PROSPECTIVE FORCING: LEARNING TO LOOK AHEAD FOR AUTOREGRESSIVE LONG VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review



Figure 1: By learning to predict future, Prospective Forcing generates long videos that remain dynamic, temporally consistent, and visually stable from a single text prompt.

ABSTRACT

Autoregressive video generators synthesize long videos through chunk-wise rollouts, yet they are typically trained to predict only the immediate future. This limited prediction horizon makes it challenging to maintain video quality over extended rollouts. To address this problem, we introduce Prospective Forcing, a multi-head prediction framework that injects long-range predictive signals into autoregressive video representations. Built on a rolling-window generator, Prospective Forcing attaches lightweight, offset-specific heads that share the same conditioning information and independently predict chunks at different future temporal offsets. By extending supervision beyond the next chunk, these heads encourage the shared history representation to retain information useful for longer-horizon prediction, supporting temporally consistent generation. At inference time, we accelerate sampling by using accepted future-chunk predictions to skip full-window backbone executions. We equip the head with learnable embeddings to generate multiple candidates for the next chunk, and a learned verifier either selects a candidate or rejects all and falls back to standard autoregressive generation.

1 INTRODUCTION

> *"The future enters into us, in order to transform itself in us, long before it happens."*
> — Rainer Maria Rilke, *Letters to a Young Poet*

Autoregressive video generation provides a natural framework for modeling dynamic visual worlds by extending a video one chunk at a time, which supports generative interactive environments (Bruce et al., 2024), real-time game simulation (Valevski et al., 2025), robot control (Li et al., 2026), and other applications that require coherent long-term visual rollouts. However, maintaining temporal consistency over long videos remains an open challenge. A rollout may remain locally plausible while subject identity, background appearance, spatial layout, color, and object structure gradually drift. These failures become more visible as the generation proceeds. This drift stems from a lack of long-horizon predictive capability, as next-chunk supervision prioritizes immediate continuation without requiring representations to preserve the information needed to anticipate more distant futures. We refer to this myopic behavior as *temporal shortsightedness*.

Existing approaches improve the temporal consistency from different perspectives. Conditioning the model on self-generated histories during training makes later predictions more robust to an imper-

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

fectly generated context (Huang et al., 2025a). Rolling-window generation jointly denoises adjacent chunks with progressive noise levels and retains initial states as global anchors, reducing error accumulation while preserving a reference to the beginning of the video (Liu et al., 2026). Training-free methods extend pretrained video generators by rescheduling noise across overlapping temporal windows or maintaining a queue of partially denoised frames (Qiu et al., 2024; Kim et al., 2024). Other approaches combine short-term conditioning on features from the preceding chunk with long-term preservation of scene and object features from the initial chunk (Henschel et al., 2025).

These approaches primarily address exposure bias, inference-time context preservation, or temporal denoising schedules. We introduce Prospective Forcing to address temporal shortsightedness through a complementary training objective. Specifically, the backbone encodes preceding chunks into a shared history representation, and independent heads combine it with target-specific diffusion noise to predict chunks at distinct future offsets. We train these heads through cross-chunk self-distillation using the backbone's own subsequent outputs as the teacher. When a target chunk is reached later in the same causal rollout, the backbone's detached feature and velocity predictions supervise the corresponding longer-offset prediction from an earlier context. These losses propagate through both the heads and the backbone, encouraging the shared representation to retain information needed across multiple future chunks. The resulting representation captures rapid changes in local motion while preserving relatively stable properties such as object identity and scene structure, consistent with the fast-and-slow perspective of hierarchical causal representation learning (Li et al., 2025). We call it a *speculative representation* to reflect its long-term predictive capability. Section 5 further analyzes this representation in an idealized setting.

At inference, our default configuration uses the speculative representation to produce multiple candidate continuations. A lightweight learned verifier selects one candidate or rejects them all, in which case generation falls back to the standard autoregressive path. Each accepted candidate extends the video by one additional chunk without another sequential execution of the backbone, reducing the sampling cost. We evaluate on 30-second video generation with multiple metrics, where Prospective Forcing improves video quality, while accelerating sampling. Notably, the improvements persist even when the prediction heads are removed at inference, indicating that the speculative representation itself captures information that benefits long-horizon generation.

Our main contributions are as follows. First, we introduce Prospective Forcing to address temporal shortsightedness by learning a speculative representation that supports explicit prediction of multiple future chunks from the current state, thereby improving long-term video quality. Second, we develop a verifier-guided speculative inference procedure that reuses the prediction heads to extend rollouts without additional sequential backbone executions. Third, we demonstrate improved video quality across multiple video generation metrics alongside sampling speedups.

2 Related Work

Autoregressive video generation. Unlike bidirectional video diffusion models which jointly denoise a fixed temporal window (Blattmann et al., 2023; Yang et al., 2025; Kong et al., 2024; Wan et al., 2025), autoregressive models generate frames or chunks sequentially, enabling low-latency streaming with KV caching. Existing methods differ mainly in how the causal generator is trained: with per-token noise levels (Chen et al., 2024a), by distilling few-step causal students from bidirectional or causal teachers via distribution matching (Yin et al., 2024; 2025; Huang et al., 2025a; Zhu et al., 2026; Zhao et al., 2026), or end-to-end without a teacher (Guo et al., 2025).

Long video generation. To mitigate drift over long rollouts, prior methods either align training with long-horizon inference (Cui et al., 2026b; Yang et al., 2026; Chen et al., 2026; Lin et al., 2026), design denoising schedules and anchor frames that preserve early context (Liu et al., 2026; Henschel et al., 2025), or manage the KV cache and positional encoding at inference time without additional training (Yi et al., 2026; Cui et al., 2026a). Instead, our method improves long video generation by learning representations with long-horizon predictive capabilities.

Future-predictive representations. Predicting future observations has long served as an auxiliary objective for learning sequential representations in reinforcement learning (Guo et al., 2020; Schwarzer et al., 2021), image generation (Yu et al., 2026b), language modeling and speculative decoding (Gloeckle et al., 2024; Li et al., 2024; Cai et al., 2024). Concurrent video works such as

Video-Mirai (Yu et al., 2026a) and Next Forcing (Xu et al., 2026) incorporate future information via JEPA-style representation alignment (Bardes et al., 2024) and recurrent future-chunk prediction, respectively. In contrast, Prospective Forcing learns a shared representation for parallel multi-chunk prediction with independent heads, and reuses these heads for verifier-guided speculative inference. A more detailed discussion is provided in Appendix B.

3 PRELIMINARIES

Autoregressive video generation. We formulate long-video generation as autoregressive prediction over latent video chunks. Let $\mathbf{x}_{1:T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ denote a sequence of video chunks and $\mathbf{z}_{1:T} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ their corresponding latents encoded by a pretrained VAE. Given a text condition $\mathbf{c}$, a causal generator factorizes the conditional distribution as

$$p_\theta\left(\mathbf{z}_{1:T} \mid \mathbf{c}\right) = \prod_{t=1}^{T} p_\theta\left(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{c}\right), \tag{1}$$

where $\mathbf{z}_{<t} = \{\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}\}$ denotes the preceding latent chunks and t indexes chunks in the autoregressive rollout. This factorization specifies the causal dependence among committed video chunks, but does not require each chunk to be denoised in isolation. We use $\tau \in [0, 1]$ to denote diffusion time. Under the rectified-flow parameterization (Liu et al., 2023), the forward corruption process is

$$\mathbf{z}_t(\tau) = (1 - \tau)\mathbf{z}_t + \tau\boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2}$$

A velocity-prediction network transforms noisy latents toward clean samples while conditioning on the text prompt and the available causal context. Sampling evolves from Gaussian noise at $\tau = 1$ toward clean latents at $\tau = 0$, with the resulting video chunks progressively committed to the autoregressive context. The key–value states of previously committed chunks can be cached to avoid repeated computation.

Distribution matching distillation. CausVid (Yin et al., 2025) distills a pretrained bidirectional video diffusion model into a few-step causal generator using distribution matching distillation (DMD) (Yin et al., 2024). Let $\hat{\mathbf{z}} \sim p_\theta(\cdot \mid \mathbf{c})$ denote a latent video sampled from the causal generator, and let $\hat{\mathbf{z}}(\tau)$ denote its noisy version obtained using equation 2. Let $\theta^\star$ denote the frozen parameters of the teacher model. DMD trains the generator to match the teacher model by minimizing the reverse KL divergence between their noise-perturbed sample distributions:

$$\mathcal{L}_{\mathrm{DMD}} = \mathbb{E}_{\mathbf{c},\tau}\left[w(\tau)\, D_{\mathrm{KL}}\left(p_{\theta,\tau}(\cdot \mid \mathbf{c}) \,\|\, p_{\theta^\star,\tau}(\cdot \mid \mathbf{c})\right)\right], \tag{3}$$

where $p_{\theta,\tau}$ and $p_{\theta^\star,\tau}$ denote the distributions of noisy generator and teacher samples, respectively. Here, $w(\tau)$ is a timestep-dependent weighting function. The gradient of this objective is estimated using the difference between the generator and teacher scores, where a score function is the gradient of the log probability density with respect to the noisy latent.

To bridge the train–test distribution gap, Self Forcing (Huang et al., 2025a) extends this framework with autoregressive self-rollout training and a DMD loss computed on the entire generated sequence. Rolling Forcing (Liu et al., 2026) further jointly denoises multiple chunks at different noise levels (Chen et al., 2024a) within a rolling window to enhance temporal consistency. However, the resulting generator still lacks an explicit mechanism for anticipating future chunks, a limitation we term *temporal shortsightedness*, as shown in Figure 2.

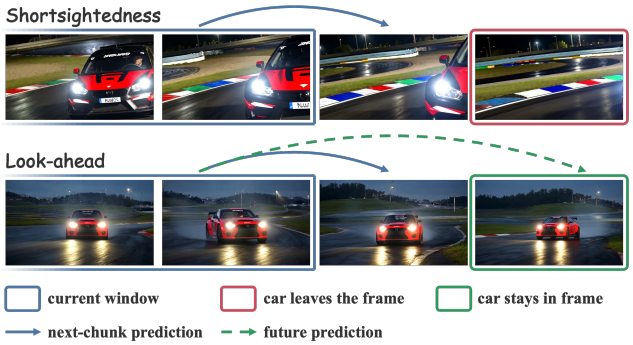

Figure 2: **Temporal shortsightedness.** The car gradually drives out of view with only next-chunk prediction, and foreseeing future chunks keeps it in frame.

