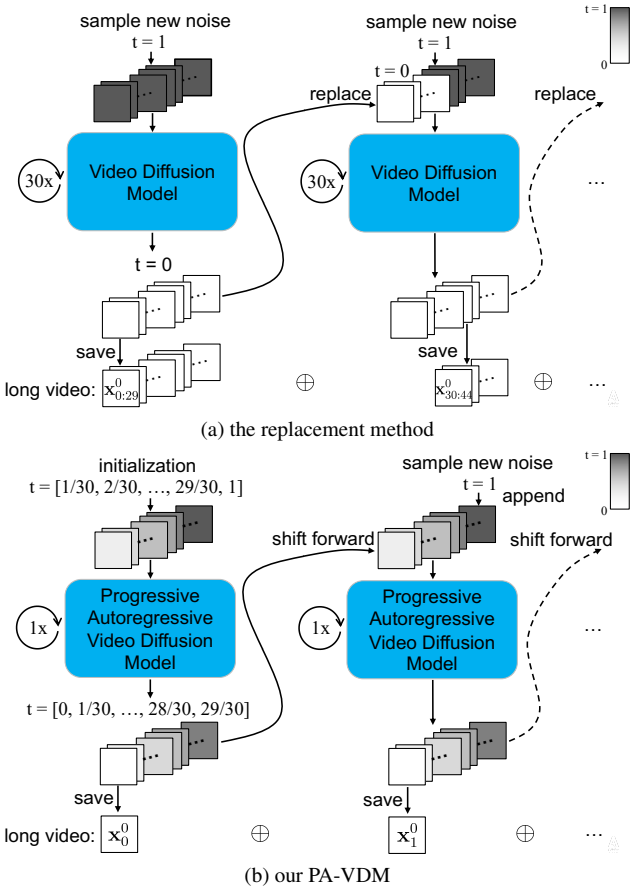
Progressive Autoregressive Video Diffusion Models

Desai Xie[†1] Zhan Xu[2] Yicong Hong[2] Hao Tan[2] Difan Liu[2]
Feng Liu[2] Arie Kaufman[1] Yang Zhou[2]

[1]Stony Brook University [2]Adobe Research

Abstract

Current frontier video diffusion models have demonstrated remarkable results at generating high-quality videos. However, they can only generate short video clips, normally around 10 seconds or 240 frames, due to computation limitations during training. Existing methods naively achieve autoregressive long video generation by directly placing the ending of the previous clip at the front of the attention window as conditioning, which leads to abrupt scene changes, unnatural motion, and error accumulation. In this work, we introduce a more natural formulation of autoregressive long video generation by revisiting the noise level assumption in video diffusion models. Our key idea is to ***1.*** *assign the frames with per-frame, progressively increasing noise levels rather than a single noise level and* ***2.*** *denoise and shift the frames in small intervals rather than all at once. This allows for smoother attention correspondence among frames with adjacent noise levels, larger overlaps between the attention windows, and better propagation of information from the earlier to the later frames.* ***Video diffusion models*** *equipped with our* ***progressive*** *noise schedule can* ***autoregressively*** *generate long videos with much improved fidelity compared to the baselines and minimal quality degradation over time. We present the first results on text-conditioned 60-second (1440 frames) long video generation at a quality close to frontier models. Code and video results are available at* https://desaixie.github.io/pa-vdm/.



(a) the replacement method

(b) our PA-VDM

Figure 1. Comparison of autoregressive long video generation methods. Top: the replacement method, which *replaces* the front of noisy latent frames with the ending of previous clip as condition and denoising all the frames at once. Bottom: our PA-VDM, which applies *progressive* noise levels and denoises and shifts the frames in small intervals. The final long video consists of autoregressively generated clean frames. $\oplus$ denotes concatenation. The noise level t for each frame is illustrated by the solid color of the frame, where darker colors are closer to 1 and lighter colors are closer to 0.

1. Introduction

Frontier video diffusion models [3, 9, 21, 22, 24, 29, 31, 34, 39, 51, 55] have recently demonstrated remarkable success in generating high-quality video contents by scaling up transformer-based [32, 48] architectures. However, they can only generate videos of relatively short duration, typically up to about 10 seconds or 240 frames, due to the demanding computation cost of long-sequence training. This temporal restriction leads to challenges for broader applications that require longer, more continuous video outputs.

†This work is done while Desai is an intern at Adobe Research.

Several approaches [1, 7, 12, 15, 58] have been proposed to autoregressively apply video diffusion models for long video generation; they generate short video clips in a windowed fashion, where each subsequent clip conditions on the final frames of the previous one. One solution [7, 58] directly places the conditioning frames into the input frames, replacing the noisy frames. Another solution [15, 43] additionally adds the same level of noise to the conditioning frames as the noisy frames. This naive way of conditioning suffers from various flaws, including temporal inconsistency, abrupt scene changes, unnatural motion dynamics, and accumulated errors that lead to divergence.

In this work, we propose *Progressive Autoregressive Video Diffusion Models* (PA-VDM) for high-quality long video generation. The core innovation of our method lies in the denoising process: instead of applying a single noise level across all frames used in traditional video diffusion models [2, 15], we apply progressively increasing noise levels across the frames; correspondingly, we denoise and shift the frames in small intervals, instead of denoising and shift them all at once. We illustrate our method in Fig. 1. Such progressive noise levels and autoregressive video denoising benefit from larger overlaps between subsequent attention windows, smoother attention correspondence among frames with adjacent noise levels, and better propagation of information from the earlier to the later frames. When applying our *variable length* progressive noise schedule, our models can start or end the autoregressive generation at arbitrary video lengths. Our *chunked frames* and *overlapped conditioning* techniques prevent divergent results and chunk-to-chunk discontinuity. Together, our method can autoregressively generate long videos while maintaining the initial quality over time.

PA-VDM provides a range of benefits for the video generation community. It can be easily implemented by changing the noise scheduling and finetuning pre-trained video diffusion models without changing the original model architecture; this allows our method to be easily reproduced and combined with orthogonal methods, such as external memory modules [12] and multiple text prompts [11, 59]. While we choose to demonstrate PA-VDM on Diffusion Transformer (DiT)-based [3, 30, 32] models, PA-VDM is model agnostic and can be extended to UNet-based [15, 37] models. As shown in Sec. 4.2, our method can work training-free, if the model has been trained on varied noise levels [58]. Moreover, the additional inference computational cost of PA-VDM is minimal without sacrificing any generation quality, as opposed to previous works [12, 36, 49] that need to trade off quality for efficiency, making this approach more efficient for practical use in long video generation.

