CAUSALFOLD: LETTING LATER VIDEO BLOCKS READ AND TRAIN EARLIER ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive video diffusion models generate a video one latent block at a time, and each new block attends to earlier blocks through a key–value (KV) cache. In every attention layer, the cache returns the Values of earlier tokens, which are projections of their input to that layer. The output that an earlier token computes in the same layer, for example the corridor content that a doorway token selects with its own Query, reaches later blocks only through the next layer, and a later block's loss reaches that attention only through deeper layers. We introduce CausalFold, which keeps each cached Key and stores the token's attention output in place of its Value. A token then reads the Values of its own block, whose tokens are denoised jointly, and the outputs of earlier blocks, which are already final. The result is a recursion over blocks that ends after finitely many reads, without truncation or discounting. Each output remains a convex combination of Values, and the Jacobian of a later output with respect to an earlier token's attention logits is the probability that attention paths visit that token times the differences among its candidates. CausalFold adds no parameters and keeps each backbone's masks, sampler, and cache schedule. Against Standard-KV controls with identical training on five backbones, it raises transition completion by 2.3 to 4.9 points in all six text- and image-to-video comparisons, including from 37.9 to 42.7 for text-to-video on Wan2.1-1.3B at 5.4% longer generation time. In image-to-video, the trained model depends on the cached outputs, and removing the later-block gradient on earlier attention removes about half of the gain.

1 INTRODUCTION

Autoregressive video diffusion models generate a video as a sequence of latent blocks (Figure 1). Each block covers one or a few latent frames and is denoised in one to eight network function evaluations (NFE) while it attends to visible earlier blocks through a key–value (KV) cache; once the block is clean, a cache-writing call stores its Keys and Values for the blocks that follow (Yin et al., 2025; Huang et al., 2025; Sand.ai et al., 2025). Distillation and training on self-generated rollouts have made this loop fast enough for streaming and interactive generation (Zhu et al., 2026; Yang et al., 2026b; Li et al., 2026; Liu et al., 2026; Yang et al., 2026a). Temporal requests remain difficult: a camera has to finish a turn it has started, a cake has to keep changing color while it rotates, and a girl has to put the basket she has just filled on the table. On TC-Bench, which checks whether a prompted transition reaches its end state, the published few-step generators built on Wan2.1-1.3B complete 31.4% to 37.5% of transitions (Table 1). In each of these requests, what the current block draws depends on a relation among earlier blocks, and that relation reaches the current block through the cache.

Consider a camera that moves down a corridor and starts to turn right. Block $t-2$ shows the corridor, block $t-1$ shows a doorway at the start of the turn, and block t has to continue the turn. In some attention layer, the doorway tokens of block $t-1$ use their Queries to select corridor tokens, and their attention outputs combine the doorway content with the selected corridor content. The cache, however, stores the doorway's Values, which are projections of the doorway tokens' input to that layer. Block t can attend to the corridor itself, but its Queries select corridor tokens according to what block t contains; the corridor tokens that matter for the doorway are selected by the doorway's Queries, and only the doorway's attention output carries that selection. This output reaches block t one layer later, through the doorway tokens' residual stream and next-layer Values, and the output

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.

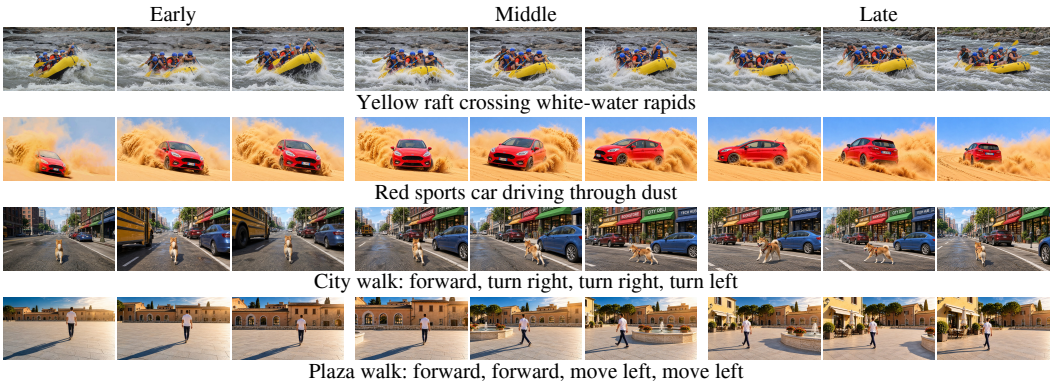

Figure 1: **CausalFold rollouts in video generation and camera-controlled navigation.** Early, middle, and late views of four videos: a raft crossing rapids, a car driving through dust, and two camera-controlled walks whose commands are listed below the rows. Each row follows one scene across many generated blocks.

of the top layer is never read. A chain of relations across k earlier blocks therefore needs k layers to reach block t. Training follows the same route: the gradient of block t's loss reaches the doorway's attention in a layer only through deeper layers.

We propose CausalFold, which folds each token's attention output back into the cache in place of its Value and keeps its Key. A token keeps its attention weights and changes what they are applied to: the Values of tokens in its own block and the cached outputs of tokens in earlier blocks. This division follows the order of generation. Tokens in the current block are denoised jointly, so their outputs are computed together; tokens in earlier blocks finished their outputs when those blocks were written to the cache. Every read therefore refers to a quantity that already exists, each chain of reads visits every earlier block at most once, and the change adds no parameters. In the corridor example, block t now reads the doorway output, which contains the corridor content selected by the doorway's Queries, and block t's loss reaches the doorway's attention weights in the same layer.

Two results describe what the change does to the forward pass and to the gradients. Each cached output is a convex combination of Values (Theorem 1), so cached outputs stay in the range of the Values that a pretrained generator already reads; we convert each generator with 200 fine-tuning updates. The Jacobian of a later output with respect to an earlier token's attention logits equals the probability that attention paths from the later token visit the earlier one, multiplied by the differences among the candidates that the earlier token chooses from (Theorem 2). Both factors depend on how a backbone writes its cache: MAGI-1 writes each block with attention restricted to that block, which sets the visit probability between cache-writing calls to zero.

We compare CausalFold with a Standard-KV control that starts from the same checkpoint and receives identical data, sampler, updates, and gradient path through the cache, on Wan2.1 (1.3B and 14B), MAGI-1 (4.5B), HunyuanVideo-1.5 (8B), and EchoWM (21.81B), across text-, image-, and camera-conditioned generation. Transition completion rises by 2.3 to 4.9 points in all six matched text- and image-to-video comparisons, most in categories where an attribute or a motion has to change over time. VBench quality and scene-preservation scores change by at most 0.3 points, and dynamic degree is 0.4 to 0.7 points lower. In image-to-video, two interventions separate the two effects: restoring cached Values in the trained model drops completion below Standard-KV, and removing the later-block gradient on earlier attention removes about half of the gain.

2 Related Work

Autoregressive video diffusion. CausVid distills a bidirectional video model into a block-causal generator with a KV cache, and Self Forcing trains such a generator on its own rollouts (Yin et al., 2025; Huang et al., 2025). Causal Forcing improves the distillation, and adversarial self-distillation and AAD-1 reduce generation to about one evaluation per block (Zhu et al., 2026; Yang et al., 2026b; Li et al., 2026); MAGI-1 scales chunk-wise generation, and Diffusion Forcing assigns independent noise levels along a causal sequence (Sand.ai et al., 2025; Chen et al., 2024). These works define training and sampling schemes; the transformer generators among them read history through cached

Keys and Values. CausalFold keeps those generators' recipes and samplers and changes what a cached entry holds.

History for long rollouts. Rolling Forcing and LongLive decide which history stays in the cache and how long rollouts are trained (Liu et al., 2026; Yang et al., 2026a). Self Gradient Forcing restores gradients to recomputed context Keys and Values (Zhuang et al., 2026), and SANA-Video replaces softmax attention with block linear attention whose cache holds cumulative Key–Value sums in constant memory (Chen et al., 2026). In the softmax designs, a cached token returns content formed before its attention in the layer. CausalFold keeps the softmax and every cached Key, and returns the token's attention output.

Reading computed states of earlier positions. The Feedback Transformer notes that a layer of a causal transformer reads only lower-layer states of earlier positions, and exposes a learned mixture of all their layers at the cost of processing positions one at a time (Fan et al., 2020). Bai et al. (2026) let later frames read causal representations of earlier frames computed apart from denoising, the Recurrent Transformer attends to its own earlier activations for greater effective depth (Oncescu et al., 2026), and ChaCAL and Infinite Self-Attention compose attention over several hops through discounted or damped attention powers (Fagnou et al., 2024; Roffo et al., 2026). CausalFold reads the same layer's outputs of earlier blocks, which a block-causal generator has already computed when it wrote those blocks to the cache. Tokens within a block remain parallel, and the block order ends every chain of reads after finitely many steps, so the composition is exact without discounting.

3 What a Video KV Cache Returns

3.1 Block-Causal Attention

A block-causal generator denoises the latent block x_t from noise level σ given the committed blocks $x_{<t}$ and the conditions c_t, which are text, a first image, or camera actions. We first write attention for a packed sequence whose tokens are arranged in ordered blocks $\mathcal{B}_1, \ldots, \mathcal{B}_S$, where $\mathcal{B}_t$ holds the tokens of block t; Section 4.2 gives the streaming form, in which earlier blocks come from the cache. For one layer and one head, let $X \in \mathbb{R}^{N\times d}$ be the layer input, $Q = XW_Q$ and $K = XW_K$ in $\mathbb{R}^{N\times d_k}$, and $V = XW_V \in \mathbb{R}^{N\times d_v}$. With the backbone's attention mask M,

$$A = \mathrm{softmax}_{\mathrm{row}}\big(QK^\top/\sqrt{d_k} + M\big) = D + L, \tag{1}$$

where the block-diagonal part D holds attention within a block and the strictly block-lower part L holds attention to earlier blocks. Thus D_{tt} is the full attention among the tokens of block t, and L_{ts} with $s < t$ is its attention to block s. A lowercase letter denotes a matrix row as a column vector, e.g., $v_i = V_{i,:}^\top$; layer and head indices are omitted.

3.2 Earlier Attention Reaches Later Blocks One Layer Late

Standard attention outputs $Y = AV$, that is, $Y_t = D_{tt}V_t + \sum_{s<t} L_{ts}V_s$, with K_s and V_s of the earlier blocks read from the cache. A later block therefore receives v_p from an earlier token p, a projection of p's input to the layer. The output y_p, which combines v_p with the history that p selects with its Query q_p, continues along p's residual stream. Two consequences follow.

Relations formed by earlier tokens arrive one layer late. A relation that p forms in layer l becomes readable in layer $l+1$ as part of p's Value, and relations formed in the top layer are never read. A chain of relations across k earlier blocks therefore needs k layers to reach a later block.

Later losses do not reach earlier attention in the same layer. Let z_p be the attention logits of p. With V, M, and all other attention rows fixed, $\partial y_i/\partial z_p = 0$ for every token i in a later block. How p attends in a layer receives a later block's gradient only through deeper layers, and in the top layer it receives none.

Both consequences call for one change: a cached entry should return the attention output of its token. The change should keep the Keys, softmax, mask, cache schedule, and parameters of the pretrained generator, and follow the order in which the generator computes, so that every read refers to an output that already exists.

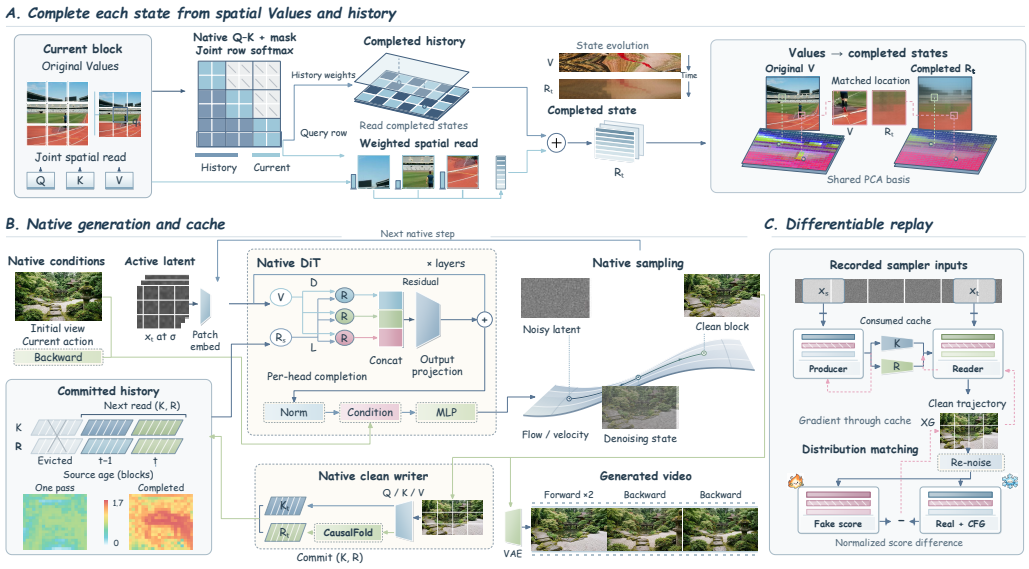

Figure 2: **CausalFold inside the generation loop.** (A) A token of the current block applies the backbone's attention weights to the Values of its own block and to the cached outputs R of earlier blocks (labeled completed states). (B) Conditions, sampler, output projection, and the cache-writing call stay native; the cache stores (K_t, R_t). (C) Training replays the recorded rollout with gradients, so the distribution-matching loss reaches every cache-writing call through the cached reads.

4 CausalFold

4.1 Values Within a Block, Outputs Across Blocks

CausalFold keeps the attention weights A of Equation (1) and changes what they are applied to. A token in block t reads the Values of its own block and the attention outputs of earlier blocks:

$$\boxed{R_t = D_{tt} V_t + \sum_{s<t} L_{ts} R_s.} \tag{2}$$

Here $R_t \in \mathbb{R}^{|\mathcal{B}_t| \times d_v}$ is the CausalFold output of block t before head concatenation and the output projection, and the cache stores (K_t, R_t) in place of (K_t, V_t). The two terms follow the order of generation. The tokens of block t are denoised jointly and attend to one another in both directions, so their outputs are computed together; the first term reads their Values, exactly as the backbone does. Earlier blocks finished their outputs when they were written to the cache; the second term reads those outputs at the same Keys and with the same weights that the backbone uses for their Values. In the corridor example, block t now receives the doorway output r_p, which contains the corridor content selected by the doorway's Query.

In matrix form, $R = DV + LR$. Each factor of L moves to a strictly earlier block, so $L^S = 0$ and

$$R = WV, \qquad W = PD, \qquad P = (I - L)^{-1} = \sum_{k=0}^{S-1} L^k. \tag{3}$$

W and P have a path interpretation. Starting at token i, choose a candidate j with probability A_{ij}; if j lies in the same block as the token the path is at, stop and return v_j, and otherwise move to j and choose again with the weights of j. W_{ij} is the probability of stopping at j, and P_{ip} is the probability of visiting p, which we call the visit probability. Standard attention stops after the first choice wherever it lands, which gives $Y = AV$; CausalFold follows each path until it ends at a Value. Forward substitution in block order evaluates Equation (2) with the backbone's projections and mask.

Theorem 1 (Exact finite solution)**.** *Let $A \geq 0$ with $A\mathbf{1} = \mathbf{1}$. Equation (2) has the unique solution $R = WV$ of Equation (3), with $W \geq 0$ and $W\mathbf{1} = \mathbf{1}$. For fixed A, $\|WV - WV'\|_{\infty,2} \leq \|V - V'\|_{\infty,2}$, where $\|Z\|_{\infty,2} = \max_i \|Z_{i,:}\|_2$.*

Every cached output is therefore a convex combination of Values, so reading across many blocks keeps cached outputs in the range of the Values that the pretrained layers already read. Appendix A gives the proof. Appendix B bounds $\|r_i - y_i\|_2$ by the probability that a path from i moves to an earlier block, times the change in where those paths end when they continue, times the largest distance between two Values.

4.2 Inside the Generator

CausalFold changes only the Value tensor of each attention call (Figure 2). Each head's R passes through the original head concatenation, output projection, residual branches, and later layers, and the denoiser returns its flow or velocity prediction to the backbone's sampler. In streaming generation, block t attends over $[K_{<t}; K_t]$ with the Value tensor $[R_{<t}; V_t]$, which is the backbone's own attention call with cached outputs in place of cached Values. When block t is committed, the backbone's cache-writing call (the clean writer of the released protocol) computes R_t in the same way and stores (K_t, R_t); eviction follows the backbone's window rule. Where one call holds several active blocks, as in MAGI-1's overlapping chunks, they are evaluated in block order. Condition tokens that share the softmax with video tokens, such as HunyuanVideo's text and image prefix, form an earlier block whose outputs the video tokens read, and EchoWM applies the change before its per-head output gate (Appendix F.2).

