

Error Analyses of Auto-Regressive Video Diffusion Models

Jing Wang^{1,2,*}

JING005@E.NTU.EDU.SG

¹Nanyang Technological University, ²A*STAR

Fengzhuo Zhang^{3,*†}

FZZHANG@U.NUS.EDU

³National University of Singapore

Xiaoli Li⁶

XIAOLI.LI@SUTD.EDU.SG

⁶Singapore University of Technology and Design

Vincent Y. F. Tan³

VTAN@NUS.EDU.SG

³National University of Singapore

Tianyu Pang⁴

TIANYUPANG@SEA.COM

⁴Sea AI Lab

Chao Du^{4,‡}

DUCHAO@SEA.COM

⁴Sea AI Lab

Aixin Sun¹

AXSUN@NTU.EDU.SG

¹Nanyang Technological University

Zhuoran Yang⁵

ZHUORAN.YANG@YALE.EDU

⁵Yale University

Editor: Jianfeng Lu

Abstract

Auto-Regressive Video Diffusion Models (AR-VDMs) have shown strong capabilities in generating long, photorealistic videos, but suffer from two key limitations: (i) *history forgetting*, where the model loses track of previously generated content, and (ii) *temporal degradation*, where frame quality deteriorates over time. Yet a rigorous theoretical analysis of these phenomena is lacking, and existing empirical understanding remains insufficiently grounded. In this paper, we introduce **Meta-ARVDM**, a unified analytical framework that studies both errors through the shared autoregressive structure of AR-VDMs. We show that history forgetting is characterized by the conditional mutual information between the generated output and preceding frames, conditioned on inputs, and prove that incorporating more past frames *monotonically alleviates* history forgetting, thereby theoretically justifying a common belief in existing works. Moreover, our theory reveals that standard metrics fail to capture this effect, motivating a new evaluation protocol based on a “needle-in-a-haystack” task in closed-ended environments (DMLab and Minecraft). We further show that temporal degradation can be quantified by the cumulative sum of per-step errors, enabling prediction of degradation for different schedulers without video rollout. Finally, our evaluation uncovers a strong empirical correlation between history forgetting and temporal degradation, a connection not previously reported.

1. * Equal contribution

2. † Project Lead

3. ‡ Corresponding author

4. Work done by Jing Wang and Fengzhuo Zhang as associate members at Sea AI Lab

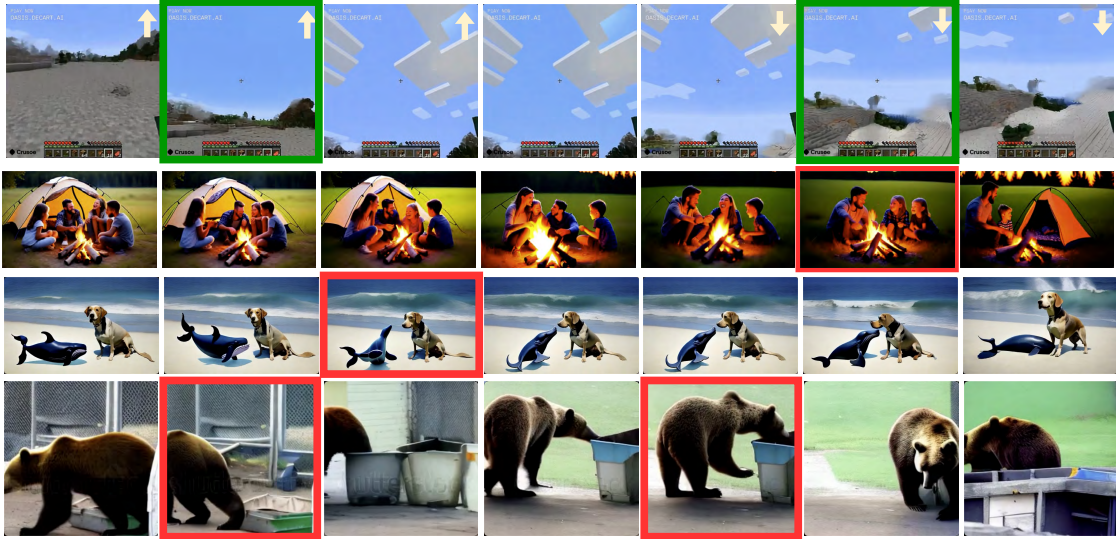


Figure 1: Example frames sampled from videos that are longer than 100 frames. We use color of bounding boxes to differentiate **History Forgetting** and **Temporal Degradation**.

Keywords: autoregressive video diffusion, history forgetting, temporal degradation, memory compression, long-horizon video generation

1. Introduction

The Auto-Regressive Video Diffusion Model (AR-VDM) has emerged as a particularly effective approach for generating long videos (Kim et al., 2024; Henschel et al., 2024). By modeling video data in an Auto-Regressive (AR) manner, these models capture complex motion patterns and rich textures that closely reflect real-world dynamics. Their versatility enables a wide range of applications, including video game synthesis (Che et al., 2024), world modeling (Chen et al., 2024a), and robotic perception and planning (Zhang et al., 2024).

However, all AR-VDMs exhibit two limitations not typically seen in Non-Auto-regressive VDMs (NAT-VDMs): (i) *history forgetting*, where the model loses track of previously generated content (e.g., subjects or backgrounds), as illustrated in the first row of Fig. 1; and (ii) *temporal degradation*, where video quality progressively deteriorates along the temporal axis. This degradation is reflected in the last three rows of Fig. 1. These issues significantly limit the utility of AR-VDMs for world modeling and realistic video generation. Yet their origins, mathematical formulations, and evaluations remain underexplored, hindering the development of effective mitigation strategies. While early studies decompose NAT-VDM errors into noise initialization, score estimation, and discretization errors (Chen et al., 2023a), it remains unclear whether history forgetting and temporal degradation fall within this framework or necessitate novel characterizations of various error terms.

In this paper, we present a unified framework, **Meta-ARVDM**, specifically designed for AR-VDMs. This framework facilitates a principled analysis, demonstrating that phenomena such as history forgetting and temporal degradation are intrinsic to the auto-regressive

characteristics of these models, rather than mere byproducts of particular implementation choices. **Meta-ARVDM** encompasses all AR-VDMs that meet our proposed auto-regressive requirement (Requirement 1), which is largely satisfied by most existing models.

Through a comprehensive mathematical error analysis of **Meta-ARVDM**, we show that history forgetting is captured by the conditional mutual information between the output and the full history given the input—a novel term absent from prior analyses. We prove that the *emergence* of this term is inevitable in AR-VDMs, however, its magnitude can be reduced in a monotonic fashion by injecting more historical context into the denoising process. This formalization implies that standard distributional metrics (e.g., Fréchet Video Distance (FVD), Fréchet Image Distance (FID)) and motion-based metrics (e.g., smoothness, motion magnitude) fail to characterize history forgetting. To remedy this problem, we propose a visual “needle-in-a-haystack” task across varying history lengths that both reveals and quantifies this phenomenon, verifying that it can mitigate the history forgetting problem in a monotonic fashion. Following this new task setting, we empirically show that some existing methods with good performances on traditional measures fail to reduce history forgetting.

Finally, we demonstrate that temporal degradation arises from the cumulative impact of noise initialization, score estimation, and discretization errors. Our theoretical analysis is robustly validated by experimental results, which, for example, accurately forecast the performance of various schedulers. The relationship between temporal degradation and these three error components suggests that metrics sensitive to them can effectively quantify temporal degradation. Furthermore, our evaluations reveal a previously unreported correlation between history forgetting and temporal degradation.

In summary, our contributions are two-fold.

- We derive the error analysis of AR-VDMs within a general framework, **Meta-ARVDM**. Our analysis quantitatively characterizes history forgetting as a novel conditional mutual information term, and temporal degradation as the cumulative accumulation of errors propagated from preceding frames. These results provide formal theoretical justifications for the common beliefs in empirical studies. Specifically, for history forgetting, we prove that this error is statistically inevitable but can be monotonically mitigated by incorporating past frames. For temporal degradation, our analysis explains why later-generated video segments exhibit lower quality than earlier ones, even when they share the same length.
- Our theoretical results yield new empirical insights into history forgetting and temporal degradation. For history forgetting, our results imply that commonly used distributional metrics (e.g., FVD, FID) and motion-based metrics (e.g., smoothness) fail to capture this error. In contrast, our proposed visual “needle-in-a-haystack” task effectively quantifies the extent of history forgetting. For temporal degradation, we show that standard distributional metrics can evaluate it reliably, and our theoretical quantification accurately predicts the performance of different step schedulers without generating videos. Leveraging these improved evaluations, we further uncover a correlation between history forgetting and temporal degradation—a relationship not previously reported in the literature.

2. Related Work

Video Diffusion Models Given the remarkable successes of diffusion models in image generation, several early works in Text-To-Video (T2V) have proposed treating the entire video as a vector to learn joint score functions (Ho et al., 2022b; Blattmann et al., 2023a; Ma et al., 2024a; Chen et al., 2024b; Guo et al., 2023; Singer et al., 2022; Yuan et al., 2024). To effectively capture the correlations between frames, these studies have developed a temporal attention module that applies attention mechanisms across the temporal dimension. The overall architecture integrates both temporal and spatial attention layers (Wang et al., 2023; Blattmann et al., 2023b; Wang et al., 2024c; Xing et al., 2025). Furthermore, to leverage the fine-grained spatial-temporal relationships between pixels, some researchers have adopted 3D attention modules as foundational components of denoising networks (Lin et al., 2024; Zheng et al., 2024; Yang et al., 2024). Beyond the use of 3D attention, additional works have investigated the combination of Mamba and attention mechanisms (Gao et al., 2024; Mo and Tian, 2024). In addition to these T2V approaches, video generation can also be conditioned on various factors, such as images (Zhang et al., 2023b; Chen et al., 2023d; Ren et al., 2024), poses (Karras et al., 2023; Ma et al., 2024b), motion (Chen et al., 2023c; Wang et al., 2024e), and sound (Liu et al., 2023).

Auto-Regressive Video Models A body of research has emerged that utilizes the AR framework to generate long videos. These studies employ two main approaches: adapting diffusion models to the AR framework and tokenizing frames for next-token prediction. We will first discuss examples of the former approach. FIFO-Diffusion (Kim et al., 2024) is a training-free method that leverages pretrained models to denoise frames at various noise levels. To address the training-inference gap, this method uses latent partitioning to reduce discrepancies between noise levels. Other notable works, including AR-Diffusion (Wu et al., 2023), Rolling Diffusion Models (Ruhe et al., 2024), Diffusion-forcing (Chen et al., 2024a), and Pyramidal Flow Matching (Jin et al., 2024), train networks specifically to denoise frames at different noise levels. In contrast, ART-V (Weng et al., 2024), StreamingT2V (Henschel et al., 2024), GameNGen (Valevski et al., 2024), Self-forcing (Huang et al., 2025), and GameGen-X (Che et al., 2024) focus on denoising frames that share the same noise level. All the above methods rely on denoised or noisy frames together with the KV cache as context. Another line of work focuses on selective compression of memory to improve generation quality, including Mixture of Contexts (Cai et al., 2025) and FramePack (Zhang et al., 2025).

The second approach involves tokenizing frames and training networks for next-token prediction. Examples include LlamaGen (Sun et al., 2024), Emu3 (Wang et al., 2024b), and Loong (Wang et al., 2024d), which generate visual tokens based on their spatial-temporal ranking. Additionally, WorldMem (Xiao et al., 2025) seeks to implement a memory mechanism by incorporating historical frames via the attention mechanism, demonstrating effectiveness in video lengths of around 10 seconds within Minecraft scenarios. Context-as-Memory (Yu et al., 2025) further enhances this by using historical context frames as memory, employing simple concatenation and a memory retrieval module to select relevant frames based on camera field of view (FOV) overlap. Lastly, MemoryPack (Wu et al., 2025) builds on this memory mechanism, extending it to a hierarchical structure to ensure both short-term consistency and long-term retention. Our work specifically analyzes the methods belonging to the first approach.

Conditional Generation of Diffusion Models AR video generation is also closely related to conditional generation in diffusion models, since later frames are generated conditional on earlier ones. Prior work has used guidance techniques to improve temporal consistency in generated videos (Guo et al., 2024; Liang et al., 2024; Song et al., 2025). The behavior of guidance in conditional diffusion has been studied from many angles. Chidambaram et al. (2024) analyze how the guidance strength affects the sample support throughout the sampling process in a simplified yet representative setting. In addition, Zhao and Schwing (2026); Wu et al. (2024); Ye et al. (2025) study diffusion models under simplified learning settings, though their focus is on broader qualitative properties. At a finer level, Gao et al. (2026); Lehman Pavasovic et al. (2025); Li et al. (2025); Jin et al. (2025); Rojas et al. (2026); Wang et al. (2024a); Kynkäänniemi et al. (2024) examine how the effect of guidance varies across sampling stages, develop non-stationary views of guidance, and propose adaptive algorithms accordingly. By contrast, Fu et al. (2024) studies the statistical difficulty of learning guidance in diffusion models. Under suitable assumptions on estimation error, Bradley and Nakkiran (2024); Xia et al. (2025); Li and Jiao (2025); Jiao et al. (2025) analyze guided sampling itself, interpreting guidance as part of the sampling algorithm and proving convergence guarantees. Most of these works treat guidance as the primary focus; our work is complementary and distinct, concentrating instead on the effects of autoregressive generation in conditional diffusion models.

3. Preliminaries: Video Diffusion Models

A Video Diffusion Model (VDM) generates video frames by reversing a forward diffusion process (Ho et al., 2022b,a). We represent a video $X_{1:w}$ as a sequence of frames along the temporal axis $(X_1, \dots, X_w) \in \mathbb{R}^{w \times d}$, where w is the number of frames and d is the dimensionality of each frame (e.g., $d = 512 \times 512$ for 512×512 -pixel frames). Starting from the clean video $X_{1:w}^0 = X_{1:w}$, the diffusion process gradually adds Gaussian noise, as described by the following process.

$$dX_{1:w}^t = f(X_{1:w}^t, t) dt + g(t) dB^t \text{ for } t \in [0, T], \quad (1)$$

where t denotes the time index of the diffusion process, $X_{1:w}^t$ denotes the noisy version of video frames at diffusion time t , and the functions $f : \mathbb{R}^{w \times d} \times \mathbb{R} \rightarrow \mathbb{R}^{w \times d}$ and $g : \mathbb{R} \rightarrow \mathbb{R}$ govern the drift and diffusion coefficients, and B^t is $(w \times d)$ -dimensional Brownian motion with identity covariance. Throughout, we use subscripts for frame indices and superscripts for diffusion time steps. To avoid confusion, we refer to t as the diffusion time (i.e., noise level) and w as the frame index.

As discussed in Song et al. (2020b), two common instantiations of Eqn. (1) are the variance-exploding SDE (SMLD) (Song and Ermon, 2019) with $f = 0$, $g(t) = \beta(t)$, and the variance-preserving SDE (DDPM) (Ho et al., 2020) with $f(X, t) = -0.5\beta(t)X$, $g(t) = \sqrt{\beta(t)}$, where $\beta(t)$ is a scalar noise schedule. In both cases, $X_{1:w}^T$ converges in distribution to a Gaussian distribution as $T \rightarrow \infty$. To generate samples, we reverse the diffusion from $t = T$ to $t = 0$, which yields the reverse-time SDE (Anderson, 1982):

$$dX_{1:w}^t = (f(X_{1:w}^t, t) - g(t)^2 \nabla \log P_t(X_{1:w}^t)) dt + g(t) d\tilde{B}^t \text{ for } t \in [0, T], \quad (2)$$

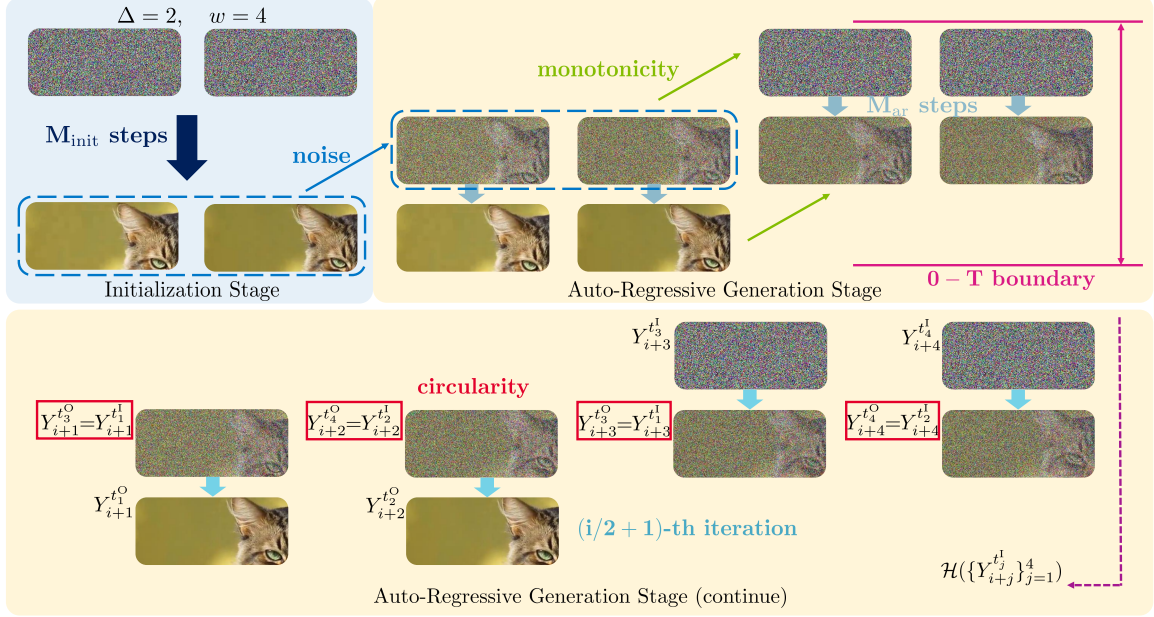


Figure 2: The Meta-ARVDM framework begins with an initialization stage (depicted on the left), which denoises the input using M_{init} steps. Noise is then added to the output of this stage to produce the starting point for the AR generation stage (shown on the right). The figure highlights key requirements for plausible implementation: **monotonicity**, **circularity**, and the **0-T boundary** condition. Here, $\mathcal{H}(\cdot)$ denotes all frames available prior to the execution of the $(i/2 + 1)$ -th iteration.

where \tilde{B}^t is *reverse-time* Brownian motion, P_t is the distribution of $X_{1:w}^t$, $\nabla \log P_t(X_{1:w}^t)$ is known as the *score function*, which is typically learned using neural networks (Ho et al., 2022b). By applying the time reversal $t \mapsto T - t$, we obtain the equivalent forward-time representation of the reverse process as

$$d\tilde{X}_{1:w}^t = (-f(\tilde{X}_{1:w}^t, T-t) + g(T-t)^2 \nabla \log P_{T-t}(\tilde{X}_{1:w}^t)) dt + g(T-t) dB^t. \quad (3)$$

We will write \tilde{X}^t (resp. \tilde{B}^t) and X^t (resp. B^t) to denote the reverse-time and original stochastic processes (Brownian motion), respectively. We will focus on this form of the reverse-time process for the theoretical analysis of our work.

Notation: For a comprehensive list of all mathematical symbols used in this paper, please refer to Appendix A.

4. Meta-ARVDM: A Unified Framework of AR-VDMs

AR-VDMs generate video frames sequentially according to their temporal positions. In contrast to the approach suggested in Eqn. (3), which denoises all frames simultaneously at the same noise level t , AR-VDMs enable the denoising of frames at varying noise levels. Importantly, the generation of each frame depends on the frames that have been generated previously. We will present AR-VDMs in a cohesive framework, utilizing outpainting and FIFO-Diffusion (Kim et al., 2024) as illustrative examples.

