FIFO-Diffusion: Generating Infinite Videos from Text without Training

Jihwan Kim[*1] **Junoh Kang**[*1] **Jinyoung Choi**[1] **Bohyung Han**[1,2]
Computer Vision Laboratory, [1]ECE & [2]IPAI, Seoul National University
{kjh26720,junoh.kang, jin0.choi, bhhan}@snu.ac.kr

(a) "A spectacular fireworks display over Sydney Harbour, 4K, high resolution."

(b) "An astronaut walking on the moon's surface, high-quality, 4K resolution."

(c) "A colony of penguins waddling on an Antarctic ice sheet, 4K, ultra HD."

Figure 1: Illustration of 10K-frame long videos generated by FIFO-Diffusion based on a pretrained text-conditional video generation model, VideoCrafter2 [3]. The number at the top-left corner of each image indicates the frame index. The results clearly show that FIFO-Diffusion can generate extremely long videos effectively based on the model trained on short clips (16 frames) without quality degradation while preserving the dynamics and semantics of scenes.

Abstract

We propose a novel inference technique based on a pretrained diffusion model for text-conditional video generation. Our approach, called FIFO-Diffusion, is conceptually capable of generating infinitely long videos without additional training. This is achieved by iteratively performing diagonal denoising, which simultaneously processes a series of consecutive frames with increasing noise levels in a queue; our method dequeues a fully denoised frame at the head while enqueuing a new random noise frame at the tail. However, diagonal denoising is a double-edged sword as the frames near the tail can take advantage of cleaner frames by forward reference but such a strategy induces the discrepancy between training and inference. Hence, we introduce latent partitioning to reduce the training-inference gap and lookahead denoising to leverage the benefit of forward referencing. Practically, FIFO-Diffusion consumes a constant amount of memory regardless of the target video length given a baseline model, while well-suited for parallel inference on multiple GPUs. We have demonstrated the promising results and effectiveness of the proposed methods on existing text-to-video generation baselines. Generated video examples and source codes are available at our project page[1].

[*]indicates equal contribution.
[1]https://jjihwan.github.io/projects/FIFO-Diffusion.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

1 Introduction

Diffusion probabilistic models have achieved remarkable success in generating high-quality images [8, 25, 5, 18]. On top of the success in the image domain, there has been rapid progress in the generation of videos [9, 22, 37, 31]. Despite the progress, long video generation still lags behind compared to image generation. One reason is that video diffusion models (VDMs) often consider a video as a single 4D tensor with an additional axis corresponding to time, which prevents the models from generating videos at scale. An intuitive approach to generating a long video is autoregressive generation, which iteratively predicts a future frame given the previous ones. However, in contrast to the transformer-based models [10, 28], diffusion-based models cannot directly adopt the autoregressive generation strategy due to the heavy computational costs incurred by iterative denoising steps for a single frame generation. Instead, several recent works [9, 7, 29, 12, 4, 1] adopt a chunked autoregressive generation strategy, which predicts several frames in parallel conditioned on few preceding ones, consequently reducing computational burden. While these approaches are computationally tractable, they often leads to temporal inconsistency and discontinuous motion, especially between the chunks predicted separately, because the model captures a limited temporal context available in the last few—only one or two in practice—frames.

To address the limitation, we propose a novel inference technique, FIFO-Diffusion, which realizes arbitrarily long video generation without training based on a pretrained video generation model for short clips. Our approach effectively alleviates the limitations of the chunked autoregressive method by enabling every frame to refer to a sufficient number of preceding frames. Our approach generates frames through diagonal denoising (Section 3.1) in a first-in-first-out manner using a queue, which contains a sequence of frames with different—monotonically increasing—noise levels over time. At each step, a completely denoised frame at the head is popped out from the queue while a new random noise image is pushed back at the tail. Diagonal denoising offers both advantage and disadvantage; noisier frames benefit from referring to cleaner ones while the model may suffer from training-inference gap because video models are generally trained to denoise frames with the same noise level. To overcome this trade-off and embrace the advantage of diagonal denoising, we propose latent partitioning (Section 3.2) and lookahead denoising (Section 3.3). Latent partitioning reduces training-inference gap by narrowing the range of noise levels in to-be-denoised frames and enables inference with finer steps. Lookahead denoising allows to-be-denoised frames to reference cleaner frames, thereby performing more accurate noise prediction. Furthermore, both latent partitioning and lookahead denoising offer parallelizability on multiple GPUs.

Our main contributions are summarized below.

- We propose FIFO-Diffusion through diagonal denoising, which is a training-free video generation technique for VDMs pretrained on short clips. Our approach denoises images with different noise levels for seamless video generation, enabling us to generate arbitrarily long videos.
- We introduce latent partitioning and lookahead denoising, which respectively reduce the training-inference gap incurred by diagonal denoising and allow the reference to less noisy frames for denoising, improving generation quality.
- FIFO-Diffusion requires a constant amount of memory regardless of the length of the generated videos given a baseline model. It is straightforward to run FIFO-Diffusion in parallel on multiple GPUs.
- Our experiments on four strong baselines, based on the U-Net [19] or DiT [16] architectures, show that FIFO-Diffusion generates extremely long videos including natural motion without degradation on quality over time.

2 Text-to-Video Diffusion Models

We summarize the basic idea of text-conditional video generation techniques based on diffusion models. They consist of a few key components: an encoder $\text{Enc}(\cdot)$, a decoder $\text{Dec}(\cdot)$, and a noise prediction network $\boldsymbol{\epsilon}_\theta(\cdot)$. They learn the distribution of videos corresponding to text conditions, and the video is denoted by $\boldsymbol{v} \in \mathbb{R}^{f \times H \times W \times 3}$, where f is the number of frames and $H \times W$ indicates the image resolution. The encoder projects each frame onto the latent image space and the decoder reconstructs the frame from the latent. A video latent $\mathbf{z}_0 = \text{Enc}(\boldsymbol{v}) = [\boldsymbol{z}_0^1;...;\boldsymbol{z}_0^f] \in \mathbb{R}^{f \times h \times w \times c}$ is obtained by concatenating projected frames and the latent diffusion model is trained to denoise its perturbed version, $\mathbf{z}_t$. For noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, a diffusion time step $t \sim \mathcal{U}([1,...,T])$, and a text

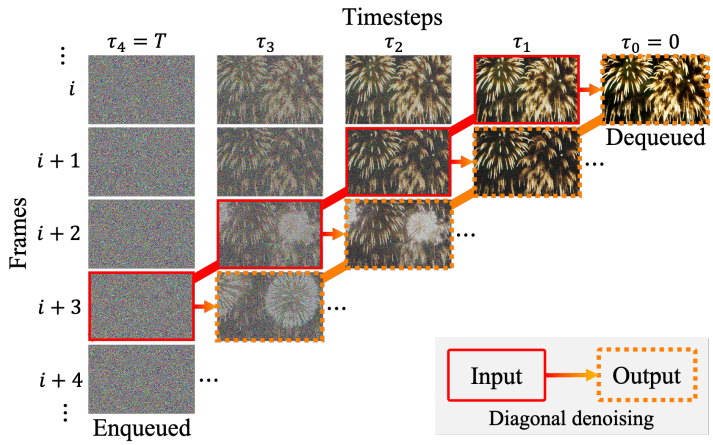


Figure 2: Illustration of diagonal denoising with $f = 4$. The frames surrounded by solid lines are model inputs while frames surrounded by dotted line are their denoised version. After denoising, the fully denoised instance at the top-right corner is dequeued while random noise is enqueued.

condition $\boldsymbol{c}$, the model is trained to minimize the following loss:

$$\mathbb{E}_{\boldsymbol{v},\boldsymbol{\epsilon},t}\left[||\boldsymbol{\epsilon}_\theta(\mathbf{z}_t;\boldsymbol{c},t)-\boldsymbol{\epsilon}||\right], \tag{1}$$

where the perturbed latent, $\mathbf{z}_t = s_t\mathbf{z}_0 + \sigma_t\boldsymbol{\epsilon}$, is obtained using predefined constants $\{s_t\}_{t=0}^T$ and $\{\sigma_t\}_{t=0}^T$, with the constraints $s_0 = 1$, $\sigma_0 = 0$ and $\sigma_T/s_T \gg 1$.
Following a time step schedule, $0 = \tau_0 < \tau_1 < ... < \tau_S = T$, initialized by a diffusion scheduler, the model generates a video by iteratively denoising $[\boldsymbol{z}^1_{\tau_S};...;\boldsymbol{z}^f_{\tau_S}] \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ over S steps using a sampler $\Phi(\cdot)$ such as the DDIM sampler. Each denoising step is expressed as

$$[\boldsymbol{z}^1_{\tau_{t-1}};...;\boldsymbol{z}^f_{\tau_{t-1}}] = \Phi([\boldsymbol{z}^1_{\tau_t};...;\boldsymbol{z}^f_{\tau_t}],[\tau_t;...;\tau_t],\boldsymbol{c};\boldsymbol{\epsilon}_\theta), \tag{2}$$

where $\boldsymbol{z}^i_{\tau_t}$ denotes the latent of the i^{th} frame at time step τ_t.

3 FIFO-Diffusion

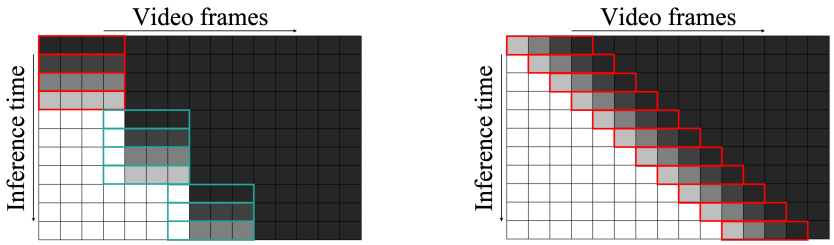
This section discusses how FIFO-Diffusion generates long videos consisting of N frames using a pretrained model only for f frames ($f \ll N$). The proposed approach iteratively employs diagonal denoising (Section 3.1) over a predefined number of frames with different levels of noise. Our method also incorporates latent partitioning (Section 3.2) and lookahead denoising (Section 3.3) to improve the output quality of FIFO-Diffusion based on diagonal denoising.