The cache-writing call determines how much history a cached output contains. Wan2.1 writes each block with attention to earlier cached entries, so its R_t composes all earlier blocks. MAGI-1 writes each block with attention restricted to that block and stores $R_t = D_{tt}V_t$: later blocks read the block's spatial aggregation in place of V_t, and no composition across written blocks takes place. Section 5.3 uses this difference to test the analysis. Storage also changes under grouped-query attention. Per cached token and layer, Standard-KV stores $H_{kv}(d_k + d_v)$ scalars and CausalFold stores $H_{kv}d_k + H_q d_v$, because each of the H_q query heads produces its own output. The two are equal under multi-head attention, and CausalFold's cache is twice as large for MAGI-1, which has 24 query heads and 8 Key/Value heads with $d_k = d_v$.

4.3 Training: Later Losses Reach Earlier Attention

Because each R_s is read by later blocks, its gradient collects contributions from those reads. Let G be the gradient of the loss ℓ that reaches R through the layer's output branch alone. The total gradient $U = \partial\ell/\partial R$ adds the contributions of all later reads and satisfies the transposed recursion

$$U = G + L^\top U, \qquad U = P^\top G, \tag{4}$$

which reverse substitution evaluates from the last block to the first; reads in later model calls add their contributions to U in the same way (Appendix E.2). The two terms of Equation (2) then give $\nabla_V \ell = D^\top U$ and the attention-weight gradients $\Pi_D(UV^\top)$ and $\Pi_L(UR^\top)$, where Π_D and Π_L keep the within-block and earlier-block entries (Appendix D, Algorithm 1). For the attention of an earlier token, this gradient takes the following form.

Theorem 2 (Gradient on an earlier token's attention)**.** *Fix V, M, and every attention row except row p. Let z_p and a_p be the logits and softmax weights of p on its unmasked candidates, and let C_p stack the rows $c_{pj}^\top$ that p reads: $c_{pj} = v_j$ for candidates in its own block and $c_{pj} = r_j$ for candidates in earlier blocks. With $\Delta_p = C_p^\top \left(\mathrm{diag}(a_p) - a_p a_p^\top\right)$ and $u_p = U_{p,:}^\top$,*

$$\frac{\partial r_i}{\partial z_p} = P_{ip}\,\Delta_p, \qquad \nabla_{z_p}\ell = \Delta_p^\top u_p, \qquad 0 \le P_{ip} \le 1. \tag{5}$$

Column j of Δ_p equals $a_{pj}(c_{pj} - r_p)$, the weighted difference between candidate j and the output of p. Hence $\partial r_i/\partial z_p \neq 0$ exactly when attention paths from i visit p ($P_{ip} > 0$) and the candidates of p differ ($\Delta_p \neq 0$). Under standard attention the same derivative is zero (Section 3.2). The strength of this training signal therefore varies across backbones with the two factors, and it vanishes between cache-writing calls that attend only within their own block. Section 5.3 measures both factors.

We train with the sample-normalized distribution-matching loss of Self Forcing on the generator's own rollouts, using each backbone's own score models (Yin et al., 2024b;a; Huang et al., 2025). A rollout is first recorded without gradients. The update then replays, with the same parameters and

Table 1: Text-to-video quality and composition across five benchmarks. Entries are means; higher is better. Scores are percentages except VideoPhy-2 semantic adherence (SA) and physical commonsense (PC), which use the native 1–5 scale. TC denotes TC-Bench; Story denotes StoryEval. CompBench columns are consistent attribute, dynamic attribute, spatial, action, motion, object-interaction, and numeracy scores. Bold marks the best displayed value within each backbone group, including ties. CausVid (Yin et al., 2025), Self Forcing (Huang et al., 2025), LongLive (Yang et al., 2026a), Rolling Forcing (Liu et al., 2026), ASD (Yang et al., 2026b), and Causal Forcing (Zhu et al., 2026) are native-system comparisons; only Standard-KV/CausalFold pairs are matched. NFE excludes clean writers; ASD uses four evaluations for the first chunk and one thereafter (1*).

Method	NFE	VBench		TC	VideoPhy-2			Story	T2V-CompBench						
		Qual.	Sem.	TCR	SA	PC	Joint	Comp.	Attr.	Dyn.	Spat.	Act.	Mot.	Inter.	Num.
Wan2.1-T2V, 1.3B (Wan et al., 2025): native external systems and the matched pair															
CausVid (Yin et al., 2025)	4	82.1	72.3	31.4	3.41	3.17	22.8	28.1	59.6	37.8	49.9	57.0	41.2	58.9	34.3
Self Forcing (Huang et al., 2025)	4	83.8	**82.0**	35.6	3.77	3.44	26.5	31.7	**68.0**	40.7	55.1	59.3	44.9	67.3	**42.9**
LongLive (Yang et al., 2026a)	4	83.4	77.5	36.8	3.64	3.45	27.2	32.4	65.1	41.4	54.7	59.5	45.3	64.7	39.1
Rolling Forcing (Liu et al., 2026)	5	83.1	73.8	35.1	3.67	3.45	25.9	32.0	63.9	40.1	54.2	59.0	44.0	65.8	38.0
ASD (Yang et al., 2026b)	1*	81.9	81.3	36.1	**3.78**	3.48	25.4	31.8	65.1	40.9	**56.4**	59.8	44.8	**67.9**	42.2
Causal Forcing (Zhu et al., 2026)	4	84.0	79.9	37.5	3.70	3.43	28.1	33.2	66.6	42.2	55.7	60.5	46.2	67.6	41.2
Standard-KV	4	84.2	80.1	37.9	3.68	**3.50**	28.4	33.8	66.3	42.5	55.8	60.7	46.8	67.3	41.7
CausalFold	4	**84.5**	80.2	**42.7**	3.72	3.40	**31.9**	**39.0**	66.6	**46.9**	56.0	**63.0**	**51.1**	66.6	41.4
MAGI-1 shortcut, 4.5B (Sand.ai et al., 2025): matched pair															
Standard-KV	8	84.8	74.2	43.8	**3.55**	3.17	33.9	39.6	**62.5**	46.8	51.6	64.1	52.3	60.7	**36.8**
CausalFold	8	**84.9**	**74.3**	**46.2**	3.54	**3.21**	**35.4**	**42.3**	61.9	**48.4**	**52.6**	**65.2**	**54.0**	**61.1**	36.7

the recorded sampler inputs, every call whose output reaches the loss, including the cache-writing calls that produced each cached entry, so the loss gradient passes through every read of the cache (Appendix F.1). Standard-KV is trained with the identical replay; the two models receive gradients through the cache in the same calls and differ only in what a cached read returns.

5 Experiments

5.1 Setup and Predictions

Backbones and matched pairs. We apply CausalFold to Wan2.1 at 1.3B and 14B, MAGI-1 shortcut at 4.5B, HunyuanVideo-1.5 TI2V at 8B through minWM, and EchoWM Flash at 21.81B (Wan et al., 2025; Sand.ai et al., 2025; Wu et al., 2025; Zhao et al., 2026; Zhang et al., 2026). For each backbone, Standard-KV and CausalFold start from the same checkpoint and share data, sampler, update schedule (200 generator updates), three paired training seeds, and the replayed gradient path; a cached read returns V in one model and R in the other. Published systems appear with their own protocols as reference points.

Tasks and benchmarks. Text-to-video tests whether a prompted change is carried out (TC-Bench, StoryEval, T2V-CompBench, VideoPhy-2, VBench). Image-to-video adds a given first frame whose scene must persist (VBench-I2V, TC-Bench-I2V). Camera-controlled navigation tests successive online actions (WBench) (Feng et al., 2024; Wang et al., 2025; Sun et al., 2025; Bansal et al., 2026; Huang et al., 2024; 2026; Ying et al., 2026). TCR denotes the TC-Bench transition completion rate. Tables report means of three seed-level benchmark aggregates; Appendix H gives splits, eligibility rules, and statistical units.

Predictions. Section 4 leads us to expect that (P1) gains concentrate on metrics that relate the current block to earlier ones while VBench quality changes little, (P2) the trained model depends on the outputs it reads, (P3) part of the gain comes from the later-block gradient on earlier attention, and (P4) the size of that gradient depends on visit probability, candidate differences, and the cache-writing call. Section 5.2 tests P1 and Section 5.3 tests P2–P4.

5.2 Gains Concentrate on Temporal Change

Text-to-video. On Wan2.1-1.3B (Table 1), TCR rises from 37.9 to 42.7 and StoryEval completion from 33.8 to 39.0. Within T2V-CompBench, the largest gains appear where an attribute or a motion has to change over time and stay bound to one entity: dynamic attributes (+4.4), motion binding (+4.3), and action binding (+2.3). Object interaction, which also involves change over time, decreases by 0.7. The categories that score a static configuration change by at most 0.3 points (con-

Table 2: Image-to-video transition completion and scene preservation. All entries are mean percentages, higher is better; subject/background denote consistency. Bold compares methods within each backbone group, not across backbones. External systems retain native protocols: CausVid (Yin et al., 2025), Self Forcing (Huang et al., 2025), Rolling Forcing (Liu et al., 2026), Causal Forcing (Zhu et al., 2026), and one-step AAD-1 (Li et al., 2026). Dashes mark three CausVid preservation measurements not supplied with the reported I2V results; no values are inferred.

Method	**NFE**	**VBench-I2V**					**TC-Bench**
		Total	Dynamic	Subject	Background	Smoothness	TCR
Wan2.1, 1.3B (Wan et al., 2025): native systems and matched pair							
CausVid (Yin et al., 2025)	4	88.4	59.2	–	–	–	38.5
Self Forcing (Huang et al., 2025)	4	89.0	61.8	92.0	93.8	98.6	42.7
Rolling Forcing (Liu et al., 2026)	5	88.8	60.9	94.8	95.7	98.7	42.0
Causal Forcing (Zhu et al., 2026)	4	89.3	62.5	95.3	**96.2**	**98.8**	44.1
Standard-KV	4	89.4	**62.8**	96.4	95.1	98.2	44.6
CausalFold	4	**89.7**	62.2	**96.5**	95.3	98.3	**49.5**
Wan2.1, 14B (Wan et al., 2025): AAD-1 and matched pair							
AAD-1 (Li et al., 2026)	1	91.1	64.8	94.3	95.1	98.2	52.0
Standard-KV	1	91.2	**65.0**	97.2	96.0	98.5	52.3
CausalFold	1	**91.3**	64.5	**97.3**	**96.1**	**98.6**	**55.8**
MAGI-1 shortcut, 4.5B (Sand.ai et al., 2025): matched pair							
Standard-KV	8	90.1	**65.4**	96.8	95.7	**98.7**	48.1
CausalFold	8	**90.3**	65.0	**96.9**	**95.9**	**98.7**	**50.4**
HunyuanVideo-1.5 TI2V, 8B via minWM (Zhao et al., 2026; Wu et al., 2025): matched pair							
Standard-KV	4	89.8	**61.1**	96.6	95.4	98.4	46.3
CausalFold	4	**90.0**	60.4	**96.7**	**95.5**	**98.5**	**50.2**

sistent attributes $+0.3$, spatial relations $+0.2$, numeracy -0.3), as do VBench quality ($+0.3$) and semantics ($+0.1$). Reading earlier outputs thus helps most where one entity's state has to be carried from block to block. MAGI-1 shows the same direction with smaller gains: $+2.4$ in TCR, $+2.7$ in StoryEval, and $+1.1$ to $+1.7$ in the dynamic-attribute, action, and motion categories. VideoPhy-2 joint accuracy, which requires both semantic adherence and physical commonsense to score at least 4, rises from 28.4 to 31.9 on Wan, while mean physical commonsense falls from 3.50 to 3.40.

Image-to-video. Table 2 adds a supplied first frame whose scene must persist during the transition. TCR rises by 4.9 points on Wan2.1-1.3B (44.6 to 49.5), 3.5 on Wan2.1-14B, 2.3 on MAGI-1, and 3.9 on HunyuanVideo-1.5. Subject consistency, background consistency, and smoothness change by 0.0 to 0.2 points in every pair, so the supplied scene is kept as well as under Standard-KV while more transitions complete. Dynamic degree, the fraction of videos that VBench's optical-flow test classifies as dynamic, is 0.4 to 0.7 points lower in every pair. In a separate paired human study on 1,003 prompts (Appendix I.3), CausalFold receives a motion-completeness preference of 71.17% over Standard-KV, so the lower dynamic degree comes with motion that raters judge more complete.

What completion looks like. In Figure 3, the frog has to leave the lily pad and the dog has to catch the frisbee while both stay recognizable. In Figure 4, the final event moves the basket filled in the



Figure 3: **Completing a temporal change.** CausalFold samples for attribute changes, object relations, and background transitions, read from left to right. The enlarged views show the frog leaving the lily pad and the dog catching the frisbee. Tables 1 and 2 measure how often such changes complete.

previous event, which holds the apples picked two events earlier, the same chain of states as in the corridor turn.

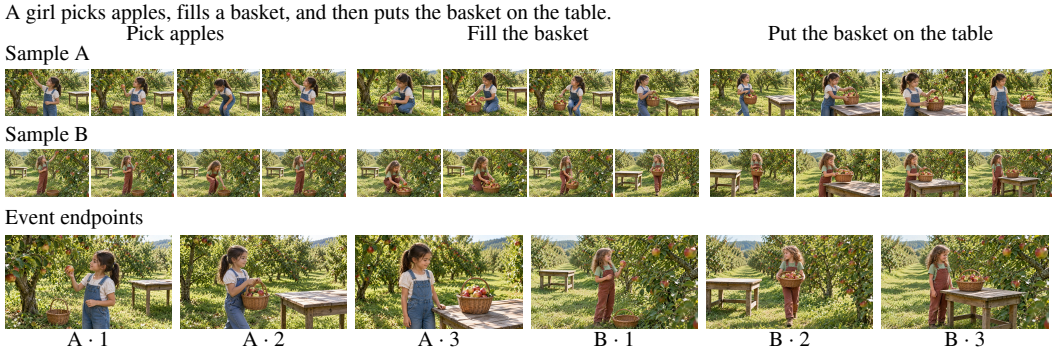

Figure 4: **An ordered story in which each event acts on the result of the previous one.** Prompt: *A girl picks apples, fills a basket, and then puts the basket on the table.* Two samples read from left to right; the lower panels enlarge the three event endpoints. The final event requires the filled basket produced by the earlier events.

5.3 Separating the Forward Read from the Later-Block Gradient

Table 3 reports image-to-video interventions, each changing one part of the mechanism.

Table 3: I2V interventions, mean percentages. Future-logit off and prefix reads: V are retrained; history reads: V changes a frozen CausalFold checkpoint. Subject consistency checks scene preservation. Higher is better; bold is per backbone. Models and native NFE follow Table 2.

Intervention	TC	VBench-I2V	
	TCR	Dynamic	Subject
Wan2.1, 1.3B; 4 NFE			
CausalFold	**49.5**	62.2	**96.5**
Future-logit off	46.9	**62.6**	96.3
History reads: V	43.8	61.4	94.7
MAGI-1 shortcut, 4.5B; 8 NFE			
CausalFold	**50.4**	65.0	**96.9**
Future-logit off	49.2	**65.2**	94.6
History reads: V	46.8	64.6	92.9
HunyuanVideo-1.5, 8B; 4 NFE			
CausalFold	**50.2**	60.4	**96.7**
Future-logit off	48.0	**60.7**	95.4
History reads: V	45.7	59.9	93.8
Prefix reads: V	49.4	60.5	96.3

Reading outputs (P2). We take the trained CausalFold model and restore raw Values on historical video reads at inference, keeping its Keys, weights, and sampler. Wan's TCR drops from 49.5 to 43.8, below the 44.6 of Standard-KV, which was trained to read Values. MAGI-1 and HunyuanVideo-1.5 drop by 3.6 and 4.5 points, and subject consistency falls by 1.8 to 4.0 points. The Queries and Keys of the trained model were learned together with output reads, so the model depends on receiving outputs at those Keys. For HunyuanVideo, a model retrained to read the Values of its text and image prefix, while still reading earlier video outputs, reaches 49.4; most of its gain comes from reading earlier video blocks.

Later-block gradients on earlier attention (P3). The future-logit-off control keeps every forward read of CausalFold and retrains with one change in the backward pass: at each video row p, the logit gradient $\Delta_p^\top u_p$ is replaced by $\Delta_p^\top g_p$, where $g_p = G_{p,:}^\top$ comes from the output branch alone. This removes the later-read term $\Delta_p^\top (u_p - g_p)$ that Theorem 2 adds, while Value gradients and gradients sent to earlier outputs keep the full U. Wan's TCR reaches 46.9, between Standard-KV (44.6) and CausalFold (49.5), and MAGI-1 (48.1, 49.2, 50.4) and HunyuanVideo-1.5 (46.3, 48.0, 50.2) show the same order. About half of each gain therefore requires the later-block gradient on earlier attention; the forward read, trained with Value-side gradients, provides the rest. Without this gradient, subject consistency also falls, most on MAGI-1 (96.9 to 94.6). With a fixed output gradient on one WBench case, the control changes only the logit-gradient map (Figure 13, Appendix E.3).