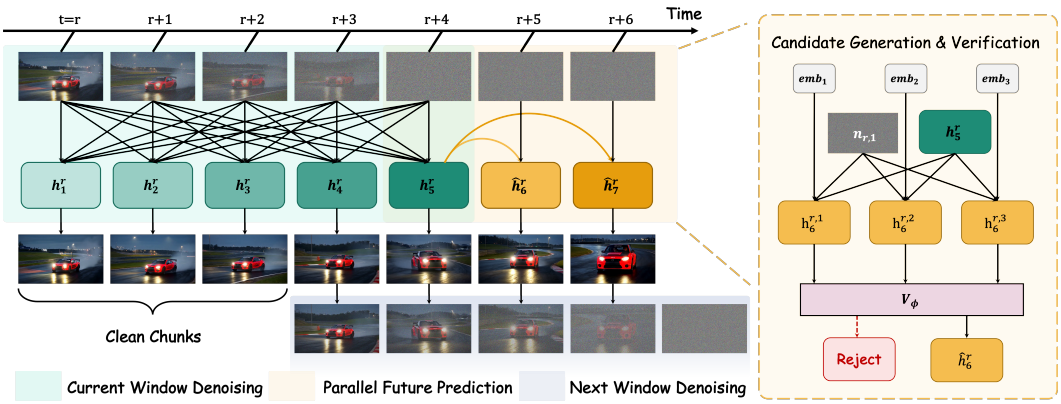

Figure 3: **Illustration of Prospective Forcing. Left**: We jointly denoise $W = 5$ chunks in window r, yielding features $\mathbf{h}_1^r$ to $\mathbf{h}_5^r$ (green). Then $K = 2$ heads predict $\hat{\mathbf{h}}_6^r$ and $\hat{\mathbf{h}}_7^r$ in parallel from $\mathbf{h}_5^r$ (orange). Leading clean chunks are committed and the window advances accordingly (blue). **Right**: $M = 3$ candidates are generated for the next chunk $\hat{h}_6^r$, and the verifier V_ϕ selects one or rejects all.

4 Prospective Forcing

We propose *Prospective Forcing*, which mitigates temporal shortsightedness by learning a speculative representation for future prediction (Figure 3). Within a rolling window, we jointly denoise W chunks at different noise levels, yielding features $\mathbf{h}_1^r, \ldots, \mathbf{h}_W^r$ from the backbone. K lightweight future prediction heads then predict the features of the next K chunks, $\hat{\mathbf{h}}_{W+1}^r, \ldots, \hat{\mathbf{h}}_{W+K}^r$, in parallel from $\mathbf{h}_W^r$, and a shared projection network maps them to velocities. For the next chunk, the first head further generates M candidates, each conditioned on a distinct learned candidate embedding, and a verifier V_ϕ is trained to score them. If the best candidate satisfies the acceptance criterion, it is accepted; otherwise, all candidates are rejected. The accepted prediction is used together with the outputs in the window to produce clean chunks. Finally, the preceding clean chunks are committed, and the last $W - 1$ chunks are renoised to form the window for the next iteration.

Speculative representation learning. Our generator processes consecutive latent chunks within a rolling window of W chunks. Let $\mathcal{W}_r$ denote the window whose first chunk is at time step r, so that it covers chunks $r, \ldots, r + W - 1$. Throughout this section, a superscript r indicates that a quantity is computed in window $\mathcal{W}_r$, and a subscript $i \in \{1, \ldots, W\}$ indicates a position within the window, corresponding to chunk $r + i - 1$. Each position is assigned a fixed diffusion time from $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_W)$, with earlier positions at lower noise levels. Let $\mathbf{z}_i^r$ denote the noisy latent obtained by applying equation 2 to chunk $r + i - 1$ at diffusion time τ_i. The window is then $\mathcal{W}_r = (\mathbf{z}_1^r, \ldots, \mathbf{z}_W^r)$. We denote the generator backbone by F_θ, and represent the cached context from previously committed chunks by $\mathcal{C}_r$. Conditioned on the text prompt $\mathbf{c}$, the backbone produces features for all positions in the window:

$$(\mathbf{h}_1^r, \ldots, \mathbf{h}_W^r) = F_\theta\left(\mathcal{W}_r, \boldsymbol{\tau}, \mathbf{c}, \mathcal{C}_r\right), \tag{4}$$

where $\mathbf{h}_i^r$ denotes the final layer features at position i of $\mathcal{W}_r$. The shared velocity projection P_θ maps the features at each position to the corresponding velocity output $\mathbf{v}_i^r$:

$$\mathbf{v}_i^r = P_\theta\left(\mathbf{h}_i^r, \tau_i\right), \qquad i = 1, \ldots, W. \tag{5}$$

We use the backbone features of the newest chunk, $\mathbf{h}_W^r$, as the shared input to K lightweight future prediction heads G_k, indexed by $k = 1, \ldots, K$. Each head estimates features for chunk $r+W+k-1$, the k-th chunk after the window, which we index as position $W + k$. We sample the Gaussian noise $\boldsymbol{\epsilon}_{r+W+k-1}$ of these future chunks in advance and embed it into tokens $\mathbf{n}_{r,k}$ with the patch embedding of the backbone. The predicted features $\hat{\mathbf{h}}_{W+k}^r$ and corresponding velocities $\hat{\mathbf{v}}_{W+k}^r$ are obtained as

$$\hat{\mathbf{h}}_{W+k}^r = G_k\left(\mathbf{h}_W^r, \mathbf{n}_{r,k}\right), \quad \hat{\mathbf{v}}_{W+k}^r = P_\theta\left(\hat{\mathbf{h}}_{W+k}^r, \tau_W\right). \tag{6}$$

The shared projection P_θ decodes each prediction at τ_W, the diffusion time assigned to an incoming chunk. The pre-sampled noise is reused when the corresponding chunk enters the window, ensuring that the prediction and its later target refer to the same noise realization.

We align these predictions with the corresponding backbone outputs in subsequent windows using cross-chunk self-distillation. After k advances, chunk $r + W + k - 1$ enters $\mathcal{W}_{r+k}$ at its last position W, where the backbone produces its features $\mathbf{h}_W^{r+k}$ and velocity $\mathbf{v}_W^{r+k}$. We use both outputs as supervision targets and stop gradients through them. Note that we do not skip any window at training time, so the supervision can always be provided. We normalize features before alignment to reduce sensitivity to their scale. Velocities are matched in their original space because both their direction and magnitude determine the denoising updates. For two feature tensors $\mathbf{a}$ and $\mathbf{b}$, we define a distance d_{feat} that combines mean squared error and cosine distance after layer normalization:

$$d_{\text{feat}}(\mathbf{a}, \mathbf{b}) = \frac{1}{2}\,\text{MSE}\left(\text{LN}(\mathbf{a}), \text{LN}(\mathbf{b})\right) + 1 - \text{Cos}\left(\text{LN}(\mathbf{a}), \text{LN}(\mathbf{b})\right), \tag{7}$$

where LN denotes layer normalization (Ba et al., 2016) applied along the feature dimension of each token. MSE averages squared errors over all elements, and Cos averages cosine similarity over corresponding tokens. Let sg denote the stop gradient operator. The feature and velocity alignment losses are

$$\mathcal{L}_{\text{feat}} = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} d_{\text{feat}}\left(\hat{\mathbf{h}}_{W+k}^{r}, \text{sg}\left[\mathbf{h}_W^{r+k}\right]\right)\right], \quad \mathcal{L}_{\text{vel}} = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} \text{MSE}\left(\hat{\mathbf{v}}_{W+k}^{r}, \text{sg}\left[\mathbf{v}_W^{r+k}\right]\right)\right], \tag{8}$$

where the expectation is taken over training rollouts. These alignment losses encourage $\mathbf{h}_W^r$ to encode information useful for predicting future chunks. We refer to these learned features as the *speculative representation*.

Candidate generation and verification. We further equip the first future prediction head G_1 with candidate generation. With learned embeddings $\mathbf{emb}_m$ that are broadcast on tokens and added to noise $\mathbf{n}_{r,1}$, G_1 produces M candidate features $\hat{\mathbf{h}}_{W+1}^{r,m}$ and velocities $\hat{\mathbf{v}}_{W+1}^{r,m}$:

$$\hat{\mathbf{h}}_{W+1}^{r,m} = G_1\left(\mathbf{h}_W^r, \mathbf{n}_{r,1} + \mathbf{emb}_m\right), \quad \hat{\mathbf{v}}_{W+1}^{r,m} = P_\theta\left(\hat{\mathbf{h}}_{W+1}^{r,m}, \tau_W\right). \tag{9}$$

We supervise the candidate $m_r^\star$ closest to the backbone features $\mathbf{h}_W^{r+1}$ computed in the next window $\mathcal{W}_{r+1}$. Using the feature distance defined in equation 7, the candidate alignment loss is

$$\mathcal{L}_{\text{cand}} = (1-\beta)\, d_{\text{feat}}\left(\hat{\mathbf{h}}_{W+1}^{r,m_r^\star}, \text{sg}\left[\mathbf{h}_W^{r+1}\right]\right) + \frac{\beta}{M-1}\sum_{m \neq m_r^\star} d_{\text{feat}}\left(\hat{\mathbf{h}}_{W+1}^{r,m}, \text{sg}\left[\mathbf{h}_W^{r+1}\right]\right), \tag{10}$$

where $m_r^\star = \arg\min_m E_m^r$ is the candidate closest to the backbone target under the normalized feature error E_m^r in equation 29. Since the closest candidate receives most of the gradient, the candidate embeddings are not pulled toward a common point and can specialize to different regions of the prediction space. The small weight β keeps every candidate activated.

We further train a lightweight multilayer perceptron (MLP) verifier V_ϕ to select one candidate or reject them all. Given the current features $\mathbf{h}_W^r$ and all candidate features $\hat{\mathbf{h}}_{W+1}^{r,m}$, V_ϕ predicts selection probabilities $\pi_1^r, \ldots, \pi_M^r$ and a rejection probability π_{rej}^r, with gradients stopped through its inputs so that the verifier loss does not update the generator. The training label y_r is $m_r^\star$, or rej if even this closest candidate exceeds an error threshold δ (see Appendix C for details). The verifier loss is

$$\mathcal{L}_{\text{ver}} = \mathbb{E}\left[-\log \pi_{y_r}^r + \left(\pi_{\text{rej}}^r - \mathbf{1}[y_r = \text{rej}]\right)^2\right], \tag{11}$$

where $\mathbf{1}[\cdot]$ is the indicator function. At inference, we take the most probable class and accept the corresponding candidate, or fall back to standard inference if it is rej.

Training loss. We train the model in two stages. Let λ_{feat}, λ_{vel}, and λ_{aux} denote the loss weights. In the first stage, we combine DMD with the future alignment losses:

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{DMD}} + \lambda_{\text{feat}}\mathcal{L}_{\text{feat}} + \lambda_{\text{vel}}\mathcal{L}_{\text{vel}}. \tag{12}$$

In the second stage, we add the candidate alignment and verifier losses. For heads with candidates, $\mathcal{L}_{\text{feat}}$ and $\mathcal{L}_{\text{vel}}$ are computed from candidate $m_r^\star$. The loss function is then

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{stage1}} + \lambda_{\text{aux}}\left(\mathcal{L}_{\text{cand}} + \mathcal{L}_{\text{ver}}\right). \tag{13}$$

Speculative inference. In standard rolling inference, each backbone evaluation advances the window by only one chunk. A new chunk enters the last position as pure noise and must be processed

once by the backbone before it can move forward. Prospective Forcing lets the future prediction head take this step in advance, since the head is trained to match the output that the backbone will produce for the same chunk in a future step under the same pre-sampled noise.