3.1 Diagonal denoising

Diagonal denoising processes a series of consecutive frames with increasing noise levels as depicted in Figure 2. To be specific, for a time step schedule $0 = \tau_0 < \tau_1 < ... < \tau_f = T$, each denoising step is defined as

$$[\boldsymbol{z}^1_{\tau_0};...;\boldsymbol{z}^f_{\tau_{f-1}}] = \Phi([\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^f_{\tau_f}],[\tau_1;...;\tau_f],\boldsymbol{c};\boldsymbol{\epsilon}_\theta). \tag{3}$$

Note that the latents along the diagonal, $[\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^f_{\tau_f}]$, are stored in a queue, Q, and diagonal denoising jointly considers the latents with different noise levels of $[\tau_1;...;\tau_f]$, in contrast to the standard method specified in Equation (2). Algorithm 1 in Appendix C illustrates how diagonal denoising in FIFO-Diffusion works. After each denoising step with $[\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^f_{\tau_f}]$, the foremost frame is dequeued as it arrives at the noise level $\tau_0 = 0$, and the new latent at noise level τ_f is enqueued. As a result, the model generates frames in a first-in-first-out manner.
Additionally, the initial diagonal latents $[\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^f_{\tau_f}]$ to initiate the diagonal denoising can be generated from f random noises at time step τ_f, similar to the the process described above. Notably, our approach does not require pregenerated videos or additional training for the initial latent construction. The detailed algorithm is presented in Algorithm 2 in Appendix C.



(a) Chunked autoregressive (b) FIFO-Diffusion

Figure 3: Comparison between the chunked autoregressive methods and FIFO-Diffusion proposed for long video generation. The random noises (black) are iteratively denoised to image latents (white) by the models. The red boxes indicate the denoising network in the pretrained base model while the green boxes denote the prediction network obtained by additional training.

FIFO-Diffusion takes f frames as input, regardless of the target video length, and generates an arbitrary number of frames by producing one frame per iteration using a sliding window approach. Note that generating N ($\gg f$) frames for a video requires $\mathcal{O}(f)$ memory in each step (see Table 2), which is independent of N.

Diagonal denoising allows us to generate consistent videos by sequentially propagating context to later frames. Figure 3 illustrates the conceptual difference between chunked autoregressive methods [9, 7, 29, 12, 4, 1] and FIFO-Diffusion. The former often struggles to maintain long-term context across chunks since their conditioning—only the last generated frame—lacks contextual information propagated from previous frames. In contrast, diagonal denoising progresses through the frame sequence with a stride of 1, allowing each frame to reference a sufficient number of preceding frames during generation. This approach enables the model to naturally extend the local consistency of a few frames to longer sequences. Additionally, FIFO-Diffusion requires no subnetworks or extra training, depending solely on a base model. This distinguishes it from existing autoregressive methods, which often require an additional prediction model or fine-tuning for masked frame outpainting.

3.2 Latent partitioning

Although diagonal denoising enables infinitely long video generation, it introduces a training-inference gap, as the model is trained to denoise all frames at uniform noise levels. To address this, we aim to reduce noise level differences in the input latents by extending the queue length n times (from f to nf with $n > 1$), partitioning it into n blocks, and processing each block independently. Note that the extended queue length increases the number of inference steps.

Algorithm 3 in Appendix C provides the procedure of FIFO-Diffusion with latent partitioning. Let a queue Q has diagonal latents $[\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^{nf}_{\tau_{nf}}]$. We partition Q into n blocks, $[Q_0;...;Q_{n-1}]$, of equal size f, then each block Q_k contains the latents at time steps $\boldsymbol{\tau}_k = [\tau_{kf+1};...;\tau_{(k+1)f}]$. Next, we apply diagonal denoising to each block in a divide-and-conquer manner (See Figure 4 (a)). At $k = 0,..., n-1$, each denoising step updates the queue as follows:

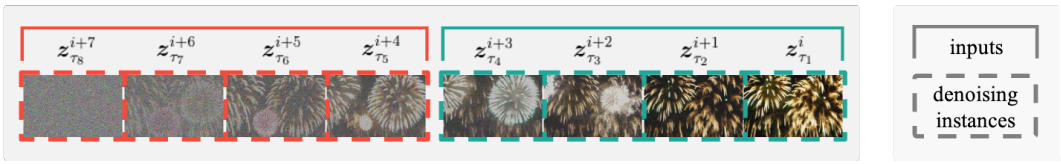
$$Q_k \leftarrow \Phi(Q_k, \boldsymbol{\tau}_k, \boldsymbol{c}; \boldsymbol{\epsilon}_\theta). \tag{4}$$

Latent partitioning offers three key advantages for diagonal denoising. First, it significantly reduces the maximum noise level difference between the latents from $|\sigma_{\tau_{nf}} - \sigma_{\tau_1}|$ to $\max_k |\sigma_{\tau_{(k+1)f}} - \sigma_{\tau_{kf+1}}|$. The effectiveness of latent partitioning is supported theoretically and empirically by Theorem 3.3 and Table 3, respectively. Second, latent partitioning improves throughput of inference by processing partitioned blocks in parallel on multiple GPUs (see Table 2). Last, it allows the diffusion process to leverage a large number of inference steps, nf ($n \geq 2$), reducing discretization error during inference. We now show in Theorem 3.3 that the gap incurred by diagonal denoising is bounded by the maximum noise level difference, which implies that the error can be reduced by narrowing the noise level differences of model inputs.

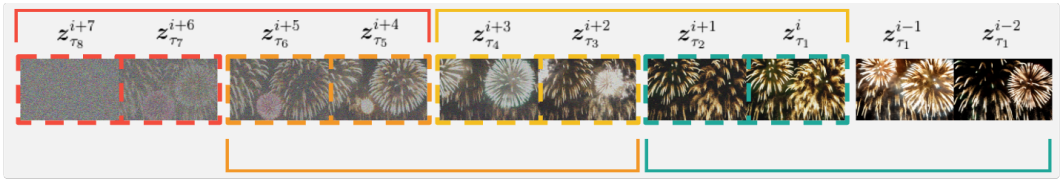
Definition 3.1. We define $\mathbf{z}^{\text{vdm}}_t := [\boldsymbol{z}^1_t;...;\boldsymbol{z}^f_t]$, where $\boldsymbol{z}^i_t$ is the latent of the i^{th} frame at time step t (noise level of $\sigma_t = ct$ for a constant c). $\mathbf{z}^{\text{vdm}}_t$ satisfies the following ODE from [11]:

$$d\mathbf{z}^{\text{vdm}}_t = c \cdot \boldsymbol{\epsilon}(\mathbf{z}^{\text{vdm}}_t, t \cdot \mathbf{1})dt, \tag{5}$$

for $\mathbf{1} = [1;...;1]$ and $\boldsymbol{\epsilon}(\cdot)$ is the scaled score function $-\sigma\nabla_{\mathbf{z}} \log p(\cdot)$.



(a) Latent partitioning



(b) Lookahead denoising

Figure 4: Illustration of latent partitioning and lookahead denoising where $f = 4$ and $n = 2$. (a) Latent partitioning divides the diffusion process into n parts to reduce the maximum noise level difference. (b) Lookahead denoising on (a) enables all frames to be denoised with an adequate number of former frames at the expense of two times more computation than (a).

Lemma 3.2. *If $\epsilon(\cdot)$ is bounded, then*

$$||\boldsymbol{z}_t^i - \boldsymbol{z}_s^i|| = O(|t-s|) \text{ for } \forall i.$$

Proof. Refer to Appendix A.1. □

Theorem 3.3. *Assume the system satisfies the following two hypotheses:*

(Hypothesis 1) $\epsilon(\cdot)$ is bounded.

(Hypothesis 2) The diffusion model $\epsilon_\theta(\cdot)$ is K-Lipschitz continuous.

Then, for diagonal latents $\mathbf{z}^{diag} = [\boldsymbol{z}_{\tau_1}^1;...;\boldsymbol{z}_{\tau_f}^f]$ and corresponding time steps $\boldsymbol{\tau}^{diag} = [\tau_1;...;\tau_f]$,

$$||\boldsymbol{\epsilon}_\theta(\mathbf{z}^{diag}, \boldsymbol{\tau}^{diag})^i - \boldsymbol{\epsilon}(\mathbf{z}_{\tau_i}^{vdm}, \tau_i \cdot \mathbf{1})^i|| = ||\boldsymbol{\epsilon}_\theta(\mathbf{z}_{\tau_i}^{vdm}, \tau_i \cdot \mathbf{1})^i - \boldsymbol{\epsilon}(\mathbf{z}_{\tau_i}^{vdm}, \tau_i \cdot \mathbf{1})^i|| + O(|\sigma_{\tau_f} - \sigma_{\tau_1}|), \tag{6}$$

where the $\boldsymbol{\epsilon}_\theta(\cdot)^i$ and $\boldsymbol{\epsilon}(\cdot)^i$ are i^{th} element of $\boldsymbol{\epsilon}_\theta(\cdot)$ and $\boldsymbol{\epsilon}(\cdot)$, and $\tau_1 < ... < \tau_f$. In other words, the error introduced by diagonal denoising is bounded by the noise level difference.