We compare our method to the baselines on a text-conditioned 60-second (1440 frames) long video generation benchmark consisting of 40 real videos and their captions. Our quantitative results demonstrate that our results have overall the best quality across various dimensions and are the best at maintaining these metrics over the entire 60-second duration. Qualitatively, our method substantially outperforms the baselines in terms of temporal consistency, motion dynamics, and maintaining quality over time. In human evaluation, our models are also favored over various baseline models. Our ablation studies demonstrate the effectiveness of our *chunked frames* and *overlapped conditioning* techniques at preventing cumulative error and temporal jittering, respectively. By applying our method to two base models and outperforming their respective baselines, we confirm its universal applicability to existing video diffusion models. We encourage readers to check out our project webpage for video results qualitatively comparing ours and the baselines. To facilitate future research, we also release our code based on Open-Sora [58].

We summarize our contribution as follows:

1. We propose a progressive noise level schedule, an autoregressive video denoising algorithm, and the chunked frames and overlapped conditioning techniques. Together, these enable high-quality long video generation building upon pre-trained video diffusion models.
2. We are the first to achieve 60-second long video generation with quality that are close to frontier models, when compared at the same resolution. On our 60-second long video generation benchmark, we achieve superior VBench and FVD scores, majority preference in human evaluations, and strong qualitative results. This marks a significant step forward in generating longer videos, a dimension that has not been explored by recent frontier video diffusion models [9, 22, 29, 31, 34, 55].
3. Our method benefits the video generation research community in many ways, including easy implementation and reproduction, training-free application, minimal additional inference cost, and universal applicability on video diffusion models.

2. Background

2.1. Video Diffusion Models

Diffusion models [13, 40] are generative models that learn to generate samples from a data distribution $q(\mathbf{x}^0)$ through an iterative denoising process. During training, data samples are first corrupted using the forward diffusion process $q(\mathbf{x}^t|\mathbf{x}^0)$

$$q\left(\mathbf{x}^t|\mathbf{x}^0\right) = \mathcal{N}(\mathbf{x}^t; \sqrt{\alpha^t}\mathbf{x}^0, (1-\alpha^t)\boldsymbol{I}) \quad (1)$$

$$\mathbf{x}^t = \sqrt{\alpha^t}\mathbf{x}^0 + \sqrt{1-\alpha^t}\epsilon \quad (2)$$

where $t \in [0, T)$ is the noise level or diffusion timestep, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is the noise, and $\boldsymbol{\alpha}^{1:T}$ is the variance schedule. With those noisy data samples $\mathbf{x}^t$, diffusion models are trained to fit to the data distribution $q(\mathbf{x}^0)$ by maximizing

the variational lower bound [20] of the log likelihood of $\mathbf{x}^0$, which can be simplified into a mean squared error loss [13]

$$\mathcal{L}(\theta) = \left\| \epsilon - \epsilon_\theta(\mathbf{x}^t, t) \right\|^2 \tag{3}$$

where t is uniform between 0 and T, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $\boldsymbol{\epsilon}_\theta$ is the noise predicted by the model with parameters θ.

At sampling time, we consider the sampling noise level schedule $\boldsymbol{\tau} = \{\tau_0, \tau_1, ..., \tau_S\}$, which is a monotonically increasing subset of $t \in [0, T)$ of length $S+1$ [42]. Starting from $\mathbf{x}^{\tau_S} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tau_S = T$, the reverse denoising process is iteratively applied as

$$p_\theta\left(\mathbf{x}^{\tau_{i-1}} | \mathbf{x}^{\tau_i}\right) = q_\sigma\left(\mathbf{x}^{\tau_{i-1}} \middle| \mathbf{x}^\tau, f_\theta(\mathbf{x}^t, t)\right) \tag{4}$$

where $\hat{\mathbf{x}}^0 = f_\theta(\mathbf{x}^t, t)$ is the $\mathbf{x}^0$ predicted by the model and $f_\theta(\mathbf{x}^t, t)$ is the DDIM [42] reverse process equation, which we omit for simplicity. This gives us a sequence of samples $\mathbf{x}^T, \mathbf{x}^{\tau_{S-1}}, \ldots, \mathbf{x}^{\tau_1}, \mathbf{x}^0$, and the last sample $\mathbf{x}^0$ is the clean output result.

Latent video diffusion models [2, 15] are diffusion models that models latent representations of video data, consisting of F latent frames $\mathbf{x}_{0:F-1} = \{\mathrm{x}_0, \mathrm{x}_1, \ldots, \mathrm{x}_{F-1}\}$. The video latent frames are usually spatially and temporally [57] compressed through a VAE [20]. For simplicity, we refer to latent video diffusion models as video diffusion models and latent frames as frames. The same forward process, reverse process, and loss (Eqs. (1) to (4)) can be applied to model these video data by treating all the frames as one entity, ignoring the correlation among the frames. Recent video diffusion models [34, 58] have employed various diffusion model variants [25–27] to improve training and inference efficiency as well as output quality. Nevertheless, our method is compatible with any diffusion model variant as long as the model corrupts the data $\mathbf{x}^t$ at the same noise levels t.

2.2. Autoregressive Long Video Generation via Replacement

Video diffusion models can only generate short video clips, because they are only trained on videos with a limited length F due to GPU memory limit. When adapted to generating $L > F$ latent frames at sampling time, their generation quality substantially degrades [36]. The straightforward solution is to autoregressively apply video diffusion models, generating each video clip while conditioning on the previous clip. In this paper, we refer to the F frames that the video diffusion model processes as the *attention window*.

