Video-Mirai: Autoregressive Video Diffusion Models Need Foresight

Yonghao Yu[1] **Lang Huang**[2,3*] **Runyi Li**[1] **Zerun Wang**[1] **Toshihiko Yamasaki**[1]
[1]The University of Tokyo [2]National Institute of Informatics [3]SOKENDAI
{y_yu, r_li, ze_wang, yamasaki}@cvm.t.u-tokyo.ac.jp, lang@nii.ac.jp

Abstract

Causal video generators must predict from the past, but they need not learn only from it. In streaming autoregressive video diffusion, each emitted segment becomes a commitment that future segments must preserve. Standard training, however, does not explicitly supervise causal states with future-informed representation targets. This creates what we call a representation-level planning gap: states that fit the current segment may discard identity, layout, and motion information needed for a consistent future. We introduce Video-Mirai, a training-only method that closes this gap without changing causal inference: the generator rolls out causally, a frozen foresight encoder reads the completed rollout non-causally, and a lightweight predictor distills the resulting stopped-gradient targets into causal states. Future frames supervise representations, never generator inputs. At inference, the encoder and predictor are discarded, leaving the original architecture, per-step FLOPs, and KV-cache behavior unchanged. Video-Mirai improves a strong Causal-Forcing baseline on 5-second VBench from 83.8 to 84.6 in terms of Total Score. On 30-second rollouts beyond the training horizon, subject consistency improves from 84.9 to 88.5 and background consistency from 90.2 to 91.9. Ablations identify future-conditioned targets as a key ingredient, and probes show that future frames become more decodable from current features. Causality should constrain inference, not representation supervision. Our study highlights that visual autoregressive models need foresight. Project Page: https://y0uroy.github.io/Video-Mirai.

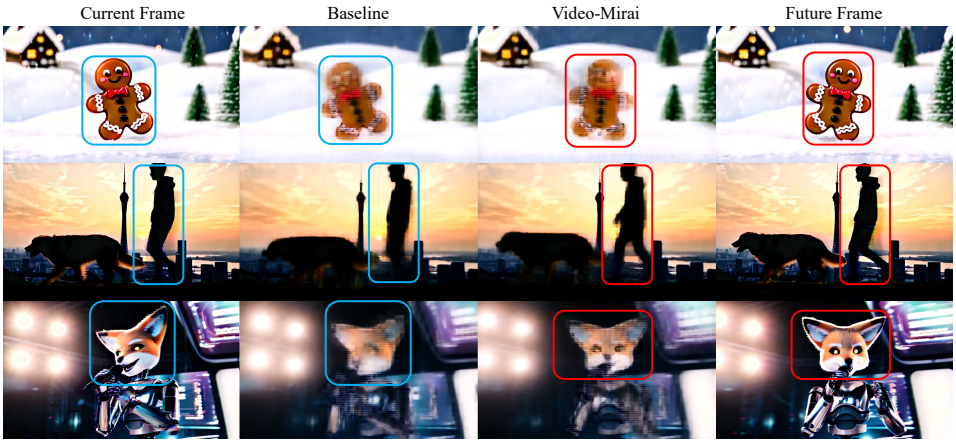


Figure 1: **Video-Mirai makes the future more decodable.** An MLP readout reconstructs future RGB from the frozen causal generator's current hidden state. Left to right: current frame, baseline readout, Video-Mirai readout (ours), future frame. blue/red: regions matching the current/future.

*Corresponding author.

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



Figure 2: **Qualitative evidence of the planning gap.** Frames at t=0s, 2.5s, 5s from the same prompt. The baseline's local plausibility per segment masks abrupt drift between segments: the bear's motion resets, and the woman's identity and background change. Video-Mirai mitigates these breaks.

1 Introduction

Streaming video generation has a hidden burden: every emitted segment becomes a promise that future segments must keep. Autoregressive (AR) video diffusion follows this interface directly: generate a frame or chunk from the previous context, emit it, and continue. This streaming ability is central to low-latency visual synthesis for interactive world modeling [6, 1], game-engine simulation [36], and embodied intelligence [8, 35, 9]. Yet, it also makes local decisions hard to revise. A segment may look correct in isolation while failing to specify what must remain true later. Figure 2 shows typical failures: in the baseline rollouts, a bear's motion unexpectedly jumps between frames; in another example, the people and the background abruptly change between segments. These errors suggest that local visual quality is not sufficient for maintaining identity, layout, and motion over time.

We view this as more than generic error accumulation. Output-level supervision is under-constrained: many hidden states can generate nearly the same plausible current segment, but only some retain the information needed for future consistency. The loss tells the model whether its output looks right; it does not directly ask whether the current state contains the information *future* segments will need. We call this missing constraint a representation-level *planning gap*.

This gap points to a simple design principle: use foresight as supervision, not as input. Future evidence is useful for deciding which current states are good, but it cannot be given to a streaming generator at inference. We introduce Video-Mirai, a training-only objective for AR video diffusion that resolves this tension. During training, Video-Mirai lets future segments supervise the current causal state; at inference, the generator remains strictly past-only. Existing AR video methods improve how models roll out, but they do not close this planning gap. Self-Forcing [19] exposes the model to self-generated histories; Causal-Forcing [45] makes distillation compatible with causal decoding. These methods modify the history distribution and the output target. Video-Mirai targets a different object: the representation carried by the current causal state.

Video-Mirai is deliberately minimal. The generator first rolls out video causally under the same mask used at inference. A frozen foresight encoder then reads the completed rollout and produces future-informed feature targets for the current states. A lightweight predictor maps each causal hidden state to its corresponding foresight target with a cosine loss. Future segments are used only to construct stopped-gradient supervision; they are never provided as generator inputs. After training, the foresight encoder and predictor are discarded. The deployed generator remains exactly causal, with identical architecture, attention pattern, per-step FLOPs, and KV-cache behavior to the baseline.

Video-Mirai improves both short- and long-horizon generation. On the 5-second VBench, Video-Mirai improves the AR video diffusion baseline Causal-Forcing from 83.82 to 84.62 in Total Score. For the 30-second generation, beyond the training horizon, subject consistency improves from 84.93 to 88.47, and background consistency from 90.22 to 91.94. The method also transfers across AR settings,

improving both frame-wise and chunk-wise generation. Beyond generation metrics, representation probes (Figures 1 and 5) show that future frames become substantially more decodable from frozen Video-Mirai features. Figure 2 confirms that this internalized foresight translates to visibly more coherent rollouts, mitigating the planning gap in practice. Unlike prior foresight objectives for image autoregression, Video-Mirai addresses a video-specific planning gap in causal diffusion rollouts, where future supervision must improve temporal coherence while preserving streaming inference and KV-cache-compatible deployment. In summary, our contributions are threefold:

- We formulate foresight prediction as a representation-level objective for causal video generation, using future-aware targets during training while preserving strictly causal inference.
- Video-Mirai improves AR video generation across frame-wise and chunk-wise settings, with gains that also extend to 30-second rollouts beyond the training horizon.
- We identify which foresight source, prediction layer, predictor design, and look-ahead window make the training signal effective, and verify through probes that future content becomes more decodable from frozen features.

2 Related Work

2.1 Video Diffusion Models

Large-scale video diffusion models such as Wan [37], CogVideoX [39], HunyuanVideo [23], and MovieGen [30] generate high-fidelity clips by jointly denoising a temporal window, allowing bidirectional interactions among frames within the clip. This full-window interface supports strong short-range consistency, but it is less natural for low-latency settings where frames or chunks should be emitted as soon as they are generated. AR video diffusion instead generates frames or chunks sequentially from past context, enabling streaming inference and supporting rollouts beyond the training horizon, albeit with increasing risk of drift. CausVid [42], Diffusion-Forcing [8], MAGI-1 [35], and SkyReels-V2 [9] exemplify this causal direction. Other approaches extend pretrained video diffusion models to long videos, including StreamingT2V [16] and FIFO-Diffusion [22]. Together, these works expose the tradeoff behind our study: streaming generation requires local causal decisions, while coherent video depends on identity, layout, and motion remaining predictable over future segments.

2.2 Distillation and Training Techniques for AR Video Diffusion

To deliver high-quality sampling in a few steps, recent work has focused on distilling bidirectional teachers [18, 33, 31, 26] into causal students via Distribution Matching Distillation (DMD) [41, 40], progressively closing several supervision gaps inherent to this process. Self-Forcing [19] closes the exposure-bias gap by performing AR rollout during training, exposing the student to its own imperfect histories under a holistic video-level loss. Causal-Forcing [45] identifies a theoretical flaw in ODE distillation initialization: distilling a bidirectional teacher into a causal student violates frame-level injectivity, thereby preventing the student from faithfully recovering the teacher's flow map. They close this injectivity gap by introducing an AR teacher for ODE initialization. Rolling-Forcing [27] targets error accumulation over long horizons through joint denoising across a rolling window, attention-sink anchors, and training over extended non-overlapping windows. These methods advance the state of the art in causal training stability and past consistency. Our work is orthogonal to theirs. We address the complementary planning gap, namely, a causal generator's inability to anticipate its own future evolution, and our foresight objective can be layered on top of AR video diffusion training paradigms, as our experiments on Self-Forcing and Causal-Forcing demonstrate.