Proof. The left-hand side of Equation (6) is bounded as:

$$\begin{aligned}
&||\boldsymbol{\epsilon}_\theta(\mathbf{z}^{\text{diag}}, \boldsymbol{\tau}^{\text{diag}})^i - \boldsymbol{\epsilon}(\mathbf{z}_{\tau_i}^{\text{vdm}}, \tau_i \cdot \mathbf{1})^i|| \\
&\qquad \leq ||\boldsymbol{\epsilon}_\theta(\mathbf{z}^{\text{diag}}, \boldsymbol{\tau}^{\text{diag}})^i - \boldsymbol{\epsilon}_\theta(\mathbf{z}_{\tau_i}^{\text{vdm}}, \tau_i \cdot \mathbf{1})^i|| + ||\boldsymbol{\epsilon}_\theta(\mathbf{z}_{\tau_i}^{\text{vdm}}, \tau_i \cdot \mathbf{1})^i - \boldsymbol{\epsilon}(\mathbf{z}_{\tau_i}^{\text{vdm}}, \tau_i \cdot \mathbf{1})^i||,
\end{aligned}$$

by triangle inequality. Then, the first term of the right-hand side satisfies the following inequality:

$$\begin{aligned}
||\boldsymbol{\epsilon}_\theta(\mathbf{z}^{\text{diag}}, \boldsymbol{\tau}^{\text{diag}})^i - \boldsymbol{\epsilon}_\theta(\mathbf{z}_{\tau_i}^{\text{vdm}}, \tau_i \cdot \mathbf{1})^i|| &\leq K||(\mathbf{z}^{\text{diag}}, \boldsymbol{\tau}^{\text{diag}}) - (\mathbf{z}_{\tau_i}^{\text{vdm}}, \tau_i \cdot \mathbf{1})|| \\
&\leq K \sum_{j=1}^{f} (||\boldsymbol{z}_{\tau_j}^j - \boldsymbol{z}_{\tau_i}^j|| + |\tau_j - \tau_i|) = O(|\sigma_{\tau_f} - \sigma_{\tau_1}|),
\end{aligned}$$

which is from the Lipshitz continuity and Lemma 3.2. Furthermore, we provide justification for *(Hypothesis 2)* in Appendix A.2. □

3.3 Lookahead denoising

Although our diagonal denoising introduces training-inference gap, it is advantageous in another respect because noisier frames benefit from observing cleaner ones, leading to more accurate denoising. As empirical evidence, Figure 5 shows the relative MSE losses in noise prediction of diagonal denoising with respect to the original denoising strategy. The formal definition of the relative MSE is given by

$$\frac{||\boldsymbol{\epsilon}_\theta(\mathbf{z}^{\text{diag}}, \boldsymbol{\tau}^{\text{diag}})^i - \boldsymbol{\epsilon}(\mathbf{z}_{\tau_i}^{\text{vdm}}, \tau_i \cdot \mathbf{1})^i||_2}{||\boldsymbol{\epsilon}_\theta(\mathbf{z}_{\tau_i}^{\text{vdm}}, \tau_i \cdot \mathbf{1})^i - \boldsymbol{\epsilon}(\mathbf{z}_{\tau_i}^{\text{vdm}}, \tau_i \cdot \mathbf{1})^i||_2}. \tag{7}$$

As depicted in Figure 4 (b), we estimate noise only for the benefited later half of the frames. In other words, we perform diagonal denoising with a stride of $f' = \lfloor \frac{f}{2} \rfloor$, updating only the last f' frames to ensure that each frame is denoised with reference to a sufficient number—at least f'—of clearer frames. Precisely, for $k = 0,...,2n-1$, each denoising step updates the queue as

$$Q_k^{f'+1:f} \leftarrow \Phi(Q_k, \boldsymbol{\tau}_k, \boldsymbol{c}; \epsilon_\theta)^{f'+1:f}. \quad (8)$$

Algorithm 4 in Appendix C outlines the detailed procedure of FIFO-Diffusion with lookahead denoising. We illustrate the effectiveness of lookahead denoising with the red line in Figure 5. Except for a few early time steps, lookahead denoising enhances the baseline models noise prediction performance, nearly eliminating the training-inference gap described in Section 3.2. Note that, this approach requires twice the computation of the original diagonal denoising since we only update the half of the queue each step. However, the concerns about the additional computational overhead are easily addressed via parallelization in the same manner as latent partitioning (see Table 2).

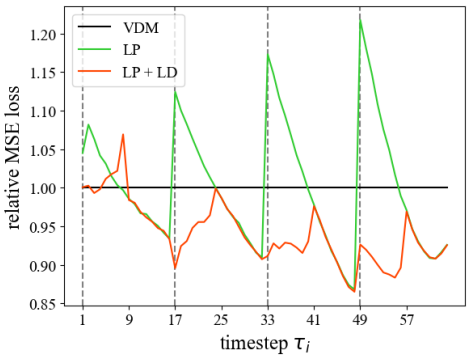


Figure 5: The relative MSE losses of the noise prediction of $\boldsymbol{z}^i_{\tau_i}$ (see Equation (7)) when $n = 4$. 'VDM' indicates the original denoising strategy as a reference line. 'LP' and 'LD' denote latent partitioning and lookahead denoising, respectively.

4 Experiment

This section presents the examples generated by existing long video generation methods including FIFO-Diffusion, and evaluates their performance qualitatively and quantitatively. We also perform the ablation study to verify the benefit of latent partitioning and lookahead denoising introduced in FIFO-Diffusion.

4.1 Implementation details

We implement FIFO-Diffusion based on existing open-source text-to-video diffusion models trained on short video clips, including three U-Net-based models, VideoCrafter1 [2], VideoCrafter2 [3], and zeroscope[2], as well as a DiT-based model, Open-Sora Plan[3]. We employ the DDIM sampling [24] with $\eta \in \{0.5, 1\}$. Appendix B provides more details about our implementations.

For quantitative evaluation, we measure FVD_{128} [27] and IS [21] scores using Latte [13] as a base model, which is a DiT-based video model trained on UCF-101 [26]. We generate 2,048 videos with 128 frames each to calculate FVD_{128}, and randomly sample a 16-frame clip from each video to measure IS score, following evaluation guidelines in [23]. To calculate computational cost, we adopt VideoCrafter2 as the baseline model, using a DDPM scheduler with 64 inference steps on A6000 GPUs.

4.2 Qualitative results

We first evaluate the performance of the proposed approach qualitatively. Figure 1 illustrates examples of long videos (longer than 10K frames) generated by FIFO-Diffusion based on VideoCrafter2. It demonstrates the ability of FIFO-Diffusion to generate significantly longer videos than the target length of pretrained baseline models—16 frames in this case. The individual frames exhibit outstanding visual quality with no perceptual quality degradation even in the later part of the videos while preserving semantic information across all frames. Figure 6 (a) and (b) present the generated videos with natural motion of scenes and cameras; the consistency of motion is effectively maintained by referencing earlier frames through the generation process.

Furthermore, Figure 6 (c) illustrates that FIFO-Diffusion can generate videos with extensive motion driven by a sequence of changing prompts. The capability to generate multiple motions and seamless transitions between scenes highlight the practicality of our method. Please refer to Appendices D and E for more examples and our project page[1] for video demos, in comparisons with the videos from other baselines.

[2] `https://huggingface.co/cerspense/zeroscope_v2_576w`

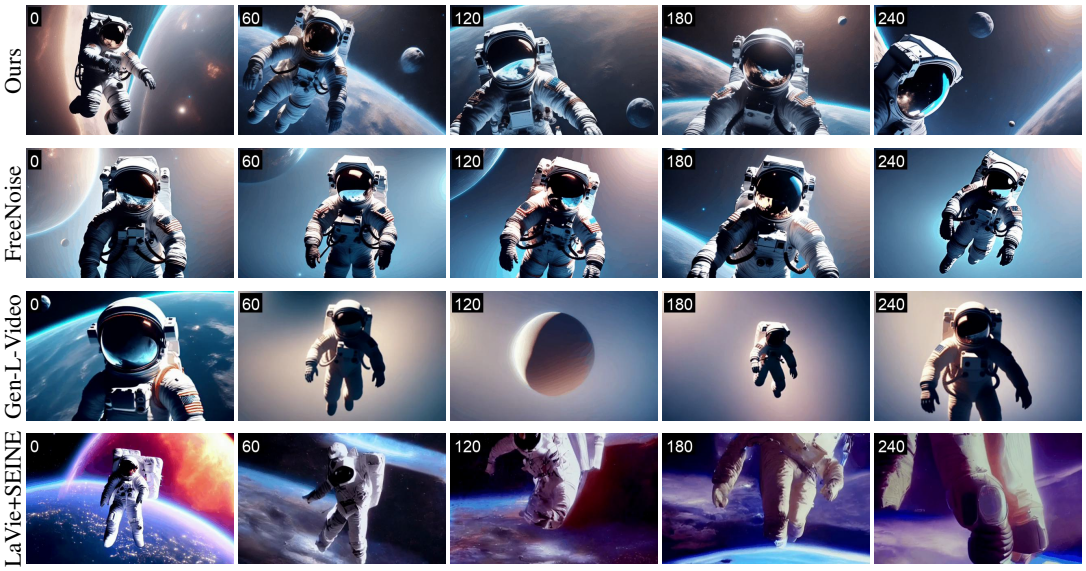
[3] `https://github.com/PKU-YuanGroup/Open-Sora-Plan`

(a) "a serene winter scene in a forest. The forest is blanketed in a thick layer of snow, which..."

(b) "A vibrant underwater scene of a scuba diver exploring a shipwreck, 2K, photorealistic."

