Snap Video: Scaled Spatiotemporal Transformers for Text-to-Video Synthesis

Willi Menapace[1,2,*] Aliaksandr Siarohin[1] Ivan Skorokhodov[1] Ekaterina Deyneka[1]
Tsai-Shien Chen[1,3,*] Anil Kag[1] Yuwei Fang[1] Aleksei Stoliar[1] Elisa Ricci[2,4]
Jian Ren[1] Sergey Tulyakov[1]

Snap Inc.[1] University of Trento[2] UC Merced[3] Fondazione Bruno Kessler[4]

snap-research.github.io/snapvideo

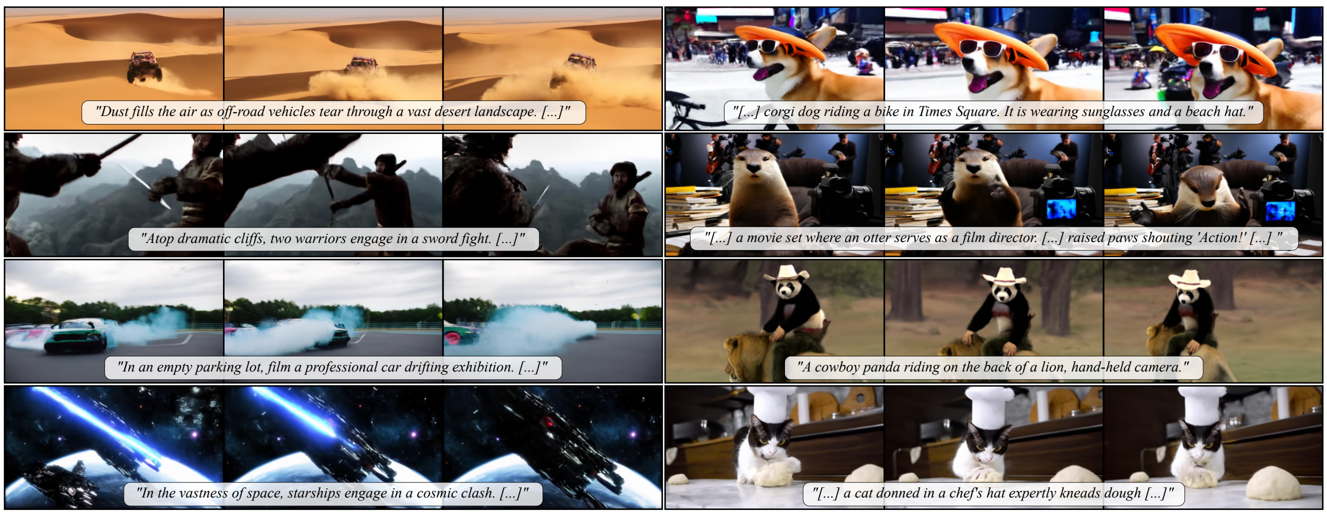


Figure 1. Samples produced by the proposed text-to-video generation method for a selection of prompts. Thanks to joint spatiotemporal video modeling, our generator can synthesize temporally coherent videos with large motion (left) while retaining the semantic control capabilities typical of large-scale text-to-video generators (right). See the *Website* for additional samples.

Abstract

Contemporary models for generating images show remarkable quality and versatility. Swayed by these advantages, the research community repurposes them to generate videos. Since video content is highly redundant, we argue that naively bringing advances of image models to the video generation domain reduces motion fidelity, visual quality and impairs scalability. In this work, we build Snap Video, a video-first model that systematically addresses these challenges. To do that, we first extend the EDM framework to take into account spatially and temporally redundant pixels and naturally support video generation. Second, we show that a U-Net—a workhorse behind image generation—scales poorly when generating videos, requiring significant computational overhead. Hence, we propose a new transformer-based architecture that trains 3.31 times faster than U-Nets (and is ∼4.5 faster at inference). This allows us to efficiently train a text-to-video model with billions of parameters for the first time, reach state-of-the-art results on a number of benchmarks, and generate videos with substantially higher quality, temporal consistency, and motion complexity. The user studies showed that our model was favored by a large margin over the most recent methods.

1. Introduction

Creating and sharing visual content is one of the key ways for people to express themselves in the digital world. Accessible to only professionals in the past, the capability to create [30, 40, 43, 69] and edit [6, 37, 42] images with stunning quality and realism was unlocked to everyone by the advent of large text-to-image models and their variations.

Fueled by this progress, large-scale text-to-video models [4, 13, 21, 48, 62] are rapidly advancing too. Current large-scale diffusion-based video generation frame-

[*] Work performed while interning at Snap Inc.

works are strongly rooted into their image counterparts [4, 13]. The availability of consolidated image generation architectures such as U-Nets [41] with publicly-available image-pretrained models [40] made them a logical foundation onto which to build large-scale video generators with the main architectural modifications focusing on the insertion of ad-hoc layers to capture temporal dependencies [4, 13, 21, 48, 62]. Similarly, training is performed under image-based diffusion frameworks with the model being applied both to videos and to a separate set of images to improve the diversity of the results [13, 21, 22, 48].

We argue that such an approach is suboptimal under multiple aspects which we systematically address in this work. First, image and video modalities present intrinsic differences given by the similarity of content in successive video frames [7, 13]. By analogy, image and video compression algorithms are based on vastly different approaches [33]. To address this issue, we rewrite the EDM [25] framework with a focus on high-resolution videos. Differently from past work where videos were treated as a sequence of images, we perform joint video-image training by treating images as *high frame-rate videos* to avoid modality mismatches introduced by the absence of the temporal dimension within purely image-based training. Second, the widely adopted U-Net [41] architecture is required to fully processes each video frame. This increases computational overhead compared to purely text-to-image models, posing a very practical limit on model scalability. The latter is a critical factor in obtaining high-quality of results [13, 21]. Extending U-Net-based architectures to naturally support spatial and temporal dimensions requires volumetric attention operations, which have prohibitive computational demands. Inability to do so affects the outputs, resulting in *dynamic images* or motion artifacts being generated instead of videos with coherent and diverse actions.