2.3 Representation Alignment and Foresight

Aligning a model's intermediate representations with those of a strong external encoder, dating back to knowledge distillation [17] and intermediate-layer hints [32], has proven more effective for training efficiency and sample quality than pixel-level objectives. REPA [43] regresses mid-layer DiT features onto a pretrained self-supervised encoder such as DINOv2 [28], with follow-ups [25] extending this to end-to-end VAE tuning. A separate line uses representation prediction as a self-supervised pretext task: V-JEPA and its successors [5, 2], building on JEPA [24, 3] and masked image modeling [15], predict masked spatiotemporal regions, with the action-conditioned V-JEPA-2-AC training a predictor

invoked at inference. The intuition that future-aware signals improve training extends beyond vision, e.g., multi-token prediction [13] in language models.

Closest in spirit, Mirai [44] exposes AR *image* generators to future-position information. Video-Mirai extends this to causal *video diffusion*, where the planning gap is sharper: temporal coherence over multi-second rollouts is more demanding than spatial coherence within an image, and the bidirectional/causal architectural asymmetry inherent to video distillation has no counterpart in the image setting. Video-Mirai draws on REPA-style predictor alignment and JEPA-style latent prediction, recombined for a different purpose: future-aware supervision for a strictly causal video model. Unlike REPA, our target is the temporally shifted hidden state from a bidirectional video model. Unlike V-JEPA, our target is the encoder's representation of the model's own future rollout rather than masked regions, and our predictor is discarded after training. The causal model remains strictly causal at inference. To our knowledge, Video-Mirai is the first to use future-conditioned representation alignment as a training-only signal for causal video diffusion.

3 Method

3.1 Preliminaries

We consider AR video diffusion that distills a pretrained bidirectional diffusion model into a few-step causal generator G_θ. A video is a sequence of temporal segments $\mathbf{x} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, where a segment is a single latent frame (frame-wise) or a chunk of consecutive latent frames (chunk-wise). The causal constraint requires:

$$p_\theta(\mathbf{x}) \;=\; \prod_{i=1}^{N} p_\theta(\mathbf{X}_i \mid \mathbf{X}_{<i}), \tag{1}$$

so segment $\mathbf{X}_i$ is generated from $\mathbf{X}_{<i}$ alone, with no access to $\mathbf{X}_{>i}$ at inference. Standard distillation pipelines such as Causal-Forcing [45] realize this through three stages: AR teacher fine-tuning, causal ODE distillation, and asymmetric DMD [41, 40]. We adopt this pipeline as our base.

The factorization in Eq. 1 only constrains each $\mathbf{X}_i$ to be plausible given $\mathbf{X}_{<i}$. It says nothing about whether the hidden state $\mathbf{h}_i^L$ used to generate $\mathbf{X}_i$ retains the information needed to generate *future* segments coherently. For example, a per-segment training objective can be written as follows:

$$\mathcal{L}_{\text{present}}(\theta) \;=\; \mathbb{E}_{\mathbf{x}} \sum_{i=1}^{N} \ell\big(G_\theta(\mathbf{X}_i \mid \mathbf{X}_{<i})\big)\,, \tag{2}$$

where ℓ is any per-segment supervision applied to the model's output distribution (DMD score, flow-matching, etc.). Existing objectives supervise generated outputs, including entire video rollouts, but do not explicitly align intermediate causal states with future-informed representation targets. Video-Mirai adds this representation-level supervision. Many distinct hidden states $\mathbf{h}_i^L$ can drive nearly the same $\mathbf{X}_i$ to optimum [12, 34]: some retain identity, layout, and motion cues that future segments will need; others discard them. The loss has no preference between the two [10]. We call this missing constraint the *representation-level planning gap*: output-level losses do not ask whether $\mathbf{h}_i^L$ is a good state from which to continue. Recent advances in AR video diffusion address adjacent yet distinct gaps, including exposure bias [19], ODE-distillation injectivity [45], and long-horizon error accumulation [27]. None of them, however, explicitly supervises $\mathbf{h}_i^L$. They reshape the history distribution, the distillation initialization, or the rollout protocol, not the supervision applied to the causal state itself. The planning gap is therefore orthogonal to all three. Closing it requires supervising $\mathbf{h}_i^L$ with a *foresight* signal: a target derived from the model's own future rollout.

3.2 Video-Mirai

Figure 3 illustrates an overview of Video-Mirai: while denoising the current segment, the causal DiT generator's mid-depth feature is fed to a predictor ϕ_ω to predict the foresight encoder's feature on a future segment of the same rollout, via a cosine similarity loss:

$$\ell_i^{\text{F}}(\delta) \;=\; 1 - \cos\big(\phi_\omega(\mathbf{h}_i^L),\ \text{sg}[\mathbf{H}_{i+\delta}^{L'}]\big). \tag{3}$$

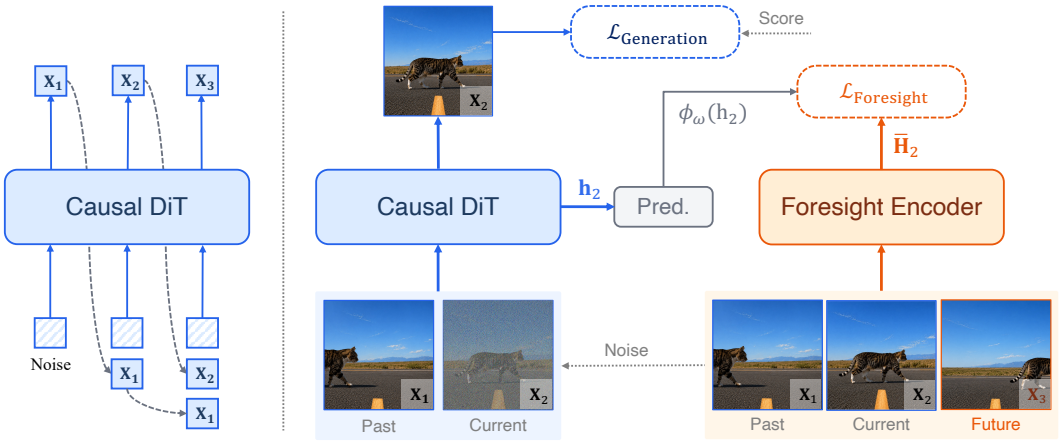


Figure 3: **Overview of Video-Mirai.** We take a three-segment video as an example. The causal DiT denoises $\mathbf{X}_2$ from its noisy version conditioned on $\mathbf{X}_1$ via KV-cache. A frozen foresight encoder processes the causal DiT's clean rollout $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3\}$, including the future segment $\mathbf{X}_3$. A predictor maps the causal DiT's hidden state $\mathbf{h}_2$ into the encoder's space, where the foresight loss aligns it with the encoder's fused hidden state $\bar{\mathbf{H}}_2$, which contains foresight information.

Let $\mathbf{h}_i^L \in \mathbb{R}^{T \times d_c}$ denote the causal generator's hidden state at layer L when generating $\mathbf{X}_i$, and $\mathbf{H}_{i+\delta}^{L'} \in \mathbb{R}^{T \times d_f}$ the foresight encoder's hidden state on the causal generator's rollout segment $\mathbf{X}_{i+\delta}$ at a matched mid-depth layer L', computed when the encoder processes the full rollout $\mathbf{x}$. Two design questions follow immediately: (i) is foresight useful, and how far ahead should the alignment look, and (ii) should multiple offsets be combined by fusing targets or by averaging losses. We answer both before assembling the full objective.

Foresight window. We compare four offset configurations: current only ($\{0\}$), one-segment ahead only ($\{1\}$), current plus one-segment ($\{0, 1\}$), and current plus two-segment ($\{0, 1, 2\}$). In the chunk-wise setting, δ=1 corresponds to a 3-latent-frame look-ahead. As shown in Table 1, current plus one-segment ($\{0, 1\}$) yields the best Quality and Total Scores. Crucially, even current-only alignment carries foresight information because the foresight encoder here is bidirectional, so its hidden state on the current chunk is already conditioned on future chunks. Thus, foresight arises not only from explicit future-offset alignment but also from implicit future information through bidirectional attention. Extending the window to $\{0, 1, 2\}$ further improves Semantic but degrades Quality, indicating that longer-range alignment trades visual fidelity for global coherence.

Target fusion vs. per-offset losses. Given a window $\Delta = \{0, 1, \ldots, K\}$ (K=1 by default), we can either compute one loss per offset and average, or fuse the encoder's features across Δ into a single target and align once. Table 2 compares four predictor designs spanning these two regimes: the pointwise MLP and the DiT predictor, each on either the fused target or a per-offset variant. Two findings emerge. First, fusing across offsets combines features from the current and future segments of the same rollout into a single target, and Table 2 shows that this approach outperforms averaging per-offset losses: the single-head MLP beats Multi-MLP, and the single-head DiT beats DiT-AdaLN. Second, DiT outperforms the pointwise MLP at matched depth, suggesting the alignment benefits from token mixing. We also vary DiT depth among $\{2, 3, 4\}$ blocks and find that 3 blocks give the best Total Score. We therefore adopt a 3-block DiT with target fusion as our default. With this choice, the fused target and per-segment loss are:

$$\bar{\mathbf{H}}_i \;=\; \frac{1}{|\Delta|} \sum_{\delta \in \Delta} \mathbf{H}_{i+\delta}^{L'}, \quad \ell_i^{\mathrm{F}} \;=\; 1 - \cos\big(\phi_\omega(\mathbf{h}_i^L),\; \mathrm{sg}[\bar{\mathbf{H}}_i]\big). \tag{4}$$

Training. At each step, the causal generator unrolls a video $\mathbf{x}$ segment by segment from noise, following the standard few-step denoising loop of Causal-Forcing with the usual KV-cache, and we cache its mid-depth states $\{\mathbf{h}_i^L\}$. The frozen foresight encoder then processes the full rollout $\mathbf{x}$ in a single forward pass at zero diffusion timestep, yielding $\{\mathbf{H}_i^{L'}\}$. For every segment with a near-future target available ($i \le N - K$), $\phi_\omega(\mathbf{h}_i^L)$ is pulled toward the fused target $\bar{\mathbf{H}}_i$ via Eq. 4, combined with

Table 1: **Foresight window comparison.** Effect of the offset set Δ on top of Causal-Forcing.