At each step, the backbone processes $\mathcal{W}_r$ while G_1 proposes M candidates for the upcoming chunk and V_ϕ evaluates them. An accepted candidate is used directly as an additional clean chunk within the same backbone evaluation, similar to Medusa (Cai et al., 2024); unlike Medusa, verification uses a lightweight learned verifier rather than an extra backbone pass. If all candidates are rejected, the step reduces to a standard rolling step. Either way, the leading chunks are committed, the remaining $W - 1$ chunks are re-noised to their new positions, and a new chunk with its pre-sampled noise enters the last position, so accepted and rejected steps can be freely interleaved. In the worst case, the method reduces to standard autoregressive generation with negligible overhead.

To better maintain temporal consistency during generation, the KV cache $\mathcal{C}_r$ retains the key–value states of the first chunk as global context and the most recent committed chunk, which immediately precedes the window, as local context. We update the latter whenever the window advances.

5 Theoretical Guarantee

The alignment losses train $\mathbf{h}_W^r$ to anticipate the features and velocities of future chunks. We now characterize what this guarantees about the learned representation in an idealized single-chunk window ($W = 1$). We analyze the video chunks $\mathbf{x}_t$, each decoded from its latent $\mathbf{z}_t$, and assume they are generated from a first-order Markov state process $\{\mathbf{s}_t\}$, with $\mathbf{x}_t$ depending on $\mathbf{s}_{t:t-L}$. $\mathbf{h}_W^r$ then plays the role of an estimated state $\widehat{\mathbf{s}}_t = \widehat{\mathbf{m}}(\mathbf{x}_{t:t-L})$, from which the heads predict the future window $\mathbf{x}_{t+2L:t+L}$ under pre-sampled noise, which for $L = 1$ covers exactly the offsets $\{1, 2\}$ used in our experiments. Letting ϑ collect the backbone and head parameters, distribution matching loss in equation 3 makes the estimated model observationally equivalent to the distribution it matches, which plays the role of the true process here. Future alignment enters through the selection indicator $S_\alpha(\vartheta) = \mathbf{1}[\mathcal{L}_{\text{feat}}(\vartheta) + \mathcal{L}_{\text{vel}}(\vartheta) \le \alpha]$: exact alignment ($\alpha = 0$) forces the future window to depend on the past only through $\widehat{\mathbf{s}}_t$ (Lemma 1), which is the conditional independence our argument requires of the estimated model. Under the following conditions, this speculative representation recovers the true state up to a smooth invertible transformation.

Theorem 1 (Block-wise identifiability). *Let $\mathcal{S} = \prod_{k=1}^n I_k$ and $\widehat{\mathcal{S}} = \prod_{i=1}^n \widehat{I}_i$, products of open intervals in $\mathbb{R}^n$, be the true and estimated state spaces, and let d_x be the dimension of a chunk. Let $\mathcal{U}, \mathcal{Y} \subseteq \mathbb{R}^{(L+1)d_x}$ be open domains for the recovery window $\mathbf{x}_{t:t-L}$ and the predictive window $\mathbf{x}_{t+2L:t+L}$, respectively. Suppose that the true process is time homogeneous, that its states $\{\mathbf{s}_t\}$ form a first-order Markov process, that each $\mathbf{x}_t$ depends only on $\mathbf{s}_{t:t-L}$, and that the chunks are conditionally independent given the states. Suppose further that:*

1. *(Smooth and positive density) All state marginal densities and the transition density are strictly positive on their respective domains. For some t_0, $\mathbf{x}_{t_0:t_0-L}$ has a Lebesgue density positive almost everywhere on $\mathcal{U}$. For every $\mathbf{s} \in \mathcal{S}$, the conditional distribution of $\mathbf{x}_{t+2L:t+L}$ given $\mathbf{s}_t = \mathbf{s}$ admits a jointly Borel Lebesgue density.*
2. *(Smooth finite-context recovery) There is a surjective C^3 map $\mathbf{m} : \mathcal{U} \to \mathcal{S}$ such that $\mathbf{s}_t = \mathbf{m}(\mathbf{x}_{t:t-L})$ almost surely for every t, and its Jacobian has full row rank: $\operatorname{rank} D\mathbf{m}(\mathbf{u}) = n$ for every $\mathbf{u} \in \mathcal{U}$.*
3. *(Predictive density separation) For any distinct states $\mathbf{s}, \mathbf{s}' \in \mathcal{S}$, the set $\{\mathbf{y} \in \mathcal{Y} : p(\mathbf{y}|\mathbf{s}) \neq p(\mathbf{y}|\mathbf{s}')\}$ has positive probability under the marginal distribution of $\mathbf{x}_{t+2L:t+L}$.*

Suppose the estimated model $(\widehat{\mathbf{m}}, \widehat{p})$, generated by the backbone and prediction heads with $W = 1$, satisfies conditions 1–3 with $(\widehat{\mathbf{m}}, \widehat{p}, \widehat{\mathcal{S}})$ in place of $(\mathbf{m}, p, \mathcal{S})$, is time homogeneous, is observationally equivalent to the true model, and satisfies $S_0(\vartheta) = 1$. Then there is a fixed C^3 diffeomorphism $\varphi : \widehat{\mathcal{S}} \to \mathcal{S}$ such that $\mathbf{m}(\mathbf{u}) = \varphi(\widehat{\mathbf{m}}(\mathbf{u}))$ for every $\mathbf{u} \in \mathcal{U}$. Hence $\mathbf{s}_t = \varphi(\widehat{\mathbf{s}}_t)$ almost surely, simultaneously for all t.

Discussion on assumptions. The assumptions combine identification conditions without imposing parametric constraints. Smoothness and positivity provide regularity for the proof. Predictive density separation parallels Assumption 4 of (Hu & Schennach, 2008): distinct states must induce

distinct future distributions. Following (Chen et al., 2024b), we assume that the state can be recovered from a finite observation window; smoothness and full rank of the recovery maps support identification through a smooth invertible transformation. Two remarks delimit the scope of this result. The true process is the distribution matched by the training objective; under DMD this is the teacher's distribution rather than the data distribution. Moreover, the regularity conditions on the estimated model are assumed rather than derived, the selection is exact ($\alpha = 0$), and the analysis covers $W = 1$ without the KV-cache anchor. We do not formally relate identifiability to drift. However, future alignment consistently reduces drift and improves video quality empirically.

6 Experiments

6.1 Setup

Implementation. We build on Wan2.1-T2V-1.3B (Wan et al., 2025), initialized from the causal consistency distillation checkpoint of Causal Forcing++ (Zhao et al., 2026) and distilled from Wan2.1-T2V-14B, with $W = 5$, $K = 2$, and $M = 2$. Training details are given in Appendix D.1.

Evaluation. Following Liu et al. (2026), we generate 30-second videos for the same 200 prompts from Movie Gen Video Bench (Polyak et al., 2024) and report VBench (Huang et al., 2024), VBench-Long (Huang et al., 2025b), VideoScore (He et al., 2024), MUSIQ (Ke et al., 2021) drift, and FPS. Please refer to Appendices D.2 and D.3 for more details. We compare against six autoregressive generators built on the same base model: CausVid (Yin et al., 2025), Self Forcing (Huang et al., 2025a), Causal Forcing++ (Zhao et al., 2026), LongLive (Yang et al., 2026), Rolling Forcing (Liu et al., 2026), and Video-Mirai (Yu et al., 2026a). We use the released checkpoint for Video-Mirai and retrain the others with their official code.

6.2 Main Results

Figure 5 illustrates the advantage of Prospective Forcing. Several baselines lose visual fidelity over time, such as the overexposed close-up of CausVid and the color corruption of Video-Mirai, or fail to keep the subject in view, as in Rolling Forcing. Others keep the car intact but gradually become static: in Self Forcing and LongLive, the car stays at almost the same position and pose relative to the camera, as if driving in place. In contrast, our method keeps the same car throughout the 30 seconds while it actually travels along the track and through the turn, with the viewpoint and surroundings changing accordingly. It demonstrates that Prospective Forcing generates long videos that are visually faithful, temporally consistent, and rich in motion. Human evaluation reflects the same trend. In our user study as shown in Figure 4, participants prefer our method over Rolling Forcing, LongLive, and Video-Mirai in visual quality, long-term consistency, and dynamics. Quantitatively, our method achieves the highest VBench Total and Quality, VBench-Long Quality, and MUSIQ scores in Table 1. LongLive attains slightly better drift and VideoScore, but with much less motion (Dynamic Degree 62.0 vs. 100.0, Table 7); since static videos inflate consistency metrics, our method achieves a better balance between motion and consistency, as the user study also confirms.

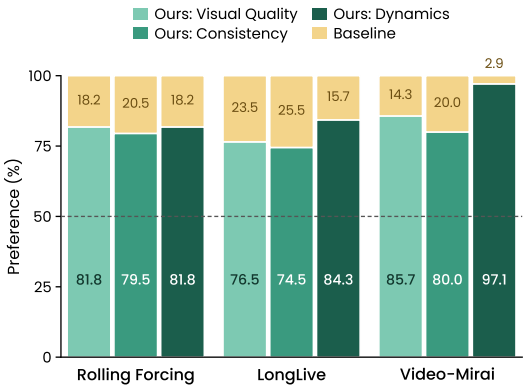

Figure 4: **User study.** Bars show the win rate of pairwise comparisons of human preference.

6.3 Ablation Study

Training and inference. Table 3 compares inference modes under the two training stages. Even with standard inference ($\ell = 0$), the stage-1 model already outperforms most baselines across multiple metrics and is preferred over them in most user study comparisons as shown in Table 6. The learned representation also supports accelerated inference. Committing the predictions of $G_1, \ldots, G_\ell$ without verification speeds up sampling by $1.56\times$ and $1.89\times$ for $\ell = 1$ and $\ell = 2$, while quality is largely preserved. In exchange, drift grows with ℓ. Stage 2 further introduces can-



Figure 5: Qualitative comparison on a 30-second video generated from the prompt "A rally car taking a fast turn on a track". Video comparison is provided in the supplementary materials.

Table 1: Quantitative comparison on 200 MovieGen prompts (30-second videos, 832×480, 16 fps). **Bold** and underline denote the best and second-best results.