Following our compression analogy, we propose to leverage repetition between frames and introduce a scalable transformer architecture that treats spatial and temporal dimensions as a single, compressed, 1D latent vector. This highly compressed representation allows us to perform spatio-temporal computation jointly and enables modelling of complex motions. Our architecture is inspired by FIT [8], which we scale to billions of parameters for the first time. Compared to U-Nets, our model features a significant $3.31\times$ reduction in training time and $4.49\times$ reduction in inference time while achieving higher generation quality.

We evaluate Snap Video on the widely-adopted UCF101 [55] and MSR-VTT [65] datasets. Our generator shows state-of-the-art performance across the range of benchmarks with particular regard to the quality of the generated motion. Most interestingly, we performed a number of user studies against the most recent open- and close-source methods and found that according to the participants of the study our model features photorealism comparable to Gen-2 [11], while being significantly better than Pika [1] and Floor33 [17]. Most excitedly, the preference of user-study participants favoured Snap Video by a large margin when text alignment and motion quality were assessed. Compared to Gen-2 [11] on prompt-video alignment our model was preferred in 81% of cases (80% against Pika [1], 81% against Floor33 [17]), generated most dynamic videos with most amount of motion (96% against Gen2 [11], 89% against Pika [1], 88% against Floor33 [17]) and had the best motion quality (79% against Gen-2 [11], 71% against Pika [1], 79% against Floor33 [17]).

2. Related Work

Video Generation Video generation is a challenging and long-studied task. Due to its complexity, a large number of works focus on modeling narrow domains [5, 9, 12, 28, 35, 44, 47, 49, 56, 58, 59, 66, 70, 71] and adopt adversarial training [5, 9, 28, 44, 47, 49, 58, 59, 71] or autoregressive generation techniques [12, 35, 56, 66, 70]. To address the narrow domain limitation, the task of text-to-video generation was proposed [34] and both autoregressive models [23, 34, 61, 63, 64] and GANs [29] emerged.

The recent success of diffusion models in the context of text-to-image generation [3, 40, 43] fostered tremendous progress in the task [2, 4, 13, 16, 17, 21, 22, 32, 48, 62, 67, 72]. ImagenVideo [21] and Make-A-Video [48] propose a deep cascade of temporal and spatial upsamplers to generate videos and jointly train their models on image and video datasets. PYoCo [13] introduces a correlated noise model to capture similarities between video frames. Video LDM [4] adopts a latent diffusion paradigm where a pre-trained latent image generator and latent decoder are fine-tuned to generate temporally coherent videos. AnimateDiff [16] freezes a pre-trained latent image generator and trains only a newly inserted motion modeling module. These works employ U-Nets with separable spatial and temporal computation which poses a limitation on motion modeling capabilities. VideoFactory [62] improves upon this paradigm by proposing a Swapped Spatiotemporal Cross-Attention that improves interactions between the spatial and temporal modalities along 3D windows.

Differently from this corpus of works which adapts the U-Net [41] architecture to the video generation task, we show that employing transformer-based FIT [8] architectures results in significant training time savings, scalability improvements, and performance increase thanks to their learnable compressed video representation. In particular, we show that the global joint spatiotemporal modeling strategy enabled by our compressed video representation results in significant improvements in temporal consistency and motion modeling capabilities.

High-Resolution Generation Different approaches have been proposed to enable the generation of high-resolution outputs. Cascaded diffusion models [3, 13, 21, 43, 48] adopt a set of independent diffusion models designed to successively upsample the results of the previous step. Latent diffusion models [2, 4, 17, 40, 72] make use of a pretrained autoencoder to encode the input into a low-dimensional set of latent vectors and learn a diffusion model on this latent representation.

A different family of methods generates high-resolution outputs end-to-end without employing cascades of models or latent diffusion. Simple Diffusion [24] and *Chen* [7] directly generate high-resolution images by adapting the noise schedule of the diffusion process. f-DM [14] and RDM [57] design a diffusion process that seamlessly transitions between different resolutions. MDM [15] proposes a strategy where a single model is trained to simultaneously denoise inputs at progressively higher resolutions.

In this work, we adopt a two-stage cascaded model out of two considerations: (i) it avoids temporal inconsistencies in the forms of flickering of high-frequency details that may be introduced by latent autoencoders [4], (ii) it increases model capacity with respect to an end-to-end model by creating two specialized models, one for the low resolution focusing on motion modeling and scene structure, and one for the high-resolution, focusing on high-frequency details.

Diffusion Frameworks Diffusion generative models are a set of techniques modeling generation as a pair of processes: a forward process progressively destructing a sample with noise, and a reverse process modeling generation as the progressive denoising of a sample. Different formulations of diffusion models have been proposed in the literature. Denoising Diffusion Probabilistic Models (DDPMs) [20, 50] formulate the forward and backward process as Markov chains. Score-based Generative Models (SGMs) [51, 52] model the score of the probability density function of a series of data distributions perturbed with increasing levels of noise, *i.e.* the direction of largest increase in the data log probability density function. An avenue of works [53, 54] generalizes DDPMs and SGMs to infinite noise levels through Stochastic Differential Equations (SDEs). In this work, we adopt the SGM framework of EDM [25] which we reformulate for the generation of high-resolution videos.

3. Method

We propose the generation of high-resolution videos by rewriting the EDM [25] diffusion framework for high-dimensional inputs and proposing an efficient transformer architecture based on FITs [8] which we scale to billions of parameters and tens of thousands input patches. Sec. 3.1 provides an introduction to the EDM framework, Sec 3.2 highlights the challenges of applying diffusion frameworks to high dimensional inputs and proposes a revisited EDM-based diffusion framework. Sec. 3.3 proposes a method to reduce the gap between image and video modalities for joint training. Finally, Sec. 3.4 describes our scalable video generation architecture, while Sec. 3.5 and Sec. 3.6 respectively describe the training and inference procedures.