Algorithm 1 Meta-ARVDM

Input: VDM s_θ , AR step-size $\Delta \in \mathbb{N}$, effective window size $w \in \mathbb{N}$, length of the video $N \in \mathbb{N}$, input noise levels $\mathcal{L}^I = (t_1^I, \dots, t_w^I)$, output noise levels $\mathcal{L}^O = (t_1^O, \dots, t_w^O)$, reference frames sets $\{\mathcal{R}_i\}_{i \in [N]}$

Procedure:

- 1: $(Y_1^{t_1^I}, \dots, Y_{w-\Delta}^{t_{w-\Delta}^I}) = \text{Initialization}(s_\theta, \mathcal{L}^I)$. // Initialization Stage
 - 2: **for** $k = 1, \dots, \lceil N/\Delta \rceil$ **do**
 - 3: $Y_{(k-1)\Delta+1:k\Delta}^0, (Y_{k\Delta+1}^{t_1^I}, \dots, Y_{k\Delta+w-\Delta}^{t_{w-\Delta}^I})$ // Auto-Regressive Generation Stage
 $= \text{Auto-Regressive Step}((Y_{(k-1)\Delta+j}^{t_j^I})_{j=1}^{w-\Delta}, s_\theta, w, \Delta, \mathcal{L}^I, \mathcal{L}^O, Y_{\mathcal{R}_{(k-1)\Delta+1}}^0)$
 - 4: **end for**
 - 5: Output $Y_{1:N}^0$.
-

Our framework operates in two stages. The first stage, **Initialization** (Algorithm 2, detailed in Section 4.1), utilizes a VDM to generate i_0 frames, with all frames being denoised at the same noise level. The second stage, **Auto-Regressive Step** (Algorithm 3, detailed in Section 4.2), denoises w frames at varying noise levels. During each AR step, the algorithm takes as input reference frames in \mathcal{R} , $w - \Delta$ past noisy frames, and Δ Gaussian noise instances. It outputs Δ clean frames together with $w - \Delta$ noisy frames for subsequent iterations. The input and output noise levels are denoted as $\mathcal{L}^I = (t_1^I, \dots, t_w^I)$ for input levels and $\mathcal{L}^O = (t_1^O, \dots, t_w^O)$ for output levels. To initialize the AR step, we introduce different noise levels to the i_0 frames generated in the initial stage. Combining these two stages results in our framework, **Meta-ARVDM**, shown in Algorithm 1.

We now formally define this framework to facilitate further analysis. While the following notations may appear complex, they are essential for ensuring mathematical rigor.¹ In the following, we will explain the two stages via two examples: outpainting and FIFO-Diffusion. Other AR-VDMs covered by the framework are explained in Appendix F.

4.1 Initialization Stage

The initialization stage adopts a VDM to generate i_0 frames via the Euler-Maruyama scheme.

$$d\tilde{X}_{1:w}^t = (-f(X_{1:w}^{\tilde{t}_n^{\text{init}}}, T - \tilde{t}_n^{\text{init}}) + g(T - \tilde{t}_n^{\text{init}})^2 s_\theta(\tilde{X}_{1:w}^{\tilde{t}_n^{\text{init}}}, T - \tilde{t}_n^{\text{init}}))dt + g(T - \tilde{t}_n^{\text{init}})dB^t \quad (4)$$

for $t \in [\tilde{t}_n^{\text{init}}, \tilde{t}_{n+1}^{\text{init}}]$ with $n = 0, \dots, M_{\text{init}} - 1$, where the partition $0 = t_0^{\text{init}} \leq \dots \leq t_{M_{\text{init}}}^{\text{init}} = T$ divides $[0, T]$ into M_{init} intervals, and $\tilde{t}_n^{\text{init}} = T - t_{M_{\text{init}}-n}^{\text{init}}$ denotes the reversed noise levels. The score function $\nabla \log P$ is estimated by a neural network s_θ with parameters θ . This process is illustrated in Figure 2 (left). In outpainting, this initialization corresponds to generating the first block of videos. In FIFO-Diffusion, this corresponds to generating the initial w frames (comprising several blocks) and injecting Gaussian noise at varying levels.

1. Readers not interested in the technical details may skip this section and proceed directly to Requirement 1, which specifies the conditions of AR-VDM satisfying the proposed framework.

Algorithm 2 Initialization

Input: VDM s_θ , size of the effective window $w \in \mathbb{N}$, input noise levels \mathcal{L}^I

Procedure:

- 1: Generate $i_0 \geq w$ independent Gaussian vectors $Y_{1:i_0}^T$.
 - 2: Denoise $Y_{1:i_0}^T$ to $Y_{1:i_0}^0$ with Eqn. (4).
 - 3: Add noises to $Y_{1:i_0}^0$ to derive $\{Y_i^{t_i}\}_{i=1}^{w-\Delta}$.
 - 4: Return $\{Y_i^{t_i}\}_{i=1}^{w-\Delta}$.
-

4.2 Auto-Regressive Generation Stage

The *auto-regressive generation stage* in methods such as FIFO-Diffusion and StreamingT2V simultaneously denoises frames at various noise levels, often using additional reference frames. To address this, we generalize the ground-truth reverse equation (Eqn. (2)) along two dimensions. First, to accommodate multiple noise levels, we introduce the extended time vector $\bar{t}(t) = (t + \delta_1, \dots, t + \delta_w)$, where each $\delta_i \in \mathbb{R}$ represents the offset of the i -th frame’s noise level relative to the shared evolution variable t . For clarity, we will refer to $\bar{t}(t)$ simply as t when the context is unambiguous. Second, we condition the initial distribution of $X_{1:w}^0$ in Eqn. (1) on the reference clean frames $X_{\mathcal{R}}^0$, where $\mathcal{R} \subseteq \mathbb{N}$ denotes the set of indices of the reference frames. The extended reverse-time SDE can then be written as

$$d\tilde{X}_{1:w}^{\bar{t}} = (-f(\tilde{X}_{1:w}^{\bar{t}}, \bar{T} - \bar{t}) + g(\bar{T} - \bar{t})^2 \nabla \log P_X^{\bar{T}-\bar{t}}(\tilde{X}_{1:w}^{\bar{t}} | X_{\mathcal{R}}^0)) dt + g(\bar{T} - \bar{t}) dB_{1:w}^{\bar{t}} \quad (5)$$

for $t \in [T - t^I, T - t^O]$, where $\bar{T} = (T, \dots, T)$, and w is the AR *window size*. The frame set $X_{1:w}^{\bar{t}}$ represents frames at varied noise levels, i.e., $X_{1:w}^{\bar{t}} = (X_1^{t+\delta_1}, \dots, X_w^{t+\delta_w}) = \{X_i^{t+\delta_i}\}_{i=1}^w$ and $P_X^{\bar{t}}$ is the distribution of $X_{1:w}^{\bar{t}}$. The functions $f: \mathbb{R}^{w \times d} \times \mathbb{R}^w \rightarrow \mathbb{R}^{w \times d}$ and $g: \mathbb{R}^w \rightarrow \mathbb{R}$ determine the evolution of the noisy frames.

Assuming all frames evolve at the same rate for variable rates, we fix input and output noise levels respectively as $\mathcal{L}^I = (t_1^I, \dots, t_w^I)$ and $\mathcal{L}^O = (t_1^O, \dots, t_w^O)$, and define $\delta_i = t_i^I - t_1^I = t_i^O - t_1^O$ (see Appendix A for detailed notation). Then each AR step approximates Eqn. (5) via the following Euler–Maruyama scheme

$$d\tilde{Y}_{1:w}^{\bar{t}} = (-f(\tilde{Y}_{1:w}^{\bar{t}}, \bar{T} - \bar{t}_n^{\text{ar}}) + g(\bar{T} - \bar{t}_n^{\text{ar}})^2 s_\theta(\tilde{Y}_{1:w}^{\bar{t}}, \bar{T} - \bar{t}_n^{\text{ar}}, Y_{\mathcal{R}}^0)) dt + g(\bar{T} - \bar{t}_n^{\text{ar}}) dB_{1:w}^{\bar{t}} \quad (6)$$

for $t \in [\tilde{t}_n^{\text{ar}}, \tilde{t}_{n+1}^{\text{ar}}]$ with $n = 0, \dots, M_{\text{ar}} - 1$. Similar to Eqn. (4), $\tilde{t}_n^{\text{ar}} = T - t_{M_{\text{ar}}-n}^{\text{ar}}$ is the reverse noise level, and $\bar{t}_n^{\text{ar}} = \bar{t}(\tilde{t}_n^{\text{ar}})$ is the reverse noise levels of all frames induced by \tilde{t}_n^{ar} . The full denoising schedule is discretized over $[t_1^O, t_1^I]$ into M_{ar} intervals. For ease of notation, we denote the denoising neural network in both initialization and AR stages as s_θ . The score estimate s_θ takes the current noisy frames $\tilde{Y}_{1:w}^{\bar{t}(\tilde{t}_n^{\text{ar}})}$, current noise level $T - \tilde{t}_n^{\text{ar}}$, and the reference frames $Y_{\mathcal{R}}^0$ as inputs. This process is implemented in the **Auto-Regressive Step** module.

The AR generation stage consists of iterative calls to **Auto-Regressive Step** (Algorithm 3), as shown on the right side of Figure 2. For outpainting that utilizes 12 already generated frames to denoise 4 frames, the effective window size is 4. For FIFO-Diffusion, the latent partitioning enables the VDM network to denoise $n \cdot f$ frames concurrently, giving an effective window of $w = n \cdot f$. Each AR step jointly denoises frames from noise levels

Algorithm 3 Auto-Regressive Step

Input: VDM s_θ , size of the effective window $w \in \mathbb{N}$, AR step-size $\Delta \in \mathbb{N}$, input noise levels \mathcal{L}^I , output noise levels \mathcal{L}^O , input noisy video $\{Y_i^{t_i^I}\}_{i=1}^{w-\Delta}$, reference frames $Y_{\mathcal{R}}^0$

Procedure:

- 1: Generate Δ independent Gaussian vectors $\{Y_i^{t_i^I}\}_{i=w-\Delta+1}^w$.
 - 2: Concatenate $\{Y_i^{t_i^I}\}_{i=1}^{w-\Delta}$ with the generated noise to form $\{Y_i^{t_i^I}\}_{i=1}^w$.
 - 3: Denoise $\{Y_i^{t_i^I}\}_{i=1}^w$ to $\{Y_i^{t_i^O}\}_{i=1}^w$ with the reference frames $Y_{\mathcal{R}}^0$ as Eqn. (6).
 - 4: Return $\{Y_i^{t_i^O}\}_{i=1}^w = (\{Y_i^{t_i^O}\}_{i=1}^\Delta, \{Y_i^{t_i^O}\}_{i=\Delta+1}^w)$
-

$\mathcal{L}^I = (t_1^I, \dots, t_w^I)$ to $\mathcal{L}^O = (t_1^O, \dots, t_w^O)$. To ensure coherent AR generation, we impose specific requirements on \mathcal{L}^I , \mathcal{L}^O , and the reference set \mathcal{R} .

Requirement 1 (*AR Requirements*)

- (Monotonicity) The noise levels are monotone, i.e., $t_{i-1}^I \leq t_i^I$, $t_{i-1}^O \leq t_i^O$, and $t_i^O < t_i^I$ for $i \in [w]$.
- ($0 - T$ Boundary) The boundaries of \mathcal{L}^I and \mathcal{L}^O are T and 0 , respectively, i.e., $t_i^I = T$ for $i \geq w - \Delta + 1$, and $t_i^O = 0$ for $i \leq \Delta$.
- (Circularity) The output noise levels are the same as the input noise levels up to a position shift, i.e., $t_i^O = t_{\Delta+i}^O$ for all $i \in [w - \Delta]$.
- (Constant Pace) The difference between the input and output noise levels are constant, i.e., $t_i^O - t_i^I = t_j^O - t_j^I$ for all $i, j \in [w]$.
- (Causality) When denoising frames indexed from $i + 1$ to $i + w$, the reference frames set is a subset of $[i]$, i.e., $\mathcal{R}_i \subseteq [i]$ for all $i \in \mathbb{N}$.

Figure 2 illustrates these conditions for $w = 4, \Delta = 2$. *Monotonicity* ensures that earlier frames (with smaller index i) are assigned lower noise levels t_i^I and t_i^O during denoising. The $0 - T$ boundary requirement ensures that the input to **Auto-Regressive Step** contains Δ new frames, i.e., there are Δ Gaussian vectors, and that the output contains Δ clean frames. *Circularity* ensures that the output of one step can serve as input to the next, enabling seamless iteration. The *constant pace* constraint is introduced to simplify theoretical analysis and is relaxed at the end of this section. The *causality* condition is essential for practical generation, ensuring that current frames are not conditioned on future ones. Both outpainting and FIFO-Diffusion satisfy these conditions. For outpainting, we have $t_i^I = T$ and $t_i^O = 0$ for all $i \in [w]$, with $\Delta = w$, denoising w noise vectors into clean frames per step. For FIFO-Diffusion, we have $\Delta = 1$, and $(0, t_1^I, \dots, t_{w-1}^I) = (t_1^O, \dots, t_{w-1}^O, T)$ are set to the noise levels of a pre-chosen scheduler. With all conditions met, each iteration of **Auto-Regressive Step** takes Δ new noise vectors and the previous outputs as input, producing Δ clean frames via Eqn. (6).

We can also consider an extension of the constant pace and scalar function $g(\cdot)$ from Eqn. (5). Specifically, we can extend the constant pace setting to accommodate various paces by introducing a new definition, namely that of the extended time $\bar{t}(t) = (\alpha_1 t + \delta_1, \alpha_2 t + \delta_2, \dots, \alpha_w t + \delta_w)$. In this expression, the coefficients α_i for $i \in [w]$ represent the different paces applied to each frame. To extend the scale function $g(\bar{t})$ into matrix form $G(X_{1:2}, \bar{t})$,

we will utilize the same approach as presented in (Song et al., 2020b). In fact, we have,

$$\begin{aligned} dX_{1:w}^{\bar{t}} = & \left(f(X_{1:w}^{\bar{t}}, \bar{t}) - \nabla \cdot [G(X_{1:w}^{\bar{t}}, \bar{t})G(X_{1:w}^{\bar{t}}, \bar{t})^\top] \right. \\ & \left. - G(X_{1:w}^{\bar{t}}, \bar{t})G(X_{1:w}^{\bar{t}}, \bar{t})^\top \nabla \log P_X^{\bar{t}}(X_{1:w}^{\bar{t}} | X_{\mathcal{R}}^0) \right) dt + G(X_{1:w}^{\bar{t}}, \bar{t}) d\tilde{B}_{1:w}^{\bar{t}}. \end{aligned}$$

5. Error Analysis of AR-VDMs through Meta-ARVDM

All algorithms encompassed by Meta-ARVDM utilize a shared analytical framework to analyze their errors. Building on previous theoretical analyses of diffusion processes (Chen et al., 2022, 2023a,b), we adopt a simplified model for clarity, specifically choosing $f(\tilde{Y}_{1:w}^t, t) = -0.5 \cdot \tilde{Y}_{1:w}^t$ and $g(t) = 1$. This corresponds to a DDPM model characterized by a constant $\beta(t) = 1$, as discussed in Section 3. Our findings can be easily generalized to accommodate other configurations. For our analysis, we impose three mild assumptions: the boundedness of pixel values, Lipschitz continuity of the score function, and a bounded score estimate error. These assumptions are standard in diffusion theory.

Assumption 1 (Boundness of Pixel Values) *For the frames of size $l = \max\{\Delta, i_0\}$, the ℓ_2 -norm of $X_{i+1:i+l}^0$ is almost surely bounded by a constant $B > 0$, i.e., $\|X_{i+1:i+l}^0\|_2 \leq B$ a.s. for any $i \in \mathbb{N}$.*

In latent diffusion models (Chen et al., 2024b; Guo et al., 2023), frames are denoised in the latent space, where pixel values are quantized and therefore bounded. For diffusion models in the pixel space, the pixel values are bounded by 255. Thus, this assumption is satisfied by most VDMs.

Assumption 2 (Lipschitz continuity of Score Function) *For any $t \in [0, T]$, both scores $\nabla \log P_X^t$ and $\nabla \log P_X^{\bar{t}(t)}$ are L -Lipschitz with respect to the ℓ_2 -norm.*

This assumption controls the discretization error by ensuring that the score function does not change too rapidly. It is standard in numerical analysis and diffusion theory (Gautschi, 2011; Süli and Mayers, 2003; Chen et al., 2023a).

Assumption 3 (Score Estimate Error) *The average score estimation error evaluated at the discretization steps is upper bounded by $\epsilon_{est}^2 > 0$ for both the initialization and AR stages, i.e., the following hold for any $k \in \mathbb{N}$:*

$$\begin{aligned} T^{-1} \sum_{n=1}^{M_{init}} (t_n^{init} - t_{n-1}^{init}) \mathbb{E} \left[\left\| \nabla \log P_X^{t_n^{init}}(X_{1:i_0}^{t_n^{init}}) - s_\theta(X_{1:i_0}^{t_n^{init}}, t_n^{init}) \right\|^2 \right] &\leq \epsilon_{est}^2 \quad \text{and} \\ (t_1^I - t_1^O)^{-1} \sum_{n=1}^{M_{ar}} (t_n^{ar} - t_{n-1}^{ar}) \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}_n^{ar}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{ar}} | X_{\mathcal{R}_{k\Delta+1}}^0) \right. \right. \\ &\left. \left. - s_\theta(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{ar}}, \bar{t}_n^{ar}, X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] \leq \epsilon_{est}^2. \end{aligned}$$

These conditions quantify the discrepancies between the true and estimated score functions along the discretized diffusion trajectory, stemming from the finite training of the denoising

networks. Similar assumptions are widely used in theoretical analyses of diffusion models (Chen et al., 2022, 2024d). We note that the estimation error of the diffusion models is averaged along the whole trajectory of the diffusion process. To state our results concisely, we write $x \lesssim y$ to mean that $x \leq Cy$ for some absolute constant $C > 0$ and write $\text{KL}(X \parallel Y)$ for the KL divergence between the distributions of X and Y .

Theorem 1 *Under regularity assumptions (Assumptions 1, 2, and 3), the KL-divergence between the distributions of the video generated by **Meta-ARVDM** and the nominal video is*

$$\text{KL}(X_{1:K\Delta}^0 \parallel Y_{1:K\Delta}^0) = \text{IE} + \sum_{k=1}^K \text{ARE}_k. \quad (7)$$

Here the error of the initialization stage IE and the error of k -th AR step ARE_k are respectively bounded as follows.

$$\text{IE} \lesssim \text{NIE} + \text{SEE} + \text{DE} \quad \text{and} \quad \text{ARE}_k \lesssim \text{NIE}_{\text{AR}} + \text{SEE}_{\text{AR}} + \text{DE}_{\text{AR}} + \text{HF}_k \quad (8)$$

where the Noise Initialization Errors (NIEs) are $\text{NIE} := (di_0 + B^2) \exp(-T)$ and $\text{NIE}_{\text{AR}} := (\Delta d + B^2) \exp(-T)$, the Score Estimation Errors (SEEs) are $\text{SEE} := T \epsilon_{\text{est}}^2$ and $\text{SEE}_{\text{AR}} := (t_1^{\text{I}} - t_1^{\text{O}}) \epsilon_{\text{est}}^2$, the Discretization Errors (DEs) are $\text{DE} := wdL^2 \sum_{n=1}^{M_{\text{mit}}} (t_n^{\text{init}} - t_{n-1}^{\text{init}})^2$ and $\text{DE}_{\text{AR}} := wdL^2 \sum_{n=1}^{M_{\text{ar}}} (t_n^{\text{ar}} - t_{n-1}^{\text{ar}})^2$, and finally, the history forgetting error is

$$\text{HF}_k := I(\text{Output}_k; \text{Past}_k \mid \text{Input}_k). \quad (9)$$

Here, the Output_k , Past_k , and Input_k are defined as $\text{Output}_k = \{X_{k\Delta+j}^{t_j^{\text{O}}}\}_{j=1}^w$, $\text{Past}_k = \mathcal{H}(\{X_{k\Delta+j}^{t_j^{\text{O}}}\}_{j=1}^w) \setminus \{X_{\mathcal{R}_{k\Delta+1}}^0\}$, and $\text{Input}_k = \{X_{\mathcal{R}_{k\Delta+1}}^0\} \cup \{X_{k\Delta+j}^{t_j^{\text{I}}}\}_{j=1}^w$. The history $\mathcal{H}(\cdot)$ is formally defined in Appendix B, and an example is shown in Figure 2. If $\mathcal{R} = \emptyset$, the KL-divergence between the distributions of the K -th video clip generated by **Meta-ARVDM** and the nominal video is

$$\text{KL}(X_{K\Delta+1:(K+1)\Delta}^0 \parallel Y_{K\Delta+1:(K+1)\Delta}^0) \lesssim \text{IE} + K[\text{NIE}_{\text{AR}} + \text{SEE}_{\text{AR}} + \text{DE}_{\text{AR}}]. \quad (10)$$

The proof is provided in Appendix B. This theorem characterizes the KL-divergence of the errors for both the generated long videos (Eqn. (7)) and the short clips (Eqn. (10)). The error for long videos comprises contributions from both the initialization stage (Algorithm 2) and the AR step (Algorithm 3). Each stage introduces three types of error: noise initialization, score estimation, and discretization. These error sources also arise in image diffusion models (Chen et al., 2022, 2023b), and are further discussed in Appendix G. In the remainder of this section, we focus on how these errors specifically relate to AR-VDMs.