(c) "A tiger ***walking*** → ***standing*** → ***resting*** on the grassland, photorealistic, 4k, high definition"

Figure 6: Illustrations of long videos generated by FIFO-Diffusion based on (a) Open-Sora Plan and (b) VideoCrafter2, as well as (c) multiple prompts based on VideoCrafter2. The number on the top-left corner of each frame indicates the frame index.



"An astronaut floating in space, high quality, 4K resolution."

Figure 7: Sample videos generated by (first) FIFO-DIffusion on VideoCrafter2, (second) FreeNoise on VideoCrafter2, (third) Gen-L-Video on VideoCrafter2, and (last) LaVie + SEINE. The number on the top-left corner of each frame indicates the frame index.

Table 1: Comparisons of FVD_{128} and IS scores on UCF-101. FIFO-Diffusion with latent partitioning and lookahead denoising utilizes Latte [13] as its baseline, where the number of partitions is four ($n = 4$). The FVD and IS scores of the other algorithms are obtained from their respective papers, and PVDM [35] denotes PVDM-L (400-400s).

	FVD_{128} (↓)	IS (↑)
StyleGAN-V [23]	1773.4	23.94±0.73
VIDM [14]	1531.9	–
PVDM [35]	648.4	74.40±1.25
FIFO-Diffusion (ours)	**596.64**	**74.44±1.17**

Table 2: Memory usages and inference times of long video generation methods. FIFO-Diffusion utilizes latent partitioning with $n = 4$ and lookahead denoising.

	Memory usage [MB] (↓)			Inference time [s/frame] (↓)
Method	128 frames	256 frames	512 frames	
FreeNoise [17]	26,163	44,683	out of memory	6.09
Gen-L-Video [30]	10,913	10,937	10,965	22.07
FIFO-Diffusion (1 GPU)	11,245	11,245	11,245	12.37
FIFO-Diffusion (8 GPUs)	13,496	13,496	13,496	1.84

Figure 7 compares the results from FIFO-Diffusion with two training-free techniques, FreeNoise [17] and Gen-L-Video [30] based on VideoCrafter2, as well as a training-based chunked autoregressive method, LaVie [32] + SEINE [4]. Note that the chunked autoregressive method requires two models: LaVie for T2V and SEINE for I2V. We observe that our method significantly outperforms the others in terms of motion smoothness, frame quality, and scene diversity.

Among the training-free methods, Gen-L-Video often produces videos with blurred background while FreeNoise struggles to generate dynamic scenes.[4] The videos from LaVie + SEINE gradually degrade and diverge from text prompts due to error accumulation in their autoregressive generation processes. Additionally, they often exhibit discontinuities between adjacent chunks, as only the last frame of each chunk is employed to transfer contextual information to the next. Figures 18 and 19 in Appendix F provide further examples comparing these methods.

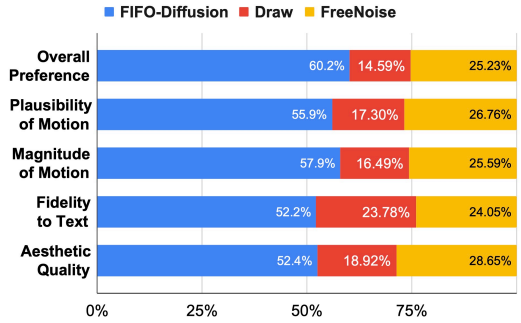


Figure 8: The results of user study between FIFO-Diffusion and FreeNoise for five criteria.

We also conduct a user study to evaluate the long video generation performance of FIFO-Diffusion compared to an existing approach, FreeNoise. As shown in Figure 8, users expressed a strong preference for FIFO-Diffusion across all criteria, particularly those related to motion. Given that motion is one of the most defining characteristics of videos as opposed to images, the strong performance of FIFO-Diffusion in these criteria is promising and highlights its potential to generate more natural, dynamic videos. Details about the user study are provided in Appendix B.1.

4.3 Quantitative results

We compare FIFO-Diffusion with the baselines trained for long video generation [23, 14, 35] in terms of the FVD_{128} and IS scores. As shown in Table 1, our approach outperforms all the compared methods including PVDM-L (400-400s) [35], which employs a chunked autoregressive generation strategy. Note that PVDM-L (400-400s) iteratively generates 16 frames conditioned on the previous outputs over 400 diffusion steps while our approach only requires 64 inference steps (with lookahead denoising) without need for additional training.

[4] We provide qauantitative evaluation on the magnitude of motion in Appendix G.

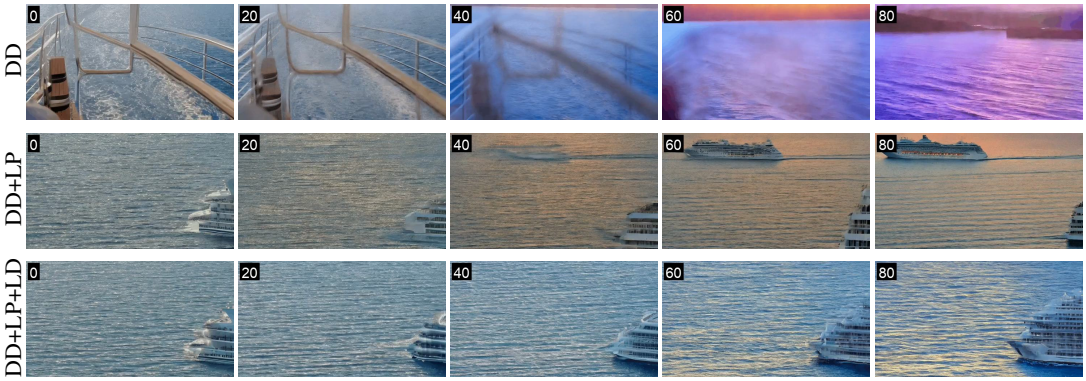


Figure 9: Ablation study. DD, LP, and LD signifies diagonal denoising, latent partitioning, and lookahead denoising, respectively. The number on the top-left corner of each frame indicates the frame index.

Table 3: Relative MSE losses of ablations. 'LP' and 'LD' denote latent partitioning and lookahead denosing, respectively.

	# of partitions	without LD	with LD
without LP	1	1.09	1.01
with LP	2	1.04	0.99
with LP	4	1.02	**0.98**

4.4 Computational cost

To evaluate computational efficiency, we assess memory usage and inference time per frame for training-free, long video generation methods. As shown in Table 2, FIFO-Diffusion generates videos of arbitrary lengths with a constant memory allocation, while FreeNoise requires memory proportional to the target video length. Although Gen-L-Video maintains nearly constant memory usage, it exhibits the slowest inference speed due to redundant computations. Notably, FIFO-Diffusion leverages parallel computation; while incorporating lookahead denoising increases computational demand, utilizing multiple GPUs for parallel processing significantly reduces sampling time.

4.5 Ablation study

We conduct ablation study to analyze the effect of latent partitioning and lookahead denoising on the performance of FIFO-Diffusion. Figure 9 shows that latent partitioning significantly improves both visual quality and temporal consistency of the generated videos. Moreover, lookahead denoising further refines the quality of generated videos by facilitating temporal coherency and reducing flickering effects. The videos on our project page[5] clearly demonstrate the benefit of FIFO-Diffusion. Additionally, Table 3 compares the relative MSE loss (see Equation (7)) averaged over all time steps across different ablation settings. The results show that latent partitioning effectively reduces the training-inference gap caused by diagonal denoising as the number of partitions increases. Furthermore, lookahead denoising enhances the model's noise prediction accuracy, achieving low relative MSE losses (below 1.0) when used in conjunction with latent partitioning.

5 Related work

This section discusses existing diffusion-based generative models for videos including long video generation techniques.

5.1 Video diffusion models

Diffusion models, originally developed for high-quality image synthesis, have become a prominent approach in video generation [2, 9, 22, 37, 31]. VDM [9] modifies the structure of U-Net [19] and proposes a 3D U-Net architecture to incorporate temporal information for denoising. On the

[5] `https://jjihwan.github.io/projects/FIFO-Diffusion`

other hand, Make-A-Video [22] employs a 1D temporal convolution layer following a 2D spatial convolutional layer to approximate 3D convolution. This design enables the model to capture visual-textual relationships by training spatial layers with image-text pairs before incorporating temporal context through 1D temporal layers. Recently, [16] introduce a transformer architecture, known as DiT, for diffusion models. Additionally, several open-sourced text-to-video models have emerged [31, 2, 32, 3], trained on large-scale text-image and text-video datasets.

5.2 Long video generation

Long video generation approaches typically involve training models to predict future frames sequentially [29, 6, 1, 4]. or generate a set of frames in a hierarchical manner [7, 34]. For instance, Video LDM [1] and MCVD [29] employ autoregressive techniques to sequentially predict frames given several preceding ones, while FDM [6] and SEINE [4] generalize masked learning strategies for both prediction and interpolation. Autoregressive methods are capable of producing indefinitely long videos in theory, but they often suffer from quality degradation due to error accumulation and limited temporal consistency across frames. Alternatively, NUWA-XL [34] adopts a hierarchical approach, where a global diffusion model generates sparse key frames with local diffusion models filling in frames using the key frames as references. However, this hierarchical setup requires batch processing, making it unsuitable for generating infinitely long videos.
There are a few training-free long video generation techniques. Gen-L-Video [30] treats a video as overlapped short clips and introduces temporal co-denoising, which averages multiple predictions for one frame. FreeNoise [17] employs window-based attention fusion to sidestep the limited attention scope issue and proposes local noise shuffle units for the initialization of long video. FreeNoise requires memory proportional to the video length for the computation of cross, limiting its scalability for generating infinitely long videos.