How the later-block gradient differs across backbones (P4). Across 500 validation cases (Figure 14, Appendix E.4), the variation of the later-block Jacobian $P_{ip}\Delta_p$ follows visit probability on Wan (correlation 0.62, versus 0.09 on HunyuanVideo) and candidate differences $\|\Delta_p\|_2$ on HunyuanVideo (0.85, versus 0.50 on Wan). MAGI-1 writes each block with attention restricted to that block, so the visit probability and the gradient between its cache-writing calls are exactly zero, and its cached outputs hold one block's spatial aggregation. The later-block logit gradients that the future-logit-off control removes on MAGI-1 therefore act inside its noisy calls, where overlap-

ping active chunks read the outputs of earlier chunks. MAGI-1 also has the smallest TCR gains in Tables 1 and 2 (2.4 and 2.3 points).

5.4 Navigation, Long Rollouts, Sampling Budget, and Cost

Table 4: Native WBench (Ying et al., 2026) navigation on EchoWM Flash (Zhang et al., 2026). Means on the official 0–100 scale; higher is better. Camera conditioning and joint audio computation remain active. Bold compares the matched pair.

Method	WBench navigation			
	Quality	Interact.	Consist.	Physical
EchoWM Flash, 21.81B; native action input				
Standard-KV	81.1	87.9	77.5	70.1
CausalFold	**81.2**	**88.6**	**79.1**	**70.8**

Navigation. With EchoWM's camera-action interface, camera conditioning, and joint audio computation active, WBench consistency rises from 77.5 to 79.1, the largest change, while interaction, physical plausibility, and quality rise by 0.7, 0.7, and 0.1 points. Figure 15 shows a left, left, right, right sequence that returns toward its starting view.

Long rollouts. Wan's cache holds 20 earlier latent frames, whereas a 60-second rollout has 241. By the path argument of Appendix B, a cached output keeps what its block read from blocks that were later evicted (Figure 10 shows this for a one-block window). On VBench-Long (Table 7), subject and background consistency rise by 3.5 and 2.7 points and quality by 1.8, with dynamic degree 0.7 lower. Among 384 sixty-second videos, the fraction with an observed failure falls from 10.42% to 5.21%, and the gap in the five-second segment score grows from 1.83 to 8.47 points over the rollout (Appendix I.2).

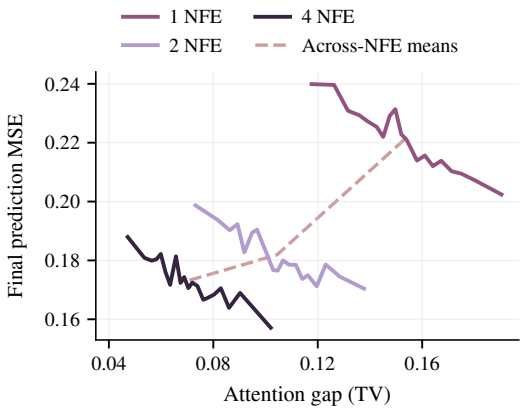

Figure 5: **Fewer evaluations per block, more weight on paths through earlier outputs.** Wan2.1-1.3B, 500 cases per NFE, three paired runs. TV ("attention gap") compares CausalFold weights W with standard weights A; MSE compares student and teacher final latents. Solid curves: 20 bin means after seed averaging; dashed line: NFE means (Appendix E.4).

Sampling budget. The distance $\mathrm{TV}(W_i, A_i)$ measures how much weight CausalFold moves onto paths through earlier outputs relative to standard attention. Wan models trained for one, two, and four evaluations per block have mean TV of 0.153, 0.104, and 0.071 (Table 5): the fewer evaluations a block receives, the more weight its trained model places on those paths. With fixed MAGI-1 weights, raising the budget from 8 to 64 evaluations lowers TV from 0.0335 to 0.0232 and final-latent error from 0.2009 to 0.1693. Within each budget, cases with larger TV end closer to the fully sampled teacher, with mean within-run correlations between TV and final-latent error from -0.56 to -0.40 (Figure 5).

Cost. Complete-output time rises by 5.4% for Wan2.1-1.3B, 8.9% for Wan2.1-14B, 13.4% for HunyuanVideo-1.5, 15.0% for EchoWM, and 18.0% for MAGI-1 (Table 6). Peak memory rises by 1.3 to 3.7 GiB, most for MAGI-1, whose cache is twice as large (Section 4.2).

6 Conclusion

CausalFold makes the KV cache of a block-causal video generator return each token's attention output. Tokens read Values within their block and outputs across blocks, so every read refers to an output that already exists. Later blocks then receive the history that earlier tokens selected with their own Queries, and their losses reach how earlier tokens attend. Against identical Standard-KV training, transition completion rises in all six text- and image-to-video comparisons with scene-preservation scores within 0.2 points, and about half of the image-to-video gain requires the later-block gradient. The change adds no parameters and 5.4% to 18.0% generation time. MAGI-1, whose cache-writing call attends only within its own block, has the smallest gains, in line with its zero visit probability between cache-writing calls.

REFERENCES

Xingjian Bai, Guande He, Zhengqi Li, Eli Shechtman, Xun Huang, and Zongze Wu. Causality in Video Diffusers is Separable from Denoising. *arXiv preprint arXiv:2602.10095*, 2026. URL https://arxiv.org/abs/2602.10095.

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.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion. In *NeurIPS*, volume 37, pp. 24081–24125, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2aee1c4159e48407d68fe16ae8e6e49e-Abstract-Conference.html.

Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, Huan Ling, Haozhe Liu, Hongwei Yi, Hao Zhang, Muyang Li, Yukang Chen, Han Cai, Sanja Fidler, Ping Luo, Song Han, and Enze Xie. SANA-Video: Efficient Video Generation with Block Linear Diffusion Transformer. In *ICLR*, 2026. URL https://openreview.net/forum?id=mzAchylAtf.

Erwan Fagnou, Paul Caillon, Blaise Delattre, and Alexandre Allauzen. Chain and causal attention for efficient entity tracking. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *EMNLP*, pp. 13174–13188, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.731. URL https://aclanthology.org/2024.emnlp-main.731/.

Angela Fan, Thibaut Lavril, Edouard Grave, Armand Joulin, and Sainbayar Sukhbaatar. Addressing Some Limitations of Transformers with Feedback Memory. *arXiv preprint arXiv:2002.09402*, 2020. URL https://arxiv.org/abs/2002.09402.

Weixi Feng, Jiachen Li, Michael Saxon, Tsu-jui Fu, Wenhu Chen, and William Yang Wang. TC-Bench: Benchmarking Temporal Compositionality in Text-to-Video and Image-to-Video Generation. *arXiv preprint arXiv:2406.08656*, 2024. URL https://arxiv.org/abs/2406.08656.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. In *NeurIPS*, volume 38, pp. 167283–167308, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f4823f831af67a3ef15e41a85434422a-Abstract-Conference.html.

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, June 2024. URL https://arxiv.org/abs/2311.17982.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and Versatile Benchmark Suite for Video Generative Models. *TPAMI*, 48(3):3268–3285, 2026. doi: 10.1109/TPAMI.2025.3633890. URL https://arxiv.org/abs/2411.13503.

Haobo Li, Yanhong Zeng, Yunhong Lu, Jiapeng Zhu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Yujun Shen, and Zhipeng Zhang. AAD-1: Asymmetric Adversarial Distillation for One-Step Autoregressive Video Generation. *arXiv preprint arXiv:2606.03972*, 2026. URL https://arxiv.org/abs/2606.03972.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling Forcing: Autoregressive Long Video Diffusion in Real Time. In *ICLR*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/935151cc6cb5d8b6816133b75233775a-Abstract-Conference.html.

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://proceedings.iclr.cc/paper_files/paper/2025/hash/0396ca5a4c628936609aa819bfbca916-Abstract-Conference.html.

Costin-Andrei Oncescu, Depen Morwani, Samy Jelassi, Alexandru Meterez, Mujin Kwun, and Sham Kakade. The Recurrent Transformer: Greater Effective Depth and Efficient Decoding. *arXiv preprint arXiv:2604.21215*, 2026. URL https://arxiv.org/abs/2604.21215.

Giorgio Roffo, Hazem Abdelkawy, Nilli Lavie, and Luke Palmer. Self-Attention And Beyond the Infinite: Towards Linear Transformers with Infinite Self-Attention. *arXiv preprint arXiv:2603.00175*, 2026. URL https://arxiv.org/abs/2603.00175.

Sand.ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, Xiaoyang Kang, Yuchen Sun, Yue Cao, Yunpeng Huang, Yutong Lin, Yuxin Fang, Zewei Tao, Zheng Zhang, Zhongshu Wang, Zixun Liu, Dai Shi, Guoli Su, Hanwen Sun, Hong Pan, Jie Wang, Jiexin Sheng, Min Cui, Min Hu, Ming Yan, Shucheng Yin, Siran Zhang, Tingting Liu, Xianping Yin, Xiaoyu Yang, Xin Song, Xuan Hu, Yankai Zhang, and Yuqiao Li. MAGI-1: Autoregressive Video Generation at Scale. *arXiv preprint arXiv:2505.13211*, 2025. URL https://arxiv.org/abs/2505.13211.

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, June 2025. URL https://arxiv.org/abs/2407.14505.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Yiping Wang, Xuehai He, Kuan Wang, Luyao Ma, Jianwei Yang, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Is Your World Simulator a Good Story Presenter? A Consecutive Events-Based Benchmark for Future Long Video Generation. In *CVPR*, pp. 13629–13638, June 2025. URL https://arxiv.org/abs/2412.16211.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, Linus, Patrol, Peizhen Zhang, Peng Chen, Penghao Zhao, Qi Tian, Songtao Liu, Weijie Kong, Weiyan Wang, Xiao He, Xin Li, Xinchi Deng, Xuefei Zhe, Yang Li, Yanxin Long, Yuanbo Peng, Yue Wu, Yuhong Liu, Zhenyu Wang, Zuozhuo Dai, Bo Peng, Coopers Li, Gu Gong, Guojian Xiao, Jiahe Tian, Jiaxin Lin, Jie Liu, Jihong Zhang, Jiesong Lian, Kaihang Pan, Lei Wang, Lin Niu, Mingtao Chen, Mingyang Chen, Mingzhe Zheng, Miles Yang, Qiangqiang Hu, Qi Yang, Qiuyong Xiao, Runzhou Wu, Ryan Xu, Rui Yuan, Shanshan Sang, Shisheng Huang, Siruis Gong, Shuo Huang, Weiting Guo, Xiang Yuan, Xiaojia Chen, Xiawei Hu, Wenzhi Sun, Xiele Wu, Xianshun Ren, Xiaoyan Yuan, Xiaoyue Mi, Yepeng Zhang, Yifu Sun, Yiting Lu, Yitong Li, You Huang, Yu Tang, Yixuan Li, Yuhang Deng, Yuan Zhou, Zhichao Hu, Zhiguang Liu, Zhihe Yang, Zilin Yang, Zhenzhi Lu, Zixiang Zhou, and Zhao Zhong. HunyuanVideo 1.5 Technical Report. *arXiv preprint arXiv:2511.18870*, 2025. URL https://arxiv.org/abs/2511.18870.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time Interactive Long Video Generation. In *ICLR*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/91a1610c6ed9e02d33f826b46f472b92-Abstract-Conference.html.

Yongqi Yang, Huayang Huang, Xu Peng, Xiaobin Hu, Donghao Luo, Jiangning Zhang, Chengjie Wang, and Yu Wu. Towards One-step Causal Video Generation via Adversarial Self-Distillation. In *ICLR*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3ae86071c169649bff21188c536163dc-Abstract-Conference.html.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved Distribution Matching Distillation for Fast Image Synthesis. In *NeurIPS*, volume 37, pp. 47455–47487, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/54dcf25318f9de5a7a01f0a4125c541e-Abstract-Conference.html.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation. In *CVPR*, pp. 6613–6623, June 2024b. URL https://arxiv.org/abs/2311.18828.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From Slow Bidirectional to Fast Autoregressive Video Diffusion Models. In *CVPR*, pp. 22963–22974, June 2025. URL https://arxiv.org/abs/2412.07772.

Kaining Ying, Hengrui Hu, Siyu Ren, Jiamu Li, Fengjiao Chen, Ziwen Wang, Xuezhi Cao, Xunliang Cai, and Henghui Ding. WBench: A Comprehensive Multi-turn Benchmark for Interactive Video World Model Evaluation. *arXiv preprint arXiv:2605.25874*, 2026. URL https://arxiv.org/abs/2605.25874.

Songchun Zhang, Yaowei Li, Junhao Zhuang, Weiyang Jin, Haoyu Wang, Xin Lu, Yilang Sun, Shiyi Zhang, Haoran Li, Xiaoxiao Ma, Yuming Li, Yijun Liu, Yaofeng Su, Yanwen Ma, Haoyu Wu, Zihan Su, Yue Ma, Lvmin Zhang, Haoyang Huang, Zeyue Xue, Anyi Rao, and Nan Duan. EchoWM: Open and Enterable Omnimodal World Models. *arXiv preprint arXiv:2608.23189*, 2026. URL https://arxiv.org/abs/2608.23189.

Min Zhao, Hongzhou Zhu, Bokai Yan, Zihan Zhou, Yimin Chen, Wenqiang Sun, Kaiwen Zheng, Guande He, Xiao Yang, Chongxuan Li, Fan Bao, and Jun Zhu. minWM: A Full-Stack Open-Source Framework for Real-Time Interactive Video World Models. *arXiv preprint arXiv:2605.30263*, 2026. URL https://arxiv.org/abs/2605.30263v1.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal Forcing: Autoregressive Diffusion Distillation Done Right for High-Quality Real-Time Interactive Video Generation. *arXiv preprint arXiv:2602.02214*, 2026. URL https://arxiv.org/abs/2602.02214.

Junhao Zhuang, Shiyi Zhang, Yuxuan Bian, Yaowei Li, Yawen Luo, Yijun Liu, Weiyang Jin, Songchun Zhang, Xianglong He, Xuying Zhang, Haoran Li, Haoyang Huang, Zeyue Xue, and Nan Duan. Self Gradient Forcing: Native Long Video Extrapolation. *arXiv preprint arXiv:2607.20368*, 2026. URL https://arxiv.org/abs/2607.20368.

AI USE STATEMENT

Generative AI assisted literature discovery, drafting and polishing manuscript sections, mathematical cross-checks, and figure and LaTeX preparation. The authors are responsible for verifying all AI-assisted content and for the final claims, references, implementation, and results.

A Exact Solution and Convex Value Weights

This appendix proves Theorem 1 of Section 4.1 and derives its path consequences; Appendix C proves Theorem 2. We work with one head at one layer before its output projection. Tokens are partitioned into ordered blocks $\mathcal{B}_1, \ldots, \mathcal{B}_S$. A row may attend within its own block and to permitted earlier blocks, but never to a later block. All statements about changing an attention row hold the layer input, Values, other attention rows, and the mask fixed. The derivatives characterize this fixed-input attention operator.

A.1 Assumptions and Exact Solution

Let $A \in \mathbb{R}^{N\times N}$ be nonnegative with $A\mathbf{1} = \mathbf{1}$. Write $D_{ij} = A_{ij}$ for tokens in the same block and zero otherwise, and let $L = A - D$. At least one unmasked candidate exists in every row. These conditions hold for pre-dropout softmax attention. Dropout outside this contraction preserves the analysis; dropping and rescaling individual attention weights generally breaks the row-stochastic assumption.

Lemma 1 (Nilpotence and uniqueness)**.** *For a strictly block-lower matrix L, $L^S = 0$. The matrix $I - L$ is invertible with inverse $P = \sum_{k=0}^{S-1} L^k$. Consequently, the forward substitution in Equation (2) yields the unique state $R = PDV$.*

Proof. A nonzero product of k entries of L follows k strictly decreasing block indices. Such a path cannot contain S edges among S blocks, proving nilpotence. Multiplying the finite sum by $I - L$ on either side gives $I - L^S = I$. At block t, all historical outputs on the right side of the recursion have already been determined. This establishes existence by substitution and uniqueness by induction. Equivalently, if two solutions differ by Z, then $Z = LZ = L^S Z = 0$. □

Proof of Theorem 1. Uniqueness follows from Lemma 1. Every term of $W = \sum_k L^k D$ is nonnegative. Since $D\mathbf{1} = (I - L)\mathbf{1}$, its row sums satisfy

$$W\mathbf{1} = PD\mathbf{1} = P(I - L)\mathbf{1} = \mathbf{1}. \tag{6}$$

For $E = V - V'$, each output row is a convex combination of the rows of E. The triangle inequality gives

$$\|(WE)_i\|_2 \le \sum_j W_{ij}\|E_j\|_2 \le \max_j \|E_j\|_2. \tag{7}$$

Taking the maximum over i proves the stated inequality. The same argument holds for any norm on the Value space. The bound controls perturbations in V at fixed A; an input perturbation also changes the attention weights through Q and K. □

A.2 Paths and Finite Substitutions

The weights admit a terminating random-path interpretation. At row i, select a candidate using row A_i. Selecting a local entry of D terminates the path and returns that candidate's original Value. Selecting an entry of L moves to the selected earlier token and repeats the procedure. The row-sum condition makes each choice a probability distribution, and strict block descent makes termination certain. If κ_i is the number of historical moves and J_i the terminal token, then

$$\Pr(\kappa_i = k, J_i = j) = (L^k D)_{ij}, \qquad \Pr(\kappa_i \ge m) = \tau_i(m) := e_i^\top L^m \mathbf{1}. \tag{8}$$

In particular, $\kappa_i \le S - 1$ and $W_{ij} = \Pr(J_i = j)$. This interpretation describes repeated composition of the attention weights within one layer evaluation.