Effect of history forgetting. In Eqn. (7), the history forgetting term for the k -th AR step is defined by the conditional mutual information between its output and all previous frames, conditioned on the input and reference frames at that step. This concept is intuitive: the output can only draw upon information from the past through its input. This phenomenon leads to a specific type of inconsistency in the generated videos. For instance, as illustrated in the first row of Figure 1, when the camera pans up to the sky and then returns downward, the AR-VDM seems to *forget* earlier frames, resulting in a completely different scene. It is

important to note that the concept of *history forgetting* is closely linked to the Information Bottleneck principle (Tishby et al., 2000).

Despite the apparent visual evidence of history forgetting demonstrated in Figure 1, a quantitative evaluation of this phenomenon remains challenging. Intuitively, commonly used video distribution metrics from previous studies, such as FID and FVD (Harvey et al., 2022), are not well-suited for this purpose. These metrics assess distributional discrepancies (e.g., through the KL-divergence) and aggregate all error contributions noted on the right-hand side of Eqn. (7), which obscures the specific impact of history forgetting. Additionally, motion-related metrics, such as smoothness and temporal flickering, employed in auto-regressive video generation (Henschel et al., 2024), concentrate on local frame characteristics and fail to capture the correlation between potentially distant generated frames and past frames—the key aspect of history forgetting. In contrast, our experimental section directly measures the dependence between Output_k and Past_k as defined in the context of history forgetting, employing “needle-in-a-haystack” problems. This approach effectively isolates and reflects the impact of forgetting.

Effect of temporal degradation. Eqn. (10) defines the error associated with the $(K + 1)$ -st short video clip of length Δ generated by an AR-VDM. This error accumulates contributions from the initialization stage and all K preceding AR steps, including noise initialization, score estimation, and discretization errors. Unlike Eqn. (7), which assesses the consistency of the entire video, Eqn. (10) focuses specifically on the quality of a local segment, omitting a history forgetting term that accounts for inconsistencies across multiple AR steps. This is exemplified in the last three cases of Figure 1, where later generated frames display unrealistic configurations. To quantitatively assess temporal degradation, our findings indicate that standard video distribution metrics applied to short clips, specifically $Y_{K\Delta+1:(K+1)\Delta}^0$, effectively capture this effect.

We have shown that the intuitive concept of *history forgetting* is evident in the *upper bound* of the error. However, it remains unclear whether this phenomenon arises from our analysis or is *intrinsic* to AR-VDMs—that is, whether it is also reflected in a *lower bound*. To address this question, we establish a lower bound under minimal assumptions, demonstrating that history forgetting is not just an artifact of our analysis, but rather an inherent limitation of the model. We will now demonstrate the inevitability of history forgetting, even in the presence of unlimited computational resources.

This information-theoretic limitation arises because the score function s_θ only conditions on the current input, without full access to past frames. We reduce the problem to the setting in Figure 13 in Appendix H, where the goal is to learn the joint distribution $P_{X,Y,Z}$. Here, the random variables X , Y and Z represent **Past**, **Input**, and **Output**, respectively. However, AR-VDMs use only **Input** to predict **Output**—i.e., partial observations. We assume access only to samples from the marginals: $\{(X_i, Y_i)\}_{i=1}^N \sim P_{X,Y}$ and $\{(Y_i, Z_i)\}_{i=N+1}^{2N} \sim P_{Y,Z}$, reflecting that s_θ is trained on segments of long videos. The goal is to estimate P using the dataset $\mathcal{D}_N(P) = \{(X_i, Y_i)\}_{i=1}^N \cup \{(Y_i, Z_i)\}_{i=N+1}^{2N}$. The resulting estimate \hat{P} captures both training and inference. For simplicity, we assume $X, Y, Z \in \{0, 1\}$.

Theorem 2 *For $s \in [0, 1]$ and any random variables X, Y, Z (representing **Past**, **Input** and **Output**, respectively), we define the conditional mutual information-constrained distribution set as $\mathcal{S}(s) = \{P = P_{X,Y,Z} \in \mathcal{P}(\Omega^3) \mid I(X; Z|Y) \leq s\}$. Then for any $N \in \mathbb{N}$, $P \in \mathcal{S}(s)$, and*

any estimate \hat{P} of the distribution P derived from $\mathcal{D}_N(P)$, we have that

$$\inf_{\hat{P} \in \sigma(\mathcal{D}_N(P))} \sup_{P \in \mathcal{S}(s)} \Pr(\text{KL}(P \parallel \hat{P}) \geq s^2/2) \geq 0.5.$$

The proof is provided in Appendix C. This theorem implies that $\Pr(\text{KL}(P \parallel \hat{P}) \geq 0.5I^2(X; Z|Y)) \geq \Pr(\text{KL}(P \parallel \hat{P}) \geq 0.5s^2) \geq 0.5$. Thus, the conditional mutual information is an information-theoretic barrier of AR-VDM. We do not impose constraints on the availability of computational resources. This implies history forgetting cannot be avoided even with unlimited computational resources.

While the *presence* of history forgetting is inevitable, its *magnitude* can be reduced. Intuitively, the more information the input carries about the past, the smaller this term becomes. This intuition is formalized in the following result.

Proposition 3 *For any random variables X, Y, Z and functions f, g, h such that $g(x) = h(f(x))$ for all x , we have that $I(X; Z|Y, f(X)) \leq I(X; Z|Y, g(X))$.*

The proof is in Appendix E.1. A special case of the result is that $I(X; Z|Y, f(X)) \leq I(X; Z|Y)$, where $h(f(x)) = 0$ is a trivial function. This result shows that adding more information from the past results in the history forgetting term to be *monotonically* non-increasing. The most direct construction of $f(\cdot)$ is to let the denoising networks take X or a part of X as the input. However, there can be redundant information in all the past frames that cannot mitigate the history forgetting.

Fact 1 *For any three random variables X, Y, Z , if $f(X)$ and Z are conditionally independent given Y , then $I(X; Z|Y, f(X)) = I(X; Z|Y)$.*

Thus, for efficient inference, it is also important to compress X , i.e., discarding conditionally independent components. We will explore this method in the next section.

6. Empirical Verifications

The primary objectives of our experiments are twofold: (1) to validate the theoretical conclusions established in Section 5, particularly regarding the monotonic relationship between history length and history forgetting, as well as the correlation between history forgetting and temporal degradation; and (2) to provide new empirical insights for the characteristics of AR-VDMs. While we compare our approach against various baselines, these comparisons serve to contextualize our findings and support our theoretical claims rather than to compete for state-of-the-art performance on generation quality metrics.

6.1 Experiments on History Forgetting

6.1.1 EXPERIMENTAL SETUP

We conduct experiments on two controlled environments: DMLab (Beattie et al., 2016) and Minecraft (Guss et al., 2019). For DMLab, we follow the TECO (Yan et al., 2023) training split and select evaluation cases from the corresponding validation split. For Minecraft, we render scenarios using MineRL while preserving the original frame rate. Our models are trained using the Adam optimizer $(\beta_1, \beta_2) = (0.9, 0.999)$ with no weight decay. Gradients

are clipped to a maximum norm of 1.0, with additional noise clipping at a norm of 6.0. For Minecraft experiments, we adopt the VAE from *stable-diffusion-3-medium* to enable latent diffusion. Across all experiments, we use fused SNR reweighting with a cumulative SNR decay of 0.96 and train with v-prediction as the target. Our denoiser is a 3D UNet with ResNet blocks alongside spatial and temporal attention modules, comprising 4 blocks each in the downsampling and upsampling paths.

The direct calculation of history forgetting—defined as the conditional mutual information between the output and the past frames—is often infeasible for high-dimensional problems due to computational and sample complexity (Belghazi et al., 2018; Paninski, 2003). To overcome this, we empirically evaluate history forgetting by measuring the inconsistency between the output and the past frames. Specifically, we design a visual “needle-in-the-haystack” task, where AR-VDMs are tasked with predicting several future frames that are determined by the past, serving as ground truth. History forgetting is quantified by the difference between the predicted frames and the ground truth. Figure 3 illustrates an example of the history retrieval task in DMLab, where the model must navigate back to previously visited locations to generate frames consistent with the past context.

Evaluation Metric. To access ground truth, we use the controlled environments—DMLab (Beattie et al., 2016) and Minecraft (Guss et al., 2019)—where frames are determined by actions. In DMLab, which features simple scenes with floors, walls, and wall decorations, we use the Successful Retrieval Rate (SRR) metric, defined as the ratio of trials in which floor and wall colors and wall decorations are correctly retrieved. For the more complex and diverse Minecraft scenes, we adopt SSIM (Wang et al., 2004). It measures the similarity of two images via the expectation, variance, and covariance of pixels. Retrieval is considered failed if SRR is 0 or SSIM is below 0.4.

To provide intuition for these SSIM thresholds, Figure 4 illustrates example frame pairs across different SSIM ranges in Minecraft, sampled from approximately 1,000 video trajectories. As shown in the figure, $SSIM < 0.4$ indicates weak visual similarity where the relationship between frames is difficult to discern; $0.4 \leq SSIM < 0.9$ represents overall resemblance with noticeable detail differences (e.g., slight camera angle variations or object placement); and $SSIM \geq 0.9$ indicates nearly identical frames with only imperceptible pixel-level differences.

We will employ these metrics to assess how injection of past information impacts history forgetting, fixing the sliding window and stride to $w = \Delta = 16$ across all experiments.

Network Architectures. We explore two structures for integrating past frame information into current AR generation on a shared U-Net backbone, as illustrated in Figure 5:

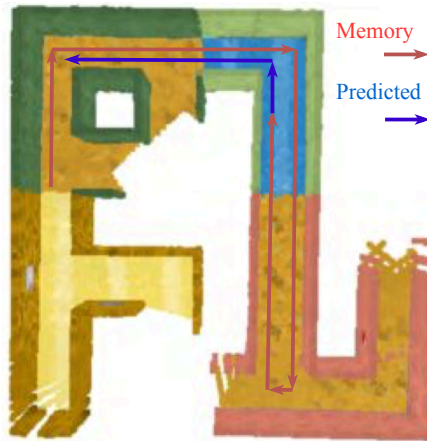


Figure 3: An example of the history retrieval task in DMLab. The agent starts at the initial position and must generate frames showing the return to previously visited areas, testing the model’s ability to retrieve historical context.



(a) $SSIM < 0.4$: The visual similarity between frames is generally weak, making it difficult to discern a clear relationship between them.



(b) $0.4 \leq SSIM < 0.9$: The two frames typically share an overall resemblance in scene composition but exhibit noticeable differences in details, such as slight variations in camera angle or object placement.



(c) $SSIM \geq 0.9$: The frames are nearly identical, with only minor pixel-level differences that are imperceptible to the human eye.

Figure 4: Visual interpretation of SSIM ranges in Minecraft. Example frame pairs demonstrate the three thresholds used for evaluating history retrieval performance.

Prepending (temporal concatenation): We directly concatenate the past w frames before the current frames along the temporal axis; past actions are layer-normalized and injected as control signals. This feeds explicit, retrievable history into the backbone.

Channel concatenation: We concatenate the past frames and their action embeddings along the channel dimension at the network input and use an initial convolutional fusion. Unlike Che et al. (2024) (which extends image diffusion to video and relies on cross-modality attention), we explicitly fuse actions with frames via a lightweight Transformer encoder to form the input tokens, and inject history at every denoised frame since VDMs denoise multiple frames per step.

History Compression (Optional). Orthogonally, to balance retrieval quality and efficiency, we optionally compress the “past frames + actions” into a compact memory with budget $B \in \{8, 16, 32, 48\}$ using a small Transformer (Figure 6). We adopt two methods to compress the past frames and actions. The joint method, which is shown in the left part of Figure 6, processes the concatenation of the frame and actions jointly. The network consists of the feedforward module and spatial and temporal attention modules. The output is also the concatenation of compressed frames and actions. The modulated method, which is shown in the right part of Figure 6, utilizes the actions to modulate the frames. In both methods, we only retain the last several frames and tokens as the final compressed memory. In the experiment, the prepending and channel concatenation structures respectively adopts the joint compression and modulated compression. The reason is that the prepending structure also requires compressed actions to transform the frames to different feature spaces, and the channel concatenation needs to merge the frame and action information.

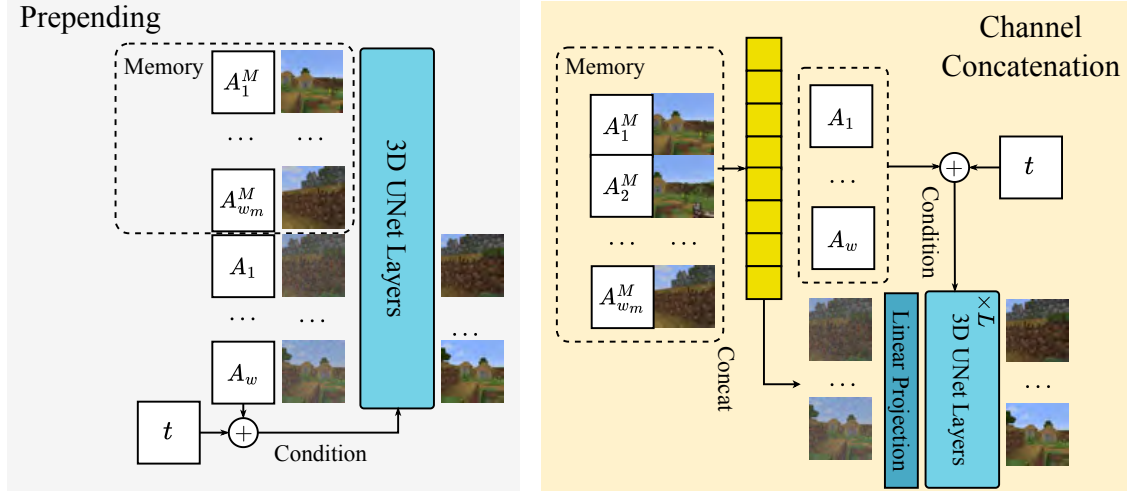


Figure 5: Network structures for adding information of previous frames into each AR step. Here w is the number of past frames provided for the denoising network. The superscript M refers to the past frames and actions (memory).

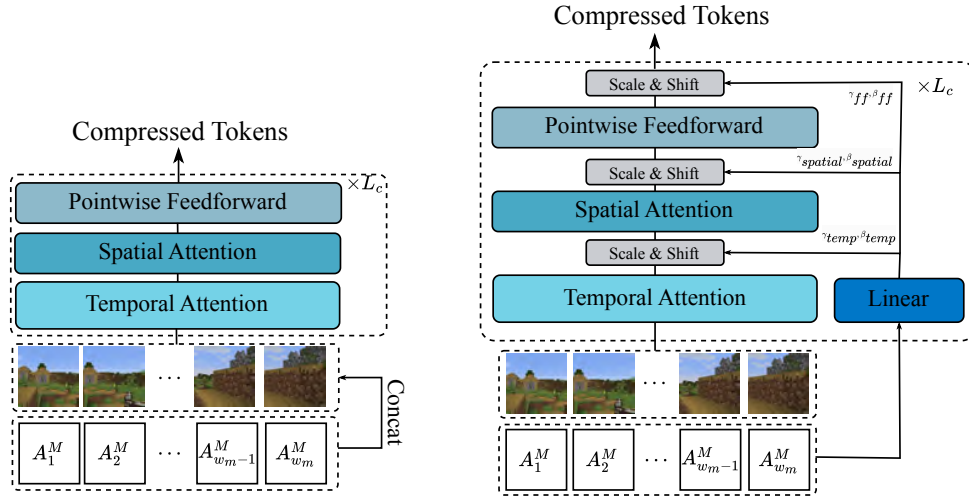


Figure 6: Network structures for compressing past frames and actions. Left: joint compression processes concatenated frames and actions through feedforward and attention modules. Right: modulated compression uses actions to modulate frame representations. Both retain only the last several tokens as the final compressed memory.

6.1.2 EXPERIMENT RESULTS.

Verification of Theorem 1, Proposition 3, and Fact 1. As discussed in Section 5, we quantify history forgetting using the conditional mutual information between the generated

Table 1: History-Aware Retrieval Metrics: SRR \uparrow for DMLab, and SSIM \uparrow for Minecraft. Given a reference timestamp, the notation “ i - j ” denotes the task of retrieving the scenes that appeared in the past i - j frames relative to the timestamp. “History” indicates the length of the provided historical context, while “Budget” refers to the network compression constraint.

Method	History	Budget	DMLab (SRR \uparrow) for $i - j$				Minecraft (SSIM \uparrow) for $i - j$			
			0-16	16-48	48-112	Avg	0-16	16-48	48-112	Avg
Prepending	16	-	0.52	0.03	0.00	0.18	0.75	0.39	0.37	0.50
	48	-	0.44	0.32	0.00	0.25	0.74	0.62	0.35	0.57
	112	-	0.48	0.28	0.23	0.33	0.72	0.60	0.55	0.62
	48	8	0.28	0.22	0.00	0.17	0.63	0.53	0.38	0.51
	48	16	0.36	0.25	0.00	0.20	0.67	0.55	0.37	0.53
	48	32	0.44	0.28	0.00	0.24	0.70	0.57	0.38	0.55
	48	48	0.40	0.34	0.00	0.25	0.72	0.61	0.35	0.56
Channel Concat	16	-	0.24	0.03	0.00	0.09	0.61	0.38	0.38	0.46
	48	-	0.16	0.19	0.00	0.12	0.58	0.57	0.37	0.51
	112	-	0.12	0.13	0.08	0.11	0.59	0.56	0.51	0.55
	48	8	0.16	0.16	0.00	0.11	0.54	0.53	0.36	0.48
	48	16	0.20	0.19	0.00	0.13	0.56	0.54	0.35	0.48
	48	32	0.24	0.22	0.00	0.15	0.57	0.56	0.36	0.50
	48	48	0.28	0.22	0.00	0.17	0.57	0.57	0.37	0.50

frames and all past frames, conditioned on the inputs. Proposition 3 and Fact 1 provide two key properties. Proposition 3 states that conditioning on more informative variables cannot worsen performance, whereas Fact 1 shows that conditioning on redundant variables—i.e., variables independent of the future frames given the input—cannot improve it. We empirically validate these intuitions. To verify Proposition 3, we condition the frame-generation process on increasingly large sets of past frames, where each larger set strictly contains the smaller ones. If the monotonicity property holds, history-forgetting error should decrease as more frames are given. To verify Fact 1, we apply different levels of compression to the conditioned frames. If redundant information does not help, then light compression (which mainly removes redundancy) should yield performance comparable to no compression, while heavy compression (which removes useful information) should significantly degrade it.