Method	FPS↑	VBench			VBench-Long	VideoScore	MUSIQ	
		Total↑	Quality↑	Semantic↑	Quality↑	Avg.↑	Avg.↑	Drift↓
CausVid	**9.28**	76.22	77.02	73.00	77.43	2.637	66.49	9.02
Causal Forcing++	7.56	78.56	79.14	76.24	79.53	2.670	66.65	5.52
Self Forcing	7.79	78.27	78.76	76.33	78.24	2.905	68.84	6.18
LongLive	8.08	80.27	80.78	**78.22**	80.61	**3.108**	69.12	**3.43**
Rolling Forcing	7.95	78.63	80.05	72.94	82.20	2.775	68.92	5.32
Video-Mirai	8.13	74.98	75.14	74.32	76.29	2.704	64.54	11.90
Ours	9.23	**81.40**	**82.71**	76.15	**83.71**	3.069	**69.14**	3.64

didate generation and the learned verifier. The resulting model achieves the highest quality, and its drift stays close to that of standard inference while still running $1.20\times$ faster. To disentangle stage-2 training from verification, we also evaluate the stage-2 model with standard and unverified inference. Both perform close to their stage-1 counterparts, indicating that stage 2 alone brings little change. With verification, the same model achieves higher quality than both settings while using fewer backbone executions than $\ell = 0$, so the quality gain comes from accepting predictions selectively rather than from additional computation. The speedup is smaller than the fraction of skipped windows would suggest, since each committed chunk still requires a cache-update pass that cannot be skipped. Please refer to Appendix D.3 for more details.

Prediction horizon. Table 2 varies the set of future offsets predicted by the heads. For example, $\{1, 2\}$ means that G_1 and G_2 predict the first and second chunks after the window. All models are initialized with causal consistency distillation, trained with stage 1 only, and evaluated with standard inference, so the heads do not participate in generation and any difference comes from the learned representation. Compared with predicting only the next chunk, adding a second head improves both quality and drift. This suggests that predicting further ahead strengthens the speculative representation and mitigates temporal shortsightedness. A longer

Table 2: Ablation on prediction horizons.

Horizons	Quality↑	Drift↓
$\{1\}$	81.00	4.08
$\{1, 2\}$ (Ours)	**81.29**	3.29
$\{1, 2, 4\}$	80.21	**2.89**

Table 3: **Ablation on inference mode.** Results of the model trained with stage 1 only and with both stages, under different inference modes.

Train	Inference	FPS↑	Quality↑	Drift↓
Stage 1	$\ell = 0$	7.69	81.29	**3.29**
	Unverified, $\ell = 1$	12.03	81.80	4.22
	Unverified, $\ell = 2$	**14.51**	81.16	4.51
Stage 1 + 2	$\ell = 0$	7.81	81.63	3.42
	Unverified, $\ell = 1$	11.95	81.98	4.14
	Verified, $\ell = 1$ (Ours)	9.23	**82.71**	3.64

Table 4: **Ablation on design choices under stage** 1 **and standard inference.** Each variant changes one component of the ODE-initialized model (row "ODE").

Factor	Setting	Quality↑	Subject↑	Dynamic↑	Imaging↑	Drift↓
Architecture	Shared recurrent head	77.14	84.72	**100.0**	63.47	5.10
	Sequential heads	78.37	85.30	**100.0**	65.01	4.16
Head input	3 chunks	79.58	85.48	**100.0**	65.74	4.04
	5 chunks	77.25	84.41	**100.0**	64.13	4.61
Head output	3 chunks	80.43	94.23	17.0	**71.40**	2.58
	5 chunks	80.87	**95.28**	23.5	70.88	**2.05**
Loss	w/o $\mathcal{L}_{\text{feat}}, \mathcal{L}_{\text{vel}}$	80.05	84.36	98.5	68.24	5.32
	w/o $\mathcal{L}_{\text{feat}}$	77.94	85.17	**100.0**	64.30	4.55
Initialization	ODE	80.64	87.01	**100.0**	65.99	3.28
	Causal CD (Ours)	**81.29**	88.05	**100.0**	68.12	3.29

horizon is not always better, however. Extending the prediction to the fourth chunk yields the lowest drift but degrades quality, likely because distant chunks are harder to predict and provide noisier supervision. We therefore use $K = 2$ heads with offsets $\{1, 2\}$, which gives the best quality with low drift and also supports the accelerated inference with $\ell \leq 2$ in Table 3.

Design choices. Table 4 ablates our design choices under the same stage-1, standard-inference setting, where each variant changes one component of the ODE-initialized model. We report the VBench Quality score together with three of its dimensions (subject consistency, dynamic degree, and imaging quality), and the MUSIQ drift. We first examine the architecture of the prediction heads. Sharing one head recursively across offsets, or using separate heads that each take the previous prediction as input, both underperform our parallel heads. We then vary what the heads read and predict. Feeding them more past chunks makes the prediction harder to learn and lowers quality, whereas supervising more positions of the future window raises the scores but collapses the dynamic degree and produces nearly static videos. Mapping the current chunk to a single future chunk therefore works best. For the loss, training the same ODE-initialized model with DMD only, which is equivalent to Rolling Forcing, yields lower quality and substantially higher drift, showing that the gain comes from future alignment rather than initialization. Removing the feature alignment loss degrades performance even below this DMD-only model, which indicates that learning the representation itself is essential. Finally, causal consistency distillation (Zhao et al., 2026) further improves over ODE (Yin et al., 2025), showing that our method is complementary to a stronger initialization.

7 Conclusion

We introduced Prospective Forcing to address temporal shortsightedness in autoregressive long video diffusion. Rather than supervising only the next chunk, it trains lightweight heads to anticipate chunks at multiple future offsets, distilling targets from the backbone's own later predictions. This turns the shared history into a speculative representation that looks beyond the immediate future. The same heads further enable verifier-guided speculative inference, which skips backbone executions whenever a predicted chunk is accepted and otherwise falls back to standard generation. Prospective Forcing outperforms baselines across multiple quantitative metrics and is preferred in human evaluations, while also achieving faster sampling. **Limitations.** Although Prospective Forcing accelerates sampling, the inference process still requires additional cache-update passes, which limits the speedup. We leave reducing this overhead to future work.

AI USE STATEMENT

We used generative AI tools to assist with editing the manuscript for grammar and readability, with writing and modifying parts of the experimental code, and with parts of the theoretical derivations. All AI-assisted text, code, and derivations were carefully reviewed, tested, and verified by the authors, who take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

Anonymized source code is provided in the supplementary material. Implementation details and evaluation protocols are given in Appendices D.1 and D.2, respectively.

ETHICS STATEMENT

Our user study involved voluntary participants who compared generated videos. No personal data was collected during the user study.

REFERENCES

Ralph Abraham, Jerrold E. Marsden, and Tudor Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75 of *Applied Mathematical Sciences*. Springer, New York, second edition, 1988. doi: 10.1007/978-1-4612-1029-0.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL https://arxiv.org/abs/1607.06450.

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. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=QaCCuDfBk2.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. doi: 10.48550/arXiv.2311.15127. URL https://arxiv.org/abs/2311.15127.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C. Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5209–5235. PMLR, 2024.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, volume 37, pp. 24081–24125. Curran Associates, Inc., 2024a. doi: 10.52202/079017-0759. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2aee1c4159e48407d68fe16ae8e6e49e-Abstract-Conference.html.

Guangyi Chen, Yifan Shen, Zhenhao Chen, Xiangchen Song, Yuewen Sun, Weiran Yao, Xiao Liu, and Kun Zhang. CaRiNG: Learning temporal causal representation under non-invertible generation process. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7236–7259. PMLR, 2024b.

Shuo Chen, Cong Wei, Sun Sun, Tiancheng Shen, Ping Nie, Kai Zou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lv86f7Qr9V.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. LoL: Longer than longer, scaling video generation to hour. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38132–38142, June 2026a.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. In *International Conference on Learning Representations*, volume 2026, pp. 85802–85822, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8a30aba6514b56d02976f49797f6338a-Paper-Conference.pdf.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Roziere, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 15706–15734. PMLR, 2024. URL https://proceedings.mlr.press/v235/gloeckle24a.html.

Yuwei Guo, Ceyuan Yang, Hao He, Yang Zhao, Meng Wei, Zhenheng Yang, Weilin Huang, and Dahua Lin. End-to-end training for autoregressive video diffusion via self-resampling. *arXiv preprint arXiv:2512.15702*, 2025. doi: 10.48550/arXiv.2512.15702. URL https://arxiv.org/abs/2512.15702.

Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Remi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 3875–3886. PMLR, 2020. URL https://proceedings.mlr.press/v119/guo20g.html.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, Kai Wang, Quy Duc Do, Yuansheng Ni, Bohan Lyu, Yaswanth Narsupalli, Rongqi Fan, Zhiheng Lyu, Bill Yuchen Lin, and Wenhu Chen. VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2105–2123. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.127. URL https://aclanthology.org/2024.emnlp-main.127/.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2568–2577, June 2025. doi: 10.1109/CVPR52734.2025.00245. URL https://openaccess.thecvf.com/content/CVPR2025/html/Henschel_StreamingT2V_Consistent_Dynamic_and_Extendable_Long_Video_Generation_from_Text_CVPR_2025_paper.html.

Yingyao Hu and Susanne M Schennach. Instrumental variable treatment of nonclassical measurement error models. *Econometrica*, 76(1):195–216, 2008. doi: 10.1111/j.0012-9682.2008.00823.x.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 167283–167308. Curran Associates, Inc., 2025a. doi: 10.52202/085713-5576. URL

https://proceedings.neurips.cc/paper_files/paper/2025/file/f4823f831af67a3ef15e41a85434422a-Paper-Conference.pdf.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024. doi: 10.1109/CVPR52733.2024.02060. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b. doi: 10.1109/TPAMI.2025.3633890.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5148–5157, 2021. doi: 10.1109/ICCV48922.2021.00510. URL https://openaccess.thecvf.com/content/ICCV2021/html/Ke_MUSIQ_Multi-Scale_Image_Quality_Transformer_ICCV_2021_paper.html.

Alexander S. Kechris. *Classical Descriptive Set Theory*, volume 156 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. doi: 10.1007/978-1-4612-4190-4.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. FIFO-Diffusion: Generating infinite videos from text without training. In *Advances in Neural Information Processing Systems*, volume 37, pp. 89834–89868. Curran Associates, Inc., 2024. doi: 10.52202/079017-2853. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a397986e0f34d4b1f0b640686ceaeff7-Paper-Conference.pdf.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. doi: 10.48550/arXiv.2412.03603. URL https://arxiv.org/abs/2412.03603.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.016. URL https://www.roboticsproceedings.org/rss22/p016.html.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28935–28948. PMLR, 2024. URL https://proceedings.mlr.press/v235/li24bt.html.

Zijian Li, Minghao Fu, Junxian Huang, Yifan Shen, Ruichu Cai, Yuewen Sun, Guangyi Chen, and Kun Zhang. Towards identifiability of hierarchical temporal causal representation learning. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 19680–19719. Curran Associates, Inc., 2025. doi: 10.52202/085713-0662. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1c6decac1477fbcc2cbf12d314ce0133-Paper-Conference.pdf.