3.1. Introduction to EDM

Diffusion models have achieved remarkable success in image and video generation. Among the proposed frameworks, *Karras et al.* [25] provide a unified view of common diffusion frameworks and formulate EDM. EDM defines a variance-exploding forward diffusion process $p(\boldsymbol{x}_{\boldsymbol{\sigma}}|\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{\sigma}^2\mathbf{I})$, where $\boldsymbol{\sigma} \in [\boldsymbol{\sigma}_{\text{min}}, \boldsymbol{\sigma}_{\text{max}}]$ represents the diffusion timestep coinciding with the standard deviation of the applied noise, and $\boldsymbol{x}_{\boldsymbol{\sigma}}$ represents the data at the current noise level. A denoiser function $\mathcal{D}_\theta$ is learned to model the reverse process using the denoising objective:

$$\mathcal{L}(\mathcal{D}_\theta) = \mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[\lambda(\boldsymbol{\sigma}) \left\|\mathcal{D}_\theta(\boldsymbol{x}_{\boldsymbol{\sigma}}) - \boldsymbol{x}\right\|_2^2\Big], \quad (1)$$

where λ is the loss weighting function, $\boldsymbol{x} \sim p_{\text{data}}$ is a data sample, $\boldsymbol{\epsilon}$ is gaussian noise, and $\boldsymbol{\sigma} \sim p_{\text{train}}$ is sampled from a training distribution. $\mathcal{D}_\theta(\boldsymbol{x}_{\boldsymbol{\sigma}})$ is defined as:

$$\mathcal{D}_\theta(\boldsymbol{x}_{\boldsymbol{\sigma}}) = c_{\text{out}}(\boldsymbol{\sigma})\mathcal{F}_\theta\left(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}\right) + c_{\text{skip}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}, \quad (2)$$

where $\mathcal{F}_\theta$ is a neural network, and c_{out}, c_{skip} and c_{in} represent scaling functions. In particular, the denoising objective $\mathcal{L}(\mathcal{F}_\theta)$ can equivalently be expressed in terms of $\mathcal{F}_\theta$ as:

$$\mathcal{L}(\mathcal{F}_\theta) = \mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[w(\boldsymbol{\sigma}) \left\|\mathcal{F}_\theta(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}) - c_{\text{nrm}}(\boldsymbol{\sigma})\mathcal{F}_{\text{tgt}}\right\|_2^2\Big], \quad (3)$$

where $\mathcal{F}_{\text{tgt}}$ represents the training target, c_{nrm} is a normalization factor, and w is a weighting function. These forms, derived in Appx. E, are presented in Tab. 1.

A second order Runge-Kutta sampler is proposed to reverse the diffusion process and produce sample $\boldsymbol{x}$ starting from gaussian noise $\boldsymbol{x}_{\boldsymbol{\sigma}_{\text{max}}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{\text{max}}^2\mathbf{I})$.

3.2. EDM for High-Resolution Video Generation

EDM is originally proposed as an image generation framework and its parameters are optimized for 64×64px image generation. Alterations in spatial resolution or the introduction of videos with shared content between frames allow the denoising network to trivially recover a noisy frame in the original resolution with higher signal-to-noise-ratio (SNR), which the original framework was designed to see at lower noise levels. To see why, consider a noisy video $\boldsymbol{x}_{\boldsymbol{\sigma}} \in \mathbb{R}^{T \times s \cdot H \times s \cdot W} \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{\sigma}^2\mathbf{I})$ where T is the number of frames and s is an upsampling factor. We build the corresponding clean and noisy frames at original resolution $\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_{\boldsymbol{\sigma}} \in \mathbb{R}^{1 \times H \times W}$ by averaging values in each $T \times s \times s$ block of pixels. As a consequence of averaging, the noise

		EDM [25]	Our
Training and Losses			
Forw. process	$\boldsymbol{x}_{\boldsymbol{\sigma}}$	$\boldsymbol{x}/\boldsymbol{\sigma}_{\text{in}} + \boldsymbol{\sigma}\boldsymbol{\epsilon}$	$\boldsymbol{x}/\boldsymbol{\sigma}_{\text{in}} + \boldsymbol{\sigma}\boldsymbol{\epsilon}$
Training target	$\mathcal{F}_{\text{tgt}}$	$\boldsymbol{\sigma}\boldsymbol{x} - \boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\epsilon} + \frac{\boldsymbol{\sigma}_{\text{data}}^2(\boldsymbol{\sigma}_{\text{in}}-1)}{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}}\boldsymbol{x}$	$-\boldsymbol{\sigma}\boldsymbol{x} + \boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\epsilon}$
Eff. loss weigh.	$w(\boldsymbol{\sigma})$	1	$(\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2)^2/(\boldsymbol{\sigma}^2 + \frac{\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}})^2$
Loss weigh.	$\lambda(\boldsymbol{\sigma})$	$1/\boldsymbol{\sigma}_{\text{data}}^2 + 1/\boldsymbol{\sigma}^2$	$1/\boldsymbol{\sigma}_{\text{data}}^2 + 1/\boldsymbol{\sigma}^2$
Network Parametrization			
Input scaling	$c_{\text{in}}(\boldsymbol{\sigma})$	$1/\sqrt{\boldsymbol{\sigma}_{\text{data}}^2 + \boldsymbol{\sigma}^2}$	$1/\sqrt{\boldsymbol{\sigma}_{\text{data}}^2/\boldsymbol{\sigma}_{\text{in}}^2 + \boldsymbol{\sigma}^2}$
Output scaling	$c_{\text{out}}(\boldsymbol{\sigma})$	$\frac{\boldsymbol{\sigma}\boldsymbol{\sigma}_{\text{data}}^2}{\sqrt{\boldsymbol{\sigma}^2+\boldsymbol{\sigma}_{\text{data}}^2}}$	$-\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}\boldsymbol{\sigma}_{\text{data}}\frac{\sqrt{\boldsymbol{\sigma}^2+\boldsymbol{\sigma}_{\text{data}}^2}}{\boldsymbol{\sigma}_{\text{data}}^2+\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}^2}$
Skip scaling	$c_{\text{skip}}(\boldsymbol{\sigma})$	$\frac{\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}^2+\boldsymbol{\sigma}_{\text{data}}^2}$	$\frac{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}^2+\boldsymbol{\sigma}_{\text{data}}^2}$
Target scaling	$c_{\text{nrm}}(\boldsymbol{\sigma})$	$1/\boldsymbol{\sigma}_{\text{data}}\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}$	$1/\boldsymbol{\sigma}_{\text{data}}\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}$