To compare W with a limited number of substitutions, initialize $R^{(0)} = V$ and define $R^{(m+1)} = DV + LR^{(m)}$. Induction yields

$$R^{(m)} = W_m V, \qquad W_m = \sum_{k=0}^{m-1} L^k D + L^m. \tag{9}$$

For $m = 0$, the empty sum gives $W_0 = I$. For $m = 1$, $W_1 = D + L = A$, so ordinary attention is exactly one substitution. For $m \geq S$, $W_m = W$. The term L^m means that paths surviving m historical moves stop early and return the raw Value at their current token. All earlier-terminating paths agree with W. This identifies precisely where a forward difference can arise.

Figure 6 gives a spatial view of Equation (9). Substitution depth measures the number of historical reads within one attention evaluation. Increasing it changes source entropy and the locations of its spatial boundaries.

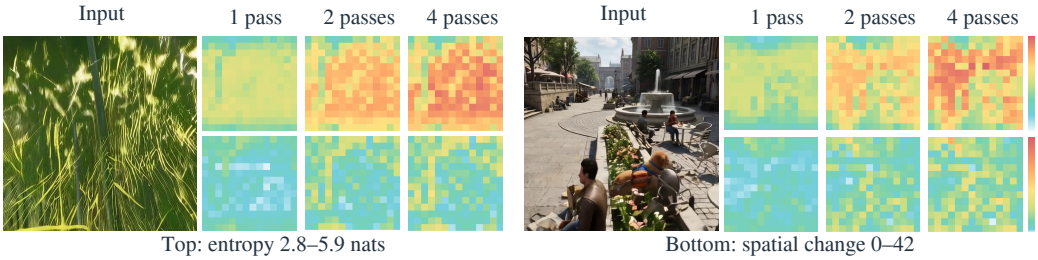

Figure 6: **Finite substitutions redistribute source uncertainty in space.** WBench cases 35 and 10, each on a 14×24 grid. Within each case, columns show the input and substitution depths $k = 1, 2, 4$, where the artwork notation $W^{[k]}$ denotes W_k of Equation (9). Depth counts historical substitutions within a layer. Upper maps show row entropy $H_p = -\sum_j W_{pj}^{[k]} \log W_{pj}^{[k]}$ in nats (shared range 2.8–5.9); lower maps show $\|\nabla_{x,y} H_p\|_2$ in normalized image coordinates (shared range 0–42). Entropy is linearly colored; derivative maps use the shared square-root transform $\sqrt{g/g_{\max}}$. The same spatial crop is used throughout, so changes localize where finite historical composition alters the endpoint distribution.

A.3 Spatial and Temporal State Readouts

The returned content is obtained by applying the induced weights to the source Values. Figure 7 therefore compares V and R through identical RGB and PCA readouts. Shared projections make the spatial redistribution visible without conflating a representation readout with a decoded generation result.

Figure 8 extends the same distinction from snapshots to a fixed image slice. Because all three x–t arrays use exactly the marked spatial row, differences trace where historical composition accumulates rather than a change in crop or observation time.

B Exact Truncation Error

Forward change depends jointly on surviving path mass and on the content returned by those paths. The following identity separates these two factors.

Proposition 1 (Exact scalar error and a vector bound)**.** *If $\tau_i(m) > 0$, define $\mu_i^{(m)} = e_i^\top L^m / \tau_i(m)$. Then*

$$e_i^\top (W - W_m) = \tau_i(m) \big(\mu_i^{(m)} W - \mu_i^{(m)} \big), \tag{10}$$

$$\sup_{v \in [0,1]^N} |e_i^\top (W - W_m) v| = \tau_i(m)\, \mathrm{TV}\big(\mu_i^{(m)} W, \mu_i^{(m)}\big), \tag{11}$$

$$\|R_i - R_i^{(m)}\|_2 \leq \tau_i(m)\, \mathrm{TV}\big(\mu_i^{(m)} W, \mu_i^{(m)}\big)\, \mathrm{diam}(V). \tag{12}$$

Here $\mathrm{TV}(u, v) = \frac{1}{2}\|u - v\|_1$ for probability vectors and $\mathrm{diam}(V) = \max_{j,k} \|V_j - V_k\|_2$. If $\tau_i(m) = 0$, all three errors are zero and no conditional distribution is needed.



Figure 7: **Spatial content under a shared state readout.** WBench cases 3 and 1 at target block 2, each on a 20×34 grid. For each case, upper and lower rows compare original Values V with CausalFold outputs R. RGB panels apply the same contraction to fixed 24×24 patch-relative RGB channels and reassemble them at their original positions. Neighboring PCA panels project the same 12-channel descriptor subspace with one shared basis and scaling. The readouts expose redistribution of spatial content by $R = (I - L)^{-1}DV$; these representation diagnostics display the resulting mixtures. Blur in an RGB readout shows spatial averaging under that mixture.

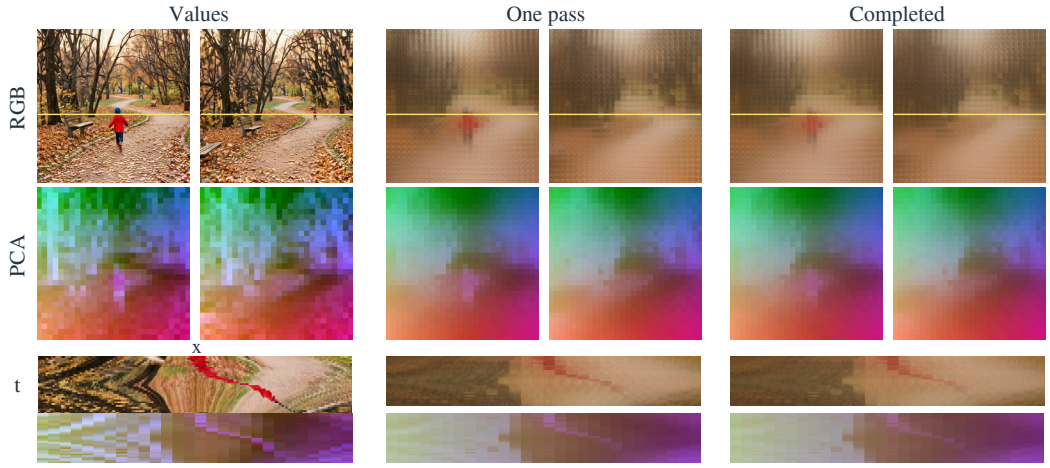

Figure 8: **A fixed spatial slice reveals how state composition accumulates over time.** WBench case 9 uses 32 sampled video frames and a 28×48 grid. Groups compare Values V_t, standard outputs $Y_t = D_{tt}V_t + L_{t,t-1}V_{t-1}$, and CausalFold outputs $R_t = D_{tt}V_t + L_{t,t-1}R_{t-1}$ with one direct historical block. RGB uses fixed 16×16 patch-relative channels; PCA shares one basis and scaling across groups. Upper snapshots use sampled-frame indices 8 and 24. Yellow horizontal lines identify the exact image row used in the lower x–t arrays, with t indexing the 32 observations. RGB and PCA slices use the corresponding observed readouts and diagnose state evolution along the marked row.

Proof. Repeatedly expanding $W = D + LW$ gives $W = \sum_{k=0}^{m-1} L^k D + L^m W$. Subtracting Equation (9) yields $W - W_m = L^m(W - I)$, proving Equation (10). The vector $\mu_i^{(m)}$ is a probability distribution by its definition, and $\mu_i^{(m)}W$ is also one because W is row stochastic. Let $q = \mu_i^{(m)}W - \mu_i^{(m)}$. Its entries sum to zero, so the total positive and negative masses both equal $\frac{1}{2}\|q\|_1$. Over the unit cube, the maximum of qv sets $v_j = 1$ exactly where $q_j > 0$, and the maximum of its negative

selects $q_j < 0$. Both equal TV, proving the scalar equality. For vector Values, write $q = \alpha(u - v)$ using its normalized positive and negative parts, where $\alpha = \mathrm{TV}$. Both uV and vV belong to the convex hull of the Value rows. Their separation is at most $\mathrm{diam}(V)$, proving the vector inequality. If $\alpha = 0$, the result is immediate. □

Proposition 2 (Path length and conditional geometric bounds)**.** *The expected number of historical moves is*

$$\mathbb{E}\kappa_i = \sum_{m=1}^{S-1} \tau_i(m) = e_i^\top LP\mathbf{1}, \quad e_i^\top P\mathbf{1} = 1 + \mathbb{E}\kappa_i \le S. \tag{13}$$

Proof. For any nonnegative integer-valued variable bounded by $S-1$, summing the survival indicators gives its expectation. Substitution of Equation (8) and the finite expansion of P proves the first identities. Under the additional condition $L\mathbf{1} \le \rho\mathbf{1}$ with $\rho < 1$, nonnegativity preserves inequalities. Induction gives $L^m\mathbf{1} \le \rho^m\mathbf{1}$, and summing the geometric series gives $\mathbb{E}\kappa_i \le \rho/(1-\rho)$. □

The number ρ is an upper bound derived from the attention matrix. It may approach one, making the geometric bound loose even though the finite solution remains exact. Likewise, positive path probability does not guarantee changed content. If all Value rows equal a common vector, both W_mV and WV equal that vector for every m. More generally, a nonzero signed weight difference can cancel when applied to the actual Values. The supremum in Equation (11) maximizes over scalar Value assignments; the realized change depends on the learned Values.

Figure 9 isolates the distributional part: similar maximum-weight maps can mask a substantial redistribution of endpoint mass. Actual output change still depends on the Values.

A terminal token has $W_{ij} > 0$ exactly when some permitted sequence of historical edges from i, followed by a positive local edge, terminates at j. Nonnegativity rules out cancellation among such paths. If the original mask already exposes every earlier token, this characterization does not enlarge visibility. Under a non-transitive sparse mask, it can carry information along an indirect path that is absent as a direct edge. Under physical cache eviction, an already cached output can retain information from an evicted ancestor. All direct reads follow the native cache mask.

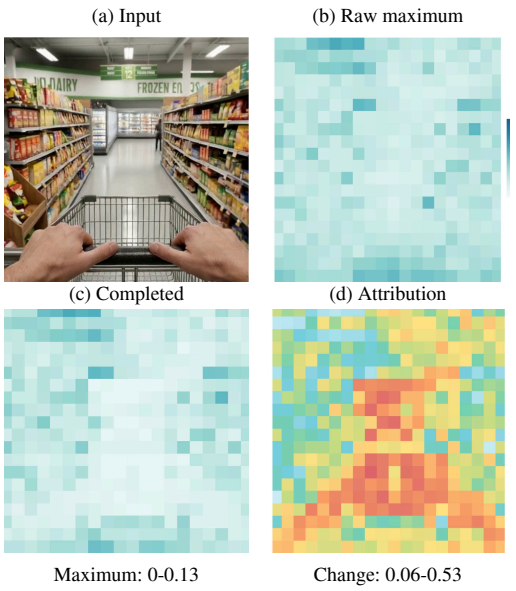

Figure 9: **Similar concentration can accompany redistributed attribution.** WBench case 12, final block, 20×34 grid. Panels read in a 2×2 grid: (a) input; (b) $\max_j A_{pj}$; (c) $\max_j W_{pj}$; (d) $\mathrm{TV}(A_p, W_p) = \frac{1}{2}\sum_j |A_{pj} - W_{pj}|$. The first two maps share the range 0–0.13; TV uses 0.06–0.53. All maps use the same spatial crop and fixed routing. Similar maximum weights can coexist with redistributed endpoint mass. TV measures the redistribution of source weights.

Figure 10 illustrates the sparse-window case of this path argument. At the later target block, CausalFold attribution reaches older sources through the same direct window. This inherited source mass quantifies information carried through retained cache entries.

C Gradient on an Earlier Token's Attention

For a historical row p, let $\mathcal{N}_p$ be its unmasked candidates. The matrix $C_p \in \mathbb{R}^{|\mathcal{N}_p| \times d_v}$ stacks original Values for candidates in p's block and CausalFold outputs for candidates in earlier blocks, so that $r_p = C_p^\top a_p$. The matrix C_p is independent of z_p: local Values are fixed, and outputs of earlier blocks

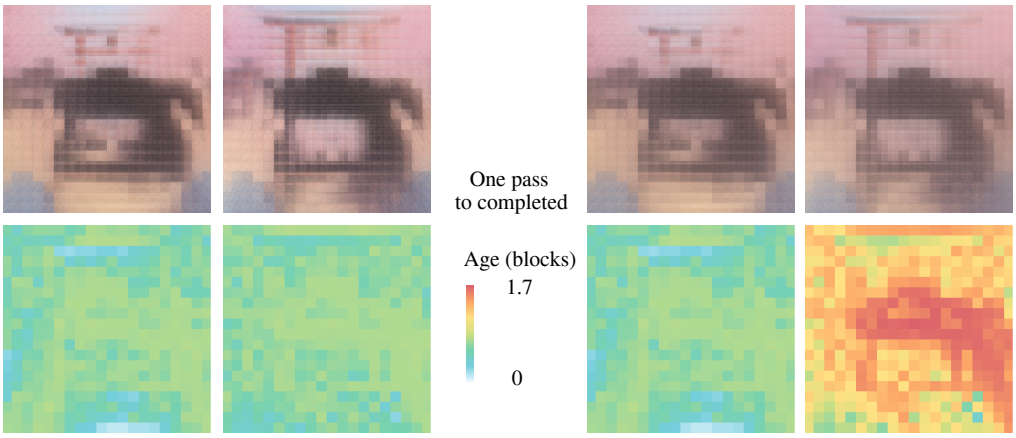

Figure 10: **Effective source age can exceed the direct cache window.** WBench case 75, target blocks 1 and 3, on a 20×34 grid with one direct historical block. Each pair compares standard-attention and CausalFold RGB readouts above their source-age maps. For a query p, age is $\sum_s m_{p,s}(t-s)$, where $m_{p,s}$ sums original-Value attribution in source block s under A or W. All four age maps share the range 0–1.7 blocks. Standard-attention age is at most one; cached outputs carry older attribution through retained entries. RGB uses identical patch-channel readouts. Older source mass measures indirect provenance through retained states.

cannot depend on a row in a later block. This independence is the structural reason the derivative factorizes.

C.1 Proof of Theorem 2

The softmax Jacobian is $\Sigma_p = \mathrm{diag}(a_p) - a_p a_p^\top$, so a perturbation of z_p injects the local change $\Delta_p\, dz_p = C_p^\top \Sigma_p\, dz_p$ at row p. Linearizing the recursion with the other attention rows fixed gives $dR = L\, dR + e_p(\Delta_p\, dz_p)^\top$. Left multiplication by P proves

$$dr_i = P_{ip}\Delta_p\, dz_p. \tag{14}$$

For output-branch gradients $g_i = G_{i,:}^\top$, the chain rule gives $\nabla_{z_p}\ell = \Delta_p^\top \sum_i P_{ip} g_i = \Delta_p^\top u_p$. This proves the derivative part of Theorem 2. For a loss that has upstream gradients only at a set of future rows, the sum restricts to that set. In a full network, later layers may also inject gradients at historical rows, so that restriction cannot be assumed merely because the final task predicts the future.

To show $P_{ip} \le 1$, view $(L^k)_{ip}$ as the probability of being at p after k historical moves without having terminated. A path visits any block at most once, so the events of reaching p at different values of k are disjoint. Their sum is the probability of ever visiting p. For $i = p$, the zero-length visit contributes one and a return is impossible. For a later i, this is the visit probability of Section 4.1. Therefore $0 \le P_{ip} \le 1$, completing the theorem.