Jingyu Lin, Xinyi Shang, Peng Sun, Cunjian Chen, and Zhiqiang Shen. DynaMem: Consistent long video generation via hierarchical memory and motion priors. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=T8UsggiFIw.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IAyzXjbfwo.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with Rectified Flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

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. URL https://arxiv.org/abs/2410.13720.

Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. FreeNoise: Tuning-free longer video diffusion via noise rescheduling. In *International Conference on Learning Representations*, volume 2024, pp. 5260–5274, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/15ce8e7afe5ee95bad56e3b9be28d3d1-Paper-Conference.pdf.

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uCQfPZwRaUu.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, volume 2025, pp. 73754–73776, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/b71ecea210f7159f31e46631fe5c838f-Paper-Conference.pdf.

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

Wenhao Wang and Yi Yang. VidProM: A million-scale real prompt-gallery dataset for text-to-video diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 65618–65642, 2024. doi: 10.52202/079017-2096. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/78b3e7836e3b7dea79d809b0c99cb097-Abstract-Datasets_and_Benchmarks_Track.html.

Gangwei Xu, Qihang Zhang, Jiaming Zhou, Xing Zhu, Yujun Shen, Xin Yang, and Yinghao Xu. Next forcing: Causal world modeling with multi-chunk prediction. *arXiv preprint arXiv:2606.11187*, 2026. URL https://arxiv.org/abs/2606.11187.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Ying-Cong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time interactive long video generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nCAODkpsPJ.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.06072.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. In *International Conference on Machine Learning*, 2026. URL https://cvlab-kaist.github.io/DeepForcing/.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, June 2024. doi: 10.1109/CVPR52733.2024.00632. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.html.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22963–22974, June 2025. doi: 10.1109/CVPR52734.2025.02138. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yin_From_Slow_Bidirectional_to_Fast_Autoregressive_Video_Diffusion_Models_CVPR_2025_paper.html.

Yonghao Yu, Lang Huang, Runyi Li, Zerun Wang, and Toshihiko Yamasaki. Video-Mirai: Autoregressive video diffusion models need foresight. *arXiv preprint arXiv:2606.03971*, 2026a. URL https://arxiv.org/abs/2606.03971.

Yonghao Yu, Lang Huang, Zerun Wang, Runyi Li, and Toshihiko Yamasaki. Mirai: Autoregressive visual generation needs foresight. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30511–30520, June 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Yu_Mirai_Autoregressive_Visual_Generation_Needs_Foresight_CVPR_2026_paper.html.

Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal Forcing++: Scalable few-step autoregressive diffusion distillation for real-time interactive video generation. *arXiv preprint arXiv:2605.15141*, 2026. URL https://arxiv.org/abs/2605.15141.

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 *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.02214.

A PROOF OF THEOREM 1

We first show how future alignment yields the conditional independence required of the estimated model, and then prove the case $L = 2$, for which the recovery window is $\mathbf{x}_{t:t-2}$ and the predictive window is $\mathbf{x}_{t+4:t+2}$. The extension to arbitrary finite $L \geq 1$ is given at the end of this section.

Lemma 1 (Future alignment induces a predictive bottleneck)**.** *Consider the estimated model with $W = 1$, in which the noise of each chunk is drawn independently of the past, each chunk is a measurable function of its backbone velocity and its noise (through the denoising update and the VAE decoder), and the prediction heads map $\widehat{\mathbf{s}}_t$ and this noise to the features and velocities of the chunks in $\mathbf{x}_{t+2L:t+L}$. If $S_0(\vartheta) = 1$, then $\mathbf{x}_{t+2L:t+L} \perp\!\!\!\perp \mathbf{x}_{t:t-L} \mid \widehat{\mathbf{s}}_t$ in the estimated model.*

Proof. $S_0(\vartheta) = 1$ means that the population alignment losses vanish, so almost surely the head outputs coincide with the backbone features and velocities computed when the corresponding chunks enter the window under the same noise realization. Hence $\mathbf{x}_{t+2L:t+L} = g(\widehat{\mathbf{s}}_t, \boldsymbol{\epsilon}_{t+L:t+2L})$ almost surely for a measurable map g. The noise $\boldsymbol{\epsilon}_{t+L:t+2L}$ is independent of $\mathbf{x}_{t:t-L}$, and $\widehat{\mathbf{s}}_t = \widehat{\mathbf{m}}(\mathbf{x}_{t:t-L})$ is a function of the recovery window, so the conditional law of $\mathbf{x}_{t+2L:t+L}$ given $\mathbf{x}_{t:t-L}$ depends on it only through $\widehat{\mathbf{s}}_t$. □

Proof of Theorem 1. Fix t. The latent variables entering $\mathbf{x}_{t:t-2}$ belong to $\mathbf{s}_{t:t-4}$, whereas those entering $\mathbf{x}_{t+4:t+2}$ belong to $\mathbf{s}_{t+4:t}$. By first-order Markovity of $\{\mathbf{s}_t\}$, the latent blocks $\mathbf{s}_{t-1:t-4}$ and $\mathbf{s}_{t+4:t+1}$ are conditionally independent given $\mathbf{s}_t$. Together with the conditional independence of the chunks given the states, this gives

$$\mathbf{x}_{t+4:t+2} \perp\!\!\!\perp \mathbf{x}_{t:t-2} | \mathbf{s}_t. \tag{14}$$

The future window starts at $t + 2$ because $\mathbf{x}_{t+1}$ may still depend on $\mathbf{s}_{t-1}$. In the estimated model, Lemma 1 gives $\mathbf{x}_{t+4:t+2} \perp\!\!\!\perp \mathbf{x}_{t:t-2} | \widehat{\mathbf{s}}_t$; this is the only step that uses the selection $S_0(\vartheta) = 1$.

Enumerate the lower rectangles in $\mathbb{R}^{3d_x}$ with rational endpoints, intersected with $\mathcal{Y}$, as $(A_j)_{j\geq 1}$. Their probabilities determine a distribution on $\mathcal{Y}$. Define

$$\Phi(\mathbf{s}) = \left(\int_{A_j} p(\mathbf{y}|\mathbf{s})\, d\mathbf{y} \right)_{j\geq 1}, \qquad \widehat{\Phi}(\widehat{\mathbf{s}}) = \left(\int_{A_j} \widehat{p}(\mathbf{y}|\widehat{\mathbf{s}})\, d\mathbf{y} \right)_{j\geq 1}. \tag{15}$$

These are Borel maps into $[0, 1]^{\mathbb{N}}$. They are independent of t by time homogeneity, and injective by condition 3 and its counterpart for the estimated model. They record the complete predictive distributions; no differentiability of Φ or $\widehat{\Phi}$ is used.

For each j, the tower property, Equ. (14), and exact recovery give

$$\begin{aligned} \mathbb{P}(\mathbf{x}_{t+4:t+2} \in A_j | \mathbf{x}_{t:t-2}) &= \mathbb{E}\left[\mathbb{P}(\mathbf{x}_{t+4:t+2} \in A_j | \mathbf{s}_t, \mathbf{x}_{t:t-2}) \,|\, \mathbf{x}_{t:t-2} \right] \\ &= \mathbb{E}[\Phi_j(\mathbf{s}_t) | \mathbf{x}_{t:t-2}] \\ &= \Phi_j(\mathbf{m}(\mathbf{x}_{t:t-2})) \quad \text{almost surely.} \end{aligned} \tag{16}$$

The estimated model gives the corresponding expression $\widehat{\Phi}_j(\widehat{\mathbf{m}}(\mathbf{x}_{t:t-2}))$. Observational equivalence makes the left-hand sides equal under the common observation law. Countability of the event collection gives

$$\Phi(\mathbf{m}(\mathbf{x}_{t:t-2})) = \widehat{\Phi}(\widehat{\mathbf{m}}(\mathbf{x}_{t:t-2})) \quad \text{almost surely.} \tag{17}$$

Equivalently, for almost every recovery window, the predictive densities satisfy

$$p(\mathbf{y}|\mathbf{m}(\mathbf{x}_{t:t-2})) = \widehat{p}(\mathbf{y}|\widehat{\mathbf{m}}(\mathbf{x}_{t:t-2})) \quad \text{for Lebesgue-almost every } \mathbf{y} \in \mathcal{Y}. \tag{18}$$

By the Lusin–Souslin theorem, an injective Borel map between standard Borel spaces has a Borel image and a Borel inverse on that image (Kechris, 1995). Hence $\mathcal{P} = \Phi(\mathcal{S}) \cap \widehat{\Phi}(\widehat{\mathcal{S}})$, $\mathcal{S}_0 = \Phi^{-1}(\mathcal{P})$ and $\widehat{\mathcal{S}}_0 = \widehat{\Phi}^{-1}(\mathcal{P})$ are Borel sets. Equ. (17) shows that $\mathcal{S}_0$ and $\widehat{\mathcal{S}}_0$ have probability one for the respective states at every time. On these sets, the maps $\varphi_0 = \Phi^{-1} \circ \widehat{\Phi} : \widehat{\mathcal{S}}_0 \to \mathcal{S}_0$ and $\varphi_0^{-1} = \widehat{\Phi}^{-1} \circ \Phi$ are mutually inverse Borel maps. In particular, $\mathbf{m}(\mathbf{x}_{t:t-2}) = \varphi_0(\widehat{\mathbf{m}}(\mathbf{x}_{t:t-2}))$ almost surely. Equality of the full prediction ranges is not needed.

We next obtain a smooth correspondence using the recovery maps. Extend φ_0 and φ_0^{-1} arbitrarily as Borel maps to the remaining states. These extensions need not be inverses there. At time t_0 in condition 1, positivity of the window density makes the almost-sure relation above valid for Lebesgue-almost every $\mathbf{u} \in \mathcal{U}$:

$$\mathbf{m}(\mathbf{u}) = \varphi_0(\widehat{\mathbf{m}}(\mathbf{u})). \tag{19}$$

By the counterpart of condition 2 for the estimated model, $\widehat{\mathbf{m}}$ is C^3 with full row rank. The local submersion theorem for C^3 maps gives, near every window, local C^3 coordinates $\kappa : O_1 \times O_2 \to \mathcal{U}$, with $O_1 \subseteq \widehat{\mathcal{S}}$ and $O_2 \subseteq \mathbb{R}^{3d_x - n}$ open, such that $\widehat{\mathbf{m}}(\kappa(a, b)) = a$, see (Abraham et al., 1988). The coordinate change preserves null sets, so Equ. (19) becomes $\mathbf{m}(\kappa(a, b)) = \varphi_0(a)$ for Lebesgue-almost every (a, b). Fubini's theorem gives this equality for almost every b at almost every a. For each such a, continuity in b extends it to all b. Fix $b_0 \in O_2$ and define $\varphi_{O_1}(a) = \mathbf{m}(\kappa(a, b_0))$. This map is C^3 and satisfies $\mathbf{m}(\kappa(a, b)) = \varphi_{O_1}(a)$ for almost every a and all b. Continuity in a then extends this identity to all $(a, b) \in O_1 \times O_2$.