Table 1. Definitions of functions in Eq. (1), Eq. (2) and Eq. (3) for the EDM and our proposed diffusion framework as derived in Appx. E and Appx. F, where we highlight the terms induced by the input scaling factor $\boldsymbol{\sigma}_{\text{in}}$. Our framework is equivalent to EDM for $\boldsymbol{\sigma}_{\text{in}} = 1$ but avoids the unstable term $\frac{\boldsymbol{\sigma}_{\text{data}}^2(\boldsymbol{\sigma}_{\text{in}}-1)}{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}}\boldsymbol{x}$ induced by $\boldsymbol{\sigma}_{\text{in}} \neq 1$ in $\mathcal{F}_{\text{tgt}}$. This form highlights that the train target and loss weight match the $\boldsymbol{v}$-prediction [45] framework for $\boldsymbol{\sigma}_{\text{data}} = 1$. All other framework parameters are unaltered with respect to EDM.

variance is reduced by a factor Ts^2, *i.e.* $\tilde{\boldsymbol{x}}_{\boldsymbol{\sigma}} \sim \mathcal{N}(\tilde{\boldsymbol{x}}, \frac{\boldsymbol{\sigma}^2}{Ts^2}\mathbf{I})$, thus $\tilde{\boldsymbol{x}}_{\boldsymbol{\sigma}}$ has an increased signal-to-noise-ratio with respect to $\boldsymbol{x}_{\boldsymbol{\sigma}}$ (see Fig. 2): $SNR_{\tilde{\boldsymbol{x}}_{\boldsymbol{\sigma}}} = Ts^2 SNR_{\boldsymbol{x}_{\boldsymbol{\sigma}}}$. If pixels in each block share similar content, a typical situation in high-resolution videos, then the information in the averaged frame is useful for recovering $\boldsymbol{x}$ and can be exploited at training time by the denoiser function. This creates a train-inference mismatch during the initial sampling steps as the average frame does not yet contain a well-formed signal, yet the denoiser is reliant on its presence. Thus, for best performance, any alteration to T or s should instead maintain the same signal-to-noise ratio at the original resolution for which the diffusion framework was designed.

To restore the optimal SNR at the original resolution, the magnitude of the input signal can be reduced [7] by a corresponding factor $\boldsymbol{\sigma}_{\text{in}} = s\sqrt{T}$ as illustrated in Fig. 2. Consequently, we redefine the forward process as $p(\boldsymbol{x}_{\boldsymbol{\sigma}}|\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{x}/\boldsymbol{\sigma}_{\text{in}}, \boldsymbol{\sigma}^2\mathbf{I})$.

We rewrite the EDM framework to introduce the input scaling factor in Appx. F and highlight the changes in Tab. 1. We notice that a naive introduction of the scaling factor would alter the training target $\mathcal{F}_{\text{tgt}}$ in a way that makes the objective explode for small noise values (see Appx. E). We thus leverage the training objective expressed in the form of Eq. (3) to rewrite the EDM process in a way that ensures $\mathcal{F}_{\text{tgt}}$ remains unchanged, the effective loss weight $w(\boldsymbol{\sigma})$ is such that it keeps the loss weight $\lambda(\boldsymbol{\sigma})$ unchanged, $c_{\text{in}}(\boldsymbol{\sigma})$ and $c_{\text{nrm}}(\boldsymbol{\sigma})$ normalize the input and training target to have unit variance, and the framework is equivalent to the original EDM formulation for $\boldsymbol{\sigma}_{\text{in}} = 1$ (see Appx. F).

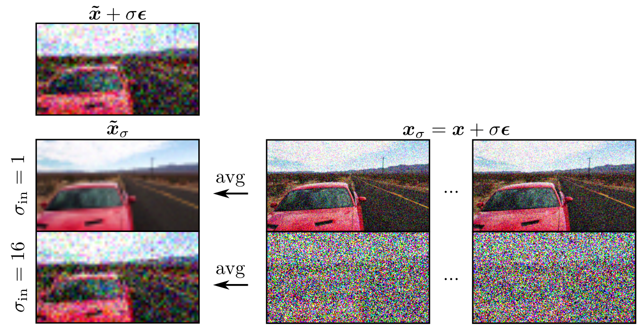


Figure 2. **Analysis of Signal-to-Noise Ratio** (SNR). Top: noise $\boldsymbol{\sigma}$ is applied to an image. Middle: the same noise $\boldsymbol{\sigma}$ is applied to a 16-frames-long video $\boldsymbol{x}$ without scaling. A clean image can be easily restored by simply taking average, indicating an increased SNR. Bottom: to maintain the original SNR, we scale down the 16 frames by $\boldsymbol{\sigma}_{\text{in}}$ before noise application. Averaging is not able to restore the images, indicating the SNR is maintained as $\tilde{\boldsymbol{x}} + \sigma\boldsymbol{\epsilon}$.

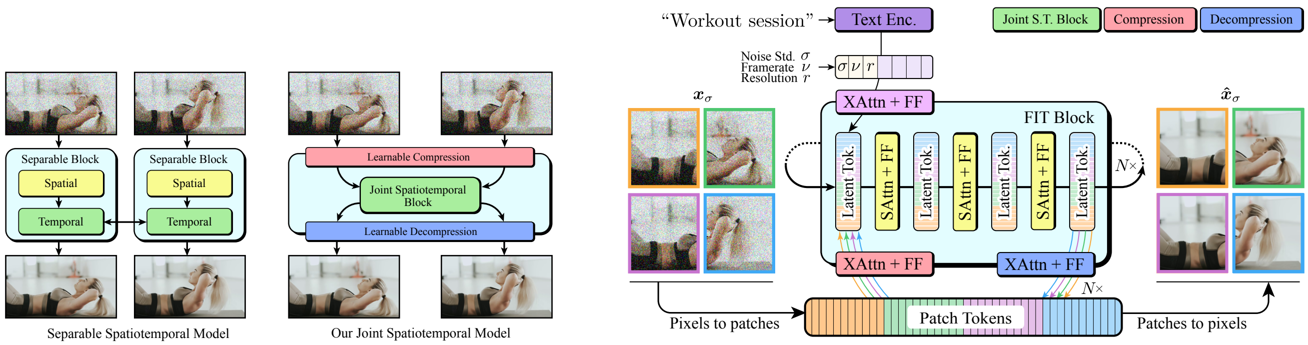
Finally, we modify the sampler according to the newly defined forward process that requires the signal component in $\boldsymbol{x}_{\boldsymbol{\sigma}}$ to be scaled by $\boldsymbol{\sigma}_{\text{in}}$. This is achieved by dividing the $\mathcal{D}_\theta(\boldsymbol{x}_{\boldsymbol{\sigma}})$ by $\boldsymbol{\sigma}_{\text{in}}$ and multiplying the final denoised sample $\boldsymbol{x}_0$ by $\boldsymbol{\sigma}_{\text{in}}$ to restore the signal magnitude.