Proposition 3 (Finite replacement of one attention row)**.** *Replace a_p by any other probability row $\widetilde{a}_p$ on the same allowed candidates, leaving the mask, Values, and other rows fixed. Then*

$$\widetilde{r}_i - r_i = P_{ip}C_p^\top(\widetilde{a}_p - a_p) = P_{ip}(\widetilde{r}_p - r_p). \tag{15}$$

Proof. No candidate in C_p is affected by the change, so the local state difference is exactly $C_p^\top(\widetilde{a}_p - a_p)$. Every later row has unchanged weights, and its difference is the weighted sum of earlier state differences. Propagating the local change by forward substitution gives the claimed expression. Equivalently, subtracting the two recurrences introduces $(\widetilde{D} - D)V + (\widetilde{L} - L)R$ at row p. The extra term $(\widetilde{L} - L)(\widetilde{R} - R)$ is zero because it reads only earlier unchanged states. Finally, P_{ip} itself does not change: every path contributing to it ends upon reaching p and never uses an outgoing edge from p. □

C.2 What Makes the Derivative Informative

Column j of Δ_p is $a_{pj}(c_{pj} - r_p)$, where $c_{pj} = (C_p)^\top_{j,:}$. Thus $\Delta_p = 0$ if and only if every positive-weight candidate equals the output r_p. With finite unmasked softmax logits all candidate weights are positive. A later-block Jacobian is therefore nonzero exactly when its visit probability is positive and the candidate states are not all equal. A nonzero Jacobian permits a loss-sensitive direction, but the actual upstream gradient may be orthogonal to that direction. Contributions from several future outputs can also cancel. These are distinct failure modes from lack of reach.

Figure 11 complements this content condition with a fixed-query comparison. The cosine map isolates candidate similarity, while visit probability and realized loss gradients are measured separately.

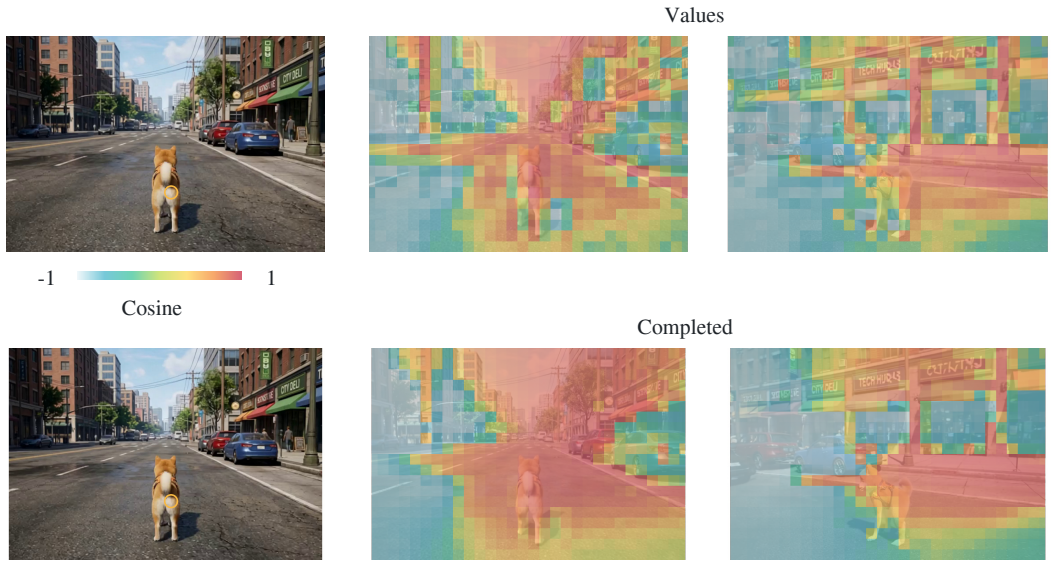

Figure 11: **Keeping the historical query fixed.** WBench case 1, 20×34 grid. The first-frame patch $(x, y) = (17, 14)$ defines the same query $q = V_{\text{query}}$ in both rows. Future blocks 1 and 3 compare $\cos(q, V_j)$ (top) with $\cos(q, R_j)$ (bottom). All maps share $[-1, 1]$ and overlay opacity 0.72 on the corresponding target image. Candidate states vary under a fixed query and source location. The plotted quantity is feature-space cosine similarity.

For a set $\mathcal{F}$ of later rows, stack the outputs and define $\pi = P_{\mathcal{F},p}$. The Jacobian is $J_{\mathcal{F},p} = \pi \otimes \Delta_p$. Direct multiplication gives $J^\top_{\mathcal{F},p} J_{\mathcal{F},p} = \|\pi\|_2^2 \Delta_p^\top \Delta_p$, and hence

$$\|J_{\mathcal{F},p}\|_2 = \|P_{\mathcal{F},p}\|_2 \|\Delta_p\|_2, \qquad \operatorname{rank}(\Delta_p) \le \min(d_v, |\mathcal{N}_p| - 1). \tag{16}$$

The rank bound follows from $\Delta_p \mathbf{1} = 0$ and its output dimension. When $\pi \neq 0$, the later-block Jacobian has the same rank and row space as Δ_p. Later reads scale and combine the routing directions defined by the local softmax and candidate content.

C.3 A Complete Three-Block Example

Take one token per block and

$$A = \begin{pmatrix} 1 & 0 & 0 \\ a & 1-a & 0 \\ c & b & 1-b-c \end{pmatrix}, \quad V = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad 0 < a < 1,\ b, c > 0,\ b + c < 1. \tag{17}$$

Ordinary attention gives $Y_2 = a$ and $Y_3 = c$. Completion gives $R_1 = 1$, $R_2 = a$, and $R_3 = c + ab$. Let $a = \operatorname{sigmoid}(\lambda)$, where λ is the logit difference in row two. Then $\partial R_3 / \partial \lambda = ba(1-a)$, while $\partial Y_3 / \partial \lambda = 0$. The factor b is the visit probability from the third row to the second, and $a(1-a)$ is its local content-sensitive derivative. The difference vanishes if $b = 0$ or the first two candidate

Values coincide. With $a = 0.4$, $b = 0.3$, and $c = 0.2$, the exact outputs are 0.20 and 0.32, and the CausalFold derivative is 0.072. These values follow analytically from the three-block matrix.

D Reverse Substitution and Numerical Error

Let $\langle Z_1, Z_2\rangle = \mathrm{tr}(Z_1^\top Z_2)$ be the Frobenius inner product and let G collect ordinary output-branch gradients for the packed layer. Differentiating $(I - L)R = DV$ gives

$$(I - L)\, dR = (dD)V + D\, dV + (dL)R. \tag{18}$$

Set $U = P^\top G$. Then $d\ell = \langle G, dR\rangle = \langle U, (dD)V + D\, dV + (dL)R\rangle$. Moving factors across the inner product gives

$$\nabla_V \ell = D^\top U, \qquad \nabla_D \ell = \Pi_D(UV^\top), \qquad \nabla_L \ell = \Pi_L(UR^\top), \tag{19}$$

where Π_D and Π_L retain only the permitted block-local and historical entries, respectively. The adjoint is evaluated in reverse block order as $U_s = G_s + \sum_{t>s} L_{ts}^\top U_t$. The D and L derivatives are placed back into one attention-gradient matrix before applying the original row-softmax Jacobian and Query/Key projections. The reverse substitution evaluates the adjoint directly.

Historical gradient accumulation depends on path length. For $\|Z\|_{1,1} = \sum_{i,j} |Z_{ij}|$ and the induced matrix norm $\|P^\top\|_1$, nonnegativity and Equation (13) give

$$\|U\|_{1,1} \le \|P^\top\|_1 \|G\|_{1,1} = \big(1 + \max_i \mathbb{E}\kappa_i\big)\|G\|_{1,1} \le S\|G\|_{1,1}. \tag{20}$$

Thus one output can distribute its gradient over several historical visits even though no single hitting probability exceeds one. The optional geometric assumption in Proposition 2 replaces S by $1/(1-\rho)$. The bounds are conditional on the entering gradient G.

Numerical substitution has a related path-length dependence. If a computed state $\widehat{R}$ has residual $E = \widehat{R} - DV - L\widehat{R}$, then $(I - L)(\widehat{R} - R) = E$, so

$$\widehat{R} - R = PE, \qquad \|\widehat{R} - R\|_{\infty,2} \le \big(1 + \max_i \mathbb{E}\kappa_i\big)\|E\|_{\infty,2}. \tag{21}$$

This is an a posteriori residual bound for fixed weights. It explains why a small per-step arithmetic residual can accumulate along long paths. The residual and expected path length jointly bound the numerical error at fixed weights.

E Cache-Writing and Reading Calls

In streaming generation, a cache-writing call (a *writer*, the clean writer of the native protocols) runs after a block has been denoised, and several noisy calls that read the cache (*readers*) consume its cached outputs later. Each produced output is a separate node identified by its call, layer, and head. Holding the inputs and attention weights of these calls fixed, every node depends only on nodes produced before it, which gives a finite directed acyclic graph even when several calls process the same video block.

E.1 Extension to a Finite Call Graph

Let $\mathcal{V}$ stack the original Values of the involved calls and $\mathcal{R}$ the outputs of all nodes. In topological order,

$$\mathcal{R} = \mathcal{D}\mathcal{V} + \mathcal{L}\mathcal{R}, \qquad \mathcal{D} \ge 0, \quad \mathcal{L} \ge 0, \quad \mathcal{D}\mathbf{1} + \mathcal{L}\mathbf{1} = \mathbf{1}. \tag{22}$$

The rectangular matrix $\mathcal{D}$ holds the weights on Values read within a node's own block, and the strictly lower matrix $\mathcal{L}$ holds the weights on earlier outputs. With q nodes, $\mathcal{L}^q = 0$, so $\mathcal{R} = \mathcal{P}\mathcal{D}\mathcal{V}$

with $\mathcal{P} = (I - \mathcal{L})^{-1}$. Nonnegativity and unit row sums follow as in Theorem 1: $\mathcal{P}\mathcal{D}\mathbf{1} = \mathcal{P}(I - \mathcal{L})\mathbf{1} = \mathbf{1}$. Theorem 2 extends whenever the changed row cannot affect its own candidates.

Restrict the graph to writers and let L_{ww} hold writer-to-writer weights and D_w the weights on the writers' own Values. Then $P_w = (I - L_{ww})^{-1}$ and $W_w = P_w D_w$. A writer that attends only to its own block has $L_{ww} = 0$, hence $P_w = I$ and $W_w = D_w$; the selected MAGI-1 writers are of this form, and their writer-to-writer Jacobians are exactly zero. A later noisy reader still consumes the writer's spatial aggregation $D_w V_w$, which differs from V_w, and its gradient reaches the writer through those reads.

Figure 12 makes the writer boundary spatially concrete. Changing the permitted support changes the output returned by a writer, while the call graph separately specifies the writer-to-writer edges.

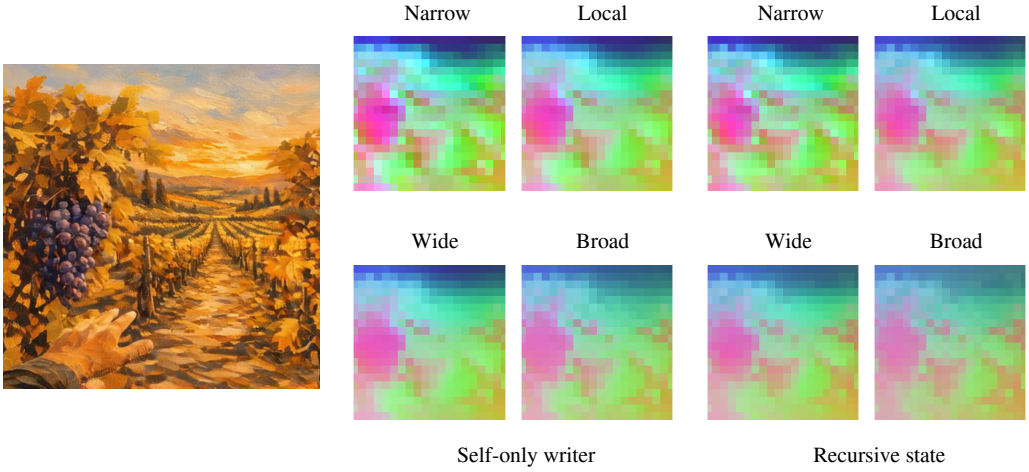

Figure 12: **Writer protocol and spatial support are separate choices.** WBench case 95, target block 3, 20×34 grid. The shared image is followed by self-only-writer and recursive-output groups. Within each group, Narrow, Local, Wide, and Broad use spatial widths $\sigma_{\mathrm{sp}} = 0.025, 0.045, 0.085, 0.16$ in normalized coordinates. Values, semantic affinity, change-dependent history weighting, and the history window are held fixed. Self-only weights are renormalized on local support, which defines a protocol-specific attention matrix. All eight maps use one PCA basis, one color range, and the same brightness mapping. The maps show spatial responses; the zero writer-to-writer Jacobian follows from the self-only writer mask.

E.2 Gradients from Readers in Later Calls

For each node, separate the gradient that enters through its ordinary attention-output branch, G as in Section 4.3, from the gradient that enters because other nodes read it from the cache. G includes the output projection and all downstream effects of that branch. Let $\mathcal{W}$ be the writer rows and $\mathcal{Q}$ the reader rows that read them directly. Then

$$G_{\mathcal{W}}^{\mathrm{read}} = A_{\mathcal{Q}\mathcal{W}}^{\top} U_{\mathcal{Q}}, \qquad U_{\mathcal{W}} = P_w^{\top}\left(G_{\mathcal{W}} + G_{\mathcal{W}}^{\mathrm{read}}\right), \tag{23}$$

where $A_{\mathcal{Q}\mathcal{W}}$ holds the reader-to-writer weights. A reader's total gradient U_r already includes its own later consumers, so each direct edge contributes exactly once, and $P_w^{\top}$ propagates writer-to-writer contributions. For a writer that attends only to its own block, $U_{\mathcal{W}} = G_{\mathcal{W}} + G_{\mathcal{W}}^{\mathrm{read}}$, which still gives a nonzero later-block contribution to its logits.

For a video row p, let $u_p = U_{p,:}^{\top}$ and $g_p = G_{p,:}^{\top}$. We record the full logit gradient $\Delta_p^{\top} u_p$, the output-branch part $\Delta_p^{\top} g_p$, and the later-read part $\zeta_p = \Delta_p^{\top}(u_p - g_p)$. The cached output of p receives the full u_p. We measure g_p at the output branch before cache-read contributions are added, which separates supervision through the token's own output from supervision through later reads.

E.3 Exact Definitions of the Controls

Future-logit off. This training control keeps all CausalFold forward outputs, masks, and cache reads. In its backward pass, the vector used for the softmax-logit derivative at each video row is g_p in place of u_p, while Value gradients and gradients sent to earlier outputs keep the full U. Every condition-prefix row keeps the full logit derivative. The intervention thus deletes ζ_p from video logit gradients and keeps the forward information and Value-side learning. Parameter gradients combine these local derivatives across all rows. The diagnostic $\Delta_p^\top g_p$ is evaluated at CausalFold weights, while the control adapts its own weights during retraining.

History reads: V. This deployment control uses frozen CausalFold weights and restores raw V only on historical video reads. It keeps the original Keys, the sampling schedule, and all condition-prefix reads, and keeps raw Values alongside the cached outputs for the replaced reads. **Prefix reads: V.** The HunyuanVideo condition control is trained and evaluated with only the video-to-prefix reads restored to the prefix's original Values, leaving historical video reads of R intact. The prefix still computes its output for its ordinary output branch. This control uses the original joint softmax without renormalization and receives the same initialization and training as CausalFold. Its comparison with Standard-KV isolates reading earlier video outputs while prefix reads return Values. Standard-KV supplies the fully retrained standard-attention comparison.

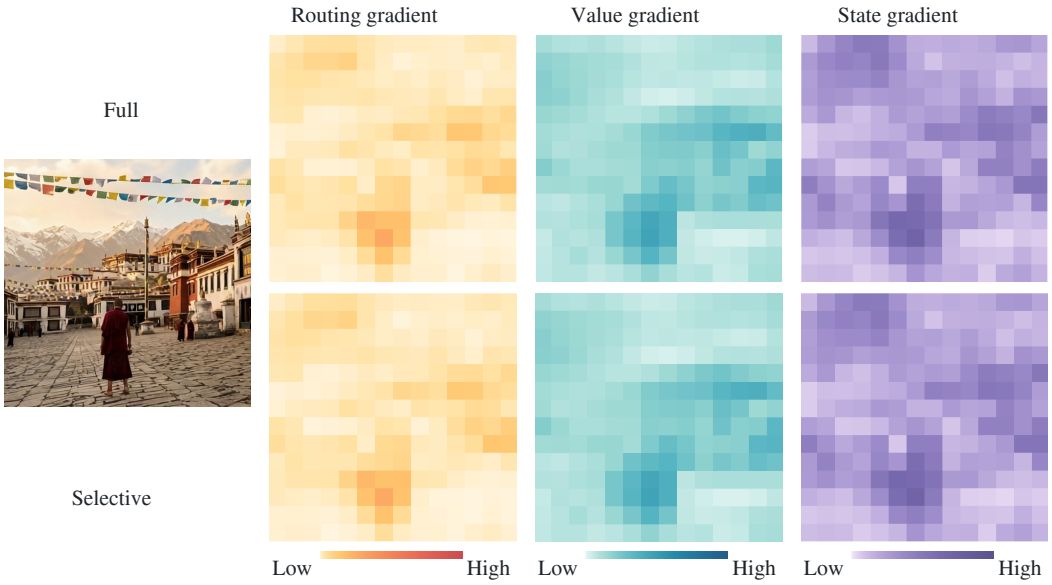

Figure 13: **The selective backward pass changes only the logit gradients.** WBench case 6, 14×24 spatial grid. Rows: full and selective backward passes. Columns: row norms of logit gradients, Value gradients, and total output gradients U. The fixed output-gradient probe is $G = 2(R-V)/(nd)$ over n rows and d descriptor channels. The selective pass replaces $\Delta_p^\top u_p$ by $\Delta_p^\top g_p$ only in the logit gradient; Value and output gradients remain unchanged. Each column shares one square-root color scale across its two rows; each scale is specific to its gradient quantity. See Appendix E.3 for the training intervention.