Setting	Δ	Quality ↑	Semantic ↑	Total ↑
Causal-Forcing	–	84.54	80.93	83.82
Current only	$\{0\}$	85.07	81.55	84.36
1-segment ahead only	$\{1\}$	84.82	81.50	84.15
Current + 1-segment ahead	$\{0, 1\}$	**85.38**	81.59	**84.62**
Current + 2-segment ahead	$\{0, 1, 2\}$	85.11	**82.00**	84.49

Table 2: **Predictor architecture comparison.** Fusion means encoder targets are averaged across Δ before the loss; per-offset means each offset is handled separately.

Architecture	Target handling	Depth	Quality ↑	Semantic ↑	Total ↑
Causal-Forcing	–	–	84.54	80.93	83.82
MLP	Fusion	3	85.09	81.58	84.38
Multi-MLP	Per-offset	3	85.00	81.12	84.22
DiT	Fusion	2	85.14	**81.85**	84.48
DiT	Fusion	3	85.38	81.59	**84.62**
DiT	Fusion	4	**85.41**	81.19	84.57
DiT-AdaLN	Per-offset	3	85.34	81.46	84.56

the asymmetric DMD generation loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Generation}} + \lambda \underbrace{\frac{1}{N-K} \sum_{i=1}^{N-K} \ell_i^{\text{F}}}_{\mathcal{L}_{\text{Foresight}}}, \tag{5}$$

where we use $\lambda = 0.2$ by default. The DMD term scores a re-noised version of the same rollout against a frozen Wan-14B real-score teacher and a Wan-1.3B fake-score critic, following Self-Forcing [19]. The remaining designs, such as the foresight encoder, injection depth L, and loss form, are ablated in §4.2. We also discuss applying Video-Mirai in a different stage in Appendix D.4.

4 Experimental Results

4.1 Setup

Implementation details. We use Wan2.1-T2V-1.3B [37] (henceforth Wan-1.3B) as the causal generator backbone, which generates 81-frame videos at a resolution of 832 × 480. Our main results are reported on top of the three-stage Causal-Forcing [45] pipeline under the chunk-wise setting, where each segment is a chunk of 3 consecutive latent frames. The training procedure in every stage follows Causal-Forcing. We also use Video-Mirai as a drop-in addition to Self-Forcing [19] and to the frame-wise Causal-Forcing variant. Unless otherwise noted, Video-Mirai uses Wan2.1-T2V-14B (henceforth Wan-14B) as the foresight encoder. All distillation runs use 8× H100 GPUs with gradient accumulation 8. The training prompts come from the filtered VidProM [38] extension released by Self-Forcing. Other training settings follow the baseline's recipe. More details are in Appendix A.

Evaluation. We evaluate on VBench [20]. For 5-second generation, each score is the mean over 5 generated videos per prompt across the full VBench prompt set, following the benchmark's standard protocol. For the 30-second long-horizon evaluation, we roll out the same trained checkpoints on 200 randomly selected MovieGen [30] prompts, following the Rolling-Forcing [27] protocol, and report the subset of VBench dimensions that are computable under its prompt-agnostic protocol. Due to the high cost of video distillation, we follow prior work and report paired prompt-level bootstrap significance instead of variance over multiple training seeds. Details are provided in Appendix E.

Table 3: **Injection depth comparison.** Each row pairs a causal generator layer with a foresight encoder layer at matched relative depth α.

Layers (α)	Quality ↑	Semantic ↑	Total ↑
Causal-Forcing	84.54	80.93	83.82
9→12 (0.3)	85.14	81.37	84.39
15→20 (0.5)	**85.38**	**81.59**	**84.62**
24→32 (0.8)	84.77	81.54	84.12

Table 4: **Foresight encoder comparison.** Total Score on 5-second VBench. *Implicit*: bidirectional attention lets the target see later segments; *explicit*: the target includes the next segment. All variants use the default 3-block DiT predictor, layered on Causal-Forcing chunk-wise.

Foresight encoder	Temporal attention	Future access	Offsets Δ	Total ↑
Causal-Forcing	–	–	–	83.82
EMA	causal	none	$\{0\}$	83.71
		explicit	$\{0, 1\}$	83.88
Wan-1.3B	bidirectional	implicit	$\{0\}$	83.86
		implicit + explicit	$\{0, 1\}$	83.96
Wan-14B	causal	none	$\{0\}$	84.06
		explicit	$\{0, 1\}$	84.53
Wan-14B	bidirectional	implicit	$\{0\}$	84.36
		implicit + explicit	$\{0, 1\}$	**84.62**

4.2 Component-wise Analysis

Injection depth. Foresight injection requires choosing both a causal generator layer L and an encoder layer L' . Since the causal generator (30 blocks) and the encoder (40 blocks) differ in depth, we pair layers at matched relative depth: $L = 30\alpha$, $L' = 40\alpha$. Table 3 shows that mid-depth (α=0.5) is best. Shallow layers have not yet accumulated the abstract planning features that benefit from foresight, while deep layers are close to pixel-level reconstruction and have less freedom to be reshaped by an auxiliary objective. We adopt α=0.5 throughout.

Foresight encoder. To examine how much the future contributes, we compare foresight encoders of different sizes with and without access to it: the EMA copy of the causal generator itself, the bidirectional Wan-1.3B and Wan-14B models, and Wan-14B under a 3-latent-frame chunk-wise causal mask that hides later segments. The results in Table 4 show that without future access, alignment does not reproduce the gain: the causal EMA falls below the baseline (83.71 vs. 83.82), and the causal Wan-14B reaches only 84.06. Explicit future access improves every encoder, implicit access lifts Wan-14B from 84.06 to 84.36, and combining both is best (84.62). Encoder capacity matters as well: the causal Wan-14B (84.06) exceeds the bidirectional Wan-1.3B with explicit future access (83.96). We adopt Wan-14B as our default, since it is the strongest encoder and is already the real-score model in the baseline DMD pipeline. A much smaller image encoder from outside the model family, DINO ViT-S/16 [7], also improves the baseline (Appendix D.5).

Foresight loss. Figure 4 compares different foresight losses with or without a Sketched Isotropic Gaussian Regularization term (SIGReg) [4]. Cosine similarity alone works best. We attribute this to two effects. First, MSE forces the generator's feature to match the encoder's direction, scale, and full coordinate distribution, pulling it away from its native feature geometry and dropping the Semantic Score. Second, adding SIGReg consistently degrades the Total Score under both losses, indicating that explicitly shaping the projected feature distribution toward isotropic Gaussianity conflicts with the alignment objective. We adopt cosine similarity without SIGReg as the default. Loss weight ablations are in Appendix D.3.

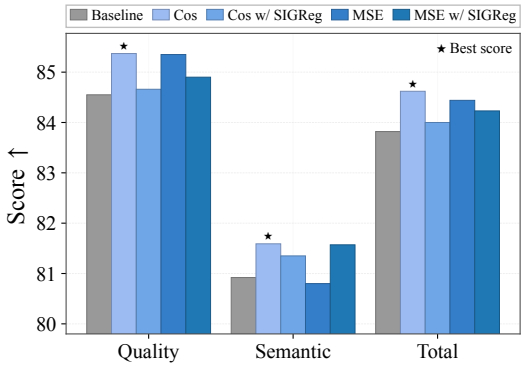


Figure 4: **Foresight loss comparison.** Cosine similarity vs. MSE, with and without the SIGReg.

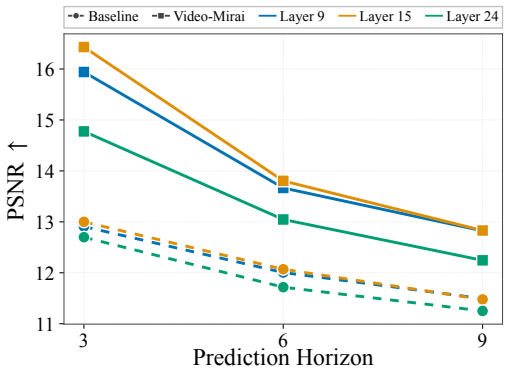


Figure 5: **Future-frame readout fidelity across layers and horizons.**

4.3 Foresight Visualization

All evidence so far is behavioral: VBench scores improve after foresight training. A sharper question is whether foresight is actually *internalized* into the causal generator's representations. We design a representational probing experiment to answer this directly.

Probe setup. We take two frozen causal generators with identical architectures: the Causal-Forcing baseline and Video-Mirai. For each model, we train a small MLP decoder to reconstruct the clean RGB future frame δ chunks ahead from the model's layer-15 hidden state on the *current* chunk. No foresight predictor is used at probing time; the decoder is purely a readout. Decoders for the two models are trained separately within an identical budget. If Video-Mirai has internalized future information, its features should support substantially more faithful future reconstructions.