3.3. Image-Video Modality Matching

Due to the limited amount of captioned video data with respect to images, joint image-video training is widely adopted [13, 21, 22, 48] with the same diffusion process typically applied to both modalities. However, as shown in Sec. 3.2, the presence of T frames in videos calls for a different process with respect to an image with the same resolution. A possibility would be to adopt different input scaling factors for the two modalities. We argue that this solution is undesirable in that it increases the complexity of the framework and image training would not foster the denoising model to learn temporal reasoning, a fundamental capability of a video generator. To sidestep these issues while using a unified diffusion process, we match the image and video modalities by treating images as T frames videos with infinite frame-rate and introduce a variable frame-rate training procedure blending the gap between the image and video modalities.

3.4. Scalable Video Generator

U-Nets [41] have shown success in video generation where they are typically augmented with temporal attention or convolutions for modeling the temporal dimension [4, 13, 21, 22, 48]. However, such an approach requires a full U-Net forward pass for each of the T video frames, rapidly becoming prohibitively expensive (see Fig. 3a). These factors pose a practical limit on model scalability—a primary factor in achieving high generation quality [13, 17, 21, 48]—and



(a) Computational Paradigms for Videos

(b) Snap Video FIT Architecture

Figure 3. (a-left) U-Net-based text-to-image architectures are adapted to do video generation by inserting temporal layers applied sequentially with spatial layers, creating separable spatiotemporal blocks. Spatial computation is repeated for each frame independently, limiting scalability. (a-right) Our scalable transformer-based model jointly performs spatial and temporal computation on a learnable compressed video representation for improved motion modeling and scalability. (b) The proposed Snap Video FIT architecture. Given a noisy input video $\boldsymbol{x}_\sigma$, the model estimates the denoised video $\hat{\boldsymbol{x}}_\sigma$ by recurrent application of FIT blocks. Each block reads information from the patch tokens into a small set of latent tokens on which computation is performed. The results are written to the patch tokens. Conditioning information in the form of text embeddings, noise level σ, frame-rate ν and resolution r is provided through an additional read operation.

similarly limit possibilities for joint spatio-temporal modeling [62]. We argue that treating spatial and temporal modeling in a separable way [4, 13, 21, 48] causes motion artifacts, temporal inconsistencies or generation of *dynamic images* rather than videos with vivid motion. Video frames, however, contain spatially and temporally redundant content that is amenable to compression [33]. We argue that learning and operating on a compressed video representation and jointly modeling the spatial and temporal dimensions are necessary steps to achieve the scalability and motion-modeling capabilities required for high-quality video generation.

FITs [8] are efficient transformer-based architectures that have recently been proposed for high-resolution image synthesis and video generation. Their main idea, summarized in Fig. 3 is that of learning a compressed representation of their input through a set of learnable latent tokens and of focusing computation on this learnable latent space, allowing input dimensionality to grow with little performance penalty. First, FITs perform patchification of the input and produce a sequence of patch tokens which are later divided into groups. A set of latent tokens is then instantiated and a sequence of computational blocks is applied. Each block first performs a cross attention "read" operation between latent tokens and conditioning signals such as the diffusion timestep, then an additional groupwise "read" cross attention operation between latent and patch tokens of corresponding groups to compress patch information, applies a series of self attention operations to the latent tokens, and performs a groupwise "write" cross attention operation that decompresses information in the latent tokens to update the patch tokens. Finally, the patch tokens are projected back to the pixel space to form the output. Self conditioning is applied on the set of latent tokens to preserve the compressed video representation computed in previous sampling steps.

While promising, these architectures have not yet been scaled to the billion-parameters size of state-of-the-art U-Net-based video generators, nor they have been applied to high-resolution video generation. In the following, we highlight the architectural considerations necessary to achieve these goals. Temporal modeling is a fundamental aspect of a high-quality video generator. FITs produce patch tokens by considering three dimensional patches of size $T_p \times H_p \times W_p$ spanning both the spatial and temporal dimensions. We find values of $T_p > 1$ to limit temporal modeling performance, so we consider patches spanning the spatial dimension only. In addition, similarly to patches, FITs group patch tokens into groups spanning both the temporal and spatial dimensions, and perform cross attention operations group by group. We observe that the temporal size of each group should be configured so that each group covers all T video frames for best temporal modeling. Furthermore, videos contain more information with respect to images due to the presence of the temporal dimension, thus we increase the number of latent tokens representing the size of the compressed space in which joint spatiotemporal computation is performed. Finally, FITs make use of local layers which perform self attention operations on patch tokens corresponding to the same group. We find this operation to be computationally expensive for large amounts of patch tokens (147.456 for our largest resolution) and replace it with a feed forward module after each cross attention "read" or "write" operation.

Our model makes use of conditioning information represented by a sequence of conditioning tokens to control the generation process. In addition to the token representing the current $\boldsymbol{\sigma}$, to enable text conditioning, we introduce a T5-11B [39] text encoder extracting text embeddings from the input text. To support variable video framerates and large differences in resolution and aspect ratios in the training data, we concatenate additional tokens representing the framerate and original resolution of the current input.