Given $E < F$ clean frames $\mathbf{x}^0_{0:E}$ as condition, there are two methods for autoregressively applying video diffusion models. [1, 7, 58] place the clean condition frames $\bar{\mathbf{x}}^0_{0:E-1}$ directly at the front of the attention window, directly replacing the sampled frames $\mathbf{x}^{\tau_i}_{0:E-1}$ at each denoising step

$$p_\theta\left(\bar{\mathbf{x}}^0_{0:E-1}, \mathbf{x}^{\tau_{i-1}}_{E:F-1} | \bar{\mathbf{x}}^0_{0:E-1}, \mathbf{x}^{\tau_i}_{E:F-1}\right) \tag{5}$$

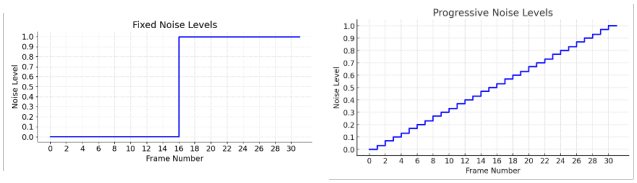


Figure 2. Comparison of noise levels of a sequence of video frames when using the *replacement without noise* method (left) and ours (right).

We will refer to this method as the *replacement-without-noise* method.

[15, 43] additionally add noise to the condition frames

$$p_\theta\left(\bar{\mathbf{x}}^{\tau_{i-1}}_{0:E-1}, \mathbf{x}^{\tau_{i-1}}_{E:F-1} \middle| \bar{\mathbf{x}}^{\tau_i}_{0:E-1}, \mathbf{x}^{\tau_i}_{E:F-1}\right) \tag{6}$$

where $\bar{\mathbf{x}}^{\tau_i}_{0:E-1}$ are the condition frames $\bar{\mathbf{x}}^0_{0:E-1}$ noised via the forward process (Eqs. (1) and (2)). This maintains the same noise level distribution and training objective as regular video diffusion models. We will refer to this method as the *replacement-with-noise* method. Note that [15] proposes reconstruction guidance for the *replacement-with-noise* method but is not widely adopted.

Both the *replacement-with-noise* method and the *replacement-without-noise* method allow a video diffusion model to autoregressively generate video frames by conditioning on previous frames. We consider them as baselines in our experiments in Sec. 4.2.

See Appendix B for a detailed discussion of two parallel works [19, 38] that share a high-level idea similar to our work. Please refer to Appendix C for related works.

3. Progressive Autoregressive Video Diffusion Models

We consider long video generation with video diffusion models. As discussed in Sec. 2.2, existing video diffusion models can only generate short video clips up to a limited length F, and the replacement methods [7, 15, 58] suffer from various flaws. We describe a more natural formulation of autoregressive long video generation, which we call *Progressive Autoregressive Video Diffusion Models* (PA-VDM). We propose a per-frame progressively increasing noise schedule, which is inspired by [4]. During training, we finetune pre-trained video diffusion models to adapt to our noise schedule; during sampling, our models adopt such noise schedule and autoregressively generate video frames.

3.1. Progressive Noise Levels and Autoregressive Generation

Conventional video diffusion methods assign a single noise level t to all the latent frames. Inspired by [4], we

Algorithm 1 Inference procedure of progressive autoregressive video diffusion models

Require: Initial video latent frames $\mathbf{x}^0_{0:F-1} = \{\mathrm{x}^0_0, \mathrm{x}^0_1, ..., \mathrm{x}^0_{F-1}\}$, maximum noise level T, number of inference steps S, and attention window size $F = S$

1: $\boldsymbol{\tau}_{0:S} = \{\tau_0, \tau_1, \ldots, \tau_S\} = \left\{0, \frac{T}{S}, \ldots, T\right\}$ ▷ Eq. (7), linear sampling noise level schedule
2: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
3: $\mathbf{x}^{\boldsymbol{\tau}_{1:S}}_{0:F-1} = \sqrt{\alpha^{\boldsymbol{\tau}_{1:S}}}\mathbf{x}^0_{0:F-1} + \sqrt{1-\alpha^{\boldsymbol{\tau}_{1:S}}}\boldsymbol{\epsilon}$ ▷ Eq. (2), add noise and set to progressive noise levels
4: **for** each autoregressive generation step $i = 1, 2, \ldots, N$ **do**
5: $\quad \mathbf{x}^{\boldsymbol{\tau}_{0:S-1}}_{0:F-1} = \left\{\mathrm{x}^0_0, \mathrm{x}^{\tau_1}_1, \ldots, \mathrm{x}^{\tau_{S-1}}_{F-1}\right\} \sim p_\theta\left(\mathbf{x}^{\boldsymbol{\tau}_{0:S-1}}_{0:F-1} \middle| \mathbf{x}^{\boldsymbol{\tau}_{1:S}}_{0:F-1}\right)$ ▷ Eq. (8), one sampling step
6: $\quad \mathrm{x}^T_{F-1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ ▷ Sample a new noisy frame
7: $\quad$ Append x^0_0 to the list of clean frames
8: $\quad \mathbf{x}^{\boldsymbol{\tau}_{0:S}}_{0:F-1} = \left\{\mathrm{x}^{\tau_1}_1, \ldots, \mathrm{x}^{\tau_{S-1}}_{F-2}, \mathrm{x}^T_{F-1}\right\}$ ▷ Remove x^0_0, shift frames forward, and append x^T_{F-1}
9: **end for**
10: **return** List of clean frames

adopt per-frame noise levels $\mathbf{t}_{0:F-1} = \{t_0, t_1, ..., t_{F-1}\}$ to the F latent frames in the attention window. In particular, we consider monotonically increasing noise levels for each frame, where earlier frames are less noisy and later frames are more noisy. In this work, we consider the linear sampling noise schedule with S sampling steps

$$\boldsymbol{\tau}_{0:S} = \left\{0, \frac{T}{S}, \frac{2T}{S}, \ldots, \frac{(S-1)T}{S}, T\right\} \tag{7}$$

which is monotonically increasing. Given a sampling noise schedule, instead of all the frames sharing a noise level and jointly going through the schedule as in conventional video diffusion models, each frame now goes through the schedule independently; at each step, the per-frame noise levels $\boldsymbol{\tau}$ still maintain the progressively increasing pattern.

Since both the sampling noise schedule and our target per-frame noise levels are monotonically increasing, we can now set per-frame noise levels $\mathbf{t}_{0:F-1}$ to be an interpolation of the sampling noise schedule $\boldsymbol{\tau}$. Let us first consider the simple case of $F = S$, when our per-frame progressive noise levels can equal to either $\mathbf{t} = \boldsymbol{\tau}_{0:S-1}$ or $\mathbf{t} = \boldsymbol{\tau}_{1:S}$. At each sampling step, the video diffusion model takes $\boldsymbol{\tau}_{0:S-1}$ as input and predicts $\boldsymbol{\tau}_{1:S}$

$$p_\theta\left(\mathrm{x}^{\tau_0}_0, \mathrm{x}^{\tau_1}_1, ..., \mathrm{x}^{\tau_{S-2}}_{F-2}, \mathrm{x}^{\tau_{S-1}}_{F-1} \middle| \mathrm{x}^{\tau_1}_0, \mathrm{x}^{\tau_2}_1, ..., \mathrm{x}^{\tau_{S-1}}_{F-2}, \mathrm{x}^{\tau_S}_{F-1}\right) \tag{8}$$

We illustrate progressive noise levels when $F = S$ in Fig. 2.