Qualitative and quantitative readout. Figure 1 shows three probe samples. We can see that Video-Mirai faithfully reconstructs the future frame, whereas the baseline readout remains closer to the current frame, as expected for a feature that lacks foresight. We then quantify the readout fidelity at scale across three feature-extraction layers $L \in \{9, 15, 24\}$ and three prediction horizons of $\{3, 6, 9\}$ latent frames ahead, which means $\delta \in \{1, 2, 3\}$ chunks, reporting MSE, PSNR, and LPIPS. For each layer, we train a separate MLP decoder on the causal generator's hidden state and evaluate on held-out prompts. For brevity, we plot PSNR in Figure 5. The MSE and LPIPS curves, which show the same trend, are provided in Appendix F. Video-Mirai consistently outperforms the baseline at every layer and every horizon across all three metrics. The gap is largest at near horizons, where foresight supervision applies most directly, and persists at a horizon of 9 latent frames (δ=3 chunks), well beyond the training-time foresight window, indicating that foresight internalization generalizes rather than memorizes.

Distribution of futures. The probes above show that Video-Mirai's features support future readout, but the future of a given current is not deterministic. Running the causal generator from the same current with different noise seeds yields a distribution of valid futures. So do the features encode *a* specific future, or the *distribution* of futures? Figure 6 visualizes this. We sample four independent rollouts from the same current under different noise seeds, confirming that the future is genuinely stochastic. For example, the purple ermine's tail ends up at different positions across rollouts. The readout from Video-Mirai's current segment resembles the *average* of these four rollouts. The readout can already capture identity, layout, and motion statistics of the future before any of those future frames are actually generated. This suggests that Video-Mirai's mid-depth features summarize what is about to happen rather than a single sampled continuation.

4.4 System-Level Comparison

Table 5 situates Video-Mirai within the broader landscape of video diffusion models, comparing against bidirectional, AR, and distilled-AR video diffusion models on VBench. We apply Video-Mirai as a drop-in addition on top of three distilled-AR baselines: Self-Forcing chunk-wise and Causal-Forcing in both frame-wise and chunk-wise variants. The results show that Video-Mirai improves

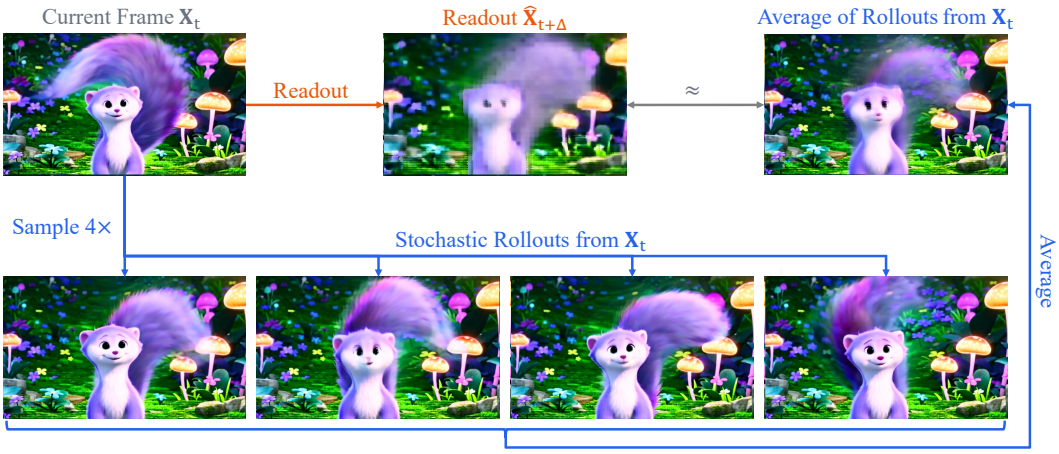


Figure 6: **The readout resembles the average of multiple future rollouts.**

Table 5: **Quantitative comparisons with existing methods.** On 5-second VBench, Video-Mirai achieves the best Quality, Semantic, and Total Scores while preserving the throughput and latency of the underlying AR baseline. Bold marks the best throughput and latency, and Video-Mirai scores above their baseline.

Model	#Params	Resolution	Throughput (FPS) ↑	Latency (s) ↓	Quality Score ↑	Semantic Score ↑	Total Score ↑
Bidirectional Video Diffusion Models							
LTX-Video [14]	1.9B	768×512	8.98	13.5	81.88	71.62	79.83
Wan2.1 [37]	1.3B	832×480	0.78	103	84.30	79.65	83.37
Autoregressive Video Diffusion Models							
NOVA [11]	0.6B	768×480	0.88	4.1	80.66	78.92	80.31
Pyramid Flow [21]	2B	640×384	6.70	2.5	83.41	70.11	80.75
SkyReels-V2 [9]	1.3B	960×540	0.49	112	83.96	74.01	81.97
MAGI-1 [35]	4.5B	832×480	0.19	282	81.67	67.72	78.88
Distilled Autoregressive Video Models							
CausVid [42]	1.3B	832×480	**17.0**	0.69	83.98	70.72	81.33
Self-Forcing [19] (chunk-wise)	1.3B	832×480	**17.0**	0.69	84.37	80.87	83.67
+ Video-Mirai (Ours)	1.3B	832×480	**17.0**	0.69	**84.82**	**81.45**	**84.15**
Causal-Forcing [45] (frame-wise)	1.3B	832×480	8.9	**0.45**	83.16	78.73	82.27
+ Video-Mirai (Ours)	1.3B	832×480	8.9	**0.45**	**84.59**	**79.66**	**83.60**
Causal-Forcing [45] (chunk-wise)	1.3B	832×480	**17.0**	0.69	84.54	80.93	83.82
+ Video-Mirai (Ours)	1.3B	832×480	**17.0**	0.69	**85.38**	**81.59**	**84.62**

Table 6: **Transfer to another AR model.** SkyReels-V2-DF-1.3B fine-tuned for 500 steps with and without the foresight loss.

Method	Quality ↑	Semantic ↑	Total ↑
SkyReels-V2-DF-1.3B (fine-tuned)	83.09	76.72	81.82
+ Video-Mirai (Ours)	**84.26**	**77.42**	**82.89**

Quality, Semantic, and Total Scores at every baseline, confirming that foresight internalization transfers across distillation frameworks and across frame-wise and chunk-wise generation. Figures 2 and 7 show representative qualitative comparisons. Additional examples are in Appendix G.

Beyond these distilled students, we also apply the foresight loss to SkyReels-V2-DF-1.3B [9], a released AR video model with different pretraining data and training objective. We fine-tune the official checkpoint for 500 teacher-forcing steps at batch size 32 with and without the foresight loss,



Figure 7: **Qualitative comparison.** Representative frames at t=0s, 2.5s, 5s from the rollouts of the same prompt under the baselines and their Video-Mirai counterparts. Video-Mirai rollouts preserve subject identity and scene composition more reliably across time.

Table 7: **System-level comparison on VBench.** Left: 5-second generation. Right: 30-second long-horizon generation. Video-Mirai is applied as a drop-in addition on top of chunk-wise Self-Forcing and both frame-wise and chunk-wise Causal-Forcing.

Method	5-second			30-second		
	Subject Consistency ↑	Background Consistency ↑	Overall Consistency ↑	Subject Consistency ↑	Background Consistency ↑	Overall Consistency ↑
Self-Forcing (chunk-wise)	96.18	96.43	26.70	89.83	92.72	25.02
+ Video-Mirai (Ours)	**96.77**	**96.85**	**26.84**	**91.62**	**93.77**	**25.33**
Causal-Forcing (frame-wise)	90.67	92.97	26.42	75.60	84.41	23.25
+ Video-Mirai (Ours)	**93.13**	**94.12**	**26.57**	**76.90**	**85.07**	**23.66**
Causal-Forcing (chunk-wise)	96.05	95.92	26.83	84.93	90.22	24.93
+ Video-Mirai (Ours)	**96.41**	**96.54**	**26.85**	**88.47**	**91.94**	**25.03**

keeping data and seeds identical, and evaluate on all 946 VBench prompts with one video per prompt. As shown in Table 6, the foresight loss improves Quality, Semantic, and Total Scores, raising Total from 81.82 to 82.89.

The improvement extends to 30-second long-horizon generation, on precisely the dimensions tied to causal short-sightedness: subject and background consistency improve substantially over the no-foresight baseline, and overall consistency improves slightly, as shown in Table 7. In chunk-wise settings, Video-Mirai yields larger gains on all three consistency metrics in the 30-second evaluation than in the 5-second evaluation. These results support its effectiveness beyond the training horizon, although the two evaluations use different prompt sets. The gains grow further at 60 seconds (Appendix D.6). This performance is achieved at identical inference cost, since the foresight encoder and predictor are discarded after training. Training-time overhead is analyzed in Appendix B.

5 Conclusion

We introduced Video-Mirai, a training-only paradigm that closes the representation-level planning gap in causal AR video diffusion: a single cosine alignment loss pulls the causal generator's mid-depth features toward a frozen bidirectional encoder's view of its own rollout. At inference, the encoder and predictor are discarded, leaving the architecture, per-step FLOPs, and KV-cache behavior unchanged. As a drop-in addition to AR video models, Video-Mirai lifts Causal-Forcing's VBench scores at 5 seconds and widens the margin at 30-second generation. Representation probes confirm that future information is internalized into the causal weights, turning anticipation into a property of the causal forward pass. Limitations are discussed in Appendix C. Our results argue that causality is a constraint on inference, not on what training can teach a model to anticipate.

Acknowledgments

This research is partially supported by JSPS KAKENHI Grant Number JP26K21317 and the Initiative on Recommendation Program for Young Researchers and Woman Researchers, Information Technology Center, The University of Tokyo. The computations were partly performed on the Miyabi-G supercomputer at the Information Technology Center, The University of Tokyo.