To generate high-resolution outputs, we implement a model cascade consisting of a first-stage model producing 36×64px videos and a second-stage upsampling model producing 288×512px videos. To improve upsampling quality, we corrupt the second-stage low-resolution inputs with a variable level of noise during training [21, 43] and during inference apply a level of noise to the first-stage outputs obtained by hyperparameter search.

We present detailed model hyperparameters in Appx. B.

3.5. Training

We train Snap Video using the LAMB [68] optimizer with a learning rate of $5e^{-3}$, a cosine learning schedule and a total batch size of 2048 videos and 2048 images, achievable thanks to our scalable video generator architecture. We train the first-stage model over 550k steps and finetune the second-stage model on high-resolution videos starting from the first-stage model weights for 370k iterations. Following the observations in Sec 3.2, we pose $\boldsymbol{\sigma}_{\text{in}} = s\sqrt{T}$. Considering videos with $T = 16$ frames and the original 64px resolution for which EDM was designed, we set $\boldsymbol{\sigma}_{\text{in}} = 4$ for the first-stage and $\boldsymbol{\sigma}_{\text{in}} = 32$ for the second-stage model.

We present training details and parameters in Appx. C.

3.6. Inference

We produce video samples from gaussian noise and user-provided conditioning information using the deterministic sampler of [25] and our two-stage cascade. We use 256 sampling steps for the first-stage and 40 for the second-stage model, and employ classifier free guidance [19] to improve text-video alignment (see Appx. D.1) unless otherwise specified. We find dynamic thresholding [43] and oscillating guidance [21] to consistently improve quality.

4. Evaluation

In this section, we perform evaluation of Snap Video against baselines and validate our design choices. Sec. 4.1 introduces the employed datasets, Sec. 4.2 defines the evaluation protocol, Sec. 4.3 shows ablations of our diffusion framework and architectural choices, Sec. 4.4 quantitatively compares our method to state-of-the-art large-scale video generators and Sec. 4.5 performs qualitative evaluation. We complement evaluation by showcasing samples in the *Appendix* and *Website*.

	FID ↓	FVD ↓	CLIPSIM ↑	Train Thr. ↓	Inf. Thr. ↓
U-Net 85M [10]	8.21	45.94	0.2319	133.2	49.6
U-Net 284M [10]	4.90	23.76	0.2391	230.3	105.1
Snap Video FIT 500M	3.07	27.79	0.2459	69.5	23.4
Snap Video FIT 3.9B	2.51	12.31	0.2579	526.0	130.4

Table 2. Performance of different architectures and model sizes on our internal dataset in 64×36px resolution. We observe strong performance gains with scaling and note that FITs present better performance with improved speed with respect to U-Nets. Train and inference throughputs in ms/video/GPU.

	$\boldsymbol{\sigma}_{\text{data}}$	$\boldsymbol{\sigma}_{\text{in}}$	Imgs. as Videos	FID ↓	FVD ↓	CLIPSIM ↑
(i)	0.5	1.0	✓	6.58	39.95	0.2370
(ii)	0.5	4.0	✓	4.03	31.00	0.2449
(iv)	1.0	2.0	✓	4.45	34.89	0.2428
(iii)	1.0	1/4.0	✗	3.50	24.88	0.2469
Ours	1.0	4.0	✓	3.07	27.79	0.2459

Table 3. Ablation of different diffusion process configurations varying $\boldsymbol{\sigma}_{\text{data}}$, input scaling $\boldsymbol{\sigma}_{\text{in}}$, and treatment of images as infinite-framerate videos, evaluated on our internal dataset in 64×36px resolution.

4.1. Datasets

We train our models on an internal dataset consisting of 1.265M images and 238k hours of videos, each with a corresponding text caption. Due to the difficulty in acquiring high-quality captions for videos, we develop a video captioning model that we use to produce synthetic video captions for the portion of videos in the dataset missing such annotation.

We make use of the following datasets for evaluation which are never observed during training:

UCF-101 [55] is a video dataset containing 13.320 320×240px Youtube videos from 101 action categories.

MSR-VTT [65] is a dataset containing 10.000 320×240px web-crawled videos, each manually annotated with 20 text captions. The test set contains 2.990 videos and 59.800 corresponding captions.

4.2. Evaluation Protocol

To validate the choices operated on the diffusion framework and on model architecture, present method ablations performed in 64×36px resolution using the first-stage model only, and compute FID [18], FVD [60] and CLIPSIM [63] metrics against the test set of our internal dataset on 50k generated videos.

To evaluate our method against baselines, we follow the protocols highlighted in [4, 13, 32, 48, 62, 72] for zero-shot evaluation on the UCF-101 [55] and MSR-VTT [65] datasets. We generate 16 frames videos in 512 $\times$ 288px resolution at 24fps for all settings. We evaluate both at the native 512 $\times$ 288px resolution with 16:9 aspect ratio and in the 288 $\times$ 288px square aspect ratio typically employed on these benchmarks. We note that the evaluation protocols of [4, 13, 32, 48, 62, 72] present different choices regarding the number of generated samples, distribution of class labels, choice of text prompts. We make use of the following evaluation parameters:

Zero-shot UCF-101 [55] We generate 10.000 videos [4, 62] sampling classes with the same distribution as the original dataset. We produce a text prompt for each class label [13] and compute FVD [60] and Inception Score [46].

Zero-shot MSR-VTT [65] We generate a video sample for each of the 59.800 test prompts [13, 48] and compute CLIP-FID [27] and CLIPSIM [63].

To provide a more complete performance assessment and compare against state-of-the-art closed-source methods not reporting results for these benchmarks, we perform a user study evaluating photorealism, video-text-alignment and, most importantly, the quantity and quality of the generated motion, important characteristics of a video generator that may signal the generation of *dynamic images*, *i.e.* videos with dim motion, or motion artifacts rather than videos with vivid and high-quality motion.

4.3. Ablations

To evaluate the proposed FIT architecture, we consider the U-Net of [10], which we adapt to the video generation setting by interleaving temporal attention operations. We consider two U-Net variants of different capacities and a smaller variant of our FIT to evaluate the scalability of both architectures. We detail the architectures in Appx. B and show results in Tab. 2.