Now we construct our autoregressive generation algorithm for video latent frames with progressive noise levels. Notice that the input and output noise levels in Eq. (8), $\boldsymbol{\tau}_{0:S-1}$ and $\boldsymbol{\tau}_{1:S}$, only differ by $\tau_0 = 0$ and $\tau_S = T$. We can simply transition the output frames back into the correct input noise levels by removing the clean frame x^0_0 at the front, shifting the frame sequence forward by one frame, and appending a new noisy frame $\mathrm{x}^T_{F-1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ at back, as illustrated in Fig. 1. We describe the autoregressive generation algorithm when $F = S$ in Alg. 1. The algorithm requires a clean short video $\boldsymbol{x}^0_{0:F-1}$ as initialization and extends from it. We describe how to avoid this requirement in Sec. 3.2.

More generally, when F is a multiple of S, e.g. $F = 90, S = 30$, every set of $F/S = 3$ frames would always share the same noise level during denoising and be removed from the attention window together as they reach $t = 0$; when S is a multiple of F, e.g. $F = 10, S = 30$, the save, shift, and append operations for the sequence of frames (line 6, 7, 8 in Alg. 1) would happen once every $S/F = 3$ steps.

Note that, regardless of what the noise level a frame initially has, it always goes through the same noise level schedule $\boldsymbol{\tau}_{S:0}$ as in conventional diffusion models. Thus, for each individual frame, it is still modeled under the valid assumptions in diffusion model training [13, 25–27] and sampling [42]. We only diverge from the noise level assumption in conventional video diffusion models [15]: now, each frame is modeled independently instead of jointly with the whole sequence of frames, and the progressive autoregressive video diffusion model attends to frames with different noise levels $\mathbf{t}_{0:F-1}$ instead of the same noise level t. Thus, we can obtain our progressive autoregressive video diffusion models from pre-trained video diffusion models by adapting the model to the new noise level distribution through finetuning. This saves us from the highly demanding computation cost of video diffusion model pre-training [34, 55].

Intuitively, the benefit of our progressive video denoising process is that it gradually establishes correlation among consecutive latent frames. Given some existing video frames as conditioning, it is challenging for video diffusion models to produce temporally consistent extension frames from newly sampled noisy frames [36]. In contrast to the *replacement-with-noise* method [1, 15] where the frames are denoised together at the same noise level, our progressive video denoising encourages the later frames with higher uncertainty to follow the patterns of the earlier and more certain frames, fa-

cilitating modeling a smoother temporal transition and better preserving motion velocity. Compared to the *replacement-without-noise* method where there is a large noise level gap between the clean condition frames $\bar{\mathbf{x}}^0_{0:E-1}$ and the noisy frames $\mathbf{x}^{\tau_i}_{E:F-1}$, our method provides smoother attention correspondence, where the difference between neighboring noise levels is only $\frac{T}{S}$, as illustrated in Eq. (7) and Fig. 2.

3.2. Variable Length

The above design only allows for autoregressive video extension given an initial video of length F. In addition, the noisy frames remaining in the attention window $\mathbf{x}^{\boldsymbol{\tau}_{1:S}}_{0:F-1}$ (line 8 of Alg. 1) are discarded after the end of the autoregressive inference, which can cause wasted computing resources and inaccurate handling of the ending of text prompt. To enable text-to-long-video generation without any starting condition frames and properly ending the generation without wasting computation, we extend the base design in Eq. (8) and Alg. 1 to add an initialization stage and a termination stage, where the model operates on variable attention window lengths from 1 to $F-1$. During initialization, we simply disable the "removing x^0_0" operation in line 8 of Alg. 1: starting from a noisy frame $\{\mathrm{x}^T_0\}$, we denoise and append to obtain $\{\mathrm{x}^{\tau_{S-1}}_0, \mathrm{x}^T_1\}$; we repeat this by $F-1$ times to obtain $\mathbf{x}^{\boldsymbol{\tau}_{1:S}}_{0:F-1} = \{\mathrm{x}^{\tau_1}_1, \ldots, \mathrm{x}^{\tau_{S-1}}_{F-2}, \mathrm{x}^T_{F-1}\}$, i.e. the input to line 5 of Alg. 1. During termination, we disable the "append x^T_{F-1}" operation in line 6 and 7 of Alg. 1: starting with F frames $\mathbf{x}^{\boldsymbol{\tau}_{1:S}}_{0:F-1} = \{\mathrm{x}^{\tau_1}_0, \ldots, \mathrm{x}^{\tau_{S-1}}_{F-2}, \mathrm{x}^T_{F-1}\}$, we denoise, save and remove to obtain $\mathbf{x}^{\boldsymbol{\tau}_{1:S-1}}_{0:F-2} = \{\mathrm{x}^{\tau_1}_0, \ldots, \mathrm{x}^{\tau_{S-1}}_{F-2}\}$; we repeat this by F times to save and remove all the remaining frames in the attention window. We train the model accordingly on video latent frames with variable lengths ranging from 1 to $F-1$, following the noise levels described above.

3.3. Chunked Frames

3D VAEs [20, 34, 57, 58] usually encode and decode video latent frames chunk-by-chunk. In our early experiments, we find that naively implementing our method on latent video diffusion models, i.e. when all latent frames are given different noise levels and the attention window is shifted by one frame at a time, leads to serious cumulative error and the videos diverge quickly after a few seconds, as shown in Ablation 2 in Fig. 6. We resolve the problem by *treating a chunk of latent frames as a whole*: they are assigned with the same noise level, and are added and removed from the attention window together. In other words, for a 3D VAE chunk size of C latent frames, e.g. $C=5$ as mentioned in Sec. 4, we shift the attention window by C frames every C sampling steps. Effectively, the C frames that belong to the same chunk always have the same noise level t and are added to or removed from the attention window together. Our ablation experiments shows that, for models using a 3D VAE, treating a chunk of frames as a whole effectively prevents accumulated errors that would lead to divergence.