The results of SRR for DMLab and SSIM for Minecraft in Tables 1 effectively demonstrate the impact of history budget for compression (8, 16, 32, 48), history length (16, 48, 112), and network structures. Figure 7 presents representative successful retrievals, and Appendix I provides additional successful and failed cases. Across all architectures, history forgetting errors consistently decrease with increasing history length and history budget, aligning with Proposition 3. In this context, the functions f and g serve as the submodules responsible for integrating historical information. Moreover, the compression results in Tables 1 reveal significant redundancy in the past frames and actions. Notably, when the history length is compressed from 48 to 8, the performance does not degrade proportionally, verifying Fact 1. This indicates that conditioning the model on redundant information offers limited benefits.

Proper evaluations of history forgetting. The results in Tables 1 show that our “needle-in-the-haystack” design accurately captures history forgetting. As discussed in Section 5, our

Table 2: Common Evaluation Metrics on DMLab and Minecraft across both structures. We report FVD and FID for both environments, while motion metrics (TF for Temporal Flickering and MS for Motion Smoothness) are reported only for Minecraft due to DMLab’s insufficient frame rate for reliable motion evaluation. History length = w (16/48/112); Budget = number of preserved history latents after compression (8/16/32/48).

Method	History	Budget	DMLab		Minecraft			
			FVD	FID	FVD	FID	TF	MS
Prepending	16	-	302.59	80.27	224.30	75.74	0.750	0.853
	48	-	308.92	78.44	222.78	73.96	0.754	0.853
	112	-	308.81	80.45	226.49	75.78	0.751	0.867
	48	8	310.59	79.50	222.30	76.36	0.754	0.873
	48	16	307.20	78.35	226.08	74.43	0.738	0.863
	48	32	311.94	73.65	223.45	75.35	0.760	0.863
	48	48	308.73	74.37	223.13	78.50	0.737	0.845
Channel Concat	16	-	294.01	71.07	209.12	71.49	0.757	0.861
	48	-	300.56	73.40	212.92	68.43	0.769	0.858
	112	-	298.85	74.00	206.17	70.51	0.762	0.862
	48	8	304.95	72.84	208.16	68.44	0.761	0.855
	48	16	302.86	74.78	207.02	68.58	0.773	0.859
	48	32	296.06	75.40	214.01	66.04	0.770	0.860
	48	48	298.56	74.02	209.55	71.40	0.771	0.847

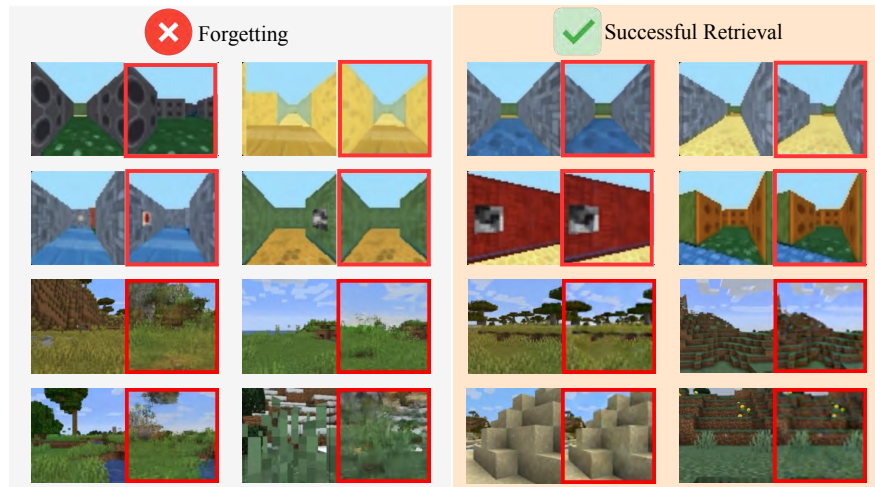


Figure 7: Examples of successful history retrieval in DMLab and Minecraft. Rows show cases where the model returns to previously visited areas or reproduces earlier scene layouts, indicating low history-forgetting error.

theory further implies that standard video-distribution metrics (e.g., FVD and FID) (Henschel et al., 2024; Harvey et al., 2022) and motion-related metrics (e.g., temporal flickering and motion smoothness) (Huang et al., 2024) are ineffective for measuring this phenomenon. Table 2 presents results for both structures across DMLab and Minecraft. Due to DMLab’s

low FPS, we assess temporal flickering and motion smoothness only for Minecraft. These metrics show minor fluctuations across both structures and settings without any clear link to history forgetting, consistent with our theoretical analysis. Notably, despite the sizable retrieval gap between prepending and channel concatenation, their FVD and FID scores remain comparable, further confirming these metrics’ limitations in capturing history forgetting.

A natural follow-up question is whether existing methods that improve distribution and motion scores can also reduce history forgetting. To investigate this, we conducted a comprehensive comparison between our soft compression approach and various baselines, including frame sampling strategies, training-free autoregressive methods, and reimplemented state-of-the-art techniques. For fair comparison, we incorporate the action-conditioned benchmark from GameNGen (Valevski et al., 2024) to enable autoregressive long video generation. We also reimplement representative baselines such as StreamingT2V (Henschel et al., 2024) and adopt various frame sampling strategies inspired by prior work (Harvey et al., 2022). All re-implemented baselines, which lack mechanisms to leverage long-horizon history, are restricted to using only the last 16 frames as input.

Table 3 presents the quantitative comparison on DMLab and Minecraft under a fixed network compression budget of 16 and history length of 48. The results reveal several important findings. First, our soft compression (prepending memory with memory compression) consistently outperforms all baselines across both short-term (0-16) and longer temporal horizons (16-32), achieving the best overall performance on both environments. Second, training-free methods (FIFO and Outpainting) show limited effectiveness in reducing history forgetting, with particularly poor performance on longer horizons. Third, while reimplemented baselines like StreamingT2V and GameNGen demonstrate reasonable short-term performance, they struggle with mid-range and long-range history retrieval.

Specifically, StreamingT2V (Henschel et al., 2024), which includes several first frames (long-term information) and near frames (short-term information), is designed to improve motion degree and warping error. However, as shown in Table 3, it cannot effectively retrieve mid-range history, achieving only 0.040 SRR on DMLab’s 16-32 frame range. Similarly, GameNGen (Valevski et al., 2024) emphasizes short-term inter-frame consistency but sacrifices long-term consistency, completely failing on longer horizons (0.000 SRR for 16-32 frames in DMLab). Frame sampling strategies (Harvey et al., 2022) merely trade off history forgetting errors across different horizons rather than fundamentally addressing the problem. These results suggest that existing approaches are not designed to effectively tackle history forgetting, and discovering optimal strategies for compressing historical information remains an open challenge.

6.2 Experiments on Temporal Degradation

Proper evaluation of temporal degradation. Our theoretical results in Section 5 imply that any video-distribution metrics are suitable to evaluate temporal degradation. In the following, we will verify this by evaluating FVD, FID, and PSNR. Similar to the concept of history forgetting, this evaluation is conducted within DMLab and Minecraft environments. Figure 9 indicates normalized different metrics for temporal degradation. All the metrics of the first 16 frames are normalized to 100%, which corresponds to the initial portion of the generated video and is expected to exhibit the highest visual quality. The

Table 3: Comparison of different history compression methods on DMLab and Minecraft. Our soft compression approach (bottom row) consistently outperforms frame sampling strategies, training-free methods, and re-implemented baselines, especially over longer temporal horizons (16-32 frames).

Method	DMLab (SRR \uparrow)			Minecraft (SSIM \uparrow)		
	0-16	16-32	Overall	0-16	16-32	Overall
Frame Sampling						
Prepending Memory + Last 16 Frames	0.520	0.000	0.260	0.610	0.380	0.495
Prepending Memory + Last 32 Frames with 1 Frame Skip	0.260	0.154	0.207	0.510	0.434	0.472
Training-free AR Methods						
FIFO	0.123	0.030	0.077	0.453	0.356	0.405
Outpainting	0.100	0.021	0.061	0.430	0.360	0.395
Re-implemented Baselines						
GameNGen	0.214	0.000	0.107	0.471	0.321	0.396
StreamingT2V	0.263	0.040	0.152	0.480	0.312	0.396
Soft Compression						
Prepending Memory + Memory Compression	0.360	0.250	0.305	0.561	0.543	0.552

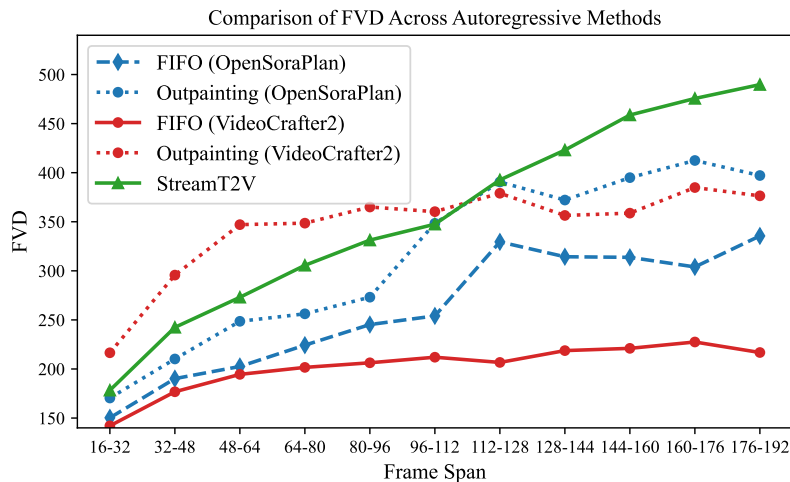


Figure 8: FVDs of *short* clips generated by different models and methods. The FVD of short clips generated by all the methods and models grows with increasing AR steps, which we coin temporal degradation.

results show that FVD, FID, and PSNR show clear trends to degrade along the temporal axis. It qualitatively corroborates our theoretical findings. Note that assessing temporal degradation does not necessitate closed-ended environments. To demonstrate the generality

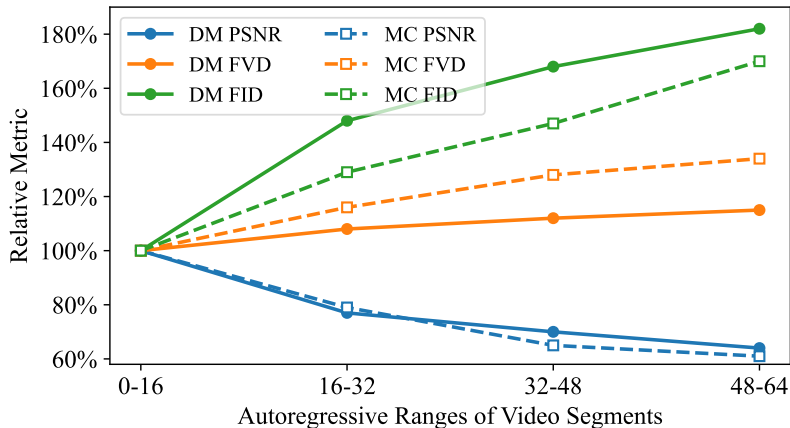
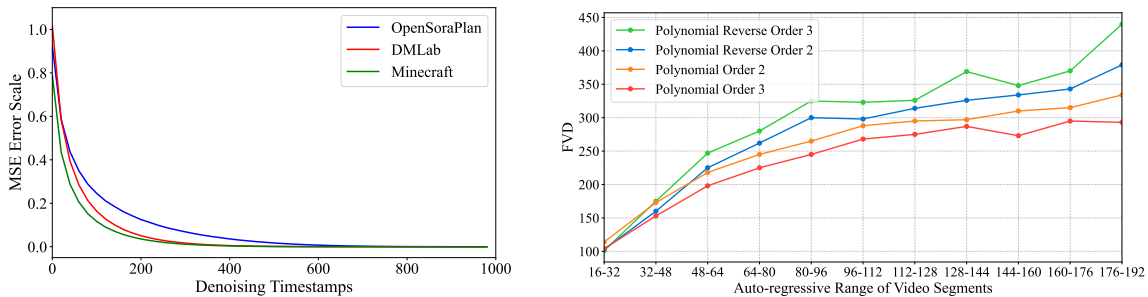


Figure 9: Temporal degradation results for DMLab and Minecraft. All values are normalized with respect to frames 0–16.

of temporal degradation beyond these controlled settings, we also evaluate it on general text-to-video generation tasks using pretrained models. Figure 8 shows FVD measurements of short clips generated by different pretrained models across multiple autoregressive steps, confirming that temporal degradation is a widespread phenomenon affecting various video generation architectures.

Quantitative verification of Eqn. (10). We highlight Eqn. (10) can also quantitatively predict the rate of temporal degradation. We measure the temporal degradation of four DDPM time-step sets concerning FVD: (a,1) $\text{Unif}([0, 999^{1/2}], N)^2$, (a,2) $999 - \text{Unif}([0, 999^{1/2}], N)^2$, (b,1) $\text{Unif}([0, 999^{1/3}], N)^3$, (b,2) $999 - \text{Unif}([0, 999^{1/3}], N)^3$. Here, $\text{Unif}([0, 999^{1/2}], N)^2$ denotes the following sampling procedure: we first draw N points uniformly from the interval $[0, 999^{1/2}]$, and then define the selected steps as the squared values of these N points. The notation $999 - \text{Unif}([0, 999^{1/2}], N)^2$ means that the selected steps are obtained by subtracting each element of $\text{Unif}([0, 999^{1/2}], N)^2$ from 999. Figure 10b shows that (*,1) is better than (*,2) for both (a) and (b), and that the performance gap in (b,1)/(b,2) is larger. These all corroborate Eqn. (10). NIE and DE are identical for (*,1) and (*,2). By Assumption 3, the weights of steps in SEE are proportional to the intervals between steps. Since (*,1) has denser steps near 0 (i.e., smaller intervals and weights), our bound predicts slower temporal degradation for (*,2). (b,1)/(b,2) have a larger weight difference, implying a larger performance gap. Similar results hold for Minecraft and pretrained T2V models (see Appendix J for detailed results).

Correlation between history forgetting and temporal degradation. Intuitively, history forgetting—the failure to retrieve relevant information from prior context—and temporal degradation—the accumulation of noise over time—are negatively correlated. A lower forgetting error indicates a stronger ability to retrieve historical information, but this may also introduce more historical noise into current outputs, resulting in more severe temporal degradation. Figure 11 illustrates this trade-off by plotting temporal degradation



(a) Pretraining errors of different time steps for OpenSoraPlan, DMLab, and Minecraft. (b) Comparison of FVD across different timestep sets for OpenSoraPlan.

Figure 10: Impact of timestep schedules on training error and temporal degradation. Left: pretraining loss evolution for different timestep sets across OpenSoraPlan, DMLab, and Minecraft. Right: FVD comparison of the same schedules on OpenSoraPlan, highlighting how denser early steps curb degradation.

Table 4: Temporal degradation quantified by PSNR across different frame segments in DMLab and Minecraft. PSNR values consistently decrease over time, indicating progressive quality loss. Higher compression budgets generally result in faster degradation. Error bars are computed from 5 independent runs with different random seeds.

Budget	DMLab				Minecraft			
	0-16	16-32	32-48	48-64	0-16	16-32	32-48	48-64
8	10.24 ± 0.11	9.47 ± 0.13	8.83 ± 0.14	7.88 ± 0.19	15.25 ± 0.15	13.43 ± 0.17	11.39 ± 0.20	10.29 ± 0.23
16	10.14 ± 0.10	8.81 ± 0.14	8.24 ± 0.17	7.39 ± 0.21	15.47 ± 0.13	12.92 ± 0.19	11.10 ± 0.23	9.98 ± 0.24
32	10.46 ± 0.14	8.19 ± 0.13	7.82 ± 0.18	6.83 ± 0.20	15.36 ± 0.16	12.64 ± 0.20	11.01 ± 0.22	9.73 ± 0.26
48	10.52 ± 0.13	8.00 ± 0.15	7.58 ± 0.18	6.76 ± 0.23	15.46 ± 0.18	12.31 ± 0.19	10.57 ± 0.26	9.47 ± 0.27

(measured as maximum PSNR decay, see Table 4) against the average SRR for compressed prepending. As SRR increases (indicating reduced history forgetting), PSNR decay also increases (indicating faster quality loss). This negative correlation is consistently observed in both DMLab and Minecraft environments, as shown in Figure 11. We leave the theoretical investigation of this phenomenon to future work.

6.3 Discussion of Mitigating History Forgetting and Temporal Degradation

The experiments in Sections 6.1 and 6.2 corroborate Theorem 1, demonstrating that AR-VDMs exhibits history forgetting and temporal degradation. These phenomena are characterized through the conditional mutual information between past and output frames and through the accumulation of initialization, discretization, and score-estimation errors. This characterization suggests that any technique that reduces these error terms can potentially improve video generation quality. To reduce discretization error, one may further employ alternative solvers such as DPM-Solver (Lu et al., 2022), DPM-Solver++ (Lu et al., 2025), and DDIM (Song et al., 2020a). From a practical perspective, to accelerate generation while preserving quality, one may adopt the parallel sampling method of Shih et al. (2023), which parallelizes diffusion-model inference with controlled score-estimation error. To generate

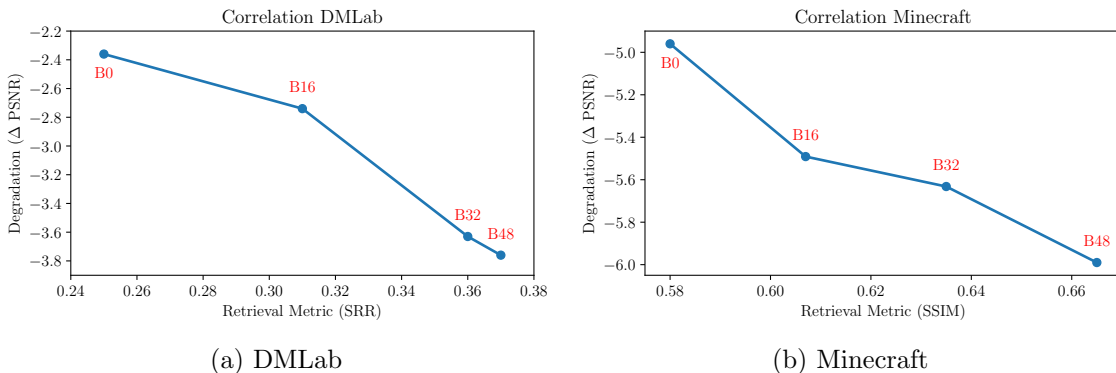


Figure 11: Correlation between history forgetting and temporal degradation in DMLab and Minecraft. Here B_i denotes the network compression budget. Both environments exhibit a consistent negative correlation: higher retrieval accuracy (reduced forgetting) leads to faster quality degradation.

high-definition videos and support arbitrary video sizes, one may represent videos as graphs and apply DiffCollage (Zhang et al., 2023a). More generally, existing theoretical analyses of initialization, discretization, and score-estimation errors for different sampling algorithms can be combined with our framework to analyze these methods in AR-VDMs (Chen et al., 2024c; Gupta et al., 2024). We leave these directions for future work.

7. Conclusion

Based on the proposed unified framework AR-VDM, we demonstrated that history forgetting corresponds to the conditional mutual information between the output and past frames, and its emergence is inherently inevitable. The temporal degradation, in turn, accumulates as the total error from all preceding AR steps. Moreover, we observed a negative correlation between history forgetting and temporal degradation. From a theoretical perspective, a limitation lies in the gap between the upper and lower bounds for history forgetting, which could be narrowed by replacing Pinsker’s inequality with tighter alternatives. From an experimental standpoint, our results do not directly indicate the most effective structure for mitigating both history forgetting and temporal degradation. Additionally, understanding why certain attention-based mechanisms struggle to integrate historical information falls outside the scope of our theoretical framework. We hypothesize that these difficulties stem from the same underlying issue that causes weak conditioning from auxiliary modalities (e.g., text or image) (Esser et al., 2024). Verifying this hypothesis and exploring optimal architectures for history integration are left for future investigations.

Acknowledgments

Vincent Y. F. Tan is supported by a Singapore Ministry of Education (MOE) AcRF Tier 2 grant under grant number A-8004062-00-00.