References

[1] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *NeurIPS*, volume 37, pages 58757–58791, 2024.

[2] Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[3] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, pages 15619–15629, 2023.

[4] Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

[5] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *TMLR*, 2024.

[6] Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *ICML*, volume 235 of *PMLR*, pages 4603–4623, 2024.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021.

[8] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token prediction meets full-sequence diffusion. In *NeurIPS*, volume 37, pages 24081–24125, 2024.

[9] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, et al. SkyReels-V2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

[10] Antoine Dedieu, Wolfgang Lehrach, Guangyao Zhou, Dileep George, and Miguel Lázaro-Gredilla. Learning cognitive maps from transformer representations for efficient planning in partially observed environments. In *ICML*, volume 235 of *PMLR*, pages 10287–10316, 2024.

[11] Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. In *ICLR*, volume 2025, pages 44730–44745, 2025.

[12] Dileep George, Rajeev V. Rikhye, Nishad Gothoskar, J. Swaroop Guntupalli, Antoine Dedieu, and Miguel Lázaro-Gredilla. Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nature Communications*, 12(1):2392, 2021.

[13] Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *ICML*, volume 235 of *PMLR*, pages 15706–15734, 2024.

[14] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. LTX-Video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 15979–15988, 2022.

[16] Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. StreamingT2V: Consistent, dynamic, and extendable long video generation from text. In *CVPR*, pages 2568–2577, 2025.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, pages 6840–6851, 2020.

[19] 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, pages 167283–167308, 2025.

[20] 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*, pages 21807–21818, 2024.

[21] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *ICLR*, volume 2025, pages 23378–23402, 2025.

[22] Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. FIFO-Diffusion: Generating infinite videos from text without training. In *NeurIPS*, volume 37, pages 89834–89868, 2024.

[23] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[24] Yann LeCun. A path towards autonomous machine intelligence. *OpenReview preprint*, 2022. Version 0.9.2.

[25] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning with latent diffusion transformers. In *ICCV*, pages 18262–18272, 2025.

[26] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

[27] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling Forcing: Autoregressive long video diffusion in real time. In *ICLR*, volume 2026, pages 91177–91196, 2026.

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024.

[29] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023.

[30] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie Gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[32] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *ICLR*, 2015.

[33] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[34] Sivaramakrishnan Swaminathan, Antoine Dedieu, Rajkumar Vasudeva Raju, Murray Shanahan, Miguel Lázaro-Gredilla, and Dileep George. Schema-learning and rebinding as mechanisms of in-context learning and emergence. In *NeurIPS*, volume 36, pages 28785–28804, 2023.

[35] Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W.Q. Zhang, Weifeng Luo, et al. MAGI-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

[36] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *ICLR*, volume 2025, pages 73754–73776, 2025.

[37] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[38] Wenhao Wang and Yi Yang. VidProM: A million-scale real prompt-gallery dataset for text-to-video diffusion models. In *NeurIPS*, volume 37, pages 65618–65642, 2024.

[39] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *ICLR*, volume 2025, pages 83048–83077, 2025.

[40] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, volume 37, pages 47455–47487, 2024.

[41] 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*, pages 6613–6623, 2024.

[42] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *CVPR*, pages 22963–22974, 2025.

[43] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, volume 2025, pages 87400–87442, 2025.

[44] Yonghao Yu, Lang Huang, Zerun Wang, Runyi Li, and Toshihiko Yamasaki. Mirai: Autoregressive visual generation needs foresight. In *CVPR*, pages 30511–30520, 2026.

[45] 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. In *ICML*, volume 306 of *PMLR*, pages 168030–168049, 2026.

Appendix

A Implementation Details

Model architecture. The causal generator backbone is Wan2.1-T2V-1.3B [37]: 30 transformer blocks, hidden dimension 1536, FFN dim 8960, 12 heads, with QK-normalization and causal attention along the temporal axis. The foresight encoder is the frozen Wan2.1-T2V-14B: 40 blocks, hidden dimension 5120, FFN dim 13,824, 40 heads, with bidirectional attention. We extract the causal generator's hidden state at layer 15 ($\alpha = 0.5$) and the encoder's hidden state at layer 20 ($\alpha = 0.5$, matched relative depth); the encoder is early-exited at layer 20, halving its forward cost. The predictor ϕ_ω is a stack of 3 DiT blocks [29], hidden dimension 1536, FFN ratio 4, 12 heads with QK-norm followed by a linear projection to dimension 5120. The predictor contains $\sim$138M parameters, accounting for $\sim 9.7\%$ of the 1.42B parameters of Wan-1.3B, and is discarded during inference.

Training pipeline. We follow the three-stage Causal-Forcing pipeline [45]: Stage 1, teacher AR fine-tuning for 2K steps; Stage 2, ODE distillation for 2K steps; Stage 3, asymmetric DMD, which is run for 100, 500, and 600 steps for chunk-wise Causal-Forcing, frame-wise Causal-Forcing, and Self-Forcing, respectively, following each baseline's recipe. Foresight is attached only during the final 100 steps of Stage 3 in our default configuration. We also discuss applying foresight in alternative stages in Appendix D.4. At each training step, the causal generator rolls out a video, with mid-depth hidden states retained, and the foresight encoder processes the same full rollout. The foresight loss is computed on the clean rollout, while the asymmetric DMD loss is computed on a re-noised version, scored against a frozen Wan-14B real-score teacher and a Wan-1.3B fake-score critic. The fake-score critic is updated 5 times per generator update using the standard flow-matching loss, while the real-score model remains frozen throughout training.

Optimization. We use AdamW for the causal generator, predictor, and fake-score critic, with linear warmup. The causal generator and critic learning rates follow the Causal-Forcing recipe; the predictor learning rate is 2×10^{-6}. All distillation runs use $8\times$ H100 GPUs with FSDP sharding and gradient accumulation of 8, with an effective batch size of 64. Mixed-precision training uses bf16. Training prompts come from the filtered VidProM [38] extension released by Self-Forcing.

Evaluation. We evaluate on VBench [20]. For the 5-second generation, each metric is the mean over 5 generated videos per prompt across the full VBench prompt set, following the benchmark's standard protocol. The Total Score follows the standard VBench $0.8\cdot$ Quality + $0.2\cdot$ Semantic weighting. For the 30-second long-horizon evaluation, we roll out the same trained checkpoints on 200 randomly selected MovieGen prompts, following the Rolling-Forcing [27] protocol, and report the subset of quality and semantic dimensions that are computable under its prompt-agnostic setting, notably subject consistency, background consistency, and overall consistency, the dimensions most directly tied to causal short-sightedness.

Prompt overlap. Training and evaluation prompts come from disjoint sources. We audited exact duplicates after Unicode, case, punctuation, and whitespace normalization between the 248,221 training prompts and both evaluation sets (946 VBench prompts, 1,003 MovieGen prompts) and found zero matches.

The pseudocode for one training step is given in Algorithm 1.

B Additional Computational Cost

Video-Mirai introduces no inference-time overhead: the foresight encoder E_ξ and predictor ϕ_ω are discarded after training, leaving the trained model as a strictly causal AR decoder with unchanged FLOPs and KV-cache behavior. Training-time overhead has two components, which we account for below by FLOP estimation on the chunk-wise Causal-Forcing pipeline at our training resolution: latent shape $1\times21\times16\times60\times104$, patch size $1\times2\times2$, giving $M_f = (60/2)\times(104/2) = 1560$ tokens per frame and $M = 21 \times M_f = 32{,}760$ tokens per training sample, and report the measured wall-clock overhead.

Algorithm 1 Video-Mirai training step

Require: causal generator G_θ, predictor ϕ_ω, frozen foresight encoder E_ξ, fake-score critic D_ψ, frozen real-score model D_{real}, text prompt c, offset set $\Delta = \{0, \dots, K\}$, causal generator/encoder layers L, L', weight λ

▷ Causal generator rollout with retained mid-depth hidden states

Initialize $\mathbf{x} \leftarrow \varnothing$, $\{\mathbf{h}_i^L\} \leftarrow \varnothing$

for segment index $i = 1, \dots, N$ **do**

  Denoise $\mathbf{X}_i$ conditioned on c using G_θ with KV-cache of $\mathbf{X}_{<i}$

  Cache $\mathbf{h}_i^L$ at layer L; append $\mathbf{X}_i$ to $\mathbf{x}$

end for

▷ Foresight encoder: one forward over the full rollout

$\{\mathbf{H}_j^{L'}\}_{j=1}^N \leftarrow E_\xi(\mathbf{x}, c)$ at layer L', early-exit.