3.4. Overlapped Conditioning

In our early experiments, naively implementing our method on video diffusion models results in temporal jittering. We hypothesize that this is because the clean frames $\mathbf{x}^0_{0:C-1}$ are immediately removed from the attention window; as the later frames cannot attend to the previous clean frames, it is hard for the model to denoise the later frames to be perfectly temporally consistent with the previous clean frames. In practice, we always keep a chunk of C clean frames by prepending it to the attention window. Our ablation study shows that overlapped conditioning helps resolving the frame-to-frame discontinuity issue.

Overlapped conditioning requires an additional inference cost at C/F ($5/50$ in our implementation) of the original cost. When using the same number of conditioning frames E and F, the replacement methods [7, 15, 58] and ours have the same inference efficiency. The key advantage of our method is that the large overlap of noisy frames enables the model to preserve the high-level information—such as motion—from prior frames. Thus, we only need a single chunk of C clean condition frames to propagate high-frequency details and prevent per-chunk temporal jittering. In contrast, the replacement methods need to balance the tradeoff between more overlap between video clips or better inference efficiency. In practice, their implementation [58] often use one chunk of frames as condition to save inference computation, but the limited overlap causes unnatural motion transition and abrupt scene changes across clips, as discussed in Sec. 4.2.

3.5. Training

As described in Sec. 3.1, PA-VDM requires change in the noise level distribution. We finetune pre-trained video diffusion model to adapt to our progressive noise level distribution. Conventional diffusion model training [13, 26, 27] involves uniformly sampling a noise level $\mathrm{t} \in [0, T)$, adding noise to the samples $\mathbf{x}^0_{0:F-1}$ via the forward diffusion process (Eqs. (1) and (2)), and computing the loss (Eq. (3)). During the finetuning process for PA-VDM, we simply continue with the conventional video diffusion model training but with our per-frame progressive training noise levels $\mathbf{t}_{0:F-1}$. In our experiment, we observed that, similar to the sampling noise levels $\boldsymbol{\tau}_{0:S}$ in Eq. (7), training on a simple linear noise schedule yielded satisfactory results for all reported experiments. During training, the noise levels $\mathbf{t}$ is perturbated by a random shift δ to fully cover of the diffusion timestep range $[0, T)$ [41]. $\delta = 0.4\epsilon(\mathrm{t}_i - \mathrm{t}_{i+1}), \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ is randomly sampled for each training iteration and remains constant for all $\mathbf{t}_{0:F-1}$ within that iteration.

Table 1. Quantitative comparison of our progressive autoregressive video generation (PA) and two baseline methods *replacement-with-noise* (RW) and *replacement-without-noise* (RN) on two base models (*M* and *O*), and other baselines StreamingT2V [12], Stable Video Diffusion (SVD) [1], and FIFO-Diffusion [19].

	Subject Consistency ↑	Background Consistency ↑	Motion Smoothness ↑	Dynamic Degree ↑	Aesthetic Quality ↑	Imaging Quality ↑	Num Scenes ↓	FVD ↓
PA-*M* (ours)	0.7923	**0.8964**	0.9896	0.8000	**0.4726**	**0.5927**	1.75	**358.020**
RW-*M*	0.8001	0.8851	0.9836	0.3958	0.4123	**0.5961**	1.10	669.747
PA-*O*-base (ours)	0.7656	0.8880	0.9859	0.5625	0.4582	0.5033	2.04	548.117
RN-*O*-base	0.7406	0.8820	0.9873	0.5750	0.4034	0.4464	5.19	600.690
StreamingSVD	**0.8172**	0.8916	**0.9929**	0.65	0.4264	0.5566	**1.08**	440.272
SVD-XT	0.6102	0.8136	0.9724	**0.9875**	0.3019	0.4814	2.10	702.343
FIFO-OSP	0.7577	0.8990	0.9731	0.75	N/A	0.5675	18.32	975.459

4. Experiments

4.1. Implementation

Our models and baseline models We implement our progressive autoregressive video diffusion models by fine-tuning from pre-trained models. Specifically, we use two video diffusion models based on the diffusion transformer architecture [3, 32]: Open-Sora v1.2 [58] (denoted as *O*) and a modified variant of Open-Sora (denoted as *M* in later experiments). Both models are latent video diffusion models [2], each utilizing a corresponding 3D VAE that encodes 17 (*O*) or 16 (*M*) raw video frames into 5 latent frames. *O* generates videos at 240×424 resolution 24 FPS with 30 sampling steps. *M* produces results at 176×320 resolution 24 FPS with 50 sampling steps. Based on *O* and *M*, we also implement two baseline autoregressive video generation methods, *replacement-with-noise* (denoted as RW) and *replacement-without-noise* (denoted as RN) (Sec. 2.2), to compare with our proposed *progressive autoregressive* (denoted as PA) video generation method (Sec. 3).

We train *M* on our progressive noise levels, as discussed in Sec. 3.5. The resulting model can perform progressive autoregressive video generation, which we denote as PA-*M*. We also train *M* with the *replacement-with-noise* method, which we will denote as RW-*M*. Starting from the same pre-trained weight of the base model, RW-*M* is trained for 3 times more training steps compared to PA-*M*.

O undergoes masked pre-training [58], where the masked latent frames $\mathbf{x}^0_{0:E-1}$ are clean without any added noise [58]. This allows the *O* base model to perform autoregressive video generation with the *replacement-without-noise* method. We denote this model as RN-*O*-base. Such training also allows *O* to learn that the noise levels $\mathbf{t}_{0:F-1}$ can be independent with respect to the latent frames and thus enables our *progressive autoregressive* video denoising sampling procedure (Alg. 1) to work training-free. We denote this model as PA-*O*-base. Please refer to Appendix E for training details.

4.2. Long video generation

The baseline methods are described in Appendix F.1.

Metrics We consider 6 metrics in VBench [17]: subject consistency, background consistency, motion smoothness, dynamic degree, aesthetic quality, and imaging quality. We compute average metrics using VBench-long, where each metric is computed on 30 2-second clips for each 60-second video; for subject and background consistency, a clip-to-clip metric is considered in addition to the average metric over the clips. We also show how the metrics vary over time by plotting the metrics over the 30 2-second clips averaged over the 80 60-second videos.