On overlapping state neighborhoods, these local maps agree almost everywhere with φ_0, so continuity makes them agree at every point. Surjectivity of $\widehat{\mathbf{m}}$ ensures that the neighborhoods cover $\widehat{\mathcal{S}}$. Thus they define a global C^3 map φ satisfying $\mathbf{m}(\mathbf{u}) = \varphi(\widehat{\mathbf{m}}(\mathbf{u}))$ for every $\mathbf{u} \in \mathcal{U}$. Applying the same argument to the reverse almost-sure relation gives a C^3 map $\boldsymbol{\rho} : \mathcal{S} \to \widehat{\mathcal{S}}$ such that $\widehat{\mathbf{m}} = \boldsymbol{\rho} \circ \mathbf{m}$ on $\mathcal{U}$. Surjectivity of both recovery maps implies $\boldsymbol{\rho} \circ \varphi = \mathrm{id}_{\widehat{\mathcal{S}}}$ and $\varphi \circ \boldsymbol{\rho} = \mathrm{id}_{\mathcal{S}}$. Consequently, φ is a C^3 diffeomorphism. It is unique because $\widehat{\mathbf{m}}$ is onto. The recovery maps and the identity on $\mathcal{U}$ are independent of time and history. Since every recovery window belongs to $\mathcal{U}$ almost surely and $\mathbb{Z}$ is countable, the recovered states satisfy the same identity simultaneously at all times on an event of probability one. □

Lemma 2 (Global disentanglement on product domains)**.** *Let $\varphi : \widehat{\mathcal{S}} \to \mathcal{S}$ be a C^3 diffeomorphism between the products of open intervals $\widehat{\mathcal{S}} = \prod_{i=1}^n \widehat{I}_i$ and $\mathcal{S} = \prod_{k=1}^n I_k$. If its Jacobian has at most one nonzero entry in each row at every point, then φ and φ^{-1} are fixed permutations composed with fixed one-dimensional C^3 diffeomorphisms.*

Proof. Write $\varphi = (\varphi_1, \ldots, \varphi_n)$. If the Jacobian of φ has at most one nonzero entry in each row at every point, invertibility implies exactly one nonzero entry in each row and each column. For a row k, its nonzero column is locally constant: a nonzero entry stays nonzero in a sufficiently small neighborhood by continuity, and no other entry in that row can become nonzero there. Since $\widehat{\mathcal{S}}$ is connected, this column is one fixed index $\sigma(k)$ throughout the domain. The map σ is a permutation.

For $j \neq \sigma(k)$, $\partial\varphi_k/\partial\hat{s}_j = 0$ on $\widehat{\mathcal{S}}$. Any two points with the same $\hat{s}_{\sigma(k)}$ can be joined by a sequence of coordinate segments in the product domain, along each of which φ_k is constant. Hence $\varphi_k(\widehat{\mathbf{s}}) = q_k(\hat{s}_{\sigma(k)})$. Each q_k is C^3 and its derivative is nonzero on $\widehat{I}_{\sigma(k)}$, so it is strictly monotone there. Surjectivity of φ implies that q_k maps onto I_k. Its inverse is therefore a global C^3 diffeomorphism. Setting $\pi(i) = \sigma^{-1}(i)$ and $\mathcal{T}_i = q_{\pi(i)}^{-1}$ gives the claimed form of φ^{-1}. □

Theorem 2 (Component-wise identifiability, $L = 2$)**.** *Consider the setting of Theorem 1 with $L = 2$, so that $\mathcal{U}, \mathcal{Y} \subseteq \mathbb{R}^{3d_x}$ are the open domains for the recovery window $\mathbf{x}_{t:t-2}$ and the predictive window $\mathbf{x}_{t+4:t+2}$, respectively. Suppose the true model and its transition density satisfy the following assumptions:*

1. *(Smooth and positive density) All state marginal densities and the transition density are strictly positive on their respective domains. Each coordinate log density $\log p(s_{k,t}|\mathbf{s}_{t-1})$ is C^3 (i.e., has continuous third-order partial derivatives). For some t_0, $\mathbf{x}_{t_0:t_0-2}$ has a Lebesgue density positive almost everywhere on $\mathcal{U}$. For every $\mathbf{s} \in \mathcal{S}$, the conditional distribution of $\mathbf{x}_{t+4:t+2}$ given $\mathbf{s}_t = \mathbf{s}$ admits a jointly Borel Lebesgue density.*

2. *(Conditional independence) The current-state coordinates are mutually independent conditional on the previous state: $p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \prod_{k=1}^n p(s_{k,t}|\mathbf{s}_{t-1})$.*

3. *(Smooth finite-context recovery) There is a surjective C^3 map $\mathbf{m} : \mathcal{U} \to \mathcal{S}$ such that $\mathbf{s}_t = \mathbf{m}(\mathbf{x}_{t:t-2})$ almost surely for every t. Its Jacobian has full row rank: $\operatorname{rank} D\mathbf{m}(\mathbf{u}) = n$ for every $\mathbf{u} \in \mathcal{U}$.*

4. *(Predictive density separation) For any distinct states $\mathbf{s}, \mathbf{s}' \in \mathcal{S}$, the set $\{\mathbf{y} \in \mathcal{Y} : p(\mathbf{y}|\mathbf{s}) \neq p(\mathbf{y}|\mathbf{s}')\}$ has positive probability under the marginal distribution of $\mathbf{x}_{t+4:t+2}$.*

5. *(Sufficient variability) Let $\eta_{k,t} = \log p(s_{k,t}|\mathbf{s}_{t-1})$ and, for $l = 1, \ldots, n$, define*

$$\boldsymbol{\omega}_{l,t}(\mathbf{s}_t, \mathbf{s}_{t-1}) = \left(\frac{\partial^2 \eta_{1,t}}{\partial s_{1,t} \partial s_{l,t-1}}, \cdots, \frac{\partial^2 \eta_{n,t}}{\partial s_{n,t} \partial s_{l,t-1}}, \frac{\partial^3 \eta_{1,t}}{\partial s_{1,t}^2 \partial s_{l,t-1}}, \cdots, \frac{\partial^3 \eta_{n,t}}{\partial s_{n,t}^2 \partial s_{l,t-1}} \right)^\top \in \mathbb{R}^{2n}. \tag{20}$$

For every fixed $\mathbf{s}_t \in \mathcal{S}$, there exist $2n$ pairs $(l_u, \mathbf{s}_{t-1}^{(u)})$, with $l_u \in \{1, \ldots, n\}$ and $\mathbf{s}_{t-1}^{(u)} \in \mathcal{S}$, such that the vectors $\boldsymbol{\omega}_{l_u,t}(\mathbf{s}_t, \mathbf{s}_{t-1}^{(u)})$ are linearly independent.

Then, for any estimated model that satisfies these assumptions with $(\widehat{\mathbf{m}}, \widehat{p}, \widehat{\mathcal{S}})$ in place of $(\mathbf{m}, p, \mathcal{S})$, meets the requirements on the estimated model in Theorem 1, and satisfies $S_0(\vartheta) = 1$, there exist a single permutation π and fixed C^3 diffeomorphisms $\mathcal{T}_i : I_{\pi(i)} \to \widehat{I}_i$ for which $\widehat{m}_i(\mathbf{x}_{t:t-2}) = \mathcal{T}_i\big(m_{\pi(i)}(\mathbf{x}_{t:t-2})\big)$ holds for $i = 1, \ldots, n$.

Proof. By Theorem 1, there is a fixed C^3 diffeomorphism $\varphi : \widehat{\mathcal{S}} \to \mathcal{S}$. On the common observed-process space, define $\mathbf{s}_t = \mathbf{m}(\mathbf{x}_{t:t-2})$ and $\widehat{\mathbf{s}}_t = \widehat{\mathbf{m}}(\mathbf{x}_{t:t-2})$. Assumption 2 ensures that these sequences have the latent-process distributions of their respective models, and Theorem 1 gives $\mathbf{s}_t = \varphi(\widehat{\mathbf{s}}_t)$ simultaneously for every t almost surely. In particular, the same function also relates the previous states: $\widehat{\mathbf{s}}_{t-1} = \varphi^{-1}(\mathbf{s}_{t-1})$.

Write $\varphi = (\varphi_1, \ldots, \varphi_n)$ and denote the Jacobian by

$$\mathbf{H}_t = \frac{\partial \varphi(\widehat{\mathbf{s}}_t)}{\partial \widehat{\mathbf{s}}_t}, \qquad H_{ki,t} = \frac{\partial \varphi_k(\widehat{\mathbf{s}}_t)}{\partial \hat{s}_{i,t}}. \tag{21}$$

The subscript t records the state at which the same fixed Jacobian function is evaluated; it does not introduce a time-dependent transformation. In the density identities below, $\widehat{\mathbf{s}}_t$ and $\mathbf{s}_{t-1}$ denote freely varying state values. Conditional change of variables gives

$$\widehat{p}(\widehat{\mathbf{s}}_t|\varphi^{-1}(\mathbf{s}_{t-1})) = p(\varphi(\widehat{\mathbf{s}}_t)|\mathbf{s}_{t-1}) \, |\det \mathbf{H}_t|. \tag{22}$$

This identity first holds for almost every $(\widehat{\mathbf{s}}_t, \mathbf{s}_{t-1})$ with respect to Lebesgue measure, since the previous-state density is strictly positive. The same condition and the smoothness of φ ensure continuity of both sides, which extends the identity to all $(\widehat{\mathbf{s}}_t, \mathbf{s}_{t-1}) \in \widehat{\mathcal{S}} \times \mathcal{S}$. Recall $\eta_{k,t} = \log p(s_{k,t}|\mathbf{s}_{t-1})$. Evaluating these functions at $\mathbf{s}_t = \varphi(\widehat{\mathbf{s}}_t)$, conditional independence Assumption 2 then gives

$$\log \widehat{p}(\widehat{\mathbf{s}}_t|\varphi^{-1}(\mathbf{s}_{t-1})) = \sum_{k=1}^{n} \eta_{k,t} + \log |\det \mathbf{H}_t|. \tag{23}$$

The estimated model also satisfies the assumptions of Theorem 2, so its transition density satisfies condition 2. For $i \neq j$, with $\mathbf{s}_{t-1}$ held fixed,

$$\frac{\partial^2 \log \widehat{p}(\widehat{\mathbf{s}}_t|\varphi^{-1}(\mathbf{s}_{t-1}))}{\partial \hat{s}_{i,t} \partial \hat{s}_{j,t}} = 0. \tag{24}$$

Using the chain rule in Equ. (23), we obtain

$$0 = \sum_{k=1}^{n} \left[\frac{\partial^2 \eta_{k,t}}{\partial s_{k,t}^2} H_{ki,t} H_{kj,t} + \frac{\partial \eta_{k,t}}{\partial s_{k,t}} \frac{\partial H_{ki,t}}{\partial \hat{s}_{j,t}} \right] + \frac{\partial^2 \log |\det \mathbf{H}_t|}{\partial \hat{s}_{i,t} \partial \hat{s}_{j,t}}, \tag{25}$$

where all derivatives of $\eta_{k,t}$ are evaluated at $(\varphi_k(\widehat{\mathbf{s}}_t), \mathbf{s}_{t-1})$.