5.3 Diffusion models with latents of different noise levels

Recent studies have adopted diffusion models for sequence generation by leveraging a sliding window approach with temporally varying noise levels [36, 20]. These methods train diffusion models from scratch to accommodate latents with different noise levels, addressing tasks such as motion generation [36] and video prediction [20]. However, training diffusion models from scratch introduces significant computational costs, especially for text-to-video generation tasks. In contrast, our approach is a training-free inference technique based on the standard diffusion models, trained on latents with uniform noise, for sequence generation within the sliding window framework. While [20] is implemented with a nested loop to deal with two different axes corresponding to video frame index and diffusion time step, FIFO-Diffusion combines these two dimensions using a 1D queue, improving efficiency with a single loop.

6 Conclusion

We introduced FIFO-Diffusion, a novel inference algorithm that enables the generation of infinitely long videos from text without tuning video diffusion models pretrained on short clips. Our approach achieves this by introducing diagonal denoising, which processes latents with increasing noise levels using a queue in a first-in-first-out fashion. While diagonal denoising presents a trade-off, we addressed its limitations with latent partitioning and leveraged its strengths with lookahead denoising. Together, these techniques allow FIFO-Diffusion to generate high-quality, long videos that maintain strong scene consistency and expressive dynamic motion. Although latent partitioning reduces the training-inference gap of diagonal denoising, the gap persists due to changes in the model's input distribution. However, we believe that this gap could be addressed by integrating the diagonal denoising paradigm into the training phase, and the benefits of FIFO-Diffusion remains for training as well. We leave this integration as future work; aligning the training and inference environments can significantly enhance FIFO-Diffusion's performance.

Acknowledgements

This work was partly supported by LG AI Research, and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2022-II220959 (No.2022-0-00959), (Part 2) Few-Shot Learning of Causal Inference in Vision and Language for Decision Making); NO.RS-2021- II211343, Artificial Intelligence Graduate School Program (Seoul National University)].

References

[1] 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 *CVPR*, 2023.

[2] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. VideoCrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

[3] 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. *arXiv preprint arXiv:2401.09047*, 2024.

[4] 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. *arXiv preprint arXiv:2310.20700*, 2023.

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

[6] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. In *NeurIPS*, 2022.

[7] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[9] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, 2022.

[10] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. CogVideo: Large-scale pretraining for text-to-video generation via transformers. In *ICLR*, 2023.

[11] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

[12] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *CVPR*, 2023.

[13] 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*, 2024.

[14] Kangfu Mei and Vishal M. Patel. Vidm: Video implicit diffusion models. In *AAAI*, 2023.

[15] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[16] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[17] Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. FreeNoise: Tuning-free longer video diffusion via noise rescheduling. *arXiv preprint arXiv:2310.15169*, 2023.

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

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[20] David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In *ICML*, 2024.

[21] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

[22] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-A-Video: Text-to-video generation without text-video data. In *ICLR*, 2022.

[23] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *CVPR*, 2022.

[24] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

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

[26] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[27] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.0171*, 2018.

[28] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. In *ICLR*, 2023.

[29] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In *NeurIPS*, 2022.

[30] Fu-Yun Wang, Wenshuo Chen, Guanglu Song, Han-Jia Ye, Yu Liu, and Hongsheng Li. Gen-L-Video: Multi-text to long video generation via temporal co-denoising. *arXiv preprint arXiv:2305.18264*, 2023.

[31] 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.

[32] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, Yuwei Guo, Tianxing Wu, Chenyang Si, Yuming Jiang, Cunjian Chen, Chen Change Loy, Bo Dai, Dahua Lin, Yu Qiao, and Ziwei Liu. LaVie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023.

[33] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[34] Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, Jianlong Fu, Gong Ming, Lijuan Wang, Zicheng Liu, Houqiang Li, and Nan Duan. NUWA-XL: Diffusion over diffusion for extremely long video generation. *arXiv preprint arXiv:2303.12346*, 2023.

[35] Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. In *CVPR*, 2023.

[36] Zihan Zhang, Richard Liu, Kfir Aberman, and Rana Hanocka. Tedi: Temporally-entangled diffusion for long-term motion synthesis. In *SIGGRAPH*, 2024.

[37] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. MagicVideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

Appendix

A Details for Lemma 3.2 and Theorem 3.3

A.1 Proof of Lemma 3.2

Lemma 3.2. If $\boldsymbol{\epsilon}(\cdot)$ is bounded, then

$$||\boldsymbol{z}_t^i - \boldsymbol{z}_s^i|| = O(|t-s|) \text{ for any } i.$$

Proof. Since $\boldsymbol{\epsilon}(\cdot)$ is bounded, there exists some $M > 0$ satisfying $||\boldsymbol{\epsilon}(\cdot)|| \leq M$.

$$\begin{aligned}
||\boldsymbol{z}_t^i - \boldsymbol{z}_s^i|| &\leq ||\mathbf{z}_t^{\text{vdm}} - \mathbf{z}_s^{\text{vdm}}|| \\
&= || \int_s^t c \cdot \boldsymbol{\epsilon}(\mathbf{z}_u^{\text{vdm}}, u \cdot \mathbf{1}) du|| \\
&\leq | \int_s^t c \cdot ||\boldsymbol{\epsilon}(\mathbf{z}_u^{\text{vdm}}, u \cdot \mathbf{1})|| du| \\
&\leq c \cdot M \cdot |t-s|.
\end{aligned}$$

□

A.2 Justification on *(Hypothesis 2)* of Theorem 3.3

We provide justification for the hypothesis, which the diffusion model is K-Lipschitz continuous. At inference, we can consider $z \in [0, B]^{f \times c \times h \times w}$ and $\sigma \in [\sigma_{\text{min}}, \sigma_{\text{max}}]$, where $\sigma_{\text{min}} > 0$ since z is pixel values and we inference for such σ. In appendix B.3 of [11], $\epsilon(z, \sigma)$ is given as the following:

$$\epsilon(z, \sigma) = -\sigma \frac{\nabla_z \sum_i \mathcal{N}(z; y_i, \sigma^2 \mathbf{I})}{\sum_i \mathcal{N}(z; y_i, \sigma^2 \mathbf{I})},$$

where $y_1, y_2, \ldots, y_n$ are data points. Note that $\mathcal{N}(z; y_i, \sigma^2 \mathbf{I})$ is twice differentiable and continuous, and $\sum_i \mathcal{N}(z; y_i, \sigma^2 \mathbf{I}) \geq c$ for $\exists c > 0$. Therefore, the differential function of $\epsilon(z, \sigma)$ is bounded and is Lipschitz continuous. Since $\epsilon_\theta(\cdot)$ estimates $\epsilon(\cdot)$, assuming Lipschitz continuity can be justified.

B Implementation details

We provide the implementation details of the experiments in Table 4. We use VideoCrafter1 [2], VideoCrafter2 [3], zeroscope[6], Open-Sora Plan[7], LaVie [32], and SEINE [4] as pre-trained models. zeroscope, VideoCrafter, and Open-Sora Plan are under CC BY-NC 4.0, Apache License 2.0, and MIT License, respectively. Except for automated results, all prompts used in experiments are randomly generated by ChatGPT-4 [15]. We empirically choose $n = 4$ for the number of partitions in latent partitioning and lookahead denoising. Also, stochasticity η, introduced by DDIM [24], is chosen to achieve good results from the baseline video generation models.

Table 4: Implementation details regarding experiments

Experiment	Model	f	Sampling Method	n	η	# Prompts	# Frames	Resolution
MSE loss (Figure 5 and Table 3)	VideoCrafter1	16	FIFO-Diffusion	4	0.5	200	-	320×512
Qualitative Result	zeroscope	24	FIFO-Diffusion	4	0.5	-	100	320×576
	VideoCrafter1	16	FIFO-Diffusion	4	0.5	-	100	320×512
	VideoCrafter2	16	FIFO-Diffusion	4	1	-	100$\sim$10k	320×512
	Open-Sora Plan	17	FIFO-Diffusion	4	1	-	385	512×512
	VideoCrafter2	16	FreeNoise	-	1	-	100	320×512
	VideoCrafter2	16	Gen-L-Video	-	1	-	100	320×512
	LaVie + SEINE	16	chunked autoregressive	-	1	-	100	320×512
User Study	VideoCrafter2	16	FIFO-Diffusion	4	1	30	100	320×512
	LaVie	16	FreeNoise	-	1	30	100	320×512
Motion Evaluation	VideoCrafter1	16	FIFO-Diffusion	4	0.5	512	100	256×256
	VideoCrafter1	16	FreeNoise	-	0.5	512	100	256×256
Ablation study	zeroscope	24	FIFO-Diffusion	$\{1, 4\}$	0.5	-	100	320×576

B.1 Details for user study

We randomly generated 30 prompts from ChatGPT-4 without cherry-picking, and generated a video for each prompt with 100 frames using each method. The evaluators were asked to choose their preference (A is better, draw, or B is better) between the two videos generated by FIFO-Diffusion and FreeNoise with the same prompts, on five criteria: overall preference, plausibility of motion, magnitude of motion, fidelity to text, and aesthetic quality. A total of 70 users submitted 111 sets of ratings, where each set consists of 20 videos from 10 prompts. We used LaVie as the baseline for FreeNoise, since it was the latest model officially implemented at that time.

[6] `https://huggingface.co/cerspense/zeroscope_v2_576w`

[7] `https://github.com/PKU-YuanGroup/Open-Sora-Plan`

C Algorithms of FIFO-Diffusion

This section illustrates pseudo-code for FIFO-Diffusion with and without latent partitioning and lookahead denoising.