Similar to [12], we also use the Adaptive Detector algorithm from PySceneDetect [35] to count the number of detected scene changes, where Num Scenes = 1 means that there is no scene change detected.

We also compute Fréchet Video Distance (FVD) [47] to measure the overall quality of the generated videos compared to real videos. We adopt the improved implementation of FVD proposed in [8] using the VideoMAE-v2 [50] model. The FVD metric usually requires a large number of video samples in order to produce a reliable value. Since our testing set includes only 40 real videos and each model only generate 80 videos, naively computing FVD on them results in erroneous values such as -3.62e+64. Instead, we compute FVD on the 2-second clips of the long videos, so that we have 1495 real videos and 2400 generated videos.

Quantitative Results We present the average metrics for each model in Tab. 1. The metrics are averaged over all the videos that each model generates from our testing set described above. Our PA-*M* has the best results overall. Notably, it surpasses other methods in FVD by a substantial margin, illustrating that its results are the most realistic. It also achieves either the best or close-to-best in other met-

rics. Its *replacement-with-noise* counterpart, RW-*M*, suffers from poor Dynamic Degree and FVD, because its videos are mostly static. Our RW-*O*-base surpasses its *replacement-without-noise* counterpart RN-*O*-base in all metrics except for being close at Dynamic Degree, while using the exact same model parameters without any finetuning. RN-*O*-base mainly suffers from a high number of scene changes.

In Fig. 3, we also illustrate the trend of metrics over the 1-minute duration of videos for each model. Our models *M*-PA and *O*-PA can best maintain the level of all metrics, while their *replacement*-method counterparts, *M*-RW and *O*-RN, both exhibit distinct reduction in dynamic degree, aesthetic quality, and imaging quality.

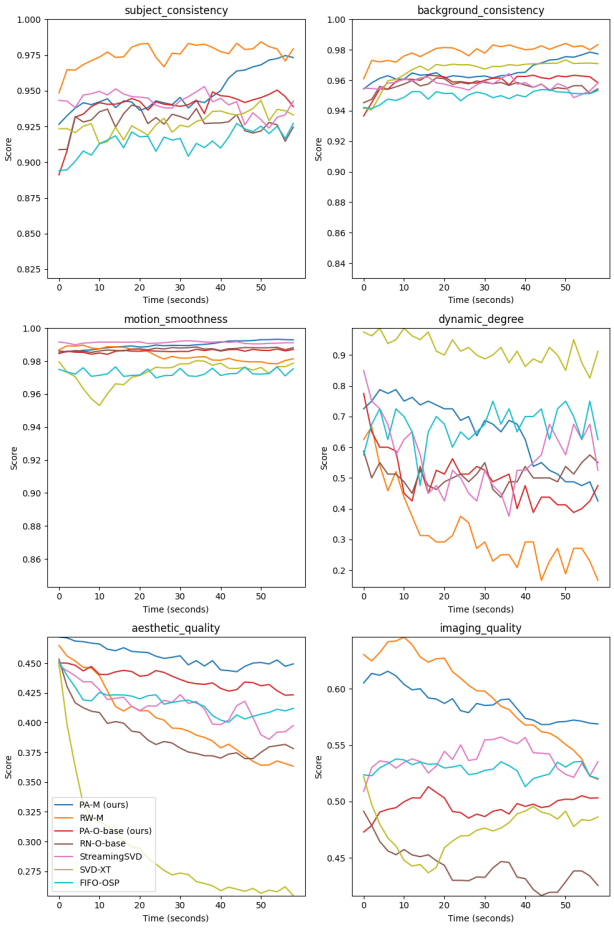


Figure 3. VBench [17] scores over the 60-second duration, which are computed on 30 2-second clips.

Qualitative Results We also show strength of our method with qualitative comparison results in Fig. 5. Both of our models demonstrate strong performance in terms of frame fidelity and motion realism (e.g. camera motion, wave motion, and running gestures) and outperforms other baselines. For more qualitative results, please refer to our project webpage.

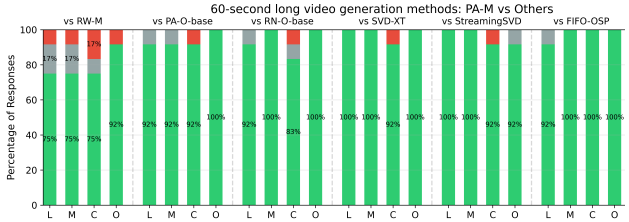


Figure 4. Human evaluation results comparing long video methods on long-shot (L), motion (M), temporal consistency (C), and overall (O).

User study We conduct a human evaluation with 12 users to compare the generated videos from each method. As shown in Fig. 4, our PA-*M* is favored in each duel by a large margin.

4.3. Ablation Study

We conduct ablation studies on the PA-*M* model to evaluate the impact of chunked frames (Sec. 3.3), and overlapped conditioning (Sec. 3.4). Qualitative comparison is shown in Fig. 6 and in the project webpage. In Ablation 1, we observe that the absence of clean frames in the input sequence prevents noisy frames from attending to previous clean frames, resulting in poor performance over a long duration. This also causes frame-to-frame discontinuity, which is more noticeable in the project webpage. In Ablation 2, not decoding the video chunk-by-chunk leads to severe cumulative errors, causing the video to diverge after only a few seconds.

See Appendix H for additional ablation study on variable length and the number of sampling steps S.

5. Conclusion

In this work, we target long video generation, a fundamental challenge of current video diffusion models. We show that they can be naturally adapted to become progressive autoregressive video diffusion models without changing the architectures. With our progressive noise levels and the autoregressive video denoising process (Sec. 3.1), we achieve state-of-the-art results on 60-second long video generation. Since our method does not require model architecture changes, it can be seamlessly combined with orthogonal works, paving the way for generating longer videos at higher quality, long-term dependency, and controllability.

6. Acknowledgments

This research was supported in part by NSF award IIS2107224 and ONR award N000142312124.