References

- Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.
- Charles Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. Deepmind lab. *CoRR*, abs/1612.03801, 2016. URL <http://arxiv.org/abs/1612.03801>.
- M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.
- Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendeleevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023a.
- Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023b.
- Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.
- Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyang Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, et al. Mixture of contexts for long video generation. *arXiv preprint arXiv:2508.21058*, 2025.
- Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. Gamegen-x: Interactive open-world game video generation. *arXiv preprint arXiv:2411.00769*, 2024.
- Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 2024a.
- H. Chen, H. Lee, and J. Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *International Conference on Machine Learning*, pages 4735–4763. PMLR, 2023a.
- Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024b.

- Haoxuan Chen, Yinuo Ren, Lexing Ying, and Grant M Rotskoff. Accelerating diffusion models with parallel sampling: Inference at sub-linear time complexity. *Advances in Neural Information Processing Systems*, 37:133661–133709, 2024c.
- Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *International Conference on Machine Learning*, pages 4672–4712. PMLR, 2023b.
- Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022.
- Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ode is provably fast. *Advances in Neural Information Processing Systems*, 36, 2024d.
- Tsai-Shien Chen, Chieh Hubert Lin, Hung-Yu Tseng, Tsung-Yi Lin, and Ming-Hsuan Yang. Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404*, 2023c.
- Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. In *The Twelfth International Conference on Learning Representations*, 2023d.
- Sinho Chewi, Murat A Erdogdu, Mufan Li, Ruoqi Shen, and Matthew S Zhang. Analysis of langevin monte carlo from poincare to log-sobolev. *Foundations of Computational Mathematics*, pages 1–51, 2024.
- Muthu Chidambaram, Khashayar Gatmiry, Sitan Chen, Holden Lee, and Jianfeng Lu. What does guidance do? a fine-grained analysis in a simple setting. *Advances in Neural Information Processing Systems*, 37:84968–85005, 2024.
- P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.
- Hengyu Fu, Zhuoran Yang, Mengdi Wang, and Minshuo Chen. Unveil conditional diffusion models with classifier-free guidance: A sharp statistical theory. *arXiv preprint arXiv:2403.11968*, 2024.
- Jiayang Gao, Tianyi Zheng, Jiayang Zou, Fengxiang Yang, Shice Liu, Luyao Fan, Zheyu Zhang, Hao Zhang, Jinwei Chen, Peng-Tao Jiang, et al. C2fg: Control classifier-free guidance via score discrepancy analysis. *arXiv preprint arXiv:2603.08155*, 2026.
- Yu Gao, Jiancheng Huang, Xiaopeng Sun, Zequn Jie, Yujie Zhong, and Lin Ma. Matten: Video generation with mamba-attention. *arXiv preprint arXiv:2405.03025*, 2024.
- Walter Gautschi. *Numerical analysis*. Springer Science & Business Media, 2011.

- Hanzhong Guo, Hongwei Yi, Daquan Zhou, Alexander William Bergman, Michael Lingelbach, and Yizhou Yu. Real-time one-step diffusion-based expressive portrait videos generation. *arXiv preprint arXiv:2412.13479*, 2024.
- Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.
- Shivam Gupta, Linda Cai, and Sitan Chen. Faster diffusion sampling with randomized midpoints: Sequential and parallel. *arXiv preprint arXiv:2406.00924*, 2024.
- William H. Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden R. Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 2442–2448. ijcai.org, 2019. doi: 10.24963/IJCAI.2019/339. URL <https://doi.org/10.24963/ijcai.2019/339>.
- W. Harvey, S. Naderiparizi, V. Masrani, C. Weillbach, and F. Wood. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35:27953–27965, 2022.
- Roberto Henschel, Levon Khachatryan, Daniil Hayrapetyan, Hayk Poghosyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. *arXiv preprint arXiv:2403.14773*, 2024.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.
- Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022b.
- Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.
- Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.
- Yuchen Jiao, Yuxin Chen, and Gen Li. Towards a unified framework for guided diffusion models. *arXiv preprint arXiv:2512.04985*, 2025.

- Cheng Jin, Qitan Shi, and Yuantao Gu. Stage-wise dynamics of classifier-free guidance in diffusion models. *arXiv preprint arXiv:2509.22007*, 2025.
- Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.
- Johanna Karras, Aleksander Holynski, Ting-Chun Wang, and Ira Kemelmacher-Shlizerman. Dreampose: Fashion image-to-video synthesis via stable diffusion. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22623–22633. IEEE, 2023.
- Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. FIFO-Diffusion: Generating infinite videos from text without training. *arXiv preprint arXiv:2405.11473*, 2024.
- Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37: 122458–122483, 2024.
- Krunoslav Lehman Pavasovic, Jakob Verbeek, Giulio Biroli, and Marc Mezard. Understanding classifier-free guidance: High-dimensional theory and non-linear generalizations. *arXiv e-prints*, pages arXiv–2502, 2025.
- Gen Li and Yuchen Jiao. Provable efficiency of guidance in diffusion models for general data distribution. *arXiv preprint arXiv:2505.01382*, 2025.
- Xiang Li, Rongrong Wang, and Qing Qu. Towards understanding the mechanisms of classifier-free guidance. *arXiv preprint arXiv:2505.19210*, 2025.
- Hanwen Liang, Yuyang Yin, Dejia Xu, Hanxue Liang, Zhangyang Wang, Konstantinos N Plataniotis, Yao Zhao, and Yunchao Wei. Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models. *arXiv preprint arXiv:2405.16645*, 2024.
- Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.
- Vivian Liu, Tao Long, Nathan Raw, and Lydia Chilton. Generative disco: Text-to-video generation for music visualization. *arXiv preprint arXiv:2304.08551*, 2023.
- Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.
- Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, 2025.
- Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024a.

- Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Siran Chen, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024b.
- Shentong Mo and Yapeng Tian. Scaling diffusion mamba with bidirectional ssms for efficient image and video generation. *arXiv preprint arXiv:2405.15881*, 2024.
- L. Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6): 1191–1253, 2003.
- Weiming Ren, Huan Yang, Ge Zhang, Cong Wei, Xinrun Du, Wenhao Huang, and Wenhui Chen. Consisti2v: Enhancing visual consistency for image-to-video generation. *arXiv preprint arXiv:2402.04324*, 2024.
- Kevin Rojas, Ye He, Chieh-Hsin Lai, Yuhta Takida, Yuki Mitsufuji, and Molei Tao. Improving classifier-free guidance in masked diffusion: Low-dim theoretical insights with high-dim impact. In *The Fourteenth International Conference on Learning Representations*, 2026.
- David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. *arXiv preprint arXiv:2402.09470*, 2024.
- Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36: 4263–4276, 2023.
- Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.
- Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025.
- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.
- Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.
- Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.
- N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

- Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.
- S. Verdú. Total variation distance and the distribution of relative information. In *2014 information theory and applications workshop (ITA)*, pages 1–3. IEEE, 2014.
- M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.
- Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.
- Xi Wang, Nicolas Dufour, Nefeli Andreou, Marie-Paule Cani, Victoria Fernández Abrevaya, David Picard, and Vicky Kalogeiton. Analysis of classifier-free guidance weight schedulers. *arXiv preprint arXiv:2404.13040*, 2024a.
- Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024b.
- Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yanan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision*, pages 1–20, 2024c.
- Yuqing Wang, Tianwei Xiong, Daquan Zhou, Zhijie Lin, Yang Zhao, Bingyi Kang, Jiashi Feng, and Xihui Liu. Loong: Generating minute-level long videos with autoregressive language models. *arXiv preprint arXiv:2410.02757*, 2024d.
- Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. URL <https://doi.org/10.1109/TIP.2003.819861>.
- Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024e.
- Wenming Weng, Ruoyu Feng, Yanhui Wang, Qi Dai, Chunyu Wang, Dacheng Yin, Zhiyuan Zhao, Kai Qiu, Jianmin Bao, Yuhui Yuan, et al. Art-v: Auto-regressive text-to-video generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7395–7405, 2024.
- Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. Ar-diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information Processing Systems*, 36:39957–39974, 2023.

- Xiaofei Wu, Guozhen Zhang, Zhiyong Xu, Yuan Zhou, Qinglin Lu, and Xuming He. Pack and force your memory: Long-form and consistent video generation, 2025. URL <https://arxiv.org/abs/2510.01784>.
- Yuchen Wu, Minshuo Chen, Zihao Li, Mengdi Wang, and Yuting Wei. Theoretical insights for diffusion guidance: A case study for gaussian mixture models. *arXiv preprint arXiv:2403.01639*, 2024.
- Mengfei Xia, Nan Xue, Yujun Shen, Ran Yi, Tieliang Gong, and Yong-Jin Liu. Rectified diffusion guidance for conditional generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13371–13380, 2025.
- Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WORLDMEM: long-term consistent world simulation with memory. *CoRR*, abs/2504.12369, 2025. doi: 10.48550/ARXIV.2504.12369. URL <https://doi.org/10.48550/arXiv.2504.12369>.
- Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pages 399–417. Springer, 2025.
- Wilson Yan, Danijar Hafner, Stephen James, and Pieter Abbeel. Temporally consistent transformers for video generation. In *International Conference on Machine Learning*, pages 39062–39098. PMLR, 2023.
- Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.
- He Ye, Rojas Kevin, and Tao Molei. What exactly does guidance do in masked discrete diffusion models. *arXiv preprint arXiv:2506.10971*, 2025.
- Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. *CoRR*, abs/2506.03141, 2025. doi: 10.48550/ARXIV.2506.03141. URL <https://doi.org/10.48550/arXiv.2506.03141>.
- Hangjie Yuan, Shiwei Zhang, Xiang Wang, Yujie Wei, Tao Feng, Yining Pan, Yingya Zhang, Ziwei Liu, Samuel Albanie, and Dong Ni. Instructvideo: instructing video diffusion models with human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6463–6474, 2024.
- Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. Diffcollage: Parallel generation of large content with diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10188–10198. IEEE, 2023a.

Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023b.

Xinyu Zhang, Yuhan Liu, Haonan Chang, Liam Schramm, and Abdeslam Boularias. Autoregressive action sequence learning for robotic manipulation. *arXiv preprint arXiv:2410.03132*, 2024.

Xiaoming Zhao and Alex Schwing. Studying classifier (-free) guidance from a classifier-centric perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 13271–13279, 2026.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Appendix A. Notation

This appendix provides a comprehensive list of mathematical notation used throughout the paper. Symbols are organized by category for ease of reference.

A.1 Basic Notation

Symbol	Description	Defined in
$X_{1:w}$	Clean video frames sequence $(X_1, \dots, X_w) \in \mathbb{R}^{w \times d}$	§3
$X_{1:w}^0$	Initial clean video (same as $X_{1:w}$)	§3
$X_{1:w}^t$	Noisy video frames at diffusion time t	§3
$\tilde{X}_{1:w}^t$	Reverse-time process for video frames	§3
$Y_{1:w}$	Noisy video frames sequence (generated by algorithm)	Algorithm 1
$Y_{1:w}^t$	Noisy video at noise level t (generated by algorithm)	Algorithm 1
$w \in \mathbb{N}$	Number of frames / effective window size	§3
d	Dimensionality of each frame (e.g., 512×512 for 512×512 -pixel frames)	§3
t	Diffusion time / noise level	§3
T	Maximum noise level of the diffusion process	§3
B^t	$(w \times d)$ -dimensional Brownian motion with identity covariance	§3
\tilde{B}^t	Reverse-time Brownian motion	§3
$f(\cdot, t)$	Drift coefficient function: $f: \mathbb{R}^{w \times d} \times \mathbb{R} \rightarrow \mathbb{R}^{w \times d}$	§3
$g(t)$	Diffusion coefficient function: $g: \mathbb{R} \rightarrow \mathbb{R}$	§3
$\beta(t)$	Scalar noise schedule (used in DDPM and SMLD)	§3
P_t	Distribution of $X_{1:w}^t$ at diffusion time t	§3
$\nabla \log P_t$	Score function (gradient of log-density)	§3

A.2 AR-VDM Framework

Symbol	Description	Defined in
$\Delta \in \mathbb{N}$	Auto-regressive step size (number of new frames generated per step)	Algorithm 1
\mathcal{L}^I	Input noise levels (t_1^I, \dots, t_w^I)	Algorithm 1
\mathcal{L}^O	Output noise levels (t_1^O, \dots, t_w^O)	Algorithm 1
t_i^I	Input noise level for frame i	Algorithm 1
t_i^O	Output noise level for frame i	Algorithm 1
δ_i	Noise level offset for frame i : $\delta_i = t_i^I - t_1^I = t_i^O - t_1^O$	§5.2
$\bar{t}(t)$	Multi-frame noise level mapping: $\bar{t}(t) = (t + \delta_1, \dots, t + \delta_w)$	§5.2
\bar{T}	Vector of maximum noise levels: $\bar{T} = (T, \dots, T) \in \mathbb{R}^w$	§5.2
$X_{1:w}^{\bar{t}}$	Frames at varied noise levels: $X_{1:w}^{\bar{t}} = (X_1^{t+\delta_1}, \dots, X_w^{t+\delta_w})$	§5.2
$\tilde{Y}_{1:w}^{\bar{t}}$	Reverse-time noisy frames at varied noise levels (AR stage)	§5.2
$P_X^{\bar{t}}$	Distribution of frames at varied noise levels	§5.2
\mathcal{R}	Reference frames index set	Algorithm 1
\mathcal{R}_i	Reference frames index set when denoising frames $i + 1$ to $i + w$: $\mathcal{R}_i \subseteq [i]$	Requirement 1
s_θ	Score function / denoising neural network with parameters θ	§3
N	Length of the video to be generated	Algorithm 1
K	Number of auto-regressive steps: $K = \lceil N/\Delta \rceil$	§5.3

A.3 Denoising Steps

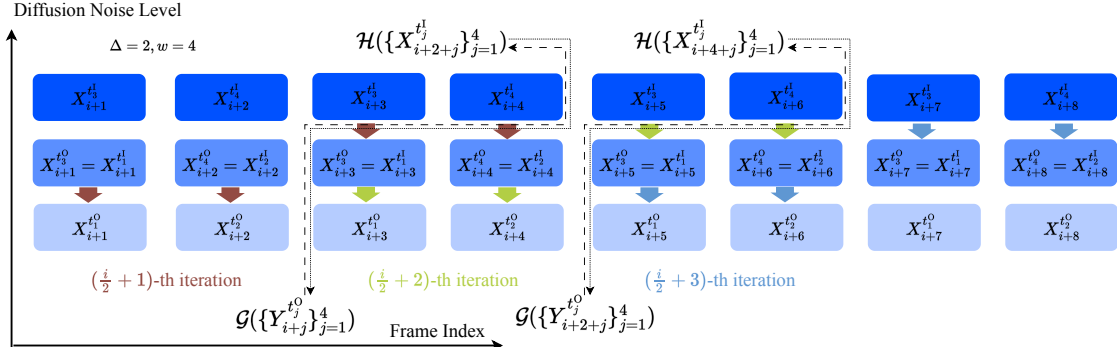
Symbol	Description	Defined in
M_{init}	Number of discretization steps in the initialization stage	§5.1
M_{ar}	Number of discretization steps in the auto-regressive stage	§5.2
t_n^{init}	Noise level at step n in the initialization stage: $0 = t_0^{\text{init}} \leq \dots \leq t_{M_{\text{init}}}^{\text{init}} = T$	§5.1
t_n^{ar}	Noise level at step n in the AR stage: $t_1^{\text{O}} = t_0^{\text{ar}} \leq \dots \leq t_{M_{\text{ar}}}^{\text{ar}} = t_1^{\text{I}}$	§5.2
$\tilde{t}_n^{\text{init}}$	Reverse noise level: $\tilde{t}_n^{\text{init}} = T - t_{M_{\text{init}}-n}^{\text{init}}$	§5.1
\tilde{t}_n^{ar}	Reverse noise level: $\tilde{t}_n^{\text{ar}} = T - t_{M_{\text{ar}}-n}^{\text{ar}}$	§5.2
\bar{t}_n^{init}	Multi-frame reverse noise levels induced by $\tilde{t}_n^{\text{init}}$: $\bar{t}(\tilde{t}_n^{\text{init}})$	§5.1
\bar{t}_n^{ar}	Multi-frame reverse noise levels induced by \tilde{t}_n^{ar} : $\bar{t}(\tilde{t}_n^{\text{ar}})$	§5.2

A.4 Theoretical Analysis

Symbol	Description	Defined in
$\mathcal{H}(\cdot)$	History frames function: all already generated frames before implementing Auto-Regressive Step	Appendix B.1
$\mathcal{G}(\cdot)$	Generated frames function: all generated frames after implementing Auto-Regressive Step	Appendix B.1
S^{I}	Set of starting indices of each input noise level	Appendix B.1
$s^{\text{I}}(t_i^{\text{I}})$	Smallest index whose input noise level equals t_i^{I}	Appendix B.1
i_0	Number of initially generated clean frames	§5.1
L	Lipschitz constant of the score function	§5.3
B	Bound on the norm of video frames: $\ X_{1:w}^{\text{O}}\ \leq B$	§5.3

A.5 Error Terms

Symbol	Description	Defined in
IE	Initialization error	§5.3
ARE $_k$	Auto-regressive error at step k	§5.3
NIE	Noise initialization error in initialization stage: $(di_0 + B^2) \exp(-T)$	§5.3
NIE _{AR}	Noise initialization error in AR stage: $(\Delta d + B^2) \exp(-T)$	§5.3
SEE	Score estimation error in initialization stage: $T\epsilon_{\text{est}}^2$	§5.3
SEE _{AR}	Score estimation error in AR stage: $(t_1^{\text{I}} - t_1^{\text{O}})\epsilon_{\text{est}}^2$	§5.3
DE	Discretization error in initialization stage: $wdL^2 \sum_{n=1}^{M_{\text{init}}} (t_n^{\text{init}} - t_{n-1}^{\text{init}})^2$	§5.3
DE _{AR}	Discretization error in AR stage: $wdL^2 \sum_{n=1}^{M_{\text{ar}}} (t_n^{\text{ar}} - t_{n-1}^{\text{ar}})^2$	§5.3
HF $_k$	History forgetting at step k : $I(\text{Output}_k; \text{Past}_k \text{Input}_k)$	§5.3
Output $_k$	Output frames from k -th AR step: $\{X_{k\Delta+j}^{t_j^{\text{O}}}\}_{j=1}^w$	§5.3
Past $_k$	Past frames: $\mathcal{H}(\{X_{k\Delta+j}^{t_j^{\text{O}}}\}_{j=1}^w) \setminus \{X_{\mathcal{R}_{k\Delta+1}}^{\text{O}}\}$	§5.3
Input $_k$	Input to k -th AR step: $\{X_{\mathcal{R}_{k\Delta+1}}^{\text{O}}\} \cup \{X_{k\Delta+j}^{t_j^{\text{I}}}\}_{j=1}^w$	§5.3
ϵ_{est}	Score estimation error bound	§5.3


 Figure 12: The examples of $\mathcal{G}(\cdot)$ and $\mathcal{H}(\cdot)$.

A.6 Divergence and Distance Measures

Symbol	Description	Defined in
$\text{KL}(\cdot \parallel \cdot)$	Kullback-Leibler divergence	§5.3
$I(\cdot; \cdot \mid \cdot)$	Conditional mutual information (used in history forgetting HF _k)	§5.3
$\mathbb{E}[\cdot]$	Expectation operator	§3
$\ \cdot\ $	ℓ_2 -norm	§3

Appendix B. Proof of Theorem 1

We provide the proof of Theorem 1 in this section. We respectively prove Eqn. (7) and (10) in Sections B.1 and B.2.

B.1 Proof of Eqn. (7)

Our proof consists of five main procedures.

- Decomposition of the concatenation of all the noisy and clean frames.
- Decomposition of the KL-divergence of all the frames into the KL-divergence of each auto-regressive generation step.
- Bounding the denoising error in the KL-divergence decomposition.
- Bounding the initialization error in the KL-divergence decomposition.
- Concluding the proof.

Step 1: Decomposition of the concatenation of all the noisy and clean frames.