Algorithm 1 FIFO-Diffusion with diagonal denoising (Section 3.1)

Require: N, f, $\epsilon_\theta(\cdot)$, $\text{Dec}(\cdot)$, $\Phi(\cdot)$
  Input: $[\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^f_{\tau_f}]$, $[\tau_1;...;\tau_f]$, $\boldsymbol{c}$
  Output: $\boldsymbol{v}$
  $\boldsymbol{v} \leftarrow []$
  $\boldsymbol{\tau} \leftarrow [\tau_1;...;\tau_f]$
  $Q \leftarrow [\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^f_{\tau_f}]$
  for $i = 1$ **to** N **do**
    $Q \leftarrow \Phi(Q, \boldsymbol{\tau}, \boldsymbol{c}; \epsilon_\theta)$ # Equation (3)
    $\boldsymbol{z}^i_{\tau_0} \leftarrow Q.\text{dequeue}()$ # Fully denoised frame
    $\boldsymbol{v}.\text{append}(\text{Dec}(\boldsymbol{z}^i_{\tau_0}))$
    $\boldsymbol{z}^{i+f}_{\tau_f} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ # New random noise
    $Q.\text{enqueue}(\boldsymbol{z}^{i+f}_{\tau_f})$
  end for
  return $\boldsymbol{v}$

Algorithm 2 Initial latent construction (Section 3.1)

Require: N, f, $\epsilon_\theta(\cdot)$, $\text{Dec}(\cdot)$, $\Phi(\cdot)$
  Input: $\boldsymbol{z}^{1:f}_{\tau_f} \sim \mathcal{N}(0, \boldsymbol{I})$, $\{\tau_i\}_{i=0}^f$, $\boldsymbol{c}$
  Output: $[\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^f_{\tau_f}]$
  $\boldsymbol{\tau} \leftarrow [\tau_f;...;\tau_f]$
  $Q \leftarrow [\boldsymbol{z}^1_{\tau_f};...;\boldsymbol{z}^f_{\tau_f}]$
  for $i = 1$ **to** f **do**
    $Q \leftarrow \Phi(Q, \boldsymbol{\tau}, \boldsymbol{c}; \epsilon_\theta)$
    $Q.\text{dequeue}()$
    $\boldsymbol{z}^i_{\tau_f} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ # New random noise
    $Q.\text{enqueue}(\boldsymbol{z}^i_{\tau_f})$
    $\boldsymbol{\tau} \leftarrow [\overbrace{\tau_{f-i};...;\tau_{f-i}}^{f-i};\overbrace{\tau_{f-i+1}...;\tau_f}^{i}]$ # Varying timestep
  end for
  return $Q = [\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^f_{\tau_f}]$

Algorithm 3 FIFO-Diffusion with latent partitioning (Section 3.2)

Require: N, f, $\epsilon_\theta(\cdot)$, $\text{Dec}(\cdot)$, $\Phi(\cdot)$, n # $n \geq 2$ if latent partitioning
Input: $[\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^{nf}_{\tau_{nf}}]$, $[\tau_1;...;\tau_{nf}]$, $\boldsymbol{c}$
Output: $\boldsymbol{v}$
$\boldsymbol{v} \leftarrow [\,]$
$\boldsymbol{\tau} \leftarrow [\tau_1;...;\tau_{nf}]$
$Q \leftarrow [\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^{nf}_{\tau_{nf}}]$
for $i = 1$ **to** N **do**
  for $k = 0$ **to** $n-1$ **do** # Parallelizable
    $\boldsymbol{\tau}_k \leftarrow \boldsymbol{\tau}^{kf+1:(k+1)f}$
    $Q_k \leftarrow Q^{kf+1:(k+1)f}$
    $Q_k \leftarrow \Phi(Q_k, \boldsymbol{\tau}_k, \boldsymbol{c}; \epsilon_\theta)$ # Equation (4)
  end for
  $Q \leftarrow [Q_0;...;Q_{n-1}]$
  $\boldsymbol{z}^i_{\tau_0} \leftarrow Q$.dequeue()
  $\boldsymbol{v}$.append($\text{Dec}(\boldsymbol{z}^i_{\tau_0})$)
  $\boldsymbol{z}^{i+nf}_{\tau_f} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Q.enqueue($\boldsymbol{z}^{i+nf}_{\tau_{nf}}$)
end for
return $\boldsymbol{v}$

Algorithm 4 FIFO-Diffusion with lookahead denoising (Section 3.3)

Require: $N, \epsilon_\theta(\cdot)$, $\text{Dec}(\cdot)$, $\Phi(\cdot)$, n # $n \geq 2$ if latent partitioning
Input: $[\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^{nf}_{\tau_{nf}}]$, $[\tau_1;...;\tau_{nf}]$, $\boldsymbol{c}$
Output: $\boldsymbol{v}$
$\boldsymbol{v} \leftarrow [\,]$
$\boldsymbol{\tau} \leftarrow [\overbrace{\tau_1;...;\tau_1}^{f'};\tau_1;...;\tau_{nf}]$
$Q \leftarrow [\overbrace{\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^1_{\tau_1}}^{f'};\boldsymbol{z}^1_{\tau_1};...;\boldsymbol{z}^{nf}_{\tau_{nf}}]$ # dummy latents are required
for $i = 1$ **to** N **do**
  $\boldsymbol{z}^i_{\tau_1} \leftarrow Q^{f'+1}$
  for $k = 0$ **to** $2n-1$ **do** # Parallelizable
    $\boldsymbol{\tau}_k \leftarrow \boldsymbol{\tau}^{kf'+1:(k+2)f'}$
    $Q_k \leftarrow Q^{kf'+1:(k+2)f'}$
    $Q_k^{f'+1:f} \leftarrow \Phi(Q_k, \boldsymbol{\tau}_k, \boldsymbol{c}; \epsilon_\theta)^{f'+1:f}$ # Equation (8)
  end for
  $\boldsymbol{z}^i_{\tau_0} \leftarrow Q_0^{f'+1}$
  $\boldsymbol{v}$.append($\text{Dec}(\boldsymbol{z}^i_{\tau_0})$)
  $Q_0^{f'+1} \leftarrow \boldsymbol{z}^i_{\tau_1}$
  $Q \leftarrow [Q_0^{1:f'}; Q_0^{f'+1:f};...; Q_{2n-1}^{f'+1:f}]$
  $Q \leftarrow [Q_0; Q_1^{f'+1:f};...; Q_{2n-1}^{f'+1:f}]$
  Q.dequeue()
  $\boldsymbol{z}^{i+nf}_{\tau_{nf}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Q.enqueue($\boldsymbol{z}^{i+nf}_{\tau_{nf}}$)
end for
return $\boldsymbol{v}$

D Qualitative results of FIFO-Diffusion

In Figures 10 to 15, we provide more qualitative results with 4 baselines, VideoCrafter2 [3], VideoCrafter1 [2], zeroscope[8], and Open-Sora Plan[9].

D.1 VideoCrafter2

(a) "A colony of penguins waddling on an Antarctic ice sheet, 4K, ultra HD."

(b) "A colorful macaw flying in the rainforest, ultra HD."

(c) "A dark knight riding on a black horse on the glassland, photorealistic, 4k, high definition."

(d) "A high-altitude view of a hang glider in flight, high definition, 4K."

(e) "A high-speed motorcycle race on a track, ultra HD, 4K resolution."

(f) "A horse race in full gallop, capturing the speed and excitement, 2K, photorealistic."

Figure 10: Videos generated by FIFO-Diffusion with VideoCrafter2. The number on the top left of each frame indicates the frame index.

[8] https://huggingface.co/cerspense/zeroscope_v2_576w

[9] https://github.com/PKU-YuanGroup/Open-Sora-Plan

(a) "A pair of tango dancers performing in Buenos Aires, 4K, high resolution."

(b) "A panoramic view of a peaceful Zen garden, high-quality, 4K resolution."

(c) "A paraglider soaring over the Alps, photorealistic, 4K, high definition."

(d) "A scenic hot air balloon flight at sunrise, high quality, 4K."

(e) "A scenic hot air balloon flight over Cappadocia, Turkey, 2K, ultra HD."

(f) "A school of colorful fish swimming in a coral reef, ultra high quality, 2K."

(g) "A spectacular fireworks display over Sydney Harbour, 4K, high resolution."

Figure 11: Videos generated by FIFO-Diffusion with VideoCrafter2. The number on the top left of each frame indicates the frame index.

(a) "A spooky haunted house, foggy night, high definition."

(b) "A time-lapse of a busy construction site, high definition, 4K."

(c) "A vibrant underwater scene of a scuba diver exploring a shipwreck, 2K, photorealistic."

(d) "A vibrant, fast-paced salsa dance performance, ultra high quality, 2K."

(e) "An astronaut floating in space, high quality, 4K resolution."

(f) "An astronaut walking on the moon's surface, high-quality, 4K resolution."

(g) "A majestic lion roaring in the African savanna, ultra HD, 4K."

Figure 12: Videos generated by FIFO-Diffusion with VideoCrafter2. The number on the top left of each frame indicates the frame index.

D.2 VideoCrafter1

(a) "A kayaker navigating through rapids, photorealistic, 4K, high quality."

(b) "A pair of tango dancers performing in Buenos Aires, 4K, high resolution."

(c) "A panoramic view of the Himalayas from a drone, high definition, 4K."

(d) "A paraglider soaring over the Alps, photorealistic, 4K, high definition."

(e) "A professional surfer riding a large wave, high-quality, 4K."

(f) "A school of colorful fish swimming in a coral reef, ultra high quality, 2K."

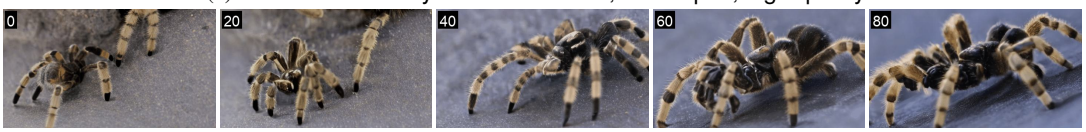
(g) "An exciting mountain bike trail ride through a forest, 2K, ultra HD."