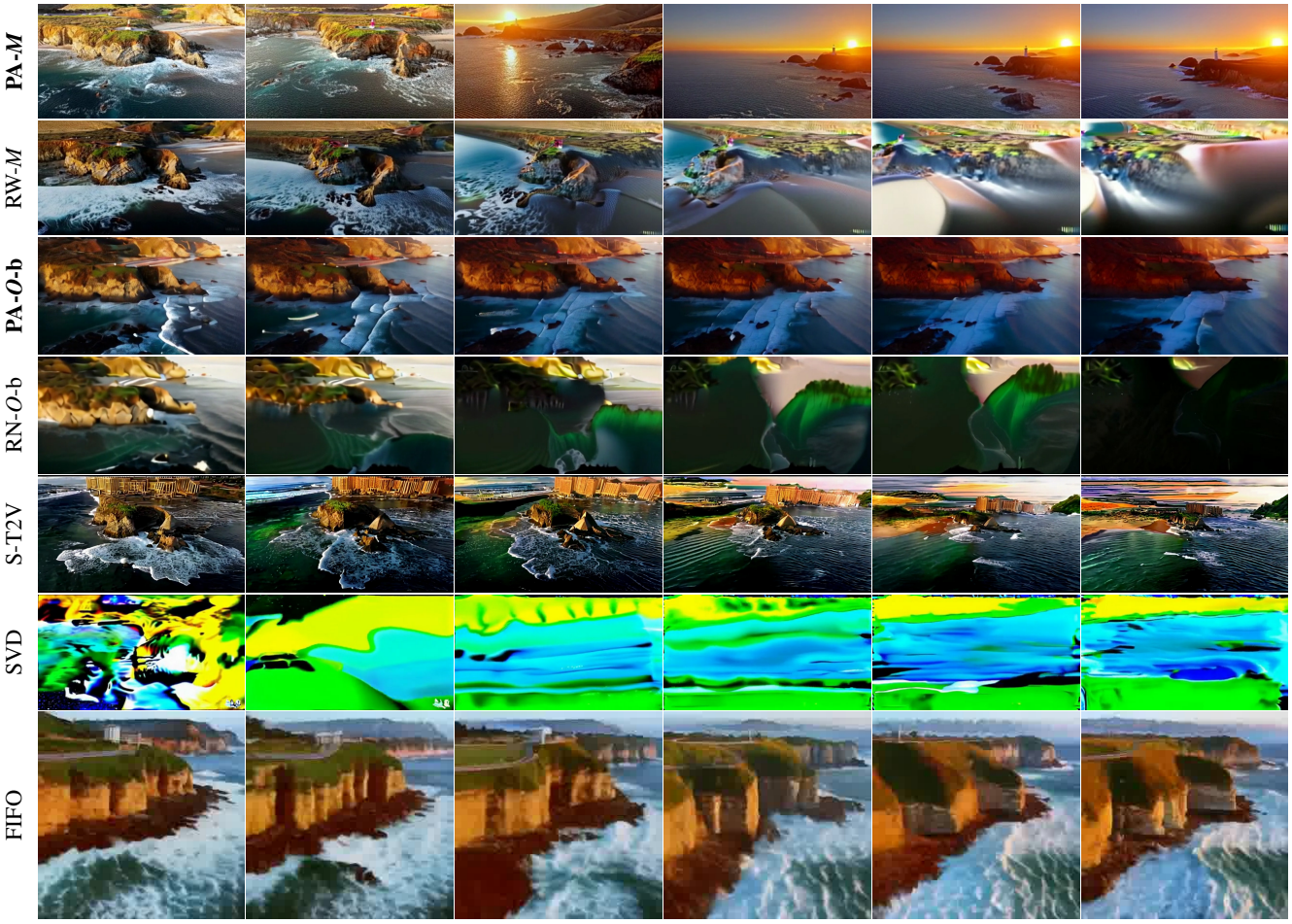


Figure 5. Qualitative comparison of PA-*M* (ours), RW-*M*, PA-*O*-base (ours), RN-*O*-base, StreamingSVD from StreamingT2V [12], SVD-XT from Stable Video Diffusion [1], and FIFO-Diffusion [19]. Frames are evenly sampled from 1 minute long generated video, i.e. at 10, 20, 30, 40, 50, and 60 seconds. Our models can autoregressively generate 60-second, 1440-frame videos without quality degradation.

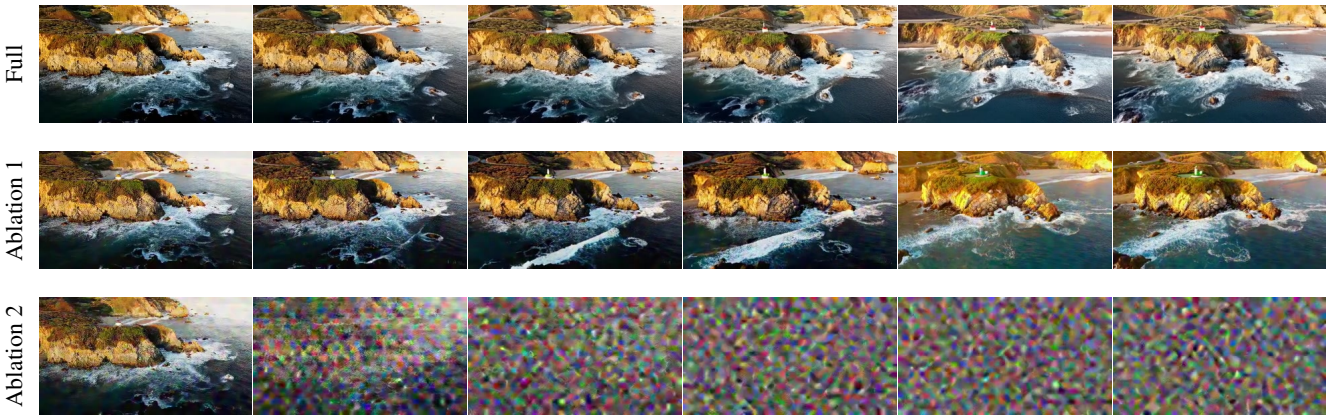


Figure 6. Qualitative comparison for ablation study. Full represents for our full solution based on PA-*M*, Ablation 1 is with *chunked frames* but without *overlapped conditioning*. Ablation 2 is without both techniques. The frames are evenly sampled from 16-second generated videos.