During each implementation of **Auto-Regressive Step**, we denoise the random vectors $\{Y_{i+j}^{t_j}\}_{j=1}^w$. We define all the already generated random vectors, including inputs and outputs of **Auto-Regressive Step**, by this implementation as

$$\mathcal{H}\left(\{Y_{i+j}^{t_j}\}_{j=1}^w\right) = \{X_{1:i}^0\} \cup \{X_{1:i+j-1}^{t_j}\}_{j \in \mathcal{S}^i} \text{ for } i \geq \Delta, \text{ and } \mathcal{H}\left(\{Y_{i+j}^{t_j}\}_{j=1}^w\right) = \emptyset \text{ for } i < \Delta, \quad (11)$$

where the set $\mathcal{S}^I \subseteq \mathcal{L}^I$ is the set of the starting indexes of each noise input noise level. It is defined as follows.

$$\mathcal{S}^I = \{s^I(t_j^I) \mid j \in [w]\}, \text{ and } s^I(t_i^I) = \min \{j \mid t_j^I = t_i^I, j \in [w]\},$$

where $s^I(t_i^I)$ is the smallest index whose input noise level is equal to t_i^I . The examples of this definition of $\mathcal{H}(\cdot)$ are provided in Figure 12. After each implementation of **Auto-Regressive Step**, we output $\{Y_{i+j}^{t_j^O}\}_{j=1}^w$. We define all the generated random vectors, including inputs and outputs of **Auto-Regressive Step**, *after this implementation* as

$$\mathcal{G}\left(\{Y_{i+j}^{t_j^O}\}_{j=1}^w\right) = \mathcal{H}\left(\{Y_{i+j}^{t_j^I}\}_{j=1}^w\right) \cup \{Y_{i+j}^{t_j^I}\}_{j=1}^w \cup \{Y_{i+j}^{t_j^O}\}_{j=1}^w. \quad (12)$$

With these definitions, we can prove the following relationship between them.

Proposition 4 *For any $i \geq \Delta$, we have that*

$$\mathcal{G}\left(\{Y_{i-\Delta+j}^{t_j^O}\}_{j=1}^w\right) = \mathcal{H}\left(\{Y_{i+j}^{t_j^I}\}_{j=1}^w\right) \cup \{Y_{i+j}^{t_j^I}\}_{j=1}^{w-\Delta}.$$

The proof of this proposition is provided in Appendix E.2.

Step 2: Decomposition of the KL-divergence of all the frames into the KL-divergence of each auto-regressive generation step.

Then we would like to decompose the upper bound of the KL-divergence between videos. In fact, we have that

$$\begin{aligned} & \text{KL}(X_{1:K\Delta}^0 \parallel Y_{1:K\Delta}^0) \\ & \leq \text{KL}\left(\mathcal{G}\left(\{X_{(K-1)\Delta+j}^{t_j^O}\}_{j=1}^w\right) \parallel \mathcal{G}\left(\{Y_{(K-1)\Delta+j}^{t_j^O}\}_{j=1}^w\right)\right) \\ & = \text{KL}\left(\mathcal{G}\left(\{X_{0+j}^{t_j^O}\}_{j=1}^w\right) \parallel \mathcal{G}\left(\{Y_{0+j}^{t_j^O}\}_{j=1}^w\right)\right) \\ & \quad + \sum_{k=1}^{K-2} \text{KL}\left(\{X_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w \mid \mathcal{G}\left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w\right) \parallel \{Y_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w \mid \mathcal{G}\left(\{Y_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w\right)\right) \\ & \quad + \sum_{k=1}^{K-2} \text{KL}\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w \mid \mathcal{G}\left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{X_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w \right. \\ & \quad \left. \parallel \{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w \mid \mathcal{G}\left(\{Y_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w\right) \\ & = \text{(I)} + \text{(II)} + \text{(III)}, \end{aligned} \quad (13)$$

where the definition of the KL-divergence between conditional distributions is provided in Appendix A, the inequality results from the data processing inequality, and the first equality results from the chain-rule of KL-divergence and Proposition 4. To see this, we have that

$$\begin{aligned} & \mathcal{G}\left(\{Y_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w \cup \{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w \\ & = \mathcal{H}\left(\{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=1}^{w-\Delta} \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w \cup \{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w \end{aligned}$$

$$\begin{aligned}
 &= \mathcal{H}\left(\{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=1}^w \cup \{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w \\
 &= \mathcal{G}\left(\{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w\right),
 \end{aligned}$$

where the first equality results from Proposition 4, the second equality results from combining the middle two terms, and the last equality results from the definition of \mathcal{G} in Eqn. (12). We note that in the right-hand side of Eqn. (13), the term (I) corresponds to the error from the initialization stage of Algorithm 1, the term (II) corresponds to the error from the noise initialization (Line 1 of Algorithm 3), and the term (III) corresponds to the error from the denoising step (Line 3 of Algorithm 3). We would like to further decompose the term (II). We first transform each term in (II) as follows.

$$\begin{aligned}
 &\text{KL}\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w \middle| \mathcal{G}\left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{X_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w\right. \\
 &\quad \left. \middle| \{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w \middle| \mathcal{G}\left(\{Y_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w\right) \\
 &= \text{KL}\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w \middle| \mathcal{H}\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{X_{k\Delta+j}^{t_j^I}\}_{j=1}^w\right. \\
 &\quad \left. \middle| \{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w \middle| \mathcal{H}\left(\{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=1}^w\right), \quad (14)
 \end{aligned}$$

where the equality follows from Proposition 4. From the denoising step in Eqn. (6), we have the following Markov chain.

$$\mathcal{H}\left(\{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w\right) \setminus \{Y_{\mathcal{R}_{k\Delta+1}^0}\} \text{---} \{Y_{\mathcal{R}_{k\Delta+1}^0}\} \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=1}^w \text{---} \{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w. \quad (15)$$

Thus, we have that

$$\begin{aligned}
 &\text{KL}\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w \middle| \mathcal{H}\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{X_{k\Delta+j}^{t_j^I}\}_{j=1}^w\right. \\
 &\quad \left. \middle| \{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w \middle| \mathcal{H}\left(\{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=1}^w\right) \\
 &= \text{KL}\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w \middle| \mathcal{H}\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w\right) \cup \{X_{k\Delta+j}^{t_j^I}\}_{j=1}^w\right. \\
 &\quad \left. \middle| \{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w \middle| \{Y_{\mathcal{R}_{k\Delta+1}^0}\} \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=1}^w\right) \\
 &= I\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w; \mathcal{H}\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w\right) \setminus \{X_{\mathcal{R}_{k\Delta+1}^0}\} \middle| \{X_{\mathcal{R}_{k\Delta+1}^0}\} \cup \{X_{k\Delta+j}^{t_j^I}\}_{j=1}^w\right) \\
 &\quad + \text{KL}\left(\{X_{k\Delta+j}^{t_j^O}\}_{j=1}^w \middle| \{X_{\mathcal{R}_{k\Delta+1}^0}\} \cup \{X_{k\Delta+j}^{t_j^I}\}_{j=1}^w \middle| \{Y_{k\Delta+j}^{t_j^O}\}_{j=1}^w \middle| \{Y_{\mathcal{R}_{k\Delta+1}^0}\} \cup \{Y_{k\Delta+j}^{t_j^I}\}_{j=1}^w\right), \quad (16)
 \end{aligned}$$

where the first equality results from the Markov chain in (15), and the second equality results from the definition of the conditional mutual information.

Step 3: Bounding the denoising error in the KL-divergence decomposition.

Eqn. (14) and (16) show that to bound the denoising error term (III) in Eqn. (13), we only need to bound the second term in the right-hand side of Eqn. (16). We first define the multiple noise-level version of Eqn. (5) and apply the change of variable $t \rightarrow T - t$ as follows.

$$d\tilde{X}_{1:w}^{\bar{t}} = \left(\frac{1}{2} \tilde{X}_{1:w}^{\bar{t}} + \nabla \log P_X^{\bar{T}-\bar{t}}(\tilde{X}_{1:w}^{\bar{t}} | X_{\mathcal{R}}^0) \right) dt + dB_{1:w}^{\bar{t}} \text{ for } T - t_1^I \leq t \leq T - t_1^O. \quad (17)$$

Recall that we define the joint time \bar{t} as $\bar{t}(t) = (t, t + t_2^I - t_1^I, \dots, t + t_w^I - t_1^I) = (t_1, \dots, t_w)$. We define the probability of $\tilde{X}_{1:w}^{\bar{t}}$ as $\tilde{P}_X^{\bar{t}}$. Then we have that $\tilde{P}_X^{\bar{t}} = P_X^{\bar{T}-\bar{t}}$. We also applied the change of variable to Eqn. (6).

$$d\tilde{Y}_{1:w}^{\bar{t}} = \left(\frac{1}{2} \tilde{Y}_{1:w}^{\bar{t}_{n}^{\text{ar}}} + s_{\theta}(\tilde{Y}_{1:w}^{\bar{t}_{n}^{\text{ar}}}, \bar{T} - \bar{t}_{n}^{\text{ar}}, Y_{\mathcal{R}}^0) \right) dt + dB_{1:w}^{\bar{t}} \text{ for } \tilde{t}_n^{\text{ar}} \leq t \leq \tilde{t}_{n+1}^{\text{ar}}. \quad (18)$$

In the following proof of this step, we will omit the symbols $X_{\mathcal{R}}^0$ and $Y_{\mathcal{R}}^0$ for ease of notation, since all the proof in this step is conditioned on them. We define the probability of $\tilde{Y}_{1:w}^{\bar{t}}$ as $\tilde{P}_Y^{\bar{t}}$ and have that $\tilde{P}_Y^{\bar{t}} = P_Y^{\bar{T}-\bar{t}}$. For notation simplicity, we define $\bar{t}^I = \bar{T} - \mathcal{L}^O$ and $\bar{t}^O = \bar{T} - \mathcal{L}^I$. Conditioned on the value at \bar{t}' , the conditional distribution of $\tilde{X}_{1:w}^{\bar{t}}$ and $\tilde{Y}_{1:w}^{\bar{t}}$ are denoted as $\tilde{P}_X^{\bar{t}|\bar{t}'}$ and $\tilde{P}_Y^{\bar{t}|\bar{t}'}$, respectively. For any $\tilde{t}_n^{\text{ar}} \leq t \leq \tilde{t}_{n+1}^{\text{ar}}$, Lemma 7 shows that

$$\begin{aligned} & \frac{d}{dt} \text{KL}(\tilde{P}_X^{\bar{t}|\bar{t}'}(\cdot|a) \| \tilde{P}_Y^{\bar{t}|\bar{t}'}(\cdot|a)) \\ &= -\mathbb{E}_{\tilde{P}_X^{\bar{t}|\bar{t}'}(\cdot|a) \cdot \tilde{P}_Y^{\bar{t}|\bar{t}'}(a)} \left[\left\| \nabla \log \frac{\tilde{P}_X^{\bar{t}|\bar{t}'}(\tilde{X}_{1:w}^{\bar{t}}|a)}{\tilde{P}_Y^{\bar{t}|\bar{t}'}(\tilde{X}_{1:w}^{\bar{t}}|a)} \right\|^2 \right] \\ & \quad + \mathbb{E}_{\tilde{P}_X^{\bar{t}|\bar{t}'}(\cdot|a) \cdot \tilde{P}_Y^{\bar{t}|\bar{t}'}(a)} \left[\left\langle \nabla \log P_X^{\bar{T}-\bar{t}}(\tilde{X}_{1:w}^{\bar{t}} | X_{\mathcal{R}}^0) - s_{\theta}(\tilde{X}_{1:w}^{\bar{t}_{n}^{\text{ar}}}, \bar{T} - \bar{t}_{n}^{\text{ar}}, X_{\mathcal{R}}^0), \nabla \log \frac{\tilde{P}_X^{\bar{t}|\bar{t}'}(\tilde{X}_{1:w}^{\bar{t}}|a)}{\tilde{P}_Y^{\bar{t}|\bar{t}'}(\tilde{X}_{1:w}^{\bar{t}}|a)} \right\rangle \right] \\ & \leq \frac{1}{2} \mathbb{E}_{\tilde{P}_X^{\bar{t}|\bar{t}'}(\cdot|a) \cdot \tilde{P}_Y^{\bar{t}|\bar{t}'}(a)} \left[\left\| \nabla \log P_X^{\bar{T}-\bar{t}}(\tilde{X}_{1:w}^{\bar{t}} | X_{\mathcal{R}}^0) - s_{\theta}(\tilde{X}_{1:w}^{\bar{t}_{n}^{\text{ar}}}, \bar{T} - \bar{t}_{n}^{\text{ar}}, X_{\mathcal{R}}^0) \right\|^2 \right], \end{aligned}$$

where the inequality results from Cauchy inequality. Then we have that

$$\begin{aligned} & \text{KL}(\tilde{P}_X^{\bar{t}^O|\bar{t}^I}(\cdot|a) \| \tilde{P}_Y^{\bar{t}^O|\bar{t}^I}(\cdot|a)) \\ &= \text{KL}(\tilde{P}_X^{\bar{t}^O|\bar{t}^I}(\cdot|a) \| \tilde{P}_Y^{\bar{t}^O|\bar{t}^I}(\cdot|a)) - \text{KL}(\tilde{P}_X^{\bar{t}^I|\bar{t}^I}(\cdot|a) \| \tilde{P}_Y^{\bar{t}^I|\bar{t}^I}(\cdot|a)) \\ & \leq \frac{1}{2} \sum_{n=1}^{M_{\text{ar}}} \int_{\tilde{t}_{n-1}^{\text{ar}}}^{\tilde{t}_n^{\text{ar}}} \mathbb{E}_{\tilde{P}_X^{\bar{t}|\bar{t}^I}(\cdot|a) \cdot \tilde{P}_Y^{\bar{t}|\bar{t}^I}(a)} \left[\left\| \nabla \log P_X^{\bar{T}-\bar{t}}(\tilde{X}_{1:w}^{\bar{t}} | X_{\mathcal{R}}^0) - s_{\theta}(\tilde{X}_{1:w}^{\bar{t}}, \bar{T} - \bar{t}_{n}^{\text{ar}}, X_{\mathcal{R}}^0) \right\|^2 \right] dt. \end{aligned}$$

Thus, for the second term on the right-hand side of Eqn. (16), we have that

$$\begin{aligned} & \text{KL} \left(\left\{ X_{k\Delta+j}^{t_j^O} \right\}_{j=1}^w \middle| \left\{ X_{\mathcal{R}_{k\Delta+1}}^{t_j^O} \right\} \cup \left\{ X_{k\Delta+j}^{t_j^I} \right\}_{j=1}^w \right\| \left\{ Y_{k\Delta+j}^{t_j^O} \right\}_{j=1}^w \middle| \left\{ Y_{\mathcal{R}_{k\Delta+1}}^{t_j^O} \right\} \cup \left\{ Y_{k\Delta+j}^{t_j^I} \right\}_{j=1}^w \right) \\ & \leq \frac{1}{2} \sum_{n=1}^{M_{\text{ar}}} \int_{\tilde{t}_{n-1}^{\text{ar}}}^{\tilde{t}_n^{\text{ar}}} \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}} | X_{\mathcal{R}_{k\Delta+1}}^0) - s_{\theta}(X_{k\Delta+1:k\Delta+w}^{\bar{t}_{n}^{\text{ar}}}, \bar{t}_{n}^{\text{ar}}, X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] dt. \end{aligned} \quad (19)$$

Then we would like to upper bound each term in the right-hand side of Eqn. (19). In fact, we have that

$$\begin{aligned}
 & \int_{t_{n-1}^{\text{ar}}}^{t_n^{\text{ar}}} \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}} | X_{\mathcal{R}_{k\Delta+1}}^0) - s_\theta(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}}, \bar{t}_n^{\text{ar}}, X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] dt \\
 & \leq 2 \int_{t_{n-1}^{\text{ar}}}^{t_n^{\text{ar}}} \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}} | X_{\mathcal{R}_{k\Delta+1}}^0) - \nabla \log P_X^{\bar{t}_n^{\text{ar}}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}} | X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] dt \\
 & \quad + 2 \int_{t_{n-1}^{\text{ar}}}^{t_n^{\text{ar}}} \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}_n^{\text{ar}}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}} | X_{\mathcal{R}_{k\Delta+1}}^0) - s_\theta(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}}, \bar{t}_n^{\text{ar}}, X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] dt \\
 & = 2 \int_{t_{n-1}^{\text{ar}}}^{t_n^{\text{ar}}} \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}} | X_{\mathcal{R}_{k\Delta+1}}^0) - \nabla \log P_X^{\bar{t}_n^{\text{ar}}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}} | X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] dt \\
 & \quad + 2(t_n^{\text{ar}} - t_{n-1}^{\text{ar}}) \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}_n^{\text{ar}}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}} | X_{\mathcal{R}_{k\Delta+1}}^0) - s_\theta(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}}, \bar{t}_n^{\text{ar}}, X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right], \tag{20}
 \end{aligned}$$

where the inequality results from the property of the ℓ_2 -norm. Then we use the following proposition to upper bound the first term in the right-hand side of Eqn. (20).

Proposition 5 *Let P^t denote the marginal distribution function of the following stochastic process at time t .*

$$dX^t = -\frac{1}{2}X^t dt + dB^t, \quad X^0 \sim P_X.$$

If $\nabla \log P^t$ is L -Lipschitz for $t \in [t_0, t_1]$ with $t_1 - t_0 \leq 1$. Then for any $t_0 \leq u < v \leq t_1$, we have that

$$\mathbb{E} \left[\left\| \nabla \log P^u(X^u) - \nabla \log P^v(X^v) \right\|^2 \right] \lesssim dL^2(v - u).$$

The proof is provided in Appendix E.4. This proposition implies that

$$\begin{aligned}
 & \int_{t_{n-1}^{\text{ar}}}^{t_n^{\text{ar}}} \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}} | X_{\mathcal{R}_{k\Delta+1}}^0) - \nabla \log P_X^{\bar{t}_n^{\text{ar}}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}} | X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] dt \\
 & \lesssim \int_{t_{n-1}^{\text{ar}}}^{t_n^{\text{ar}}} wdL^2(t_n^{\text{ar}} - t) dt \\
 & \lesssim wdL^2(t_n^{\text{ar}} - t_{n-1}^{\text{ar}})^2. \tag{21}
 \end{aligned}$$

Combining Eqn. (19), (20), (21), we have that

$$\begin{aligned}
 & \text{KL} \left(\left\{ X_{k\Delta+j}^{t_j^{\text{O}}} \right\}_{j=1}^w \middle| \left\{ X_{\mathcal{R}_{k\Delta+1}}^0 \right\} \cup \left\{ X_{k\Delta+j}^{t_j^{\text{I}}} \right\}_{j=1}^w \right\| \left\{ Y_{k\Delta+j}^{t_j^{\text{O}}} \right\}_{j=1}^w \middle| \left\{ Y_{\mathcal{R}_{k\Delta+1}}^0 \right\} \cup \left\{ Y_{k\Delta+j}^{t_j^{\text{I}}} \right\}_{j=1}^w \right) \\
 & \lesssim \sum_{n=1}^{M_{\text{ar}}} (t_n^{\text{ar}} - t_{n-1}^{\text{ar}}) \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}_n^{\text{ar}}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}} | X_{\mathcal{R}_{k\Delta+1}}^0) - s_\theta(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}}, \bar{t}_n^{\text{ar}}, X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] \\
 & \quad + wdL^2 \sum_{n=1}^{M_{\text{ar}}} (t_n^{\text{ar}} - t_{n-1}^{\text{ar}})^2. \tag{22}
 \end{aligned}$$

Step 4: Bounding the initialization error in the KL-divergence decomposition.