▷ Foresight loss on valid segments

$\bar{\mathbf{H}}_i \leftarrow \frac{1}{|\Delta|} \sum_{\delta \in \Delta} \mathbf{H}_{i+\delta}^{L'}$ for $i = 1, \dots, N-K$

$\mathcal{L}_{\text{Foresight}} \leftarrow \frac{1}{N-K} \sum_{i=1}^{N-K} \left[1 - \cos(\phi_\omega(\mathbf{h}_i^L),\ \text{sg}[\bar{\mathbf{H}}_i])\right]$

▷ Combined objective and update

$\mathcal{L}_{\text{Generation}} \leftarrow$ asymmetric DMD loss on re-noised rollout $\mathbf{x_{noise}}$, using frozen D_{real} and D_ψ

$\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{Generation}} + \lambda \cdot \mathcal{L}_{\text{Foresight}}$

Update θ and ω via $\nabla \mathcal{L}_{\text{total}}$

▷ Fake-score critic update (5 steps per generator step) and predictor replay

Update D_ψ for 5 steps on re-noised rollouts via flow-matching loss

Update ω on 4 of these 5 steps by replaying cached $(\mathbf{h}_i^L, \bar{\mathbf{H}}_i)$ pairs via $\mathcal{L}_{\text{Foresight}}$

Foresight encoder forward. Adding foresight to Stage 3, which is the DMD stage, requires one forward pass of encoder E_ξ per generator update. The encoder we analyze here is the Wan-14B: $d = 5120$, FFN dim 13,824, 40 heads, 40 layers. We early-exit at layer 20 of 40, matching the student's mid-depth layer 15 of 30, which halves the encoder's forward cost. Each Wan-14B layer at $M = 32{,}760$ tokens contributes approximately $8Md^2 + 4M^2d + 4Md \cdot d_{\text{ffn}} \approx 38.1$ TFLOPs forward, so 20 layers add $\sim$762 TFLOPs per generator-update step. The encoder is frozen and stores no gradients, so its backward cost is zero.

Predictor forward and backward. The 3-block DiT predictor ϕ_ω runs at the causal generator's hidden dimension $d_c = 1536$ with FFN ratio 4, FFN dim 6144 and 12 attention heads. The predictor processes one segment of 4,680 tokens at a time. Each block contributes $\sim$0.4 TFLOPs per segment, so forward across 3 blocks and all segments is $\sim$8.4 TFLOPs and backward is $\sim$17 TFLOPs ($\sim 2\times$ forward). A final linear projection $\mathbb{R}^{1536} \to \mathbb{R}^{5120}$ maps ϕ_ω's output into the encoder's hidden width for cosine alignment; its cost is $\sim$1.5 TFLOPs forward+backward at $M = 32{,}760$, bringing the predictor pass to $\sim$27 TFLOPs.

Parameter and optimizer-state overhead. The predictor ϕ_ω contains $\sim$138M trainable parameters in total: $\sim$130M from the 3 DiT blocks plus $\sim$7.9M from the final 1536$\to$5120 linear projection, which is $\sim 9.7\%$ of the 1.42B-parameter Wan-1.3B causal generator. Under AdamW with FSDP sharding across our 8 training GPUs, the per-GPU optimizer-state increase is approximately 207MB $\approx$ 138M $\times$ 12 bytes/param / 8.

FLOP overhead. With foresight enabled, each generator update adds $\sim$789 TFLOPs (762 from E_ξ and $\sim$27 from ϕ_ω). We additionally update ϕ_ω on four of the five critic steps using cached hidden states, adding $\sim$27 TFLOPs per replay. This replay amortizes the encoder cost across the DMD cycle: E_ξ runs only on the generator step, while the predictor benefits from more frequent updates.

Empirical wall-clock measurement. We measure per-iteration training time on $8\times$ H100 GPUs over 100 training iterations with foresight enabled and disabled, under otherwise identical Stage 3 configuration. Each asymmetric DMD cycle spans five iterations: one runs a generator update followed by a critic update, and the other four run only a critic update, so we report the two iteration types separately in Table 8. The measured per-cycle overhead is **14.3%**.

Table 8: **Empirical wall-clock training time per iteration.** Measured on 8× H100 over 100 training iterations each. One cycle is one generator iteration plus four critic-only iterations.

Iteration type	Baseline (s)	+ Foresight (s)	Overhead
Generator + critic	244	292	+20.0%
Critic only	86	95	+10.6%
Per-cycle (1 gen + 5 critic)	588	672	**+14.3%**

Inference cost. At deployment, the student runs without E_ξ, ϕ_ω, or the foresight loss. The inference cost is identical to a standard causal Wan-1.3B forward pass, and the KV cache is maintained as in any causal AR generator. Real-time streaming properties like the latency per frame or throughput are also unchanged from the Causal-Forcing baseline.

C Limitations

Our foresight window is deliberately small: the default $\Delta = \{0, 1\}$ looks at most one chunk ahead, roughly 3 latent frames (∼0.75 seconds at 16 fps). The target itself is computed by a bidirectional encoder that attends over the complete rollout. Our ablations show that naively extending the window further within the current recipe yields diminishing returns (Table 1): adding a two-step-ahead target ($\Delta = \{0, 1, 2\}$) improves Semantic but degrades Quality, and the number of valid (source, target) pairs shrinks as K grows since supervising chunk i requires access to chunk $i + K$. We hypothesize that the Quality drop at $\Delta = \{0, 1, 2\}$ arises because far-offset targets from a single stochastic rollout are noisy representatives of a multimodal future, and uniform averaging injects that variance into the target. This is consistent with Figure 6, where the readout resembles the average of multiple future rollouts. Distance-weighted fusion, multi-sample far targets, and learned fusion are natural next steps for building targets for multimodal futures. Given how clearly anticipation is internalized at short horizons (Figure 5), an open question is whether a longer training-time foresight window could deliver proportional gains at longer generation horizons. Promising directions include longer rollouts with position-reweighted foresight loss, hierarchical alignment at multiple timescales, and virtual chunks that extrapolate beyond the rollout horizon. We leave these to future work.

D Additional Results

D.1 Full VBench

Tables 9 and 10 report the detailed per-dimension breakdown of VBench Quality and Semantic Scores, covering 7 and 9 dimensions respectively, for the Causal-Forcing baseline and after applying Video-Mirai. We evaluate both 5-second generation under the VBench standard protocol and 30-second long-horizon generation using MovieGen prompts under the VBench prompt-agnostic protocol.

Table 9: **Per-dimension Quality Score on VBench.**

Method	Subject Consistency	Background Consistency	Temporal Flickering	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Quality Score
5-second								
Causal-Forcing	96.05	95.92	98.64	98.26	61.11	67.01	70.92	84.54
+ Video-Mirai	**96.41**	**96.54**	**99.42**	**98.33**	**65.00**	**67.88**	69.87	**85.38**
30-second								
Causal-Forcing	84.93	90.22	96.04	97.21	49.00	56.90	64.54	76.26
+ Video-Mirai	**88.47**	**91.94**	95.93	97.16	**49.50**	**61.52**	64.25	**77.88**

Table 10: **Per-dimension Semantic Score on VBench.** The 30-second evaluation provides only Temporal Style and Overall Consistency, which use the same ViCLIP video-text similarity on the same prompts and therefore coincide.

Method	Object Class	Multiple Objects	Human Action	Color	Spatial Relation	Scene	Temporal Style	Appearance Style	Overall Consistency	Semantic Score
5-second										
Causal-Forcing	95.98	88.66	95.60	86.89	78.40	57.05	24.62	20.63	26.83	80.93
+ Video-Mirai	**96.79**	87.96	95.40	**87.26**	**82.29**	**57.57**	**24.83**	**20.76**	**26.85**	**81.59**
30-second										
Causal-Forcing	–	–	–	–	–	–	24.93	–	24.93	–
+ Video-Mirai	–	–	–	–	–	–	**25.03**	–	**25.03**	–

D.2 Frame-wise Foresight Window

Table 11: **Frame-wise foresight window comparison.** Effect of the offset set Δ when applied to the frame-wise Causal-Forcing model.

Setting	Δ	Quality ↑	Semantic ↑	Total ↑
Causal-Forcing (frame-wise)	–	83.16	78.73	82.27
Current only	$\{0\}$	83.81	79.50	82.94
1-frame ahead only	$\{1\}$	83.79	**79.66**	82.96
Current + 1-frame ahead	$\{0, 1\}$	**84.59**	**79.66**	**83.60**
Current + 2-frame ahead	$\{0, 1, 2\}$	83.25	77.80	82.16

Table 11 extends our foresight window study (Table 1) to the frame-wise setting, where each segment is a single latent frame. The chunk-wise conclusion holds: Δ=$\{0, 1\}$ achieves the best Total Score. Unlike the chunk-wise setting, extending to $\{0, 1, 2\}$ drops the Total Score below the baseline.

D.3 Loss Weight

Table 12 sweeps the foresight weight $\lambda \in \{0.1, 0.2, 0.3\}$ on Causal-Forcing. λ=0.2 achieves the best Total Score: lower weights under-supervise the predictor and yield smaller gains, while higher weights over-pull the causal generator's mid-depth features toward the encoder and slightly degrade the Quality Score. We adopt λ=0.2 throughout.

Table 12: **Foresight weight λ ablation** on Causal-Forcing.

Weight	Quality ↑	Semantic ↑	Total ↑
Causal-Forcing	84.54	80.93	83.82
0.1	84.83	81.42	84.15
0.2	**85.38**	81.59	**84.62**
0.3	85.14	**81.71**	84.45

D.4 Foresight at Different Distillation Stages

Causal-Forcing exposes two training stages at which Video-Mirai can be attached: Stage 2, causal ODE distillation, and Stage 3, asymmetric DMD. These two stages are not interchangeable injection points because they differ in what data the causal generator consumes. In this section, we analyze how to inject foresight into Stage 2.

Stage 2 ODE: target from the ground-truth future. Stage 2 trains the causal generator's flow map on paired (noise → clean) trajectories whose clean side is a ground-truth video $\{\mathbf{X}_1^{\mathrm{GT}}, \ldots, \mathbf{X}_N^{\mathrm{GT}}\}$. The causal generator is teacher-forced on clean GT prefixes, and the ODE distillation loss is defined

on GT trajectories. The consistent foresight target is therefore the GT future segment: at each step, we feed the clean latent of $\mathbf{X}^{\text{GT}}_{i+\delta}$ to the foresight encoder. Both the ODE loss and the foresight loss live on the same clean, teacher-forced manifold.