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, 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*, 2023. 2, 3, 4, 6, 8, 1

[2] 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, 2023. 2, 3, 6

[3] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1, 2, 6

[4] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion, 2024. 3, 1

[5] 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*, 2023. 1

[6] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2

[7] Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, and Jun Xiao. ViD-GPT: introducing GPT-style autoregressive generation in video diffusion models, 2024. 2, 3, 5, 1

[8] Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in fréchet video distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7277–7288, 2024. 6

[9] Genmo. Mochi 1 preview. https://www.genmo.ai/blog, 2024. Accessed: 2024-11-13. 1, 2

[10] 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. 1

[11] Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation, 2025. 2

[12] 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, 2024. 2, 6, 8, 1, 3, 4

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3, 4, 5

[14] 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*, 2022. 1

[15] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. 2, 3, 4, 5, 1

[16] Li Hu. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8153–8163, 2024. 1

[17] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol 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. 6, 7

[18] Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions, 2024. 3

[19] Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *arXiv preprint arXiv:2405.11473*, 2024. 3, 6, 8, 1, 4

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. 3, 5

[21] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025. 1

[22] Kuaishou. Kling. https://wwwhtbprolklingai.htbprolcom-s.e 2024. Accessed: 2024-11-13. 1, 2

[23] PKU-Yuan Lab and Tuzhan AI etc. Open-sora-plan, 2024. 3

[24] Zongyu Lin, Wei Liu, Chen Chen, Jiasen Lu, Wenze Hu, Tsu-Jui Fu, Jesse Allardice, Zhengfeng Lai, Liangchen Song, Bowen Zhang, Cha Chen, Yiran Fei, Yifan Jiang, Lezhi Li, Yizhou Sun, Kai-Wei Chang, and Yinfei Yang. Stiv: Scalable text and image conditioned video generation, 2024. 1

[25] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. 3, 4

[26] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. 5

[27] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023. 3, 4, 5

[28] Yu Lu, Yuanzhi Liang, Linchao Zhu, and Yi Yang. Freelong: Training-free long video generation with spectralblend temporal attention. *arXiv preprint arXiv:2407.19918*, 2024. 1

[29] Luma. Dream machine. https://lumalabs.ai/dream-machine, 2024. Accessed: 2024-11-13. 1, 2

[30] 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. 2

[31] Runway ML. Gen-3 alpha. https://runwayml.com/research/introducing-gen-3-alpha, 2024. Accessed: 2024-11-13. 1, 2

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1, 2, 6, 3

[33] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 2

[34] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models, 2024. 1, 2, 3, 4, 5

[35] PySceneDetect. PySceneDetect. https://www.scenedetect.com/. Accessed: 2024-10-10. 6

[36] Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. FreeNoise: tuning-free longer video diffusion via noise rescheduling, 2024. 2, 3, 4, 1

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2

[38] David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. *arXiv preprint arXiv:2402.09470*, 2024. 3, 1

[39] Team Seawead, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, Feng Cheng, Feilong Zuo Xuejiao Zeng, Ziyan Yang, Fangyuan Kong, Zhiwu Qing, Fei Xiao, Meng Wei, Tuyen Hoang, Siyu Zhang, Peihao Zhu, Qi Zhao, Jiangqiao Yan, Liangke Gui, Sheng Bi, Jiashi Li, Yuxi Ren, Rui Wang, Huixia Li, Xuefeng Xiao, Shu Liu, Feng Ling, Heng Zhang, Houmin Wei, Huafeng Kuang, Jerry Duncan, Junda Zhang, Junru Zheng, Li Sun, Manlin Zhang, Renfei Sun, Xiaobin Zhuang, Xiaojie Li, Xin Xia, Xuyan Chi, Yanghua Peng, Yuping Wang, Yuxuan Wang, Zhongkai Zhao, Zhuo Chen, Zuquan Song, Zhenheng Yang, Jiashi Feng, Jianchao Yang, and Lu Jiang. Seaweed-7b: Cost-effective training of video generation foundation model, 2025. 1

[40] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2

[41] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 3, 4

[43] 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*, 2020. 2, 3

[44] K Soomro. UCF101: a dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 3

[45] Linrui Tian, Qi Wang, Bang Zhang, and Liefeng Bo. Emo: Emote portrait alive-generating expressive portrait videos with audio2video diffusion model under weak conditions. *arXiv preprint arXiv:2402.17485*, 2024. 1

[46] Ye Tian, Ling Yang, Haotian Yang, Yuan Gao, Yufan Deng, Jingmin Chen, Xintao Wang, Zhaochen Yu, Xin Tao, Pengfei Wan, et al. Videotetris: Towards compositional text-to-video generation. *arXiv preprint arXiv:2406.04277*, 2024. 1

[47] 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:1812.01717*, 2018. 6

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 1, 2

[49] 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, 2023. 2, 1

[50] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560, 2023. 6

[51] WanTeam, :, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. 1

[52] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023. 3

[53] Jay Zhangjie Wu, Xiuyu Li, Difei Gao, Zhen Dong, Jinbin Bai, Aishani Singh, Xiaoyu Xiang, Youzeng Li, Zuwei Huang, Yuanxi Sun, Rui He, Feng Hu, Junhua Hu, Hai Huang, Hanyu Zhu, Xu Cheng, Jie Tang, Mike Zheng Shou, Kurt Keutzer, and Forrest Iandola. Cvpr 2023 text guided video editing competition, 2023. 3

[54] Desai Xie, Jiahao Li, Hao Tan, Xin Sun, Zhixin Shu, Yi Zhou, Sai Bi, Sören Pirk, and Arie E Kaufman. Carve3d: Improving multi-view reconstruction consistency for diffusion models with rl finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6369–6379, 2024. 2

[55] 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. 1, 2, 4

[56] Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. *arXiv preprint arXiv:2303.12346*, 2023. 1

[57] Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation, 2024. 3, 5

[58] 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, 2024. 2, 3, 5, 6

[59] Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. In *Advances in Neural Information Processing Systems*, pages 110315–110340. Curran Associates, Inc., 2024. 2

[60] Shenhao Zhu, Junming Leo Chen, Zuozhuo Dai, Yinghui Xu, Xun Cao, Yao Yao, Hao Zhu, and Siyu Zhu. Champ: Controllable and consistent human image animation with 3d parametric guidance. *arXiv preprint arXiv:2403.14781*, 2024. 1

[61] Shaobin Zhuang, Kunchang Li, Xinyuan Chen, Yaohui Wang, Ziwei Liu, Yu Qiao, and Yali Wang. Vlogger: Make your dream a vlog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8806–8817, 2024. 1