We note that there are two initialization error terms in Eqn. (13), i.e., (I) and (II). Here (I) arises from Line 1 of Algorithm 1, while (II) arises from Line 1 of Algorithm 3. We first derive the bound for term (II). In fact, we have that

$$\begin{aligned}
 & \text{KL} \left(\{X_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w \middle| \mathcal{G} \left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w \right) \right\| \{Y_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w \middle| \mathcal{G} \left(\{Y_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w \right) \right) \\
 &= \text{KL} \left(\{X_{k\Delta+j}^T\}_{j=w-\Delta+1}^w \middle| \mathcal{G} \left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w \right) \right\| \mathcal{N}(0, I) \right) \\
 &= \text{KL} \left(\{X_{k\Delta+j}^T\}_{j=w-\Delta+1}^w \middle| \mathcal{G} \left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w \right) \cup \{X_{k\Delta+j}^0\}_{j=w-\Delta+1}^w \right\| \mathcal{N}(0, I) \right) \\
 &\quad - \text{KL} \left(\{X_{k\Delta+j}^T\}_{j=w-\Delta+1}^w \middle| \mathcal{G} \left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w \right) \cup \{X_{k\Delta+j}^0\}_{j=w-\Delta+1}^w \right. \\
 &\quad \quad \left. \left\| \{X_{k\Delta+j}^T\}_{j=w-\Delta+1}^w \middle| \mathcal{G} \left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w \right) \right\| \right) \\
 &\leq \text{KL} \left(\{X_{k\Delta+j}^T\}_{j=w-\Delta+1}^w \middle| \mathcal{G} \left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w \right) \cup \{X_{k\Delta+j}^0\}_{j=w-\Delta+1}^w \right\| \mathcal{N}(0, I) \right) \\
 &= \text{KL} \left(\{X_{k\Delta+j}^T\}_{j=w-\Delta+1}^w \middle| \{X_{k\Delta+j}^0\}_{j=w-\Delta+1}^w \right\| \mathcal{N}(0, I) \right), \tag{23}
 \end{aligned}$$

where the first equality results from Line 1 of Algorithm 3, and the last equality results from the following Markov chain for the forward process of X_i^t

$$\{X_{k\Delta+j}^T\}_{j=w-\Delta+1}^w \rightarrow \{X_{k\Delta+j}^0\}_{j=w-\Delta+1}^w \rightarrow \mathcal{G} \left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w \right).$$

With Assumption 1 and Lemma 9, we have that

$$\text{KL} \left(\{X_{k\Delta+j}^T\}_{j=w-\Delta+1}^w \middle| \{X_{k\Delta+j}^0\}_{j=w-\Delta+1}^w \right\| \mathcal{N}(0, I) \right) \leq (\Delta d + B^2) \exp(-T). \tag{24}$$

Combining these inequalities, we have that

$$\begin{aligned}
 & \text{KL} \left(\{X_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w \middle| \mathcal{G} \left(\{X_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w \right) \right\| \{Y_{k\Delta+j}^{t_j^I}\}_{j=w-\Delta+1}^w \middle| \mathcal{G} \left(\{Y_{(k-1)\Delta+j}^{t_j^O}\}_{j=1}^w \right) \right) \\
 &\leq (d\Delta + B^2) \exp(-T). \tag{25}
 \end{aligned}$$

We then upper bound the term (I). From the procedures of Algorithm 2, we have that

$$\begin{aligned}
 & \text{KL} \left(\mathcal{G} \left(\{X_{0+j}^{t_j^O}\}_{j=1}^w \right) \right\| \mathcal{G} \left(\{Y_{0+j}^{t_j^O}\}_{j=1}^w \right) \right) \\
 &= \text{KL} \left(\{X_{i+j}^{t_j^I}\}_{j=1}^w \cup \{X_{i+j}^{t_j^O}\}_{j=1}^w \right\| \{Y_{i+j}^{t_j^I}\}_{j=1}^w \cup \{Y_{i+j}^{t_j^O}\}_{j=1}^w \right) \\
 &\leq \text{KL}(X_{1:i_0}^0 \parallel Y_{1:i_0}^0) \\
 &\leq \text{KL}(X_{1:i_0}^0 \parallel X_{1:i_0}^T \parallel Y_{1:i_0}^0 \parallel Y_{1:i_0}^T) + \text{KL}(X_{1:i_0}^T \parallel Y_{1:i_0}^T), \tag{26}
 \end{aligned}$$

where the equality follows from the definition of \mathcal{G} in Eqn. (12), these two inequalities follows from the data processing inequality. For the first term in the right-hand side of Eqn. (26), we follow the same procedures in **Step 2** and derive that

$$\begin{aligned} & \text{KL}(X_{1:i_0}^0 | X_{1:i_0}^T \| Y_{1:i_0}^0 | Y_{1:i_0}^T) \\ & \lesssim \sum_{n=1}^{M_{\text{ar}}} (t_n^{\text{ar}} - t_{n-1}^{\text{ar}}) \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}_n^{\text{ar}}} (X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}} | X_{\mathcal{R}_{k\Delta+1}}^0) - s_\theta(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}}, \bar{t}_n^{\text{ar}}, X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] \\ & \quad + wdL^2 \sum_{n=1}^{M_{\text{ar}}} (t_n^{\text{ar}} - t_{n-1}^{\text{ar}})^2. \end{aligned} \quad (27)$$

For the second term in the right-hand side of Eqn. (26), Assumption (1) and Lemma (9) show that

$$\text{KL}(X_{1:i_0}^T \| Y_{1:i_0}^T) \leq (d \cdot i_0 + B^2) \exp(-T). \quad (28)$$

Step 5: Concluding the proof.

Now we have derived all the components to upper bound the KL-divergence in Eqn. (13). Combining Eqn. (26), (27) and (28), we have that

$$\begin{aligned} \text{(I)} & \lesssim (d \cdot i_0 + B^2) \exp(-T) \\ & \quad + \sum_{n=1}^{M_{\text{init}}} (t_n^{\text{init}} - t_{n-1}^{\text{init}}) \mathbb{E} \left[\left\| \nabla \log P_X^{t_n^{\text{init}}} (X_{k\Delta+1:k\Delta+w}^{t_n^{\text{init}}} | X_{\mathcal{R}_{k\Delta+1}}^0) - s_\theta(X_{k\Delta+1:k\Delta+w}^{t_n^{\text{init}}}, t_n^{\text{init}}, X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] \\ & \quad + wdL^2 \sum_{n=1}^{M_{\text{init}}} (t_n^{\text{init}} - t_{n-1}^{\text{init}})^2. \end{aligned} \quad (29)$$

Combining Eqn. (23) and (24), we have that

$$\text{(II)} \leq K(\Delta d + B^2) \exp(-T). \quad (30)$$

Combining Eqn. (14), Eqn. (16), (22), we have that

$$\begin{aligned} \text{(III)} & \leq \sum_{k=1}^{K-2} I \left(\{X_{k\Delta+j}^{t_j^{\text{O}}}\}_{j=1}^w; \mathcal{H} \left(\{X_{k\Delta+j}^{t_j^{\text{O}}}\}_{j=1}^w \right) \setminus \{X_{\mathcal{R}_{k\Delta+1}}^0\} \middle| \{X_{\mathcal{R}_{k\Delta+1}}^0\} \cup \{X_{k\Delta+j}^{t_j^{\text{I}}}\}_{j=1}^w \right) \\ & \quad + \sum_{k=1}^{K-2} \sum_{n=1}^{M_{\text{ar}}} (t_n^{\text{ar}} - t_{n-1}^{\text{ar}}) \mathbb{E} \left[\left\| \nabla \log P_X^{\bar{t}_n^{\text{ar}}} (X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}} | X_{\mathcal{R}_{k\Delta+1}}^0) \right. \right. \\ & \quad \quad \left. \left. - s_\theta(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}}, \bar{t}_n^{\text{ar}}, X_{\mathcal{R}_{k\Delta+1}}^0) \right\|^2 \right] \\ & \quad + \sum_{k=1}^{K-2} wdL^2 \sum_{n=1}^{M_{\text{ar}}} (t_n^{\text{ar}} - t_{n-1}^{\text{ar}})^2. \end{aligned} \quad (31)$$

Combining Eqn. (13), (29), (30), and (31), we conclude the proof of Eqn. (7).

B.2 Proof of Eqn. (10)

The proof of Eqn. (10) largely shares the similar procedures with the proof of Eqn. (7). Thus, we will reuse some intermediate results in the proof of Eqn. (7). Here we consider the special case $\mathcal{R}_i = \emptyset$, and the results of the general case $\mathcal{R}_i \neq \emptyset$ can be derived by our analysis with complicated calculations.

The proof of Eqn. (10) consists of the following two steps.

- Derive the recursive formula of the output error.
- Conclude the proof by aggregating the recursive formula.

Step 1: Derive the recursive formula of the output error.

To derive the KL-divergence between a generated clip and the nominal video clip, we upper bound the error as follows.

$$\begin{aligned}
 & \text{KL}(X_{K\Delta+1:(K+1)\Delta}^0 \| Y_{K\Delta+1:(K+1)\Delta}^0) \\
 & \leq \text{KL}\left(\{X_{K\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{Y_{K\Delta+i}^{t_i^O}\}_{i=1}^w\right) \\
 & \leq \text{KL}\left(\{X_{K\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{X_{K\Delta+i}^{t_i^I}\}_{i=1}^w \middle\| \{Y_{K\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{Y_{K\Delta+i}^{t_i^I}\}_{i=1}^w\right) \\
 & \quad + \text{KL}\left(\{X_{K\Delta+i}^{t_i^I}\}_{i=1}^w \middle\| \{Y_{K\Delta+i}^{t_i^I}\}_{i=1}^w\right) \\
 & = \text{KL}\left(\{X_{K\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{X_{K\Delta+i}^{t_i^I}\}_{i=1}^w \middle\| \{Y_{K\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{Y_{K\Delta+i}^{t_i^I}\}_{i=1}^w\right) \\
 & \quad + \text{KL}\left(\{X_{K\Delta+i}^{t_i^I}\}_{i=1}^{w-\Delta} \cup \{X_{K\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w \middle\| \{Y_{K\Delta+i}^{t_i^I}\}_{i=1}^{w-\Delta} \cup \{Y_{K\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w\right) \\
 & = \text{KL}\left(\{X_{K\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{X_{K\Delta+i}^{t_i^I}\}_{i=1}^w \middle\| \{Y_{K\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{Y_{K\Delta+i}^{t_i^I}\}_{i=1}^w\right) \\
 & \quad + \text{KL}\left(\{X_{K\Delta+i}^{t_i^O}\}_{i=1}^{w-\Delta} \cup \{X_{K\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w \middle\| \{Y_{K\Delta+i}^{t_i^O}\}_{i=1}^{w-\Delta} \cup \{Y_{K\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w\right) \\
 & \leq \text{KL}\left(\{X_{K\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{X_{K\Delta+i}^{t_i^I}\}_{i=1}^w \middle\| \{Y_{K\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{Y_{K\Delta+i}^{t_i^I}\}_{i=1}^w\right) \\
 & \quad + \text{KL}\left(\{X_{(K-1)\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{Y_{(K-1)\Delta+i}^{t_i^O}\}_{i=1}^w\right) \\
 & \quad + \text{KL}\left(\{X_{K\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w \middle\| \{X_{K\Delta+i}^{t_{\Delta+i}^O}\}_{i=1}^{w-\Delta} \middle\| \{Y_{K\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w \middle\| \{Y_{K\Delta+i}^{t_{\Delta+i}^O}\}_{i=1}^{w-\Delta}\right), \quad (32)
 \end{aligned}$$

where the inequalities follows from the data processing inequality, and the second equality follows from the Circularity of Requirement 1. To derive the upper bound of the right-hand side of Eqn. (32), we can summing up the following recursive equation implied by Eqn. (32). In fact, we have that

$$\begin{aligned}
 & \text{KL}\left(\{X_{k\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{Y_{k\Delta+i}^{t_i^O}\}_{i=1}^w\right) \\
 & \leq \text{KL}\left(\{X_{k\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{X_{k\Delta+i}^{t_i^I}\}_{i=1}^w \middle\| \{Y_{k\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{Y_{k\Delta+i}^{t_i^I}\}_{i=1}^w\right) \\
 & \quad + \text{KL}\left(\{X_{(k-1)\Delta+i}^{t_i^O}\}_{i=1}^w \middle\| \{Y_{(k-1)\Delta+i}^{t_i^O}\}_{i=1}^w\right) \\
 & \quad + \text{KL}\left(\{X_{k\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w \middle\| \{X_{k\Delta+i}^{t_{\Delta+i}^O}\}_{i=1}^{w-\Delta} \middle\| \{Y_{k\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w \middle\| \{Y_{k\Delta+i}^{t_{\Delta+i}^O}\}_{i=1}^{w-\Delta}\right).
 \end{aligned}$$

This inequality indicates the recursive relationship of the error of the clips at different time steps.

Step 2: Conclude the proof by aggregating the recursive formula.

Summing the inequality about the recursive relationship from $k = 1$ to $k = K$, we have that

$$\begin{aligned}
 & \text{KL}\left(\{X_{K\Delta+i}^{t_i^O}\}_{i=1}^w \parallel \{Y_{K\Delta+i}^{t_i^O}\}_{i=1}^w\right) \\
 & \leq \text{KL}\left(\{X_i^{t_i^O}\}_{i=1}^w \parallel \{Y_i^{t_i^O}\}_{i=1}^w\right) \\
 & \quad + \sum_{k=1}^K \text{KL}\left(\{X_{k\Delta+i}^{t_i^O}\}_{i=1}^w \parallel \{X_{k\Delta+i}^{t_i^I}\}_{i=1}^w \parallel \{Y_{k\Delta+i}^{t_i^O}\}_{i=1}^w \parallel \{Y_{k\Delta+i}^{t_i^I}\}_{i=1}^w\right) \\
 & \quad + \sum_{k=1}^K \text{KL}\left(\{X_{k\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w \parallel \{X_{k\Delta+i}^{t_{\Delta+i}^O}\}_{i=1}^{w-\Delta} \parallel \{Y_{k\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w \parallel \{Y_{k\Delta+i}^{t_{\Delta+i}^O}\}_{i=1}^{w-\Delta}\right). \quad (33)
 \end{aligned}$$

Then we upper bound the second and third terms in the right-hand side of Eqn. (33). Similar to Eqn. (31), we have that

$$\begin{aligned}
 & \text{KL}\left(\{X_{k\Delta+i}^{t_i^O}\}_{i=1}^w \parallel \{X_{k\Delta+i}^{t_i^I}\}_{i=1}^w \parallel \{Y_{k\Delta+i}^{t_i^O}\}_{i=1}^w \parallel \{Y_{k\Delta+i}^{t_i^I}\}_{i=1}^w\right) \\
 & \leq \sum_{n=1}^{M_{\text{ar}}}(t_n^{\text{ar}} - t_{n-1}^{\text{ar}}) \mathbb{E}\left[\left\|\nabla \log P_X^{\bar{t}_n^{\text{ar}}}(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}} \mid X_{\mathcal{R}_{k\Delta+1}}^0) - s_\theta(X_{k\Delta+1:k\Delta+w}^{\bar{t}_n^{\text{ar}}}, \bar{t}_n^{\text{ar}}, X_{\mathcal{R}_{k\Delta+1}}^0)\right\|^2\right] \\
 & \quad + wdL^2 \sum_{n=1}^{M_{\text{ar}}}(t_n^{\text{ar}} - t_{n-1}^{\text{ar}})^2 \\
 & \text{KL}\left(\{X_{k\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w \parallel \{X_{k\Delta+i}^{t_{\Delta+i}^O}\}_{i=1}^{w-\Delta} \parallel \{Y_{k\Delta+i}^{t_i^I}\}_{i=w-\Delta+1}^w \parallel \{Y_{k\Delta+i}^{t_{\Delta+i}^O}\}_{i=1}^{w-\Delta}\right) \\
 & \leq (d \cdot i_0 + B^2) \exp(-T).
 \end{aligned}$$

Thus, we conclude the proof by combining these inequalities.

Appendix C. Proof of Theorem 2

In the following, we would like to prove

$$\inf_{\hat{P}} \sup_{P \in \mathcal{S}(s)} P\left(\text{TV}(P, \hat{P}) \geq \frac{s}{2}\right) \geq \frac{1}{2}.$$

Then Pinsker inequality implies that

$$\begin{aligned}
 \inf_{\hat{P}} \sup_{P \in \mathcal{S}(s)} P\left(\text{KL}(P \parallel \hat{P}) \geq \frac{s^2}{2}\right) & \geq \inf_{\hat{P}} \sup_{P \in \mathcal{S}(s)} P\left(2[\text{TV}(P, \hat{P})]^2 \geq \frac{s^2}{2}\right) \\
 & = \inf_{\hat{P}} \sup_{P \in \mathcal{S}(s)} P\left(\text{TV}(P, \hat{P}) \geq \frac{s}{2}\right) \\
 & \geq \frac{1}{2}.
 \end{aligned}$$

We would like to follow the procedures of the proof of impossibility results in the non-parametric statistics (Wainwright, 2019). We start the proof by constructing two distributions P_0 and P_1 as follows.

$$P_0(X = x, Y = y, Z = z) = \frac{1}{8} \text{ for all } x, y, z \in \{0, 1\}, \quad (34)$$

$$P_1(X = x, Y = y, Z = z) = P_1(X = x, Z = z)P_1(Y = y) = \frac{1}{2}P_1(X = x, Z = z)P_1(Y = y), \quad (35)$$

$$P_1(X = x, Z = 1 - x) = \frac{\epsilon}{2}, \quad P_1(X = x, Z = x) = \frac{1 - \epsilon}{2},$$

where $\epsilon \in [0, 1/2]$ is a hyperparameter. For P_0 , X, Y and Z are independent Bernoulli(1/2) random variables. For P_1 , Y is a independent Bernoulli(1/2) with X, Z . The marginal distributions of X and Z are Bernoulli(1/2), but these two variables are connected by a flip channel with flip probability ϵ . We first verify that these two distributions are in $\mathcal{S}(s)$. In fact, we have that

$$I_0(X; Z|Y) = 0, \quad I_1(X; Z|Y) = I_1(X; Z) = 1 - H(\epsilon),$$

where I_i is the mutual information with respect to the distribution P_i , and $H(x) = -x \log x - (1 - x) \log(1 - x)$ is the entropy of Bernoulli distribution. Obviously, $P_0 \in \mathcal{S}(s)$. We note that $H(x) \in [0, 1]$ and is monotone on $[0, 1/2]$. Thus, if $\epsilon \in [H^{-1}(1 - s), 1/2]$, $P_1 \in \mathcal{S}(s)$. In the following, we set $\epsilon = H^{-1}(1 - s)$, then $I_1(X; Z|Y) = s$. A property of the constructed two distributions is that

$$P_0(X, Y) = P_1(X, Y), \quad P_0(Y, Z) = P_1(Y, Z). \quad (36)$$

We also show that the total variation between them can be bounded as follows.

Proposition 6 *For the distributions defined in Eqn. (34) and (35) with $\epsilon \in [0, 1/2]$, we have that*

$$\text{TV}(P_0, P_1) \geq \frac{1}{2} \text{KL}(P_1 \parallel P_0) = \frac{1}{2} I_1(X; Z|Y).$$

The proof is provided in Appendix E.3. Then we consider the estimate \hat{P} based on the data generated by these two distributions. For any \hat{P} , we define a classifier ψ that determines the data set is generated by P_0 or P_1 as

$$\psi(\hat{P}) = \underset{i \in \{0, 1\}}{\text{argmin}} \text{TV}(\hat{P}, P_i).$$

If $\psi(\hat{P}) \neq i$ for $i \in \{0, 1\}$, then the triangle inequality shows that

$$\text{TV}(\hat{P}, P_i) \geq \frac{1}{2} I_1(X; Z|Y) = \frac{s}{2}.$$

Thus, we have that

$$\inf_{\hat{P}} \sup_{P \in \mathcal{S}(s)} P \left(\text{TV}(\hat{P}, P) \geq \frac{s}{2} \right) \geq \inf_{\hat{P}} \sup_{i \in \{0, 1\}} P_i \left(\text{TV}(\hat{P}, P_i) \geq \frac{s}{2} \right) \geq \inf_{\hat{P}} \sup_{i \in \{0, 1\}} P_i(\psi(\hat{P}) \neq i). \quad (37)$$

We denote the data distribution as $P_i^{\mathcal{D}}$, i.e.,

$$\begin{aligned} P_i^{\mathcal{D}}(\{X_i = x_i, Y_i = y_i\}_{i=1}^N \cap \{Y_i = y_i, Z_i = z_i\}_{i=N+1}^{2N}) \\ = \prod_{i=1}^N P_i(X_i = x_i, Y_i = y_i) \cdot \prod_{i=N+1}^{2N} P_i(Y_i = y_i, Z_i = z_i). \end{aligned}$$

Then Eqn. (36) implies that $P_0^{\mathcal{D}} = P_1^{\mathcal{D}}$. Neyman-Pearson Theorem shows that any test ψ has the following error probability.

$$P_1(\psi = 0) + P_0(\psi = 1) \geq \int \min(dP_0^{\mathcal{D}}, dP_1^{\mathcal{D}}) = 1.$$

Then we can further lower bound Eqn. (37) as

$$\inf_{\hat{P}} \sup_{i \in \{0,1\}} P_i(\psi(\hat{P}) \neq i) \geq \inf_{\psi} \sup_{i \in \{0,1\}} P_i(\psi \neq i) \geq \frac{1}{2}.$$

Thus, we conclude the proof of Theorem 2.