(h) "A vibrant coral reef with diverse marine life, photorealistic, 2K resolution."

Figure 13: Videos generated by FIFO-Diffusion with VideoCrafter1. The number on the top left of each frame indicates the frame index.

D.3 zeroscope

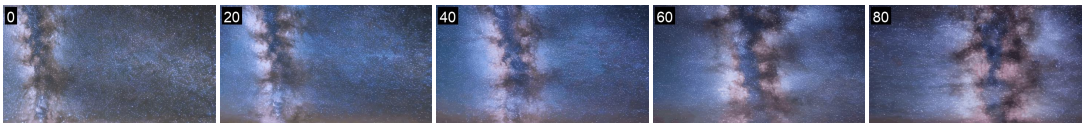
(a) "A beautiful cherry blossom festival, time-lapse, high quality."



(b) "A close-up of a tarantula walking, high definition."

(c) "A thrilling white water rafting adventure, high definition."

(d) "A detailed macro shot of a blooming rose, 4K."



(e) "A panoramic view of the Milky Way, ultra HD."

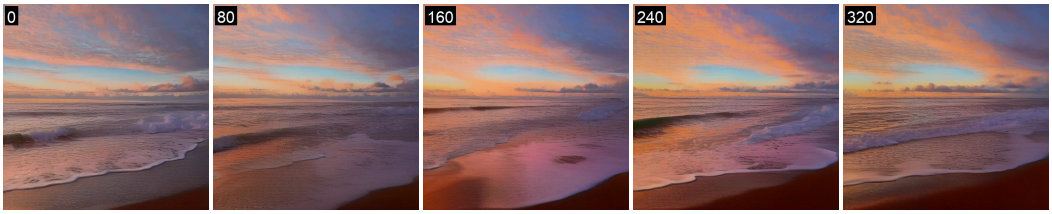
(f) "A mysterious foggy forest at dawn, high quality, 4K."

(g) "A scenic cruise ship journey at sunset, ultra HD."

(h) "A lone tree in a vast desert, sunset, high definition."

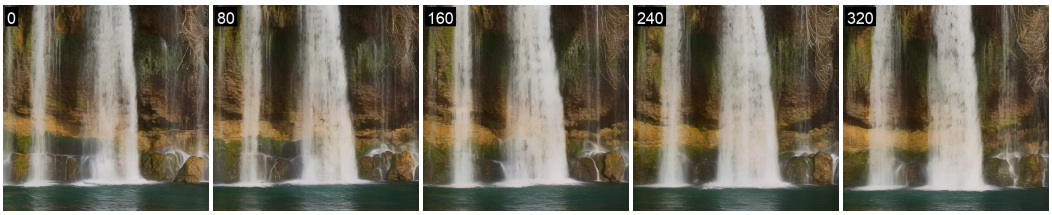
Figure 14: Videos generated by FIFO-Diffusion with zeroscope. The number on the top left of each frame indicates the frame index.

D.4 Open-Sora Plan



(a) "A quiet beach at dawn, the waves gently lapping at the shore and the sky painted in pastel hues."

(b) "A snowy forest landscape with a dirt road running through it. The road is flanked..."



(c) "The majestic beauty of a waterfall cascading down a cliff into a serene lake."

(d) "Slow pan upward of blazing oak fire in an indoor fireplace."

(e) "The dynamic movement of tall, wispy grasses swaying in the wind. The sky above is..."

(f) "a serene winter scene in a forest. The forest is blanketed in a thick layer of snow, which..."

Figure 15: Videos generated by FIFO-Diffusion with Open-Sora Plan. The number on the top left of each frame indicates the frame index.

E Multi-prompts generation for FIFO-Diffusion

E.1 Method

For multi-prompts generation, we simply change prompts sequentially during the inference. To be specific, let $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$ be k prompts, and $0 = n_0 < n_1 < \ldots < n_k$ are increasing sequence of integers. Then, we use prompt condition $\boldsymbol{c}_i$ for $(n_{i-1}+1)^{\text{th}} \sim n_i^{\text{th}}$ iterations.

E.2 Qualitative results

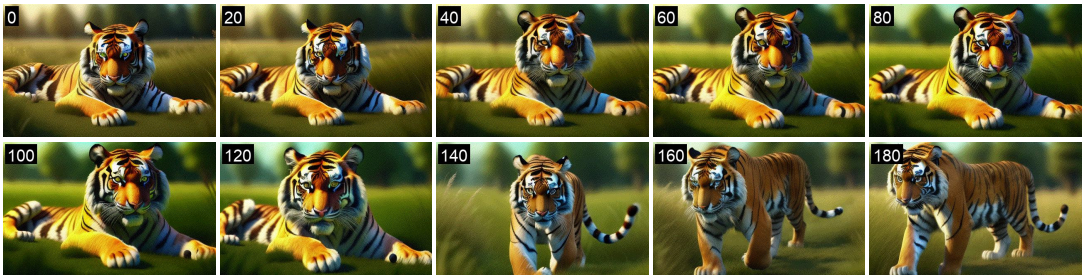
In Figures 16 and 17, we provide more qualitative results based on VideoCrafter2.

(a) "Ironman ***running*** → ***standing*** → ***flying*** on the road, 4K, high resolution."

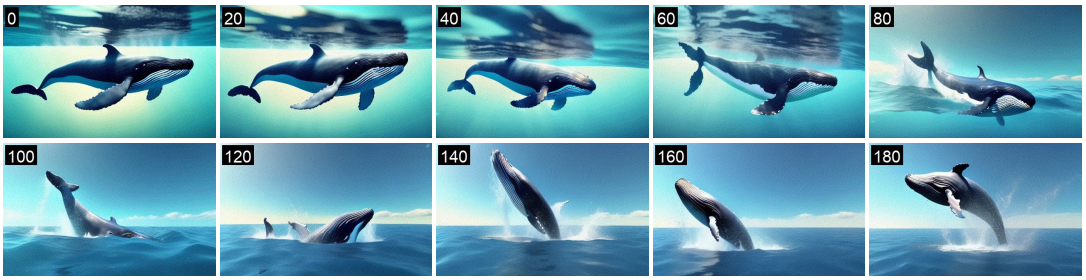
(b) "A tiger ***walking*** → ***standing*** → ***resting*** on the grassland, photorealistic, 4k, high definition"

(c) "A teddy bear ***walking*** → ***standing*** → ***dancing*** on the street, 4K, high resolution."

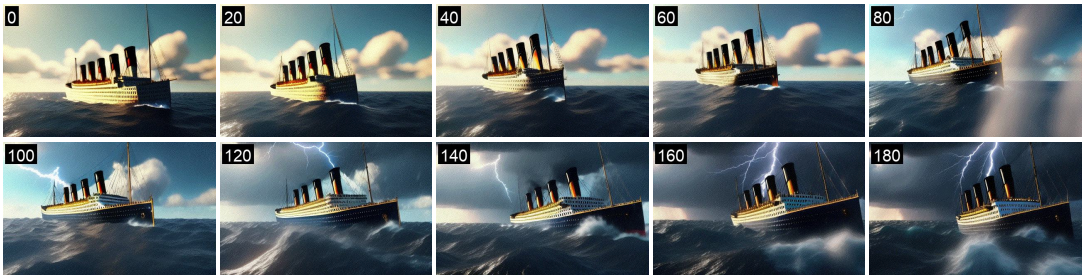
Figure 16: Videos generated by FIFO-Diffusion with three prompts. The number on the top left of each frame indicates the frame index.



(a) "A tiger ***resting*** → ***walking*** on the grassland, photorealistic, 4k, high definition"



(b) "A whale ***swimming on the surface of the ocean*** → ***jumps out of water***, 4K, high resolution."



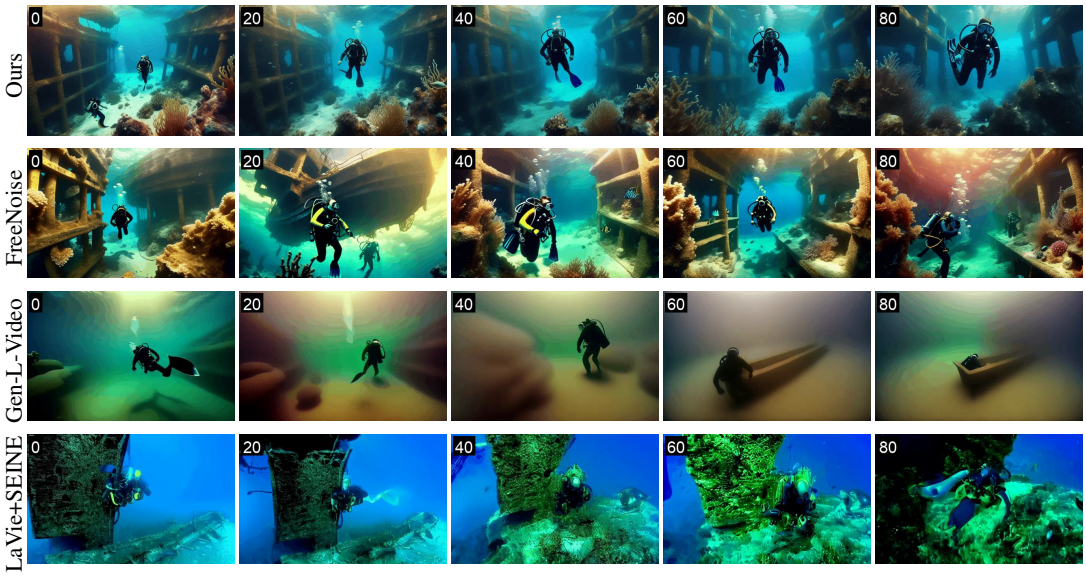
(c) "Titanic sailing through ***the sunny calm ocean*** → ***a stormy ocean with lightning***, 4K, high resolution."