E.4 Visit-Probability and Gradient Measurements

Population and measurement. We use all 500 validation sources, three paired training seeds, and EMA weights at update 200. Each source summary includes every layer, query head, and eligible row in the native call graph. The student and complete native teacher share conditions and shareable initial noise; final-clean-latent MSE uses their common VAE coordinates. This diagnostic measures agreement between the student and the fully sampled native teacher. For each eligible attention row i, TV is $\frac{1}{2}\|W_i - A_i\|_1$: the distance between CausalFold weights and standard attention weights at fixed attention logits. Case summaries average these row-wise distances across the native calls, including self-only instances with zero distance. Table 5 reports three-run means of TV, MSE, and within-run Pearson correlation, without displaying SDs or intervals. The underlying

10,000-resample paired source-cluster analysis describes case uncertainty conditional on these runs, separately at each NFE.

Table 5: Forward change and final-latent error. TV, MSE, and within-run Pearson r are three-run means. Bold marks the lowest MSE within a multi-budget model group. TV and r are diagnostics, not quality rankings. Wan budgets use separate training; MAGI budgets share weights. Backbone references are given in Tables 1–2.

NFE	TV	MSE	Mean r
Wan2.1, 1.3B; separate training by NFE			
1	0.1532	0.2211	-0.56
2	0.1044	0.1819	-0.50
4	0.0707	**0.1732**	-0.40
MAGI-1 shortcut, 4.5B; shared weights			
8	0.0335	0.2009	-0.40
64	0.0232	**0.1693**	-0.38
HunyuanVideo-1.5, 8B via minWM			
4	0.1193	0.1457	-0.33

TV measures changes in Value weights; MSE measures final-latent agreement with the teacher. Correlations describe case variation at a fixed budget. Wan budgets use separately trained pairs; MAGI holds weights fixed.

Visit probability, candidate differences, and realized gradients. The identity $\partial r_i/\partial z_p = P_{ip}\Delta_p$ holds for each permitted pair of a later row i and a writer row p. Norms are computed before averaging, so pairing keeps how often each sensitive writer is actually visited. With n_c permitted pairs in case c, Equation (24) compares paired sensitivity with the product of separate means; Figure 14(a) plots this ratio.

$$q_c = \frac{n_c \sum_{(i,p)\in c} P_{ip}\|\Delta_p\|_2}{\left(\sum_{(i,p)\in c} P_{ip}\right)\left(\sum_{(i,p)\in c} \|\Delta_p\|_2\right)}. \tag{24}$$

A ratio below one means separate averages overstate paired sensitivity. The later-block Jacobian is the case mean of pairwise spectral norms $\|P_{ip}\Delta_p\|_2$; the writer gradient averages $\|\zeta_p\|_2$, including gradients from readers in later calls (Equation (23)).

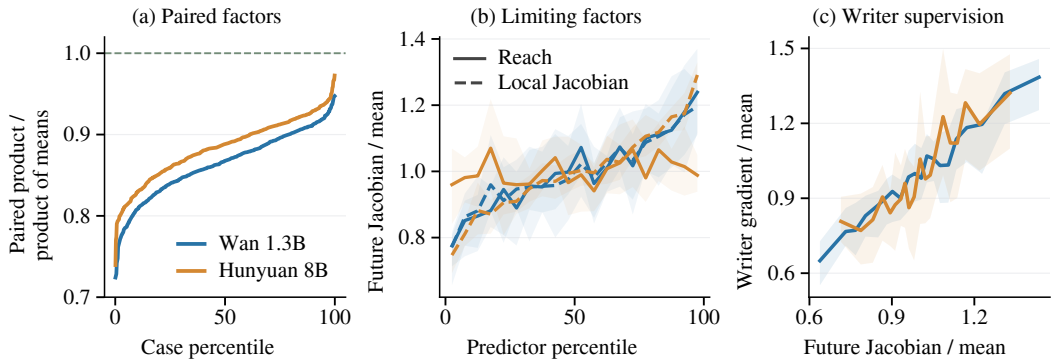

Figure 14: **Visit probability and candidate differences explain different backbone responses.** Wan 1.3B and Hunyuan 8B, four NFE; "reach" in the panels is the visit probability P_{ip}. (a) Averaging paired products $P_{ip}\|\Delta_p\|_2$ differs from multiplying their separate means. (b) Visit probability is more informative for Wan, while the candidate term $\|\Delta_p\|_2$ is more informative for Hunyuan. (c) The later-block Jacobian of writers covaries with the gradient entering through cache reads. Each model uses 500 validation cases and three paired runs. Panel (a) sorts case-wise ratios; panels (b,c) connect means in 20 equal-count predictor bins, with shading for the 25th–75th percentiles. Definitions and unbinned statistics are in Appendix E.4.

For Wan at four NFE, the within-run correlation of the later-block Jacobian with visit probability is 0.62 (95% case interval $[0.59, 0.65]$), with $\|\Delta_p\|_2$ is 0.50 ($[0.46, 0.53]$), and with the writer gradient is 0.71 ($[0.69, 0.74]$). For Hunyuan the corresponding values are 0.09 ($[0.04, 0.14]$), 0.85 ($[0.84, 0.87]$), and 0.46 ($[0.42, 0.50]$). Visit probability limits the later-block gradient on Wan, and candidate differences limit it on Hunyuan.

MAGI-1 uses writers that attend only to their own block. At eight NFE, its writer-to-writer visit probability and Jacobian are exactly zero, while $\|\Delta_p\|_2$ at writer rows is $(5.09 \pm 0.03) \times 10^{-3}$ and the cache-read contribution to writer logits is $(1.50 \pm 0.05) \times 10^{-7}$. At 64 NFE the latter is $(1.10 \pm 0.03) \times 10^{-7}$. The constant writer-to-writer quantities have undefined correlations. The nonzero cache-read contribution is the reader term $G^{\text{read}}_{\mathcal{W}}$ in Equation (23).

Figure aggregation. We average the three runs per case before plotting. Figure 5 uses 20 equal-count TV bins and joins bin means; its dashed line joins overall NFE means. Figure 14(a) sorts the 500 case ratios. Panels (b,c) use 20 equal-count predictor bins; each model's Jacobian and writer gradient are normalized by their respective case means. Shading spans the within-bin 25th–75th percentiles. Correlations use all unbinned cases separately within each run.

E.5 Training Trajectories of Forward and Backward Contributions

We measure how the contributions of output reads and of gradients through them change during training. Four models are compared: Standard-KV (KV), output reads with detached history (R), raw-Value reads with the CausalFold backward rule (V), and CausalFold (CF). The detached-history control stops the entire gradient through cached outputs; the future-logit-off control in Appendix E.3 removes only its contribution to video attention logits.

Let $\ell_{\mathrm{KV}}, \ell_{\mathrm{R}}, \ell_{\mathrm{V}}, \ell_{\mathrm{CF}}$ denote the corresponding training losses for a paired prompt, seed, and training checkpoint. The paired isolated (iso) and conditional (cond) contrasts are

$$\gamma^{\mathrm{iso}}_{\mathrm{fwd}} = \ell_{\mathrm{KV}} - \ell_{\mathrm{R}}, \qquad \gamma^{\mathrm{iso}}_{\mathrm{bwd}} = \ell_{\mathrm{KV}} - \ell_{\mathrm{V}},$$
$$\gamma^{\mathrm{cond}}_{\mathrm{fwd}} = \ell_{\mathrm{V}} - \ell_{\mathrm{CF}}, \qquad \gamma^{\mathrm{cond}}_{\mathrm{bwd}} = \ell_{\mathrm{R}} - \ell_{\mathrm{CF}}.$$

Each contrast is averaged over 180 prompts and three paired training seeds at 0, 25, 50, 75, and 100% of training. At completion, the isolated forward and backward reductions are 4.06×10^{-3} and 13.07×10^{-3}; their conditional counterparts are 7.66×10^{-3} and 16.67×10^{-3}. The larger backward contrasts emerge in the later checkpoints.

F Algorithm and Native Generation Settings

F.1 Complete Algorithm

Algorithm 1 states the complete method. Lines 2–18 implement Equations (1)–(5) for one layer, one query head, and one model call; lines 20–31 give streaming generation; the last procedure gives one generator update with its fake-score updates. Matrices $Q, K \in \mathbb{R}^{N \times d_k}$ and $V, R, G, U \in \mathbb{R}^{N \times d_v}$ hold the active tokens of a call, and $K_{\mathcal{H}}, R_{\mathcal{H}}$ hold the cached entries visible under its mask. A bar denotes a reverse-mode derivative, $\odot$ the elementwise product, and sg stop-gradient. $\mathcal{C}[t] = \{(K_t^{l,h}, R_t^{l,h})\}_{l,h}$ is the cache entry of block t over layers l and heads h. The trace $\mathcal{T} = \{(x_e, \sigma_e, c_e, M_e, \mathcal{H}_e, \mathrm{producer}(\mathcal{H}_e))\}_e$ records, for every call e, its inputs, mask, visible cache entries, and the producing call of each entry. The generator $\mathcal{G}_\theta$ returns the clean prediction of a block, and G always denotes the output-branch gradient of Section 4.3.

Algorithm 1 gives derivatives at the projection boundary; the standard projection backward pass maps them to X, W_Q, W_K, W_V and sums shared-head derivatives under grouped-query attention, where every query head has its own R. The full mode ($\chi = \mathrm{full}$) is the exact derivative of Equation (2); the selective mode is the training control of Appendix E.3. In line 45, v_ψ is the native flow output of the fake score, from which f_ψ obtains its clean prediction by the native conversion.

Native calls and cache commitment. Line 5 is the per-call form of Equation (2). For a call e with visible cache entries $\mathcal{H}_e$,

$$R_{e,t} = A_{e,tt} V_{e,t} + \sum_{s<t,\ s \text{ active}} A_{e,ts} R_{e,s} + \sum_{w \in \mathcal{H}_e} A_{e,tw} R_w. \tag{25}$$

Each candidate belongs to exactly one of the local, earlier-active, or cached sets. Joint-attention prefixes occupy their native causal block and normalization domain. Native head concatenation, output branches, and the final prediction head return the native flow or velocity, which $\mathcal{G}_\theta$ converts to its clean prediction. Separate cache-writing events and overlapping active chunks follow the native schedule; provisional outputs belong to their producing call.

Trace and differentiable replay. Lines 33–37 record a rollout without gradients and replay, with θ fixed at its recording value, every call in the ancestor closure $\mathcal{E}$ of the loss. Replay restores the

Algorithm 1 CausalFold: attention, streaming generation, and training

Input: Generator $\mathcal{G}_\theta$ whose attention layers call FOLDATTN; noise levels $\sigma_n > \cdots > \sigma_1$; sampler Ψ; native forward noising Φ; conditions $\{c_t\}_{t=1}^{S}$ with negative text c_t^-; frozen real score d_{real}; fake score f_ψ; guidance λ

Output: Generated latents X_θ; updated parameters θ, ψ

Attention of one layer and one query head in one call; active blocks $1, \ldots, S_a$ in causal order

procedure FOLDATTN($Q, K, V;\ K_{\mathcal{H}}, R_{\mathcal{H}};\ M$)
  Set $A \leftarrow \mathrm{softmax}_{\mathrm{row}}\big(Q[K_{\mathcal{H}}; K]^\top/\sqrt{d_k} + M\big)$ and split $A = [A_{\mathcal{H}}\ \ D + L]$ ▷ Eq. (1)
  $D \leftarrow$ block-diagonal part and $L \leftarrow$ strictly block-lower part over the active blocks
  for $t = 1, \ldots, S_a$ **do**
    $R_t \leftarrow D_{tt}V_t + \sum_{s<t} L_{ts}R_s + A_{\mathcal{H},t}R_{\mathcal{H}}$ ▷ Eqs. (2), (25)
  end for
  return R ▷ Standard-KV: $R_{\mathcal{H}} \to V_{\mathcal{H}}$ and $R_s \to V_s$
procedure FOLDATTNBACKWARD($G, G^{\mathrm{read}};\ \chi$) ▷ G^{read}: from reads in later calls
  $U \leftarrow G + G^{\mathrm{read}}$
  for $t = S_a, \ldots, 1$ **do**
    $U_s \leftarrow U_s + L_{ts}^\top U_t$ for all $s < t$ ▷ Eq. (4)
  end for
  $\bar{V} \leftarrow D^\top U;\ \bar{R}_{\mathcal{H}} \leftarrow A_{\mathcal{H}}^\top U$
  $\widetilde{U} \leftarrow U$; if $\chi = \text{selective}$, set $\widetilde{U}_{p,:} \leftarrow G_{p,:}$ for every video row p ▷ App. E.3
  $\bar{A} \leftarrow \big[\,\widetilde{U}R_{\mathcal{H}}^\top\ \ \Pi_D(\widetilde{U}V^\top) + \Pi_L(\widetilde{U}R^\top)\,\big]$ ▷ row p: $\bar{a}_p = C_p\tilde{u}_p$
  $\bar{Z} \leftarrow A \odot \big(\bar{A} - ((A \odot \bar{A})\mathbf{1})\mathbf{1}^\top\big)$ ▷ row p: $\bar{z}_p = \Delta_p^\top\tilde{u}_p$, Eq. (5)
  $\bar{Q} \leftarrow \bar{Z}[K_{\mathcal{H}}; K]/\sqrt{d_k};\ [\bar{K}_{\mathcal{H}}; \bar{K}] \leftarrow \bar{Z}^\top Q/\sqrt{d_k}$ ▷ $\bar{K}_{\mathcal{H}}, \bar{R}_{\mathcal{H}}$ go to the producing calls
  return $\bar{Q}, \bar{K}, \bar{V}, \bar{K}_{\mathcal{H}}, \bar{R}_{\mathcal{H}}$

Streaming generation with a cache of attention outputs

procedure GENERATE(θ)
  Initialize $X_\theta \leftarrow [\,],\ \mathcal{C} \leftarrow \emptyset$
  for $t = 1, \ldots, S$ **do**
    Sample $x_t^{\sigma_n} \sim \mathcal{N}(0, I)$
    for $k = n, \ldots, 1$ **do**
      Set $\hat{x}_t \leftarrow \mathcal{G}_\theta(x_t^{\sigma_k}; \sigma_k, c_t, \mathcal{C})$ ▷ FOLDATTN reads $(K_{\mathcal{H}}, R_{\mathcal{H}})$ from $\mathcal{C}$
      if $k > 1$ **then** Sample $\epsilon \sim \mathcal{N}(0, I)$; Set $x_t^{\sigma_{k-1}} \leftarrow \Psi(\hat{x}_t, \epsilon, \sigma_{k-1})$
    end for
    X_θ.append($\hat{x}_t$)
    Update cache $\mathcal{C}[t] \leftarrow \{(K_t^{l,h}, R_t^{l,h})\}_{l,h}$ from $\mathcal{G}_\theta(\hat{x}_t; 0, c_t, \mathcal{C})$ ▷ cache-writing call
    Evict entries of $\mathcal{C}$ by the native window rule
  end for
  return X_θ

One generator update and five fake-score updates

procedure TRAINSTEP(θ, ψ)
  Run GENERATE(θ) without gradients and record the trace $\mathcal{T}$
  for calls e in the ancestor closure $\mathcal{E}$ of the final predictions, in execution order **do**
    Set $\hat{x}_e \leftarrow \mathcal{G}_\theta(\mathrm{sg}(x_e); \sigma_e, c_e, \mathcal{C}_e)$, with $\mathcal{C}_e$ the replayed outputs of $\mathrm{producer}(\mathcal{H}_e)$
  end for
  Assemble $\widetilde{X}_G$ from the replayed final predictions $\hat{x}_e$
  Sample $\sigma, \epsilon \sim \mathcal{N}(0, I)$; Set $X_\sigma \leftarrow \Phi(\mathrm{sg}(\widetilde{X}_G), \epsilon, \sigma)$
  Set $d_c \leftarrow d_{\text{real}}(X_\sigma; \sigma, c),\ d_n \leftarrow d_{\text{real}}(X_\sigma; \sigma, c^-),\ d_f \leftarrow f_\psi(X_\sigma; \sigma, c)$ ▷ clean predictions
  Set $d_r \leftarrow d_c + \lambda(d_c - d_n);\ \delta \leftarrow \mathrm{nan_to_num}\big((d_f - d_r)/\,\mathrm{mean}\,|\widetilde{X}_G - d_r|\big)$
  $\mathcal{L}_G \leftarrow \frac{1}{2}\,\mathrm{mean}\big[(\widetilde{X}_G - \mathrm{sg}(\widetilde{X}_G - \delta))^2\big]$ ▷ Eq. (27)
  $\theta \leftarrow \mathrm{AdamW}(\theta, \nabla_\theta\mathcal{L}_G)$ with FOLDATTNBACKWARD in every replayed call; update EMA
  for $m = 1, \ldots, 5$ **do**
    Set $X \leftarrow \mathrm{sg}(\text{GENERATE}(\theta))$; Sample σ, ϵ; Set $X_\sigma \leftarrow \Phi(X, \epsilon, \sigma)$
    $\psi \leftarrow \mathrm{AdamW}\big(\psi, \nabla_\psi\,\mathrm{mean}[(v_\psi(X_\sigma; \sigma, c) - (\epsilon - X))^2]\big)$ ▷ MAGI-1: $X - \epsilon$
  end for
  return θ, ψ

recorded noise, masks, and conditions and uses the stopped input $\mathrm{sg}(x_e)$. Each cache read uses its recorded producer, and checkpoint recomputation restores producers whose physical slots have been evicted. The loss therefore sends gradients through every output branch and recorded read edge; the terms $\bar{K}_{\mathcal{H}}$ and $\bar{R}_{\mathcal{H}}$ of lines 13 and 17 carry the gradients of readers in later calls to their writers, where $\bar{R}_{\mathcal{H}}$ enters G^{read} (Appendix E.2).