Appendix D. Supporting Lemmas

Lemma 7 (Lemma 6 in Chen et al. (2023a)) *Consider the following two Ito processes*

$$\begin{aligned} dX^t &= F_1(X^t, t)dt + g(t)dB^t, & X^0 &= a \\ dY^t &= F_2(Y^t, t)dt + g(t)dB^t, & Y^0 &= a, \end{aligned}$$

where F_1 , F_2 , and g are continuous functions and may depend on a . We assume the uniqueness and regularity condition:

- These two Stochastic Differential Equation (SDE)s have unique solutions.
- The processes X^t and Y^t admit densities $p^t, q^t \in C^2(\mathbb{R}^d)$ for $t > 0$.

Define the relative Fisher information between p^t and q^t by

$$J(p^t \| q^t) = \int p^t(x) \left\| \nabla \log \frac{p^t(x)}{q^t(x)} \right\|^2 dx.$$

Then for any $t > 0$, the evolution of $\text{KL}(p^t \| q^t)$ is given by

$$\frac{d}{dt} \text{KL}(p^t \| q^t) = -g(t)^2 J(p^t \| q^t) + \mathbb{E}_{p^t} \left[\left\langle F_1(X^t, t) - F_2(X^t, t), \nabla \log \frac{p^t(X^t)}{q^t(X^t)} \right\rangle \right].$$

Lemma 8 (Theorem 7 in Verdú (2014)) *For two distributions $P, Q \in \mathcal{P}(\Omega)$, we define*

$$\beta_1^{-1} = \sup_{w \in \Omega} \frac{dP}{dQ}(w).$$

Then we have that

$$\text{TV}(P, Q) \geq \frac{1 - \beta_1}{\log_2 1/\beta_1} \text{KL}(P \| Q).$$

Lemma 9 (Lemmas 9 and 11 in Chen et al. (2023a)) For any distribution P_X on \mathbb{R}^d that has finite second moment, i.e., $\mathbb{E}_{P_X}\|X\|^2 < \infty$, and X^t evolves as

$$dX^t = -\frac{1}{2}X^t dt + dB^t, \quad X^0 \sim P_X,$$

where B^t is a Brownian motion. We denote the distribution of X^t as P^t . Then we have that

$$\text{KL}(P^T \|\mathcal{N}(0, I)) \leq (d + \mathbb{E}_{P_X}\|X\|^2) \cdot \exp(-T).$$

For any $0 \leq t \leq s \leq T$, we define $\alpha_{t,s} = \exp(-\frac{1}{2}(s-t))$. Then we have that

$$\begin{aligned} & \mathbb{E}\left[\|\nabla \log P^t(X^t) - \nabla \log P^s(X^s)\|^2\right] \\ & \leq 4\mathbb{E}\left[\|\nabla \log P^t(X^t) - \nabla \log P^t(\alpha_{t,s}^{-1}X^s)\|^2\right] + 2(1 - \alpha_{t,s}^{-1})^2\mathbb{E}\left[\|\nabla \log P^t(X^t)\|^2\right]. \end{aligned}$$

Lemma 10 (Chewi et al. (2024)) Let $P \in C^1(\mathbb{R}^d)$ be a probability distribution. If $\nabla \log P$ is L -Lipschitz, then we have that

$$\mathbb{E}_P\left[\|\nabla \log P(X)\|^2\right] \leq d \cdot L.$$

Appendix E. Proof of Supporting Propositions

E.1 Proof of Proposition 3

We first decompose $I(X, f(X), g(X); Z|Y)$ in two ways as follows.

$$\begin{aligned} I(X, f(X), g(X); Z|Y) &= I(f(X); Z|Y) + I(X; Z|Y, f(X)) + I(g(X); Z|Y, f(X), X) \\ &= I(f(X); Z|Y) + I(X; Z|Y, f(X)), \end{aligned}$$

where the first equality results from the chain rule, and the second equality results from the fact that $I(g(X); Z|Y, f(X), X) = 0$. Similarly, we have that

$$I(X, f(X), g(X); Z|Y) = I(g(X); Z|Y) + I(X; Z|Y, g(X)).$$

Thus, we have that

$$I(X; Z|Y, g(X)) - I(X; Z|Y, f(X)) = I(f(X); Z|Y) - I(g(X); Z|Y).$$

In the following, we will show that the right-hand side of this equation is non-negative. In fact, the chain rule shows that

$$I(g(X), f(X); Z|Y) = I(g(X); Z|Y) + I(f(X); Z|Y, g(X)).$$

Similarly, we have that

$$I(g(X), f(X); Z|Y) = I(f(X); Z|Y) + I(g(X); Z|Y, f(X)) = I(f(X); Z|Y),$$

where the last equality results from that $g(x) = h(f(x))$ for all x . Thus, we have that

$$I(f(X); Z|Y) - I(g(X); Z|Y) = I(f(X); Z|Y, g(X)) \geq 0.$$

We conclude the proof of Proposition 3.

E.2 Proof of Proposition 4

We begin the proof by showing the recursive relationship for \mathcal{H} . We note that

$$\mathcal{H}\left(\{Y_{i+j}^{t_j^I}\}_{j=1}^w\right) = \mathcal{H}\left(\{Y_{i-\Delta+j}^{t_j^I}\}_{j=1}^w\right) \cup \{Y_{i-\Delta+1:i}^0\} \cup \{Y_{i-\Delta+j}^{t_j^I}\}_{j=1}^w, \quad (38)$$

the equality follows from the definition of $\mathcal{H}(\cdot)$ in Eqn. (11). Then we have

$$\begin{aligned} & \mathcal{G}\left(\{Y_{i-\Delta+j}^{t_j^O}\}_{j=1}^w\right) \\ &= \mathcal{H}\left(\{Y_{i-\Delta+j}^{t_j^I}\}_{j=1}^w\right) \cup \{Y_{i-\Delta+j}^{t_j^I}\}_{j=1}^w \cup \{Y_{i-\Delta+j}^{t_j^O}\}_{j=1}^w \\ &= \mathcal{H}\left(\{Y_{i-\Delta+j}^{t_j^I}\}_{j=1}^w\right) \cup \{Y_{i-\Delta+j}^{t_j^I}\}_{j=1}^w \cup \{Y_{i-\Delta+j}^{t_j^O}\}_{j=1}^\Delta \cup \{Y_{i-\Delta+j}^{t_j^O}\}_{j=\Delta+1}^w \\ &= \mathcal{H}\left(\{Y_{i-\Delta+j}^{t_j^I}\}_{j=1}^w\right) \cup \{Y_{i-\Delta+j}^{t_j^I}\}_{j=1}^w \cup \{Y_{i-\Delta+j}^0\}_{j=1}^\Delta \cup \{Y_{i+j}^{t_j^I}\}_{j=1}^{w-\Delta} \\ &= \mathcal{H}\left(\{Y_{i+j}^{t_j^I}\}_{j=1}^w\right) \cup \{Y_{i+j}^{t_j^I}\}_{j=1}^{w-\Delta}, \end{aligned}$$

where the first equality follows from the definition of \mathcal{G} in Eqn. (12), the third equality follows from $0 - T$ boundary and circularity in Requirement 1, and the last equality follows from Eqn. (38). Thus, we conclude the proof of Proposition 4.

E.3 Proof of Proposition 6

First, we note that the KL-divergence between these two distributions is

$$\text{KL}(P_1 \| P_0) = I_1(X; Z|Y).$$

Lemma 8 shows that

$$\text{TV}(P_0, P_1) \geq \frac{1 - \beta_1}{\log_2 1/\beta_1} \text{KL}(P_1 \| P_0),$$

where $\beta_1^{-1} = 2(1 - \epsilon)$. To prove the desired result, it remains to show that

$$\frac{1 - \beta_1}{\log_2 1/\beta_1} = \frac{1 - 2\epsilon}{2(1 - \epsilon) \log_2 (2(1 - \epsilon))} \geq \frac{1}{2}.$$

Then we define the function

$$f(\epsilon) = 1 - 2\epsilon - (1 - \epsilon) \log_2 (2(1 - \epsilon)).$$

The derivative of this function is that

$$f'(\epsilon) = -1 + \frac{1}{\log 2} - \frac{\log(1 - \epsilon)}{\log 2},$$

which is a monotone function. Since $f'(0) > 0 > f'(1/2)$, we have that

$$\inf_{x \in [0, 1/2]} f(x) = \min\{f(0), f(1/2)\} \geq 0.$$

Thus, we conclude the proof of Proposition 6.

E.4 Proof of Proposition 5

For the difference between the scores to bound, we have that

$$\begin{aligned}
 & \mathbb{E} \left[\left\| \nabla \log P^u(X^u) - \nabla \log P^v(X^v) \right\|^2 \right] \\
 & \leq 4\mathbb{E} \left[\left\| \nabla \log P^u(X^u) - \nabla \log P^u(\alpha_{u,v}^{-1}X^v) \right\|^2 \right] + 2(1 - \alpha_{u,v}^{-1})^2 \mathbb{E} \left[\left\| \nabla \log P^u(X^u) \right\|^2 \right] \\
 & \lesssim L^2 \cdot \mathbb{E} \left[\left\| X^u - \alpha_{u,v}^{-1}X^v \right\|^2 \right] + d \cdot L(1 - \alpha_{u,v}^{-1})^2 \\
 & \lesssim d \cdot L^2 (\exp(v - u) - 1) + dL(v - u)^2 \\
 & \lesssim dL^2(v - u),
 \end{aligned}$$

where the first inequality results from Lemma 9, the second inequality results from the Lipschitzness and Lemma 10, the third inequality results from the definition of X^t , and the last inequality results from that $v - u \leq 1$.

Appendix F. Other Methods Covered By Meta-ARVDM

We list some methods that are special cases of our framework here to demonstrate the universality of Meta-ARVDM. The methods we list here are a strict subset of all the methods included by our framework.

- ART-V (Weng et al., 2024): It denoises under same noise-level, i.e., $t_i^O = 0, t_i^I = T$ for all $i \in [w]$. The reference frames are an anchor frame and the last two, i.e., $\mathcal{R}(i) = \{0, i - 1, i - 2\}$.
- GameNGen (Valevski et al., 2024): It repurposes T2I model in AR style to generate videos, i.e., $t_1^O = 0, t_1^I = T$ and $w = 1$. It conditions on previous 64 frames, i.e., $\mathcal{R}(i) = \{i - 64, \dots, i - 1\}$.
- StreamingT2V (Henschel et al., 2024): It denoises 16 frames with the same noise level, i.e., $t_i^O = 0, t_i^I = T$ for all $i \in [16]$. The reference frames are anchor frames and the last 8 frames, i.e., $\mathcal{R}(i) = [N_a] \cup \{i - 8, \dots, i - 1\}$. Here we use N_a anchor frames.

Appendix G. Discussion of Noise Initialization Errors, Score Estimation Errors, and Discretization Errors in Theorem 1

The *noise initialization error* originates from the fact that the random variables $X_{1:w}^T$ and $X_{w-\Delta+1:w}^T$ are not exactly Gaussian. The difference between them and the Gaussian random variables contributes to this error, which decreases exponentially with increasing T . The *score estimation error* originates from the difference between the true score function and the estimate from the network. The concrete rate of this term is not relevant to the main topic of our work, which is considered in Chen et al. (2023b). Intuitively, this term should decrease with increasing number of training data points. The *discretization error* originates from the fact that we discretize the trajectory with the Euler-Maruyama scheme. We can see that this term decreases with increasing discretization steps M_{init} and M_{ar} . If we adopt uniform discretization, the discretization error will scale as $O(M_{\text{init}}^{-1})$ or $O(M_{\text{ar}}^{-1})$. This error

term can be improved with more advanced numerical methods, such as DPM-solver (Lu et al., 2022).

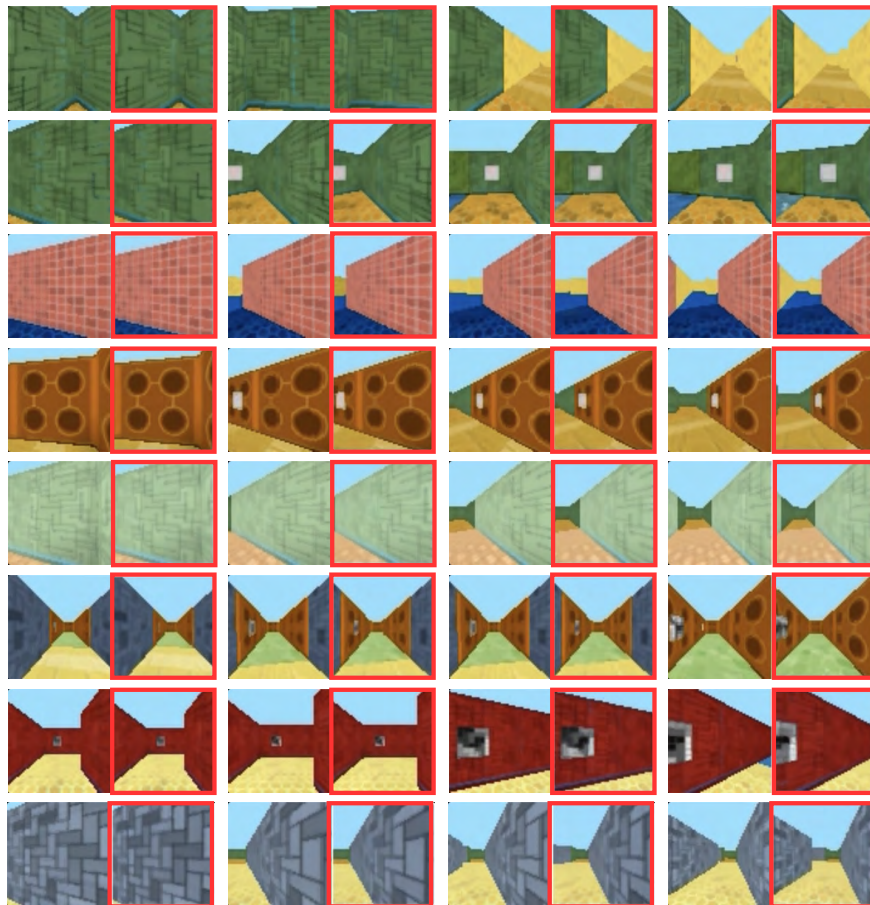
Appendix H. Demonstration of Data Model for Lower Bounds



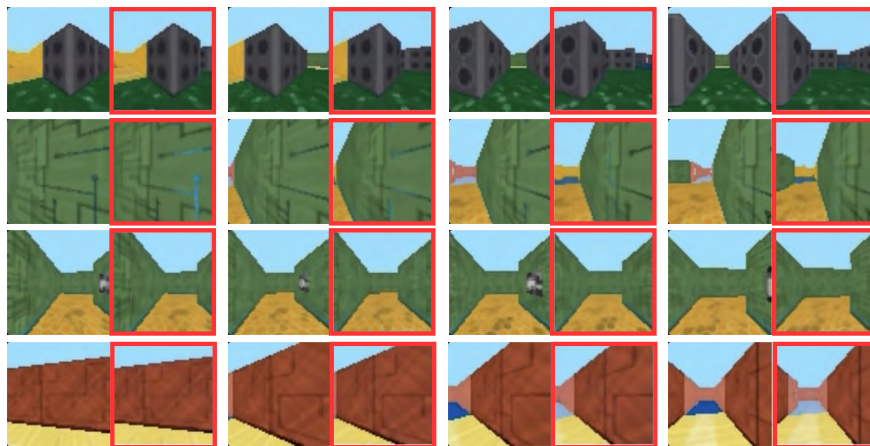
Figure 13: The simplified setting for the proof of lower bounds. We adopt three random variables X, Y, Z to represent **Past**, **Input**, and **Output**, respectively.

Appendix I. Additional Demonstration Figures

This section provides detailed visual demonstrations of successful and failed retrieval cases in both DMLab and Minecraft environments.



(a) Successful Retrieval on DMLab

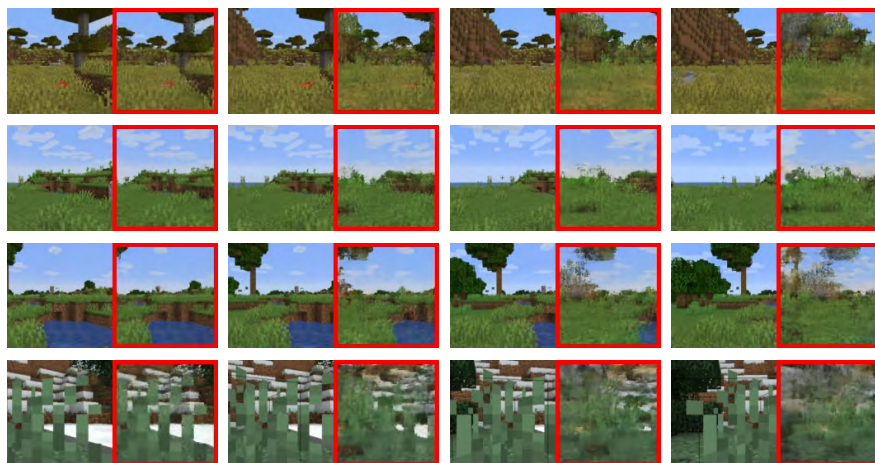


(b) Failed Retrieval on DMLab

Figure 14: Recall demonstrations on DMLab. The left frames represent the expected ground truth, while the right frames, outlined with a red square, are generated by the model.



(a) Successful Retrieval on Minecraft

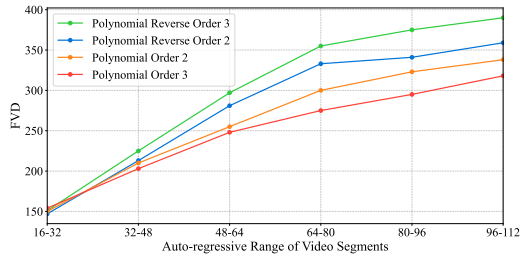


(b) Failed Retrieval on Minecraft

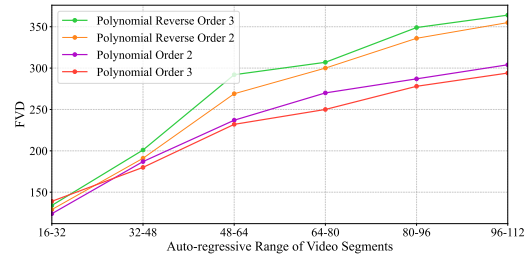
Figure 15: Recall demonstrations on Minecraft. The left frames represent the expected ground truth, while the right frames, outlined with a red square, are generated by the model. The first 4 frames without red squares are provided context.

Appendix J. Additional Experimental Results

This appendix contains additional experimental results for DMLab and Minecraft environments.



(a) DMLab



(b) Minecraft

Figure 16: Comparison of FVD across different timestep sets in DMLab and Minecraft. Polynomial Order is $(*, 1)$, while Polynomial Reverse Order is $(*, 2)$.