(d) "A pair of tango dancers ***performing*** → ***kissing*** in Buenos Aires, 4K, high resolution."

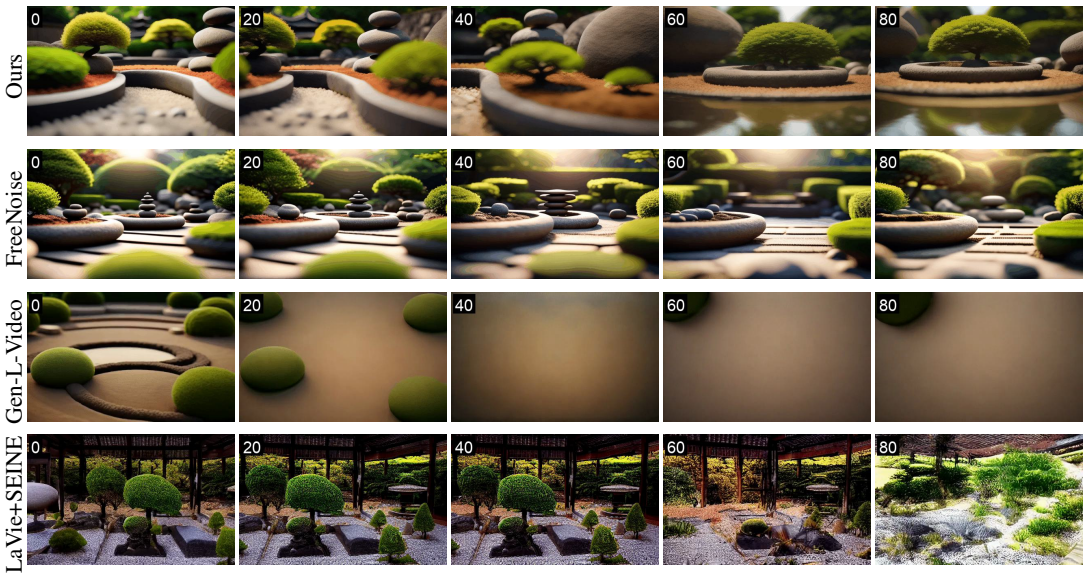
Figure 17: Videos generated by FIFO-Diffusion with two prompts. The number on the top left of each frame indicates the frame index.

F Qualitative comparisons with other long video generation methods

In Figures 18 and 19, we provide more qualitative comparisons with other longer video generation methods, FreeNoise [17], Gen-L-Video [30], and LaVie [32] + SEINE [4].



(a) "A vibrant underwater scene of a scuba diver exploring a shipwreck, 2K, photorealistic."



(b) "A panoramic view of a peaceful Zen garden, high-quality, 4K resolution."

Figure 18: Qualitative comparisons with other long video generation techniques, Gen-L-Video, FreeNoise, and LaVie + SEINE. The number in the top-left corner of each frame indicates the frame index.

(a) "A pair of tango dancers performing in Buenos Aires, 4K, high resolution."

(b) "A spooky haunted house, foggy night, high definition."

Figure 19: Qualitative comparisons with other long video generation techniques, Gen-L-Video, FreeNoise, and LaVie + SEINE. The number in the top-left corner of each frame indicates the frame index.

G Motion evaluation

We measure optical flow magnitudes (i.e. average of optical flow magnitudes) to compare the amount of motion between FIFO-Diffusion and FreeNoise, for the videos generated with randomly sampled prompts from the MSR-VTT [33] test set. Figure 20 illustrates that over 65% of videos generated by FreeNoise are located in the first bin, indicating significantly less motion compared to FIFO-Diffusion. In contrast, our method generates videos with a broader range of motion.

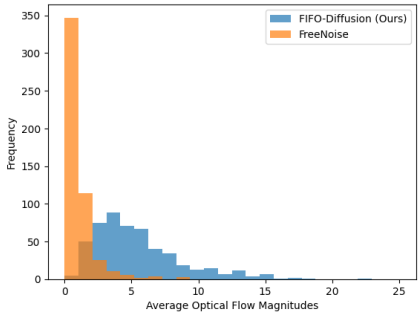
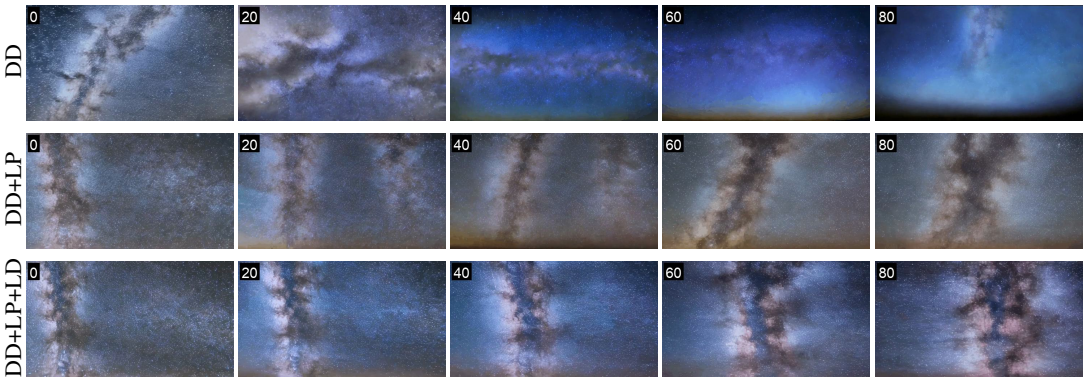


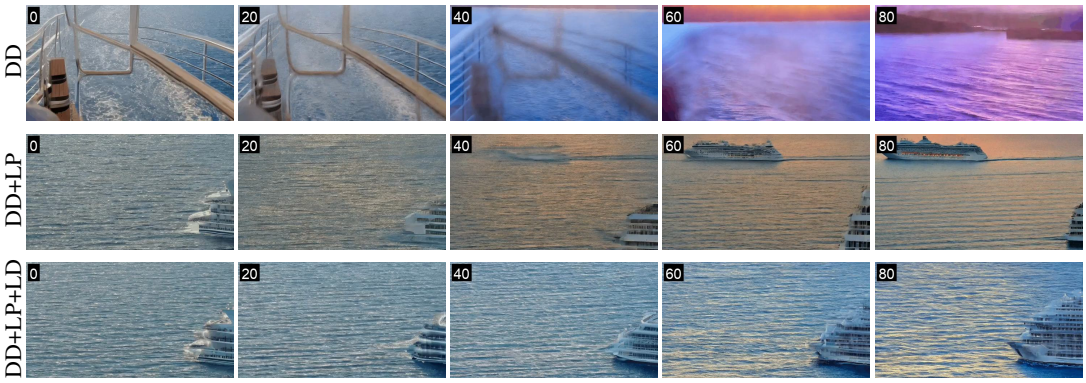
Figure 20: Comparison of optical flow magnitudes between FIFO-Diffusion and FreeNoise.

H Ablation study

In Figures 21 and 22, we conduct an ablation study to investigate the effectiveness of each component in FIFO-Diffusion. We compare the results of FIFO-Diffusion only with diagonal denoising (DD), with the addition of latent partitioning with n=4 (DD + LP), and lookahead denoising (DD + LP + LD).



(a) "A panoramic view of the Milky Way, ultra HD."

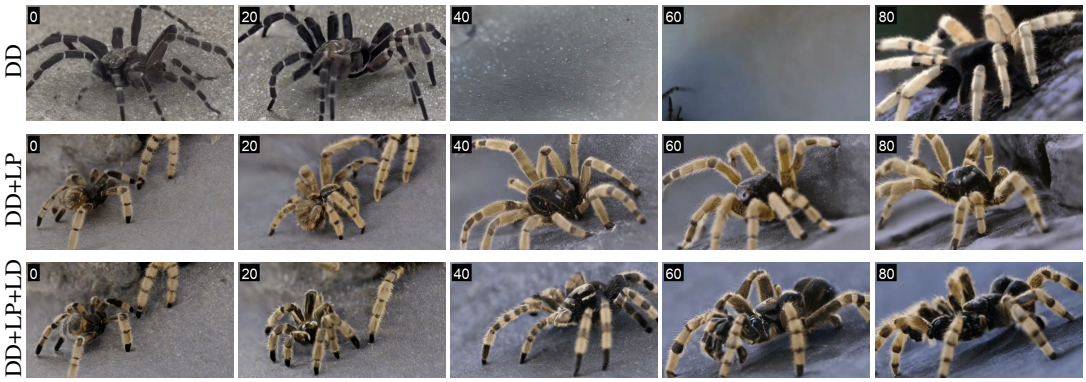


(b) "A scenic cruise ship journey at sunset, ultra HD."

(c) "A beautiful cherry blossom festival, time-lapse, high quality."

Figure 21: Ablation study. DD, LP, and LD signifies diagonal denoising, latent partitioning, and lookahead denoising, respectively. The number on the top-left corner of each frame indicates the frame index.

(a) "A detailed macro shot of a blooming rose, 4K."



(b) "A close-up of a tarantula walking, high definition."

Figure 22: Ablation study. DD, LP, and LD signifies diagonal denoising, latent partitioning, and lookahead denoising, respectively. The number on the top-left corner of each frame indicates the frame index.

I Potential Broader Impact

This paper leverages pretrained video diffusion models to generate high quality videos. The proposed method can potentially be used to synthesize videos with unexpectedly inappropriate content since it is based on pretrained models and involves no training. However, we believe that our method could mildly address ethical concerns associated with the training data of generative models.