Native update schedule. Each scheduled update draws its own rollout, and generator and fake-score updates follow the native 1:5 ratio. The fake-score target is $\epsilon - X$ for Wan, Hunyuan, and EchoWM, and $X - \epsilon$ for MAGI-1. Fixed image coordinates and EchoWM's valid joint audio/video coordinates follow native loss masks. Appendix G specifies score conversion, guidance, optimization, and EMA. Standard-KV follows the same trace and replay procedure with ordinary AV and (K, V) entries.

F.2 Model Boundaries and Inference Settings

Wan. The small model uses Wan2.1-T2V-1.3B with the framewise Causal Forcing initialization and its native mixed text/image protocol. The default sampler uses four evaluations at 480×832 and 16 frames per second. A short output contains 81 RGB frames and 21 latent frames. Its 30 attention layers have 12 query and 12 Key/Value heads of width 128, with $1,560$ spatial tokens per latent frame. The physical attention window holds 20 historical latent frames plus the current frame. BF16 Key/Value or Key/output storage across 21 slots is

$$\frac{30 \cdot 21 \cdot 1560 \cdot 12 \cdot (128 + 128) \cdot 2}{2^{30}} = 5.624 \text{ GiB}. \tag{26}$$

This analytical tensor-size calculation is reported separately from peak GPU memory. Each committed latent frame also has a cache-writing call (clean writer) at zero noise. Thus the short sequence uses 84 denoising calls and 21 writer calls, totaling 105 DiT calls. A 961-frame long output has 241 latent frames, giving $964 + 241 = 1,205$ calls. The separate large model uses the native one-evaluation Wan 14B I2V system initialized from AAD-1. The two model sizes are evaluated in separate groups.

MAGI. The selected 4.5B distilled model uses BF16, 720×720 resolution, 24 frames per second, and eight native shortcut evaluations (Sand.ai et al., 2025). Its 34 layers use 24 query heads and 8 Key/Value heads of width 128. The T2V interface requests 72 RGB frames. The I2V interface requests 68 additional frames with its initial prefix, producing three native chunks and 18 latent frames. The native overlapping schedule is retained. Clean writers use the self-only range discussed in Appendix E. The sixty-four-evaluation setting reuses the same weights.

Hunyuan. The 8B HunyuanVideo1.5 TI2V model uses the released minWM I2V route, with four evaluations, 77 RGB frames, 16 frames per second, and 480×832 resolution (Zhao et al., 2026). Each chunk contains four latent frames. The model has 54 double-stream blocks, width $2,048$, and 16 heads of width 128. Qwen, ByT5, and SigLIP conditioning tokens remain in their original prefix. Attention folding includes all original prefix and video candidates in the same softmax. This route uses native text and image conditioning.

EchoWM. Flash uses the released causal LTX-2.3-derived action model with its initial image, caption, and native camera ActionDSL (Zhang et al., 2026). It produces 1280×704 video at 24 frames per second using four evaluations, three latent frames per chunk, seven sink frames, and nineteen local frames. The main video attention has 32 heads of width 128. Folding changes only its Value contraction before the original $2\,\mathrm{sigmoid}$ head gate. UCPE camera processing and all text/audio attention remain native. Audio remains part of the joint computation even when the exported video has no audio track. The DiT contains approximately 21.81B parameters.

NFE counts denoising evaluations under each native schedule. Clean-writer calls are counted separately. Appendix F.3 reports measured complete-output latency, first-output latency, and peak memory; the cache sizes and call counts above are analytical quantities.

Figure 15 links the native-action setting to a return sequence in an architectural interior. The first and last views and matched details provide a concrete reading of scene consistency in Table 4, alongside the trajectory-level evaluation.

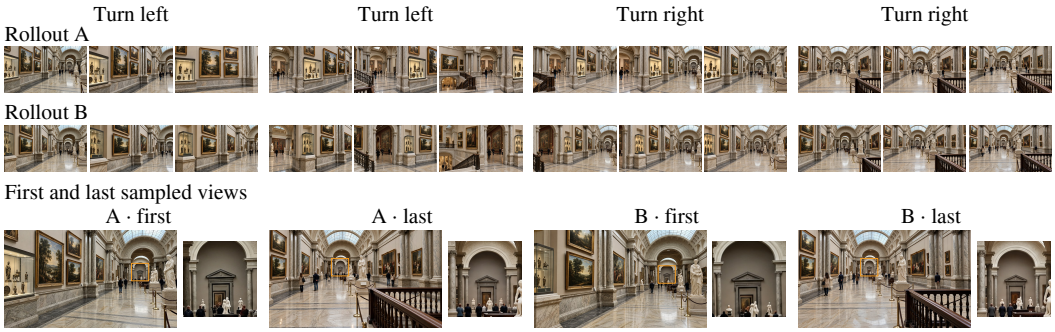

Figure 15: **Following online camera actions and revisiting scene geometry.** Two rollouts use the same sequence: turn left, turn left, turn right, turn right. The temporal strips read from left to right. First and last sampled views, together with the enlarged architectural details, support a visual comparison of the departure and return views. These examples complement the native-action scores in Table 4.

F.3 Runtime Measurement

Table 6 measures the cost of generating each model's native output, pairing CausalFold with Standard-KV at the same shape and sampling schedule. Complete-output time answers how long the entire video takes; first-output time measures when a new decodable segment becomes available. The distinction matters for systems that release a prefix while continuing the rollout.

Table 6: Execution cost at native output shapes. Entries are means of three run summaries. Complete/first are seconds to complete output/first new decodable output; peak is allocated memory (GiB) on any participating GPU. Lower is better within each matched pair. Model references are in Tables 1, 2, and 4.

Variant	GPUs	Execution		
		Complete	First	Peak
Wan2.1, 1.3B / T2V, 81 RGB				
Standard-KV	1	**18.99**	**0.955**	**21.13**
CausalFold	1	20.02	1.011	22.40
Wan2.1, 14B / I2V, 81 RGB				
Standard-KV	4	**38.17**	**8.278**	**39.28**
CausalFold	4	41.55	8.471	41.81
MAGI-1 shortcut, 4.5B / I2V, 72 RGB				
Standard-KV	4	**93.27**	**38.226**	**31.99**
CausalFold	4	110.09	45.176	35.66
HunyuanVideo-1.5, 8B / I2V, 77 RGB				
Standard-KV	4	**38.16**	**10.859**	**42.83**
CausalFold	4	43.28	11.582	45.04
EchoWM Flash, 21.81B / action, 96 new RGB				
Standard-KV	4	**208.31**	**55.857**	**58.71**
CausalFold	4	239.59	61.386	60.09

Environment and timing. NVIDIA A100 80GB SXM4, BF16, TF32 disabled, inference micro-batch one; Wan 1.3B uses one GPU and the other models four. Warm complete-output timing includes conditions, all denoising/writer calls, VAE decoding, and required transfers; loading, compilation, queueing, and MP4 writing are excluded. First output means the first new decodable output, excluding any supplied image. Each run summarizes 500 calibration sources; EchoWM instead averages all 617 turns of 158 navigation cases, retaining joint audio computation. Timing is repeated in three runs.

For the matched CausalFold/Standard-KV timings in Table 6, complete-output overhead is 5.4% for Wan T2V, 18.0% for MAGI I2V, and 13.4% for Hunyuan I2V, computed as the ratio of run means minus one.

F.4 Complete-Output and First-Output Operating Points

Figure 16 combines TC-Bench TCR from Table 1 with complete-output and first-new-output latency for seven Wan systems. Timing averages three runs, each over 500 calibration sources, on one A100 80GB (Appendix F.3). Each system keeps its native sampling protocol. Bubble areas encode complete-generation DiT calls: 105 for CausalFold/Standard-KV, 35 for Self Forcing/LongLive/Causal Forcing, 18 for Rolling Forcing, and 17 for ASD.

The analytical Wan protocol uses 105 calls and a 5.624-GiB persistent payload (Equation (26)). MAGI's analytical twofold payload ratio follows from its 24 query heads and 8 KV heads. Measured peak device memory additionally includes temporary computation and reflects native offloading.

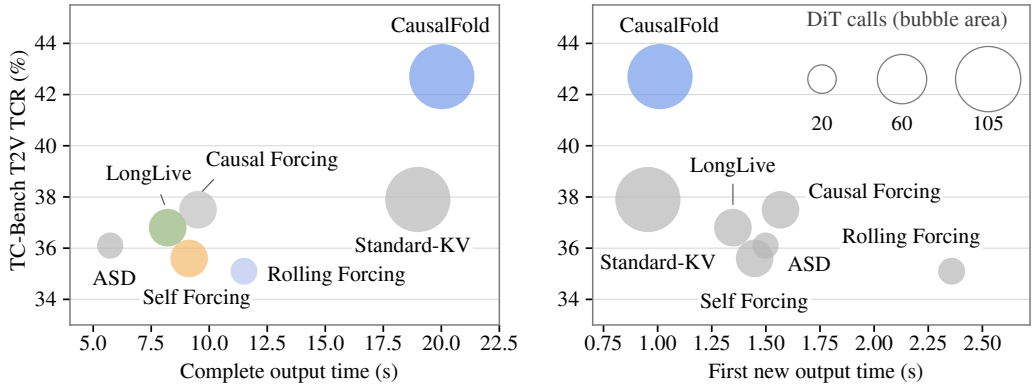

Figure 16: **The first-output and complete-output boundaries produce different latency rankings.** TC-Bench TCR from Table 1: 150 cases, five videos per case, three seeds. Left: complete-output time; right: first-new-output time; means of three calibration runs. CausalFold increases TCR by 4.8 points with 5.4% longer complete-output time (20.02 versus 18.99 seconds). Bubble area denotes physical DiT calls, including clean writers.

G Training Objective and Comparison Settings

The protocol uses a deduplicated pool of 70,000 OpenVid-1M sources for training, followed by disjoint sets of 500 sources for validation and 500 for numerical and runtime calibration (Nan et al., 2025). All frames from one source remain in the same split. Benchmark prompts, source images, actions, and supervision derived from them are excluded before selection. Wan 1.3B uses a fixed source-level text/image task assignment, while the other image-conditioned models retain their native first-image inputs.

G.1 Distribution Matching and Native Score Interfaces

The main task comparisons use Self Forcing's sample-normalized distribution-matching loss (Huang et al., 2025). Let X_G be the complete clean student trajectory, and let d_c, d_n, and d_f be clean predictions from the frozen real model with positive text, the real model with negative text, and the trainable fake-score model. Define

$$d_r = d_c + \lambda(d_c - d_n), \qquad \delta = \mathrm{nan_to_num}\left(\frac{d_f - d_r}{\mathrm{mean}\,|X_G - d_r|}\right),$$
$$\mathcal{L}_G = \tfrac{1}{2}\,\mathrm{mean}\Big[\big(X_G - \mathrm{sg}(X_G - \delta)\big)^2\Big]. \tag{27}$$

The mean is per sample over scalar latent coordinates, followed by ordinary batch averaging. Non-finite entries of the normalized difference are replaced by finite values. The guidance coefficient is the conditional-extra coefficient, with $\lambda = 3$ for Wan, MAGI, and EchoWM and $\lambda = 4$ for Hunyuan.

Negative conditioning replaces only text, retaining other native conditions. Fake predictions use their native unguided branch. The score predictions are stopped in the generator update. Their resulting output gradient is propagated through the replay graph in Section 4.3.

For Wan, the real score is the native bidirectional 14B text model, and the fake score uses the corresponding 1.3B or 14B bidirectional model. An image-conditioned student's first image reaches this text score through X_G, without adding an image encoder to the score network. Loss noise uses a uniformly drawn integer from 0 through 999, a time shift of 5, and clipping to $[20, 980]$. The original score wrappers retain their own time-to-noise lookup and flow-to-clean conversion, including their shift of 8.

For MAGI, real and fake scores start from the native 4.5B base model. The score call processes all three chunks at the shared time $t = 1 - \sigma$, preserving the base model's chunk mask and converting velocity by $d = X_\sigma + \sigma v$. The base noise-to-clean window follows its four native stages, with

time boundaries $1/46$, $1/10$, and $3/10$ and window sizes $5, 4, 3, 2$. The score network evaluates the noised trajectory at the shared time, and the fake-score update differentiates through that evaluation.

For Hunyuan, both scores use its released bidirectional TI2V model and the original image, Qwen, ByT5, and SigLIP inputs. Loss times follow its native four-interval sampling and scheduler conversion, with training guidance $d_r = 5d_c - 4d_n$. EchoWM scores preserve joint audio/video inputs, native per-token noise masks, camera conditions, and fixed initial-image coordinates. For modality m, its native noising is $X_\sigma^m = (1 - \sigma M^m)Z^m + \sigma M^m \epsilon^m$, and clean conversion uses $d^m = X_\sigma^m - \sigma M^m v^m$. The loss averages over the concatenated valid generated audio/video scalar coordinates. Fixed image coordinates are masked out of the loss.

The fake score learns the original flow target on stopped student trajectories. Its sign follows the native time direction: $\epsilon - X_G$ for Wan, Hunyuan, and EchoWM, and $X_G - \epsilon$ for MAGI. Each generator DMD evaluation uses two real-score branches and one fake-score branch without score-parameter backpropagation. Fake-score updates use independently sampled student trajectories and include fake-parameter gradients. The teacher/student diagnostic in Appendix E.4 uses complete teacher sampling trajectories.

G.2 Optimization and Matched Comparisons

The main task comparisons use AdamW, a batch of 64, and 200 generator updates with five fake-score updates per generator update. The generator learning rate is 10^{-6} (4×10^{-7} for Wan 14B and EchoWM); the fake-score learning rate is 4×10^{-7}. Evaluation uses EMA weights with decay 0.99 after update 200.

G.3 Layer Placement and Early Adaptation

Figure 17 uses a 12-layer controlled configuration. Lower, middle, upper, and alternating placements each modify four layers, allowing location to be compared at a fixed layer count. The all-layer setting modifies all twelve. Points report mean semantic-quality and temporal-quality scores over 180 prompts and three training seeds, scaled to 0–100; higher is better on both axes.

Segments connect the initial state to 25 adaptation updates. All five placements increase both scores.

At update 25, all-layer completion reaches 82.86 semantic quality and 87.96 temporal quality; alternating placement reaches 82.22 and 86.72 with four layers. Upper-layer adaptation gains 1.26 semantic points; alternating-layer adaptation gains 1.05 temporal points.

Numbering the twelve layers by forward depth from 0 to 11, the contiguous thirds are Lower $\{0, 1, 2, 3\}$, Middle $\{4, 5, 6, 7\}$, and Upper $\{8, 9, 10, 11\}$; All layers uses $\{0, \ldots, 11\}$. For each placement and update count, each axis is $100/(180 \cdot 3)$ times the sum of its scores over the 180 prompts and three seeds. This gives every prompt equal weight at both adaptation endpoints.

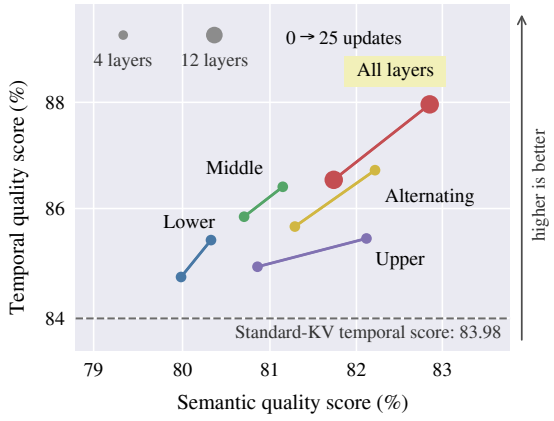

Figure 17: **Equal-count placements isolate location effects.** Zero to 25 updates; means over 180 prompts and three seeds. Bubble area encodes four or twelve layers. Dashed line: Standard-KV temporal score after 25 updates, 83.98.