Our 500M parameters FIT trains 3.31$\times$ faster than the baseline 284M parameters U-Net, performs inference 4.49$\times$ faster and surpasses it in terms of FID and CLIPSIM. In addition, both FITs and U-Nets show strong performance gains with scaling. Our largest FIT scales to 3.9B parameters with only a 1.24$\times$ increase in inference time with respect to the 284M U-Net.

To evaluate the choices operated on our diffusion framework, we ablate different configurations of the diffusion process using our 500M FIT architecture. We produce the following variations: (i) the original EDM framework, (ii) our scaled diffusion framework with EDM $\boldsymbol{\sigma}_{\text{data}}$, (iii) our framework with a reduced value of $\boldsymbol{\sigma}_{\text{in}}$, (iv) our framework with images not treated as infinite-frame-rate videos. Our framework improves over EDM under all metrics (i) and shows benefits in setting $\boldsymbol{\sigma}_{\text{data}} = 1$, an effect that we attribute to the creation of a training target and loss weighting matching the widely used $\boldsymbol{v}$-prediction formulation of *Salimans et al.* [45] (see Tab. 1). Using $\boldsymbol{\sigma}_{\text{in}} < s\sqrt{T}$ (see Sec. 3.2) impairs performance (iii). Finally, treating images as infinite-frame-rate videos consistently improves FID.

	FVD ↓	FID ↓	IS ↑
CogVideo [23] (Chinese)	751.3	-	23.55
CogVideo [23] (English)	701.6	-	25.27
MagicVideo [72]	655	-	-
LVDM [17]	641.8	-	-
Video LDM [4]	550.6	-	33.45
VideoFactory [62]	410.0	-	-
Make-A-Video [48]	367.2	-	33.00
PYoCo [13]	355.2	-	47.46
Snap Video (288 $\times$ 288 px)	260.1	39.0	38.89
Snap Video (512 $\times$ 288 px)	200.2	28.1	38.89

Table 4. Zero-shot evaluation results on UCF101 [55].

	CLIP-FID ↓	FVD ↓	CLIPSIM ↑
NUWA [64] (Chinese)	47.68	-	0.2439
CogVideo [23] (Chinese)	24.78	-	0.2614
CogVideo [23] (English)	23.59	-	0.2631
MagicVideo [72]	-	998	-
LVDM [17]	-	-	0.2381
Latent-Shift [2]	15.23	-	0.2773
Video LDM [4]	-	-	0.2929
VideoFactory [62]	-	-	0.3005
Make-A-Video [48]	13.17	-	0.3049
PYoCo [13]	9.73	-	-
Snap Video (288 $\times$ 288 px)	8.48	110.4	0.2793
Snap Video (512 $\times$ 288 px)	9.35	104.0	0.2793

Table 5. Zero-shot evaluation results on MSR-VTT [65].

4.4. Quantitative Evaluation

We perform comparison of Snap Video against baselines on the UCF101 [55], and MSR-VTT [65] datasets respectively in Tab. 4 and Tab. 5. FID and FVD video quality metrics show improvements over the baselines which we attribute to the employed diffusion framework and joint spatiotemporal modeling performed by our architecture. On UCF101, our method produces the second-best IS of 38.89, demonstrating good video-text alignment. While our method surpasses Make-A-Video [48] on UCF101, we note that it produces a lower CLIPSIM score on MSR-VTT. We attribute this behavior to the use of T5 [39] text embeddings in place of the commonly used CLIP [38] embeddings which were observed [43] to produce higher text-image alignment despite similar CLIPSIM.

To provide a comprehensive evaluation we run a user study to evaluate photorealism, video-text alignment, quantity of motion and quality of motion, important aspects of a video generator. Three publicly-accessible state-of-the-art video generators are considered: Gen-2 [11], PikaLabs [1]

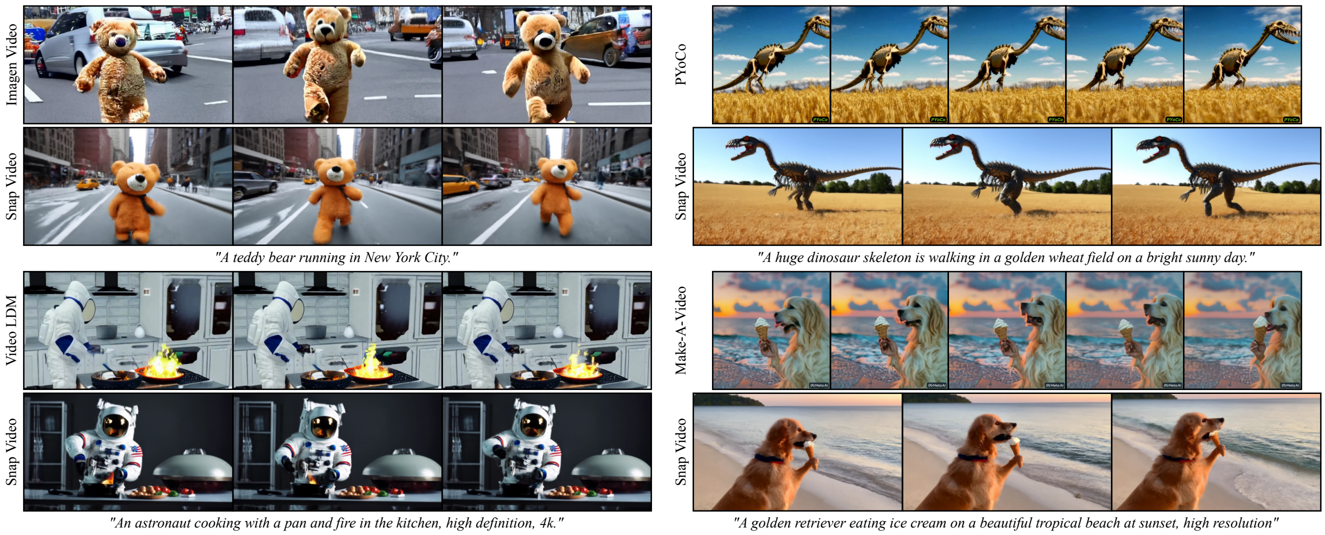


Figure 4. Qualitative results comparing Snap Video to state-of-the-art video generators on publicly available samples. While baseline methods present motion artifacts (top-left, top-right, bottom-right) or produce *dynamic images* (bottom-left), our method produces more temporally coherent motion. Best viewed in the *Website*.

