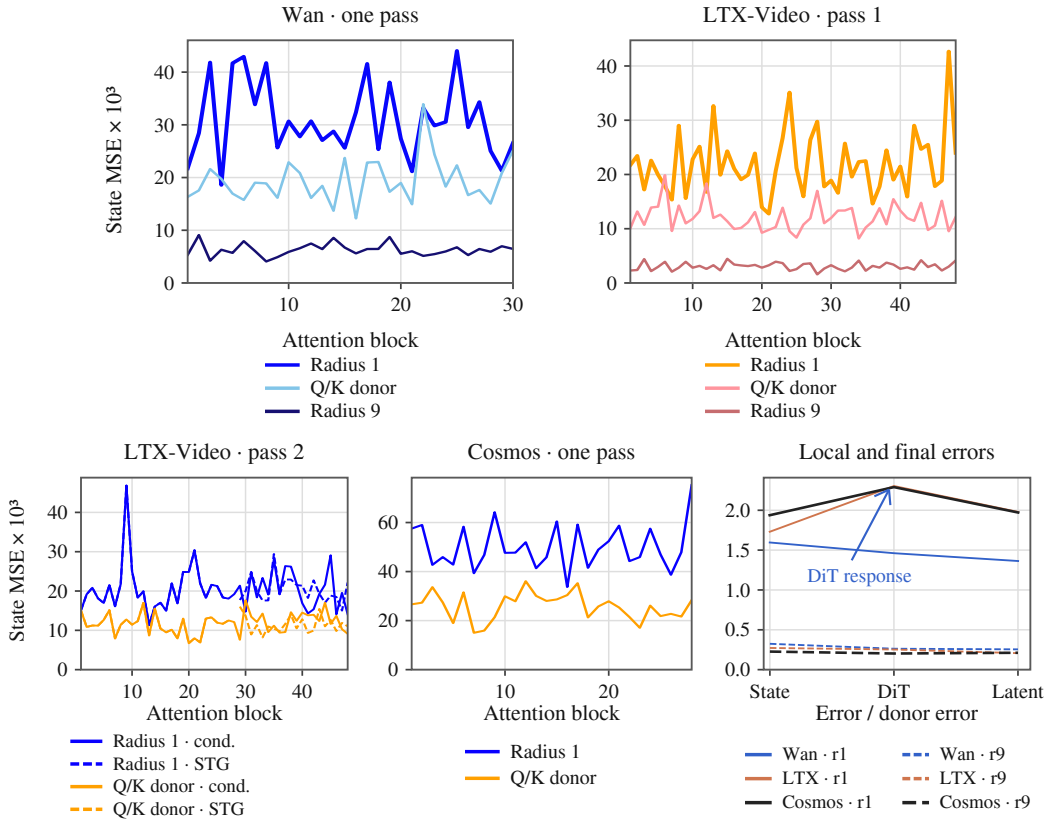


Figure 22: **State, DiT and final-latent responses measure different stages.** Fixed-weight interventions on CompBench, averaged over three roots. The first four panels show state MSE $\times 10^3$ in block order for Wan (30 blocks), LTX pass 1 (48), LTX pass 2 (48) and Cosmos (28), for the radius-one, radius-nine and donor-Q/K conditions; pass 2 separates conditional and STG branches, and the STG trace has a gap at block 28, whose native value bypass does not use the attention output. The last panel reports intervention error divided by donor error for the state, the DiT output and the final latent, with combined native guidance for the final latent.

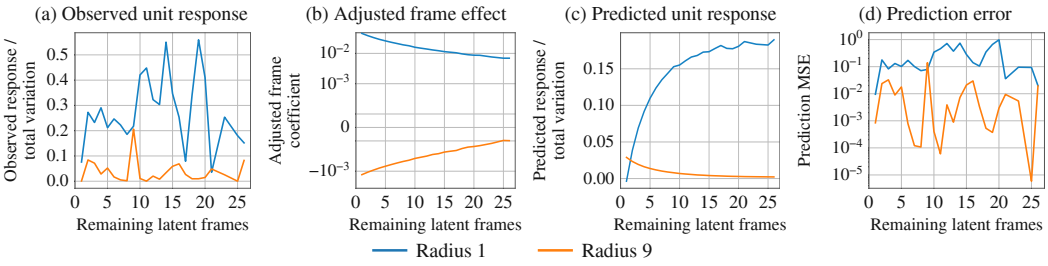


Figure 23: **Sensitivity depends on the transitions along the remaining distance.** Wan radius-one restriction and radius-nine expansion on 30 DAVIS videos and three roots. (a) Observed directional response divided by the total variation of the perturbation; (b) adjusted remaining-frame coefficient; (c) unit response predicted by the independently calibrated six-coefficient affine model; (d) squared prediction error. All 24 observed nonzero remaining-frame values are kept. Each radius aggregates 8,418,816 valid head–source–channel–root elements (24 heads, 64 channels). At $L_{\mathrm{rem}} = 0$ the perturbation is zero and the ratio undefined, so the plotted domain contains positive distances.

Training. Training uses 193,795 sources (256 held out for monitoring), 50-frame 256×256 windows at stride two, independent history noise levels and native weighted velocity prediction. AdamW uses learning rate 10^{-4}, weight decay 0.001, batch 96, FP16 and moving-average decay 0.9999, with 10,000 warmup updates; checkpoints at 100,000 and 200,000 updates are evaluated.



Figure 24: **Four action-conditioned Minecraft rollouts.** Each row follows a separate clip at frames 24, 37, 49, 74, 99, 124, 137 and 149. Frame 24 is the last observed frame; frames 49, 74, 99, 124 and 149 end the five 25-frame prediction windows, and each window passes its generated history to the next. Table 9 evaluates all frames of the complete 150-frame sequences.

Evaluation. The official 400-dimensional I3D procedure evaluates the complete 150-frame sequences, including the observed history. Table 9 reports both checkpoints. The random start improves from 159.5 to 109.9 between the two budgets while the pretrained start moves from 101.4 to 93.9, so the gap shrinks from 58.1 to about 16 FVD; the unrounded final gap of 15.9 has the paired three-root interval $[-18.3, 50.2]$, which includes zero. The native objective therefore learns PathSpan from random DiT parameters, and the remaining gap depends on the training budget.

Table 9: Complete Minecraft prediction from two DiT initializations (three-root mean FVD). Both use the same frozen VAE and new-training protocol. Evaluation covers 150-frame sequences from all 2,084 test sources. Bold compares initializations at each update budget.

DiT initialization	**Minecraft FVD ↓**	
	100k updates	200k updates
DFoT DiT-B (Song et al., 2025); shared frozen image VAE		
Random	159.5	109.9
Pretrained	**101.4**	**93.9**

Rollouts. Two starts in each of the 2,084 test sources give 4,168 clips; each predicts 125 frames from 25 observed frames through five windows with generated history, using 50 DDIM steps per window (Figure 24).