Differentiate Equ. (25) with respect to $s_{l,t-1}$, holding $\widehat{\mathbf{s}}_t$ fixed. Since φ and $\mathbf{H}_t$ depend only on $\widehat{\mathbf{s}}_t$, the determinant term vanishes and no derivative of either function with respect to the previous state appears:

$$0 = \sum_{k=1}^{n} \left[\frac{\partial^3 \eta_{k,t}}{\partial s_{k,t}^2 \partial s_{l,t-1}} H_{ki,t} H_{kj,t} + \frac{\partial^2 \eta_{k,t}}{\partial s_{k,t} \partial s_{l,t-1}} \frac{\partial H_{ki,t}}{\partial \hat{s}_{j,t}} \right]. \tag{26}$$

Fix $\widehat{\mathbf{s}}_t$, and hence $\mathbf{s}_t = \varphi(\widehat{\mathbf{s}}_t)$. The $2n$ coefficients

$$\left(\frac{\partial H_{1i,t}}{\partial \hat{s}_{j,t}}, \ldots, \frac{\partial H_{ni,t}}{\partial \hat{s}_{j,t}}, H_{1i,t}H_{1j,t}, \ldots, H_{ni,t}H_{nj,t}\right)^{\mathsf{T}}$$

do not change as $\mathbf{s}_{t-1}$ varies. Equ. (26) says that their inner product with $\boldsymbol{\omega}_{l,t}(\mathbf{s}_t, \mathbf{s}_{t-1})$ is zero for every l and every previous state. The $2n$ linearly independent vectors in sufficient variability Assumption 5 therefore force all these coefficients to vanish. In particular,

$$H_{ki,t}H_{kj,t} = 0 \quad \text{for all } k,\ i \neq j,\ \widehat{\mathbf{s}}_t \in \widehat{\mathcal{S}}. \tag{27}$$

Apply Lemma 2 to φ. There are fixed one-dimensional C^3 diffeomorphisms $\mathcal{T}_i$ and a fixed permutation π such that $\hat{s}_{i,t} = \mathcal{T}_i(s_{\pi(i),t})$ almost surely for every t. Substitution of the recovery maps proves that $\widehat{m}_i(\mathbf{x}_{t:t-2}) = \mathcal{T}_i\big(m_{\pi(i)}(\mathbf{x}_{t:t-2})\big)$ holds for $i = 1, \ldots, n$. □

Extension to arbitrary finite L. Fix $L \geq 1$. Replace $\mathbf{x}_{t:t-2}$ by $\mathbf{x}_{t:t-L}$ and $\mathbf{x}_{t+4:t+2}$ by $\mathbf{x}_{t+2L:t+L}$ throughout the alignment argument. Because each chunk $\mathbf{x}_{t'}$ depends only on $\mathbf{s}_{t':t'-L}$, the latent variables entering the recovery window belong to $\mathbf{s}_{t:t-2L}$, while those entering the predictive window belong to $\mathbf{s}_{t+2L:t}$. Thus the two latent blocks share only $\mathbf{s}_t$; first-order Markovity and conditional independence of the chunks give $\mathbf{x}_{t+2L:t+L} \perp\!\!\!\perp \mathbf{x}_{t:t-L} \mid \mathbf{s}_t$, and Lemma 1 gives the same for the estimated model. The predictive-separation and smooth-recovery steps of Theorem 1 then apply with $\mathbb{R}^{(L+1)d_x}$ in place of $\mathbb{R}^{3d_x}$ and $\mathbb{R}^{(L+1)d_x - n}$ in place of $\mathbb{R}^{3d_x - n}$, yielding the same fixed C^3 diffeomorphism φ. Lemma 2 and the change-of-variables and sufficient-variability argument above involve only the one-step transition density $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ and are unchanged. Consequently, the stated component-wise identification holds for every fixed finite $L \geq 1$.

B Related Work

Autoregressive video generation. Modern video diffusion models typically denoise all frames in a fixed temporal window with bidirectional attention (Blattmann et al., 2023; Yang et al., 2025; Kong et al., 2024; Wan et al., 2025). This design supports high-quality offline synthesis but requires the complete clip to be generated jointly. Autoregressive video models instead generate frames or chunks sequentially. This causal formulation reduces viewing latency and enables efficient streaming over extended horizons through KV caching (Yin et al., 2025; Huang et al., 2025a). Within this family, Diffusion Forcing (Chen et al., 2024a) assigns an independent noise level to each token. It connects causal next-token prediction with full-sequence diffusion and supports flexible generation horizons. Another line uses distribution matching distillation (DMD) (Yin et al., 2024) to reduce the sampling cost of causal generators. CausVid (Yin et al., 2025) combines ODE initialization with asymmetric DMD to obtain a few-step causal student for real-time generation. Self Forcing (Huang et al., 2025a) trains on self-generated rollouts and applies a holistic distribution-matching objective to mitigate exposure bias. Causal Forcing (Zhu et al., 2026) uses an autoregressive ODE teacher to bridge the architectural gap between bidirectional teachers and causal students. Causal Forcing++ (Zhao et al., 2026) replaces costly offline ODE trajectories with causal consistency distillation, making few-step initialization scalable. Beyond DMD-based post-training, Resampling Forcing (Guo et al., 2025) uses self-resampled histories and a sparse causal mask for teacher-free end-to-end training.

Long video generation. Consistency over long autoregressive rollouts has been addressed through training, denoising, and context-management strategies. Self-Forcing++ (Cui et al., 2026b) mitigates quality degradation when the student extrapolates beyond its training horizon by providing teacher guidance on segments sampled from self-generated long videos. Rolling Forcing (Liu et al., 2026) mitigates error accumulation by jointly denoising frames at progressively increasing noise levels within a rolling window and retaining initial-frame KV states as a global context anchor. StreamingT2V (Henschel et al., 2025) uses features from the preceding chunk to facilitate smooth transitions and preserves scene and object features from the first video chunk to prevent forgetting of the initial scene. LongLive (Yang et al., 2026) uses streaming long tuning to align long-video training and inference, together with short window attention and a frame sink to improve efficiency while preserving long-range consistency. Context Forcing (Chen et al., 2026) resolves

the mismatch between a long-context student and a short-context teacher by training with a long-context teacher, while its Slow-Fast Memory architecture makes the growing context computationally tractable. DynaMem (Lin et al., 2026) addresses semantic drift, motion decay, and appearance instability through Semantic-Adaptive Hierarchical Memory, Motion-Prioritized Optimization, and Reference-Anchored Perceptual Alignment. Recent training-free approaches such as Deep Forcing and LoL implement context management at inference time by modifying cache retention and/or positional encoding. Deep Forcing (Yi et al., 2026) introduces Deep Sink, which retains a substantially enlarged attention sink with temporal RoPE adjustment, and Participative Compression, which prunes redundant and degraded history from the KV cache. LoL (Cui et al., 2026a) introduces multi-head RoPE jitter to break inter-head attention homogenization and mitigate sink-collapse over extreme horizons. Instead, our method improves temporal consistency by learning representations with long-horizon predictive capabilities.

Future-predictive representations and multi-horizon generation. Predicting future observations has long served as an auxiliary supervision for learning sequential representations. PBL (Guo et al., 2020) couples forward prediction of future-observation embeddings with reverse prediction of agent states from the corresponding latents, achieving a bootstrapping objective. SPR (Schwarzer et al., 2021) instead rolls a learned transition model forward in latent space and matches its multi-step predictions to representations produced by an exponential-moving-average target encoder. In text generation, multi-token prediction (Gloeckle et al., 2024) equips a shared language-model trunk with independent output heads for predicting multiple subsequent tokens, while Medusa (Cai et al., 2024) and EAGLE (Li et al., 2024) accelerate LLM decoding by drafting future tokens with lightweight heads and verifying them against the backbone. In image generation, Mirai (Yu et al., 2026b) aligns internal representations with explicit or implicit foresight during autoregressive training while preserving the causal autoregressive inference path. The closest concurrent works in video generation are Video-Mirai and Next Forcing. Video-Mirai (Yu et al., 2026a) incorporates future information through alignment of JEPA-style representation (Bardes et al., 2024), while Next Forcing (Xu et al., 2026) uses recurrent future chunk prediction to improve performance in robotic tasks. In contrast, Prospective Forcing learns a shared representation for parallel multi-chunk prediction through independent heads and reuses these heads for verifier-guided speculative inference.

C Verifier Details

Verifier prediction. The verifier V_ϕ is a lightweight MLP with trainable parameters ϕ. It takes the current features $\mathbf{h}_W^r$ and the candidate features from equation 9 as inputs and outputs the selection and rejection probabilities. We use $\mathbf{h}_W^r$ and the candidates $\hat{\mathbf{h}}_{W+1}^{r,m}$ to decide which candidate to be selected:

$$\left(\pi_1^r, \ldots, \pi_M^r, \pi_{\text{rej}}^r\right) = \text{Softmax}\left(V_\phi\left(\text{sg}\left[\mathbf{h}_W^r\right], \left\{\text{sg}\left[\hat{\mathbf{h}}_{W+1}^{r,m}\right]\right\}_{m=1}^{M}\right)\right). \tag{28}$$

The stop-gradient operations prevent the verifier loss equation 11 from updating the generator.

Label construction. After the window advances to $\mathcal{W}_{r+1}$, we compare each candidate with the backbone features $\mathbf{h}_W^{r+1}$ and define the normalized feature error of candidate m as

$$E_m^r = \text{MSE}\left(\text{LN}\left(\hat{\mathbf{h}}_{W+1}^{r,m}\right), \text{LN}\left(\mathbf{h}_W^{r+1}\right)\right). \tag{29}$$

Let δ denote the maximum acceptable error. We set y_r to the candidate $m_r^\star = \arg\min_m E_m^r$ used in equation 10 if its error does not exceed δ, and to the rejection label rej otherwise:

$$y_r = \begin{cases} m_r^\star, & E_{m_r^\star}^r \le \delta, \\ \text{rej}, & \text{otherwise.} \end{cases} \tag{30}$$

D Additional Experimental Details and Results

D.1 Implementation Details

All models are trained on 8 NVIDIA A100 GPUs with a global batch size of eight and fully sharded data parallelism. We optimize the generator and the trainable score model with AdamW (Loshchilov

& Hutter, 2019) using $(\beta_1, \beta_2) = (0, 0.999)$ and learning rates of 1.5×10^{-6} and 4×10^{-7}, respectively, and update the score model five times per generator update. The generator uses an exponential moving average with decay 0.99 starting from iteration 200, and all reported results use the EMA weights. The VAE and text encoder remain frozen throughout training. Following Liu et al. (2026), we use the denoising timesteps $\{1000, 800, 600, 400, 200\}$ with a timestep shift of 5, and the teacher uses a classifier-free guidance scale of 3. We train on the VidProM (Wang & Yang, 2024) prompt set released by Huang et al. (2025a). Each chunk contains three latent frames, and each training rollout contains 21, 24, or 27 latent frames, i.e., 7 to 9 chunks.