The four-layer settings compare placement at a fixed modification count. Their different endpoints therefore reflect which part of the network is adapted under the same short training budget. The all-layer point also changes the number of modified layers, so its larger joint score reflects a broader intervention. Reading both axes together preserves the distinction between semantic quality and temporal quality: progress on one axis can differ in size

from progress on the other. This controlled comparison complements the native-backbone results, where layer counts and writer protocols vary together.

G.4 Distribution of Motion Gains Across Training Recipes

We compare Self-Forcing DMD and Causal-Forcing CD/DMD at update 50. For each recipe, the gain is CausalFold minus Standard-KV on VMBench Average, in percentage points. We pair the 180 prompts and three training seeds, average each prompt's three differences, and sort the gains in descending order. The associated upper-tail percentage is $100k/180$ for rank k.

Mean gains are 4.52 and 2.28 points for Self-Forcing and Causal-Forcing, respectively. Negative gains occur for $15/180$ and $46/180$ prompts. The distribution therefore identifies both the magnitude and the coverage of the changes under each recipe. Median gains are 4.51 and 2.34 points, respectively. Both are within 0.07 points of their corresponding means.

H Benchmark Aggregation and Statistical Units

H.1 Tasks and Independent Units

VBench. T2V uses 946 prompt entries (944 distinct normalized texts): 25 videos for each of 75 flickering entries and five for each other entry, totaling $6,230$ videos per seed. Evaluation retains the original 16 dimensions and aggregate weights (Huang et al., 2024); duplicate prompts share a statistical cluster. I2V uses $1,118$ image/text pairs from 355 source images, with five videos per pair, the ten native dimensions, and metric-specific eligibility rules (Huang et al., 2026). All prompt variants of a source image form one cluster.

T2V-CompBench and TC-Bench. T2V-CompBench has seven published categories with 200 prompts each, totaling $1,400$ prompts (Sun et al., 2025). Each category is evaluated separately. Table 1 reports all seven categories, retaining their separate definitions rather than combining them into a new score. TC-Bench has 150 T2V and 120 I2V cases, each with five generated videos (Feng et al., 2024). Transition completion requires the prescribed assertions to be jointly satisfied, excluding the benchmark's other-object checks. TC-Score retains the native assertion-based definition. Its I2V variant combines the specified assertion component with adjacent-frame CLIP consistency using the native two-thirds/one-third weighting. Resampling groups all assertions and generated videos from the same prompt or source-image case.

VideoPhy-2 and StoryEval. VideoPhy-2 evaluates all 591 dense captions derived from 197 actions (Bansal et al., 2026). Semantic adherence and physical commonsense use the native five-point scales. Joint accuracy counts a video only when both scores are at least four. Resampling keeps the three captions of each action together. StoryEval contains 423 stories with two to four events (Wang et al., 2025). Event correctness is averaged within each story and then across stories. The complete story is the statistical unit. The full story conditions generation from the start, and all its events remain grouped in statistical analysis.

WBench. Navigation uses 158 cases and 617 turns, each generating four seconds of new video from the current action (Ying et al., 2026). The sixteen non-navigation turns in nine mixed cases use the official zero-camera mapping. Evaluation applies the nineteen camera-compatible metrics and their native category aggregation, with Doubao Seed2.0 Lite as the semantic judge. All turns, trajectory points, and frames from a case remain grouped.

Native long-video scope. Table 7 uses the VBench-Long whole-rollout aggregation and the complete native durations recorded for each system. Standard-KV and CausalFold use 60.06-second outputs; the other listed systems use 60.56 seconds except CausVid's 63-second segmented rollout. This evaluation is separate from the 30/60-second segment cohorts in Appendix I.2. Native NFE is task-specific and excludes clean writers.

H.2 Uncertainty and Pairing

Tables report means of three seed-specific benchmark aggregates with native weights and eligibility rules; the displayed summaries are means. The diagnostic Pearson coefficient is also averaged

within runs. Primary judgment counts and execution metadata are retained where they define the experiment, as part of the experimental specification. CausalFold/Standard-KV pairs share initialization, source order, and training seed within each backbone; fixed external models vary generation seed. Diagnostic intervals use $10,000$ paired validation-source bootstrap draws conditional on the three trained models (Appendix E.4). Runtime averages calibration sources within each run.

I Temporal Composition, Long Video, and Perceptual Outcomes

I.1 Entity-Specific Temporal Composition

Figure 18 shows a dynamic-attribute transition during rotation. The corresponding surface details separate the requested color change from the change in viewpoint.

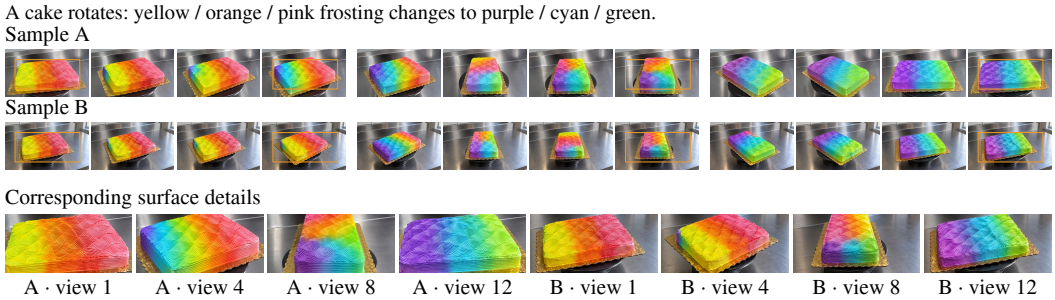

Figure 18: **Dynamic attributes evolve while the object moves.** A rotating cake changes from yellow, orange, and pink frosting to purple, cyan, and green. Two sample sequences read from left to right. The enlarged surface details correspond to views 1, 4, 8, and 12 in each row, allowing the changing colors to be inspected alongside rotation.

The seven-category panel in Table 1 includes entity-specific temporal binding and the non-temporal composition endpoints. Wan's dynamic-attribute and motion-binding scores rise from 42.5/46.8 to 46.9/51.1, respectively, under the benchmark's category-specific definitions.

I.2 Failure Accumulation and Time-Resolved Quality

Table 7 first provides a complete native long-video comparison, retaining each external method's NFE and output duration. Within the matched Wan pair, quality changes from 81.6 to 83.4, subject consistency from 91.8 to 95.3, and background consistency from 93.5 to 96.2; dynamic degree changes from 71.5 to 70.8. External rankings remain mixed: LongLive has higher quality, while Causal Forcing has higher semantics. The whole-rollout comparison therefore yields endpoint-specific rankings. CausVid uses a three-evaluation native long-video workpoint, distinct from its four-NFE short-video entry in Table 1.

Table 7: Native long-video comparison on VBench-Long. Scores are mean percentages, higher is better; bold marks the best displayed value. Subject/background denote consistency and flicker denotes the benchmark's temporal-flickering score. CausVid (Yin et al., 2025), Self Forcing (Huang et al., 2025), LongLive (Yang et al., 2026a), Rolling Forcing (Liu et al., 2026), ASD (Yang et al., 2026b), and Causal Forcing (Zhu et al., 2026) retain their complete long-video protocols; lengths are therefore explicit. Only Standard-KV/CausalFold are matched. ASD uses four first-chunk evaluations and one thereafter (1*). This whole-rollout evaluation is separate from the fixed-duration, five-second segment cohorts in Appendix I.2 and Figure 19.

Method	NFE	Length (s)	VBench-Long						
			Qual.	Sem.	Subject	Background	Smooth.	Flicker	Dynamic
Wan2.1-T2V, 1.3B (Wan et al., 2025): native long-rollout protocols, not length-matched external systems									
CausVid (Yin et al., 2025)	3	63.00	77.6	67.8	85.2	89.3	97.4	97.0	65.9
Self Forcing (Huang et al., 2025)	4	60.56	81.2	74.9	91.1	93.5	98.0	97.8	64.9
LongLive (Yang et al., 2026a)	4	60.56	**84.2**	75.0	**95.3**	**96.5**	**98.6**	**98.4**	60.3
Rolling Forcing (Liu et al., 2026)	5	60.56	83.1	71.0	94.4	95.1	98.2	97.8	66.5
ASD (Yang et al., 2026b)	1*	60.56	79.4	73.1	89.4	92.1	97.3	97.1	61.9
Causal Forcing (Zhu et al., 2026)	4	60.56	83.9	**76.2**	94.8	95.8	98.4	98.2	68.0
Standard-KV	4	60.06	81.6	75.3	91.8	93.5	97.6	97.6	**71.5**
CausalFold	4	60.06	83.4	75.4	**95.3**	96.2	98.5	97.8	70.8

Time-resolved evaluation uses separate fixed-duration cohorts: 946 prompts with five samples each for 30 seconds (4,730 videos per method), and 128 prompts with three samples each for 60 seconds (384 videos per method). Figure 19 measures quality in successive five-second segments. Cumulative failure incidence uses the 60-second CausalFold and Standard-KV cohort.

For video v, let T_v be the earliest of the twelve five-second endpoints with an observed failure. A video with no observed failure has $T_v = \infty$. At endpoint t, cumulative incidence is $100 \sum_{v=1}^{384} \mathbf{1}[T_v \le t]/384$; the fixed denominator includes all videos, and each first failure contributes to every later endpoint. At 60 seconds, CausalFold has $20/384$ failures (5.21%), compared with $40/384$ (10.42%) for Standard-KV. The gap increases from 0.78 points at 20 seconds to 4.17 points at 50 seconds, locating most of the separation later in the rollout.

For segment s of video v, the quality composite is $c_{vs} = (4q_{vs} + a_{vs})/5$, where q is the segment's quality component and a its semantic component. Figure 19 plots $100N^{-1} \sum_v c_{vs}$ at each segment endpoint, with $N = 4{,}730$ or 384 for the two durations. CausalFold's advantage over Standard-KV grows from 1.72 to 4.83 points across the 30-second cohort and from 1.83 to 8.47 points across the 60-second cohort. Both methods' scores decrease with elapsed time, with a slower decline for CausalFold in these evaluations.

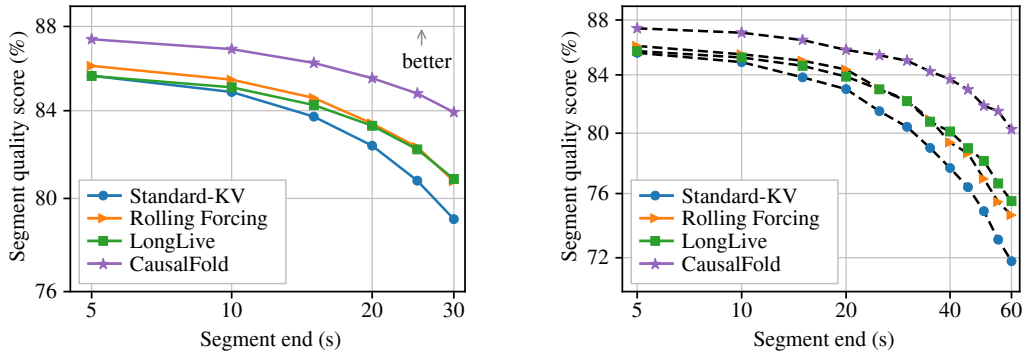

Figure 19: **Segment scores locate the growth of the long-horizon quality gap.** Standard-KV, Rolling Forcing, LongLive, and CausalFold evaluated in five-second segments. Left: 4,730 videos per method over 30 seconds. Right: a separate cohort of 384 videos per method over 60 seconds. The score is the mean of $(4q + a)/5$, scaled by 100, combining segment quality q and semantics a. Lines connect observed segment means.

I.3 Scene Preservation and Perceptual Preference

Table 2 reports changes of 0.0–0.2 points in subject consistency, background consistency, and smoothness across the matched pairs. Its consolidated completion and preservation columns show higher transition completion alongside similar scene-preservation scores and slightly lower dynamic degree. In the 60-second rollouts of Appendix I.2, which exceed the cache window, subject and background consistency rise by 3.5 and 2.7 points (Table 7).

Figure 21 complements these scores with three samples under one scene description, and Table 8 lists the primary judgment counts of the paired human study.

Figure 20 compares CausalFold and Standard-KV on 1,003 prompts and three training seeds, giving 3,009 video pairs. All three raters evaluate every pair on each question, totaling 9,027 judgments for each of the four dimensions. Prompts define the independent units; seed repetitions and multiple ratings are grouped within a prompt.

Table 8: Paired perceptual judgments. CF is CausalFold; KV is Standard-KV. Preference is the mean score with ties worth one-half, in percent. Counts are retained as primary outcomes; bold marks the larger win count within each question.

Dimension	Judgment counts			Pref.
	CF win	Tie	KV win	CF (%)
Temporal consist.	**6,122**	1,284	1,621	74.93
Motion complete.	**5,737**	1,375	1,915	71.17
Text alignment	**5,222**	1,526	2,279	66.30
Visual realism	**4,257**	1,854	2,916	57.43

The preference score is $100(n_{\text{win}} + n_{\text{tie}}/2)/9,027$. CausalFold receives $6,122$ wins and $1,284$ ties for temporal consistency (74.93%), and $4,257$ wins and $1,854$ ties for visual realism (57.43%).

Motion completeness and text alignment receive 71.17% and 66.30% preference, respectively, between temporal consistency and visual realism. Temporal preference aligns with higher transition completion in the task comparison.

For dimension d, let $o_{qkr}^{(d)}$ be 1 for a CausalFold win, $1/2$ for a tie, and 0 for a Standard-KV win on prompt q, training seed k, and rater r. The prompt-level score is $s_q^{(d)} = \frac{1}{9}\sum_{k=1}^{3}\sum_{r=1}^{3} o_{qkr}^{(d)}$; the reported percentage is $\frac{100}{1,003}\sum_{q=1}^{1,003} s_q^{(d)}$. Table 8 reproduces all four plotted percentages from the primary judgment counts.

Each question has 9,027 judgments from 3,009 video pairs. The counts distinguish wins, ties, and losses, with one-half credit for a tie. Each prompt has nine judgments per question, so averaging all judgments and averaging the prompt-level scores give the same point estimate. The four questions describe different aspects of the same generated pairs; their ordering places temporal consistency and motion completeness above visual realism, consistent with the task-level completion gains.

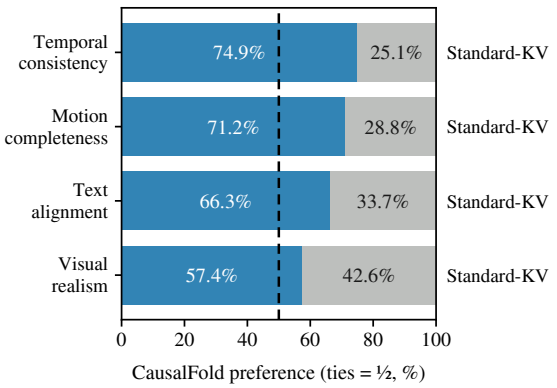

Figure 20: **Paired preference is strongest for temporal consistency.** Blue is CausalFold's preference share over Standard-KV; gray is its complement. Ties contribute one-half. The dashed line marks equal preference at 50%.

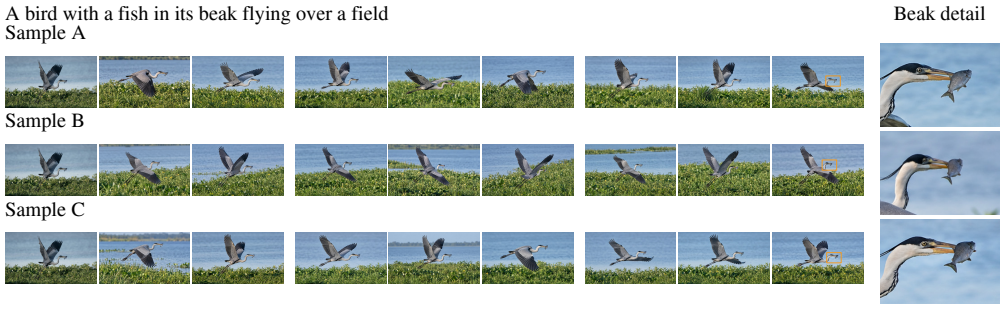

Figure 21: **Scene preservation across repeated samples.** Three CausalFold samples (A–C) for the prompt *A bird with a fish in its beak flying over a field.* Each row shows nine sampled frames in temporal order from left to right. Wing poses and the visible background change along each sequence. Orange boxes mark the beak regions enlarged in the rightmost column, where the fish–beak relation can be inspected across the three samples. The temporal strips show the full scene, while the close-ups expose the local object relation that should be preserved during flight.