Stage 3 DMD: target from the causal generator's own rollout. Stage 3 removes the paired data: the causal generator self-rolls a video $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ from noise, and asymmetric DMD matches the causal generator's output distribution to a bidirectional teacher's score. The consistent foresight target is the foresight encoder's representation of the causal generator's own future rollout. Foresight then regularizes the same trajectory and distribution that DMD already matches, ensuring the two objectives operate on a shared signal.

Ablation study of foresight in Stage 2. The main paper studies foresight at Stage 3 (DMD). For completeness, we conduct two parallel ablations within Stage 2 (ODE) along the same axes as the main paper's Stage 3 counterparts: the foresight encoder and the predictor's fusion strategy. All Stage 2 ablation runs use a shortened budget of 1K distillation steps and no Stage 3 refinement.

Table 13 fixes the encoder to Wan-14B and sweeps three fusion strategies for the offset set $\Delta{=}\{0, 1\}$: no fusion (one loss per offset), per-frame averaging (encoder features averaged across frames within each segment), and per-δ averaging (encoder features averaged across offsets, i.e., the default in the main paper, Eq. 4). Table 14 fixes the fusion strategy to per-frame averaging and sweeps the encoder among EMA, Wan-1.3B, and Wan-14B. Two findings emerge. First, at Stage 2, per-δ averaging is actively harmful, while per-frame averaging is best. This contrasts with Stage 3, where per-δ target fusion is optimal. The two stages train on different data: ODE pairs of GT videos versus the causal generator's own rollouts. The fusion strategy that aligns with each stage's data is what wins. Second, Wan-14B remains the strongest encoder at Stage 2, as at Stage 3 (Table 4).

Table 13: **Stage 2 fusion-strategy ablation.** Encoder fixed to Wan-14B after 1K training steps.

Fusion strategy	Quality ↑	Semantic ↑	Total ↑
Causal-Forcing (stage 2 reproduction)	82.56	**78.30**	81.71
no fusion	82.65	77.51	81.62
per-frame averaging	**84.17**	**77.68**	**82.87**
per-δ averaging	81.45	74.32	80.02

Table 14: **Stage 2 encoder ablation.** Fusion strategy fixed to per-frame averaging after 1K training steps.

Encoder	Quality ↑	Semantic ↑	Total ↑
Causal-Forcing (stage 2 reproduction)	82.56	78.30	81.71
EMA	83.96	78.01	82.77
Wan-1.3B	83.40	**78.76**	82.47
Wan-14B	**84.17**	77.68	**82.87**

Foresight injection stage comparison. Causal-Forcing's three-stage pipeline exposes two natural injection points for foresight: ODE distillation (Stage 2) and asymmetric DMD (Stage 3). Whether to attach foresight at Stage 2, Stage 3, or both is not *a priori* obvious. Table 15 sweeps all three options. The results show that ODE-Foresight and DMD-Foresight yield identical total gains over Causal-Forcing, but split the credit very differently: ODE-Foresight drives the largest quality gain while slightly reducing the Semantic Score, consistent with pulling the flow map toward a clean, well-formed future, whereas DMD-Foresight drives the largest semantic gain while retaining most of the quality improvement, consistent with regularizing the very distribution DMD is already matching, on the same rollout trajectory.

Stacking the two does not combine these profiles, collapsing to a Total Score only marginally above the baseline. The reason is that the two instantiations do not disagree about *whether* the causal generator should anticipate the future, but about *which* future: ODE-Foresight trains ϕ_ω to predict a

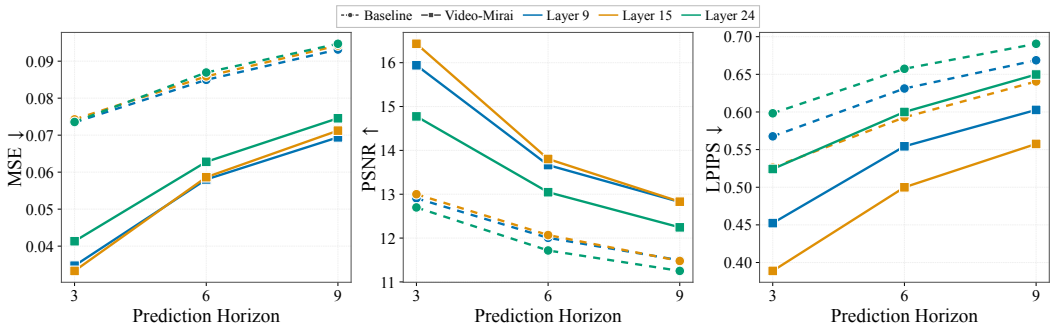


Figure 8: **Future-frame readout fidelity across layers and horizons.** Solid: Video-Mirai; dashed: Causal-Forcing baseline. Video-Mirai dominates at every layer and horizon.

clean GT future from a GT prefix, while DMD-Foresight trains it to predict the future from the causal generator's *own* rollout, and the two trajectories differ whenever the causal generator is imperfect. Stacked naively, Stage 3 must rewrite the foresight mapping Stage 2 just installed, swapping a clean target for a moving one while simultaneously running asymmetric DMD; these pressures pull mid-depth features in conflicting directions.

DMD-Foresight is the most practical choice since it is equally effective and cheaper: it drops into any pretrained Causal-Forcing Stage 2 checkpoint without modifying Stage 2 training, and requires only 100 DMD steps, 20× fewer than the 2K Stage 2 distillation steps. We therefore use it throughout the main paper.

Table 15: **Foresight injection stage comparison.** Foresight applied at Stage 2 (ODE-Foresight), Stage 3 (DMD-Foresight), or both.

Stage	Quality ↑	Semantic ↑	Total ↑
Causal-Forcing	84.54	80.93	83.82
Stage 2	**85.56**	80.84	**84.62**
Stage 3	85.38	**81.59**	**84.62**
Stage 2 + 3	84.78	81.10	84.05

D.5 Foresight Encoder from Outside the Model Family

To test whether the foresight encoder must come from the same model family as the generator, we replace Wan-14B with a frozen 21M-parameter DINO ViT-S/16 [7]. DINO is a self-supervised image encoder that is neither a diffusion model nor trained on video, and it receives no text conditioning. Its patch grid matches the causal generator's latent token grid one-to-one, so the rest of the recipe is unchanged: offsets $\{0, 1\}$ and the same 100 DMD steps. Table 16 shows that this much smaller encoder recovers about two thirds of the gain of Wan-14B, so the foresight encoder need not be a Wan model or a video diffusion model. We keep Wan-14B as the default because it is already resident in the pipeline as the DMD real-score teacher and shares the VAE latent space.

Table 16: **Foresight encoder from outside the model family.** Causal-Forcing chunk-wise on 5-second VBench; DINO ViT-S/16 replaces Wan-14B with the rest of the recipe unchanged.

Encoder	Quality ↑	Semantic ↑	Total ↑
Causal-Forcing	84.54	80.93	83.82
Wan-14B	**85.38**	**81.59**	**84.62**
DINO ViT-S/16	85.09	81.33	84.34

D.6 60-Second Generation

We extend the same checkpoints to 60-second rollouts on the same 200 MovieGen prompts and seeds as the 30-second evaluation, under the same prompt-agnostic protocol. Comparing Table 17 with the 5- and 30-second results in Tables 7 and 9 shows that the gains increase beyond the training horizon. From 5 to 30 to 60 seconds, the gain grows from $+0.36$ to $+3.54$ to $+10.35$ on Subject Consistency, from $+0.62$ to $+1.72$ to $+4.51$ on Background Consistency, from $+0.02$ to $+0.10$ to $+0.77$ on Overall Consistency, and from $+0.84$ to $+1.62$ to $+7.49$ on the seven-dimension Quality Score.

Table 17: **60-second generation.** 200 MovieGen prompts under the prompt-agnostic protocol, 969 decoded frames per video.

Method	Subject Consistency ↑	Background Consistency ↑	Overall Consistency ↑	Quality Score ↑
Causal-Forcing	73.74	85.03	23.32	65.81
+ Video-Mirai	**84.09**	**89.54**	**24.09**	**73.30**

E Statistical Significance

We test whether Video-Mirai's gains over Causal-Forcing chunk-wise are statistically significant via a paired prompt-level bootstrap. For each VBench dimension, we resample its N_d prompts with replacement, using the same indices for both methods and recompute the dimension's mean, repeating 10,000 times. Each replicate is then aggregated into Quality, Semantic, and Total Scores using VBench's standard $\frac{4}{5}$ quality $+$ $\frac{1}{5}$ semantic weighting [20], yielding bootstrap distributions of the three summary metrics. We report the mean paired difference with its standard error; a * marks a $[2.5\%, 97.5\%]$ percentile 95% CI that excludes zero (two-sided $p < 0.05$). Per-prompt scores are obtained by averaging the 5 inference seeds per prompt.

Table 18 shows that Video-Mirai's gains on all three aggregated metrics have 95% CIs strictly excluding zero, confirming the improvements are not driven by a small subset of favorable prompts.

Table 18: **Significance of Video-Mirai gains over Causal-Forcing on VBench.** $10{,}000$ paired bootstrap resamples by prompt; cells: $\Delta_{\text{mean}} \pm$ SE. * marks 95% confidence intervals excluding zero, equivalent to a two-sided $p < 0.05$ test.