In the first stage, we train for 3,000 iterations with the DMD objective and the feature and velocity alignment losses, without candidate embeddings or the verifier. In the second stage, we initialize from the stage-1 checkpoint, enable the candidate embeddings and the verifier, and continue training all parameters for 1,000 iterations. Each future prediction head is a lightweight transformer block with 12 attention heads and an MLP ratio of 2. It adds $\mathbf{h}_W^r$, $\mathbf{n}_{r,k}$ and a learned offset embedding, applies self-attention over the three latent frames of the chunk independently at each spatial location, and updates $\mathbf{h}_W^r$ through a noise-gated residual attention branch followed by an MLP. The heads do not attend to other window positions or to the text condition. The candidate embeddings are initialized from $\mathcal{N}(0, 0.02^2)$, with the first embedding set to zero so that the first candidate coincides with the original prediction. The verifier is a three-layer MLP with a hidden size of 96. The loss weights are $\lambda_{\text{feat}} = 0.1$, $\lambda_{\text{vel}} = 0.01$ and $\lambda_{\text{aux}} = 0.1$. For the candidate alignment loss we use $M = 2$ candidates with $\beta = 0.05$, and the verifier acceptance threshold is $\delta = 0.12$.

Since our training setup follows Rolling Forcing, it is equivalent to our stage-1 model with ODE initialization and $\lambda_{\text{feat}} = \lambda_{\text{vel}} = 0$, and serves as the DMD-only baseline in Table 4.

D.2 Evaluation Metrics

All metrics are computed on the same 200 videos per method.

VBench. The Quality score follows the official aggregation (Huang et al., 2024): each of the seven quality dimensions is normalized with the official minimum and maximum values, Dynamic Degree is weighted by 0.5 and the other six dimensions by 1, and the weighted sum is divided by 6.5 and multiplied by 100. None of the 200 prompts is applicable to the Color dimension, so we exclude Color and report the Semantic score as the average of the remaining eight normalized semantic dimensions. The Total score is $0.8 \times$ Quality $+ 0.2 \times$ Semantic.

VBench-Long. We evaluate six quality dimensions of VBench-Long (Huang et al., 2025b) in the `long_custom_input` mode: subject consistency, background consistency, motion smoothness, dynamic degree, aesthetic quality, and imaging quality. We aggregate them in the same way as the VBench Quality score, dividing the weighted sum by 5.5.

VideoScore. We average the five dimensions of VideoScore v1.1 (He et al., 2024) for each video and then average over videos.

MUSIQ and quality drift. Following Liu et al. (2026), we evaluate MUSIQ (Ke et al., 2021) on all frames of the first and last 5 seconds of each video. Let H_i and T_i be the mean MUSIQ scores of the first and last 5 seconds of video i. The MUSIQ avg. score averages H_i and T_i over all videos, and the quality drift is $\frac{1}{N}\sum_{i=1}^{N} |T_i - H_i|$.

D.3 Efficiency Measurement

We measure all methods on the same NVIDIA A100 80GB GPU with batch size one and bfloat16 precision. Table 5 breaks down the per-video inference cost. Generating a 30-second video involves two kinds of backbone passes: sampling passes that denoise the rolling window, and cache-update passes that write the key–value states of each newly committed clean chunk into the cache. An accepted prediction advances the window by one extra chunk and skips one sampling pass. However, every committed chunk still requires its own cache-update pass, and the first windows cannot be skipped because the rolling window must first be filled. As a result, even when every prediction is accepted ($\ell = 1$, unverified), the number of sampling passes only drops from 46 to 27, while

the 46 cache-update passes, which account for about 20% of the standard inference time, remain unchanged. This bounds the speedup of $\ell = 1$ at $1.56\times$ rather than $2\times$. Our verifier accepts 32.1% of the proposals (9 out of 28 rounds per video), which skips 9 sampling passes and yields a $1.20\times$ speedup. The overhead of the prediction heads and the verifier is negligible, below 0.2 seconds per video.

Table 5: Per-video inference cost breakdown on a single A100 GPU (30-second video, 480 frames).

Inference	Accept rate	Sampling passes	Cache-update passes	Speedup
Standard ($\ell = 0$)	–	46	46	$1.00\times$
Unverified, $\ell = 1$	100%	27	46	$1.56\times$
Verified (Ours)	32.1%	37	46	$1.20\times$

D.4 User Study

We use randomly selected prompts from the evaluation set and generate 30-second videos at 832×480 and 16 fps, compressed with H.264 at CRF 30 for web playback. Each participant completes five comparisons. In each comparison, the baseline is sampled uniformly from Rolling Forcing, LongLive, and Video-Mirai, our video is sampled uniformly from Ours and the stage-1 model with standard inference ($\ell = 0$), and the left–right order is randomized. Participants choose the better video separately for visual quality, long-term consistency, and dynamics. We only count submitted responses that pass automatic and manual quality checks, which yields 59 valid participants and 295 comparisons, of which 130 involve Ours and 165 involve the stage-1 model. All preference rates of Ours in Figure 4 are significantly above 50% under a two-sided binomial test ($p < 0.001$).

Table 6 reports the results of the stage-1 model. Even without accelerated inference, participants prefer it over all three baselines on every criterion. The preferences are significant ($p < 0.05$) in all cases except for visual quality ($p = 0.24$) and consistency ($p = 0.56$) against LongLive.

Table 6: User study of the stage-1 model with standard inference ($\ell = 0$). Each entry is the percentage of pairwise comparisons in which participants preferred this model over the baseline. Average is computed over all comparisons of the three criteria.

Stage 1 vs.	Visual Quality	Consistency	Dynamics	Avg.
Rolling Forcing	81.8%	86.4%	87.9%	85.4%
LongLive	59.6%	55.3%	68.1%	61.0%
Video-Mirai	78.8%	73.1%	86.5%	79.5%

D.5 Per-Dimension Results

Tables 7–11 report the individual dimensions behind the aggregate scores in Table 1. **Bold** and underline denote the best and second-best results. Please note that some individual scores can not reflect the true visual quality. For example, in Table 9, the background consistency score is high, but the sampled videos have a quality shift and tend to stay static.

Table 7: Per-dimension VBench quality scores (%).

Method	Subject↑	Background↑	Flickering↑	Smoothness↑	Dynamic↑	Aesthetic↑	Imaging↑
CausVid	87.59	90.60	<u>98.13</u>	**99.06**	21.00	57.35	67.47
Causal Forcing++	85.33	90.48	95.93	97.71	72.00	60.33	66.13
Self Forcing	<u>89.59</u>	<u>92.27</u>	**98.17**	<u>98.77</u>	24.50	61.07	**69.57**
LongLive	**89.70**	**92.97**	96.20	97.89	62.00	**62.95**	<u>69.34</u>
Rolling Forcing	84.36	88.72	94.34	97.86	<u>98.50</u>	58.19	68.24
Video-Mirai	82.85	88.02	95.99	96.92	47.00	57.30	64.44
Ours	87.95	91.91	95.69	97.87	**100.00**	<u>61.92</u>	68.86

Table 8: Per-dimension VBench semantic scores (%). The first row reports the number of prompts to which each dimension applies.

Method	Object↑	Multi-Obj.↑	Action↑	Spatial↑	Scene↑	App. Style↑	Temp. Style↑	Overall Cons.↑
Valid videos	119	20	36	5	54	8	36	200
CausVid	92.12	77.50	**94.44**	77.28	35.07	21.29	22.19	23.51
Causal Forcing++	95.06	76.88	**94.44**	87.15	42.82	21.35	22.72	24.46
Self Forcing	95.48	65.31	91.67	**90.20**	48.73	21.17	24.17	24.85
LongLive	95.43	**80.94**	**94.44**	80.74	49.54	21.17	24.95	25.97
Rolling Forcing	87.61	53.44	91.67	78.49	**51.62**	21.34	24.06	25.05
Video-Mirai	90.81	65.62	91.67	81.84	45.49	**21.71**	23.53	25.00
Ours	**97.43**	75.94	91.67	74.61	46.76	20.13	**25.56**	**26.23**

Table 9: Per-dimension VBench-Long scores (%).

Method	Subject↑	Background↑	Smoothness↑	Dynamic↑	Aesthetic↑	Imaging↑
CausVid	98.15	**96.99**	**99.08**	21.80	56.30	67.18
Causal Forcing++	96.09	95.76	97.74	60.77	58.86	65.39
Self Forcing	**98.19**	96.90	98.82	21.77	59.73	**69.20**
LongLive	97.20	96.26	97.96	55.50	**61.43**	68.66
Rolling Forcing	96.16	95.59	97.92	88.20	57.05	67.73
Video-Mirai	96.41	96.21	96.93	36.43	56.54	63.84
Ours	96.64	95.51	97.94	**96.47**	60.32	68.07

Table 10: Per-dimension VideoScore v1.1 scores.

Method	Visual↑	Temporal↑	Dynamic↑	T2V Align.↑	Factual↑
CausVid	2.683	2.450	2.985	2.738	2.330
Causal Forcing++	2.698	2.472	3.044	2.803	2.332
Self Forcing	2.908	2.731	3.242	3.046	2.597
LongLive	**3.157**	**2.985**	3.345	3.206	**2.847**
Rolling Forcing	2.843	2.608	3.095	2.876	2.452
Video-Mirai	2.679	2.461	3.187	2.752	2.441
Ours	3.117	2.890	**3.379**	**3.217**	2.742

Table 11: MUSIQ scores of the first and last 5 seconds and the drift measure.

Method	Head↑	Tail↑	Avg.↑	$\frac{1}{N}\sum_i \lvert T_i - H_i \rvert \downarrow$
CausVid	70.25	62.73	66.49	9.02
Causal Forcing++	68.00	65.31	66.65	5.52
Self Forcing	**71.06**	66.61	68.84	6.18
LongLive	69.37	**68.87**	69.12	**3.43**
Rolling Forcing	69.48	68.35	68.92	5.32
Video-Mirai	69.49	59.58	64.54	11.90
Ours	70.23	68.04	**69.14**	3.64