Method	ΔQuality	ΔSemantic	ΔTotal
Video-Mirai (vs Causal-Forcing)	$+0.84 \pm 0.19^*$	$+0.66 \pm 0.24^*$	$+0.80 \pm 0.15^*$

F More Probing Results

We provide additional visualization results of our probing analysis in Figures 8 and 9.

G More Comparison Results

We provide additional visual comparisons in Figure 10.

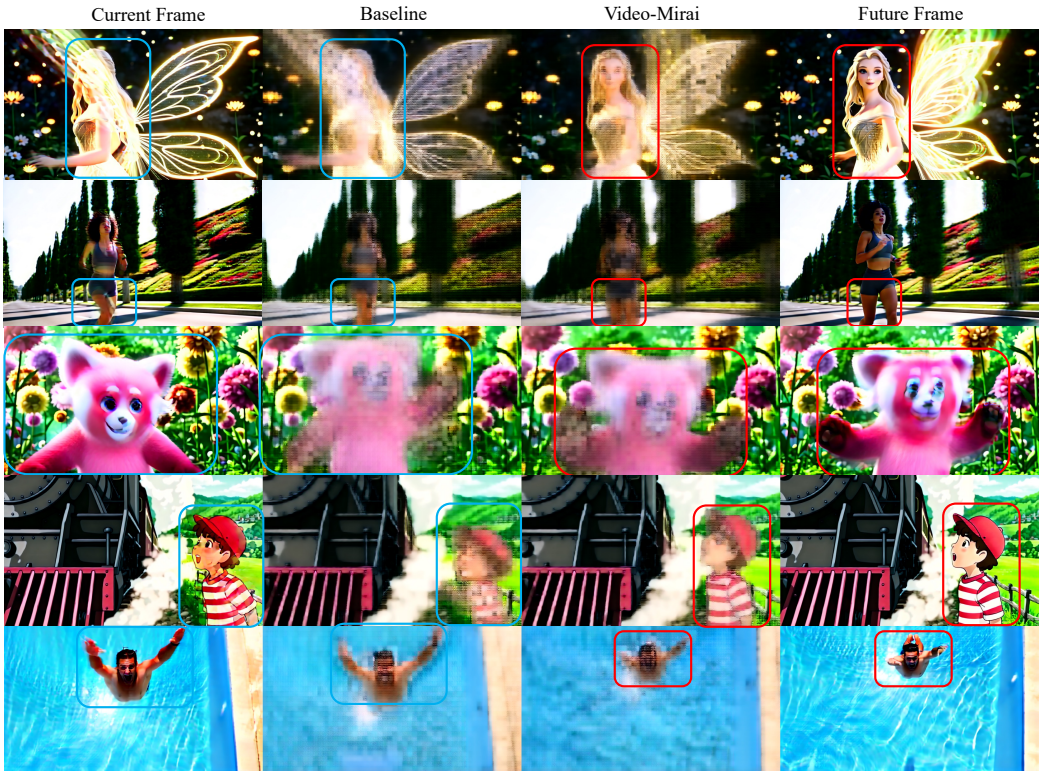


Figure 9: **Video-Mirai's representations encode the future.** An MLP readout reconstructs future RGB from the frozen causal generator's current hidden state (layer 15). Left to right: current frame, baseline readout, Video-Mirai readout (ours), future frame. blue/red: regions matching the current/future frame.

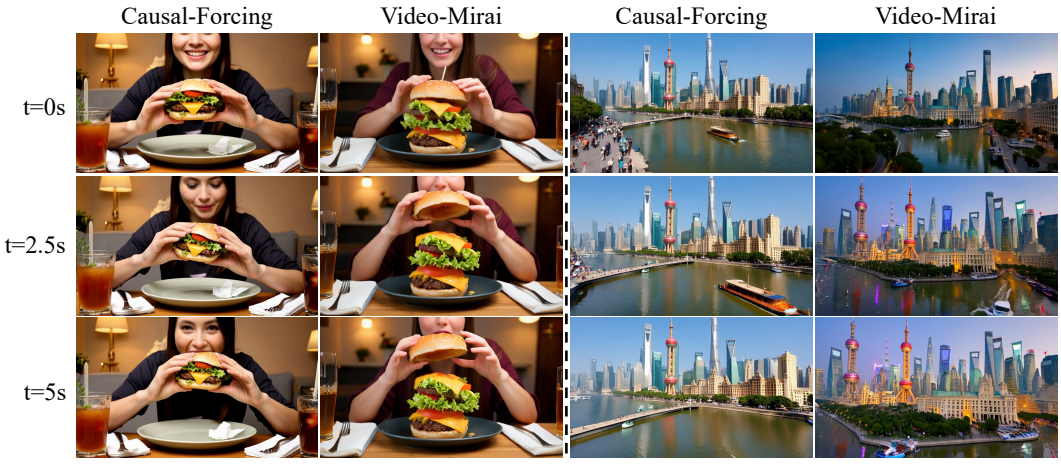


Figure 10: **More qualitative comparison.** Representative frames at t=0s, 2.5s, 5s from the rollouts of the same prompt under the baseline and its Video-Mirai counterpart. Video-Mirai rollouts preserve subject identity, scene composition, and motion coherence more reliably across time.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and §1 state three contributions: a systematic design study of foresight in AR video diffusion models, the Video-Mirai training paradigm, and consistent VBench gains on top of Causal-Forcing, each of which is directly supported by §3 and 4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the main limitations in Appendix C: the default foresight window is deliberately small (Δ=$\{0, 1\}$, $\sim$0.75s at 16 fps), and naively extending it under the current recipe yields diminishing returns. We discuss promising directions like longer rollouts, hierarchical alignment, and virtual chunks, but leave them to future work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper is empirical. It introduces a training-time representation-alignment objective and validates it on VBench, and provides visualization results. No formal theorems or proofs are claimed.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: §3 and Algorithm 1 fully specify the Video-Mirai training step. §4 and Appendix A report the detailed training and evaluation settings.

Guidelines:

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

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The training code, inference scripts, configs, and model checkpoints are publicly released at https://github.com/y0ur0y/Video-Mirai and https://huggingface.co/y0ur0y/Video-Mirai (linked from the project page), with instructions for environment setup, training, and inference. All upstream assets needed for reproduction are publicly available: Wan-1.3B, Wan-14B, the Causal-Forcing training data, and the VBench prompt set. Our codebase is built directly on the open-source Causal-Forcing / Self-Forcing / Wan2.1 repositories, which we cite. Algorithm 1 together with §3 and Appendix A specify the training step and all hyperparameters.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: §4 and Appendix A report the training and test details, including the optimizer (AdamW), learning rates for the generator and predictor, gradient accumulation, training steps per stage, foresight weight, offset set, and the VBench evaluation dimensions and number of samples. Hyperparameters that are not specific to Video-Mirai follow the Causal-Forcing / Self-Forcing defaults, which we cite.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Table 5 reports the 5-second results on the full VBench prompt set, with five videos per prompt. Table 7 additionally reports the 30-second results on 200 MovieGen prompts. For the main 5-second configuration, Video-Mirai on Causal-Forcing chunk-wise, Appendix E gives a paired prompt-level bootstrap: B=10,000 resamples, percentile 95% CIs, no Normality assumption; $\pm$ values are 1-sigma bootstrap standard errors, and * in Table 18 marks CIs excluding zero (two-sided $p < 0.05$). All configurations share the same inference seeds.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: §4 and Appendix A specify that all distillation runs use 8$\times$ H100 GPUs (FSDP, bf16, gradient accumulation 8). The training step for each stage follows the Causal-Forcing recipe. As analyzed in Appendix B, foresight is attached only during the final 100 Stage 3 steps, adding $\sim$14.3% wall-clock overhead per training cycle (Table 8). The foresight encoder (Wan-14B) is frozen; the predictor adds $\sim$9.7% trainable parameters and is discarded at inference.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

Justification: The research follows the NeurIPS Code of Ethics. No human subjects are involved, no personally identifying data is used, and all training data and pretrained backbones are publicly released assets that we use under their stated licenses.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: On the positive side, causal AR video models with stronger long-horizon coherence are foundational for interactive applications. For example, they enable streaming generation, simulation for embodied AI, and real-time content creation, where latency matters. On the negative side, improvements in video generation quality can amplify misuse risks, such as non-consensual synthetic media and disinformation. We build on publicly released base models whose developers have discussed similar concerns, and Video-Mirai is a training-time modification that does not introduce new misuse vectors beyond those inherent to AR video diffusion.

Guidelines:

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

11. **Safeguards**

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

Answer: [Yes]

Justification: Our release consists of weights fine-tuned from publicly released Wan-1.3B checkpoints and a training-time auxiliary loss; we do not release new base generative models or scraped datasets. As safeguards, (i) the released weights inherit Wan-1.3B's license, intended-use guidelines, and usage restrictions, and (ii) Video-Mirai is a training-time modification that does not expand the upstream model's misuse surface.

Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All external assets are cited in the paper, including the Wan-1.3B / Wan-14B backbones [37], Causal-Forcing [45], Self-Forcing [19], and the VBench evaluation suite [20]. We use each asset in accordance with its stated license and its terms of use.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The released Video-Mirai training code, inference scripts, configs, and model checkpoints (`https://github.com/y0ur0y/Video-Mirai`, `https://huggingface.co/y0ur0y/Video-Mirai`) are accompanied by a README and a model card describing the environment setup, training and inference commands, training data, hyperparameters, and license, consistent with the documentation style of the upstream Causal-Forcing and Self-Forcing releases.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject research.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects or study participants are involved.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were used only to assist with polishing the writing. They are not a component of the method, training pipeline, or evaluation.

    Guidelines:

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