	Photorealism	Video-Text Align.	Mot. Quant.	Mot. Qual.
Gen-2 [11]	44.3	81.0	96.0	78.7
PikaLab [1]	61.5	80.3	89.2	70.5
Floor33 [17]	76.3	80.9	88.0	79.1

Table 6. User study on photorealism, video-text alignment, motion quantity and quality against publicly-accessible video generators on 65 dynamic scene prompts. % of votes in favor of our method.

and Floor33 [17]. We filter a set of 65 prompts from [31] describing scenes with vivid motions, and generate a video for each method with default options. We ask the participants to express preference between paired samples from Snap Video and each baseline, gathering votes from 5 users for each sample. Results are shown in Tab. 6 and video samples provided along with the employed prompt list in Appx. D.2 and in the *Website*. Our method produces results with photorealism comparable to Gen-2, while surpassing PikaLab and Floor33, and outperforms all baselines with respect to video-text alignment. Most importantly, we note that baselines often produce *dynamic images*, *i.e*. videos with dim motion, or videos with motion artifacts, a finding we attribute to the challenges in modeling large motion. In contrast, our method, thanks to the joint spatiotemporal modeling approach, produces vivid and high-quality motion as shown by the motion metrics.

4.5. Qualitative Evaluation

In this section, we perform qualitative evaluation of our framework. In Fig. 4, Appx. D.3 and the *Website*, we present qualitative results comparing our method to state-of-the-art generators [4, 13, 21, 48] on samples publicly released by the authors. While such prompts might have been selected to highlight strengths of the baselines, our method produces more photorealistic samples aligned to the text descriptions. Most importantly, our samples present vivid and high-quality motion avoiding flickering artifacts that are present in the baselines due to temporal inconsistencies. We accompany qualitative evaluation with a user study performed on the same set of samples in Appx. D.2.

5. Conclusions

In this work, we highlight the shortcomings of diffusion processes and architectures commonly used in text-to-video generation, and systematically address them by treating videos as first-class citizens. First, we propose a modification to the EDM [25] diffusion framework for the generation of high-resolution videos and treat images as high frame-rate videos to avoid image-video modality mismatches. Second, we replace U-Nets [41] with efficient transformer-based FITs [8] which we scale to billions of parameters. Thanks to their learnable compressed representation of videos, they significantly improve training times, scalability and performance with particular regards to temporal consistency and motion modeling capabilities due to the joint spatiotemporal modeling on the compressed representation. When evaluated on UCF101 [55] and MSR-VTT [65] and in user studies, Snap Video attains state-of-the-art performance with particular regard to the quality of the modeled motion.

References

[1] Pika lab discord server. https://www.pika.art/. Accessed: 2023-11-01. 2, 7, 8, 3, 6

[2] Jie An, Songyang Zhang, Harry Yang, Sonal Gupta, Jia-Bin Huang, Jiebo Luo, and Xi Yin. Latent-shift: Latent diffusion with temporal shift for efficient text-to-video generation. *arXiv*, 2023. 2, 3, 7

[3] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *ArXiv*, 2022. 2, 3

[4] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 15

[5] Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei A Efros, and Tero Karras. Generating long videos of dynamic scenes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[6] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[7] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv*, 2023. 2, 3, 4

[8] Ting Chen and Lala Li. Fit: Far-reaching interleaved transformers. *arXiv*, 2023. 2, 3, 5, 8, 1

[9] Aidan Clark, Jeff Donahue, and Karen Simonyan. Efficient video generation on complex datasets. *arXiv*, 2019. 2

[10] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 6, 7

[11] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 7, 8, 3, 6

[12] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *Proceedings of the European Conference of Computer Vision (ECCV)*, 2022. 2

[13] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 15

[14] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Miguel Angel Bautista, and Josh Susskind. f-dm: A multi-stage diffusion model via progressive signal transformation. *International Conference on Learning Representations (ICLR)*, 2023. 3

[15] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Josh Susskind, and Navdeep Jaitly. Matryoshka diffusion models. *arXiv*, 2023. 3

[16] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv*, 2023. 2

[17] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv*, 2023. 2, 3, 4, 7, 8, 6

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 6

[19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv*, 2022. 6, 2

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[21] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *arXiv*, 2022. 1, 2, 3, 4, 5, 6, 8, 15

[22] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022. 2, 4, 3

[23] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv*, 2022. 2, 7

[24] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023. 3

[25] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3, 4, 6, 8

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2015. 2

[27] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance. In *International Conference on Learning Representations (ICLR)*, 2023. 7

[28] Alex X. Lee, Richard Zhang, Frederik Ebert, P. Abbeel, Chelsea Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv*, abs/1804.01523, 2018. 2

[29] Yitong Li, Martin Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2

[30] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *arXiv*, 2023. 1

[31] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. *arXiv*, 2023. 8, 3, 7

[32] Z. Luo, D. Chen, Y. Zhang, Y. Huang, L. Wang, Y. Shen, D. Zhao, J. Zhou, and T. Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 6, 7

[33] Siwei Ma, Xinfeng Zhang, Chuanmin Jia, Zhenghui Zhao, Shiqi Wang, and Shanshe Wang. Image and video compression with neural networks: A review. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 2, 5

[34] Gaurav Mittal, Tanya Marwah, and Vineeth N. Balasubramanian. Sync-draw: Automatic video generation using deep recurrent attentive architectures. In *Proceedings of the 25th ACM International Conference on Multimedia*, 2017. 2

[35] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. CCVS: Context-aware controllable video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. 1

[37] Chenyang QI, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 1

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 7

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 2022. 6, 7, 15

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv*, 2021. 1, 2, 3

[41] 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. 2, 4, 8, 1

[42] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv*, 2022. 1, 2, 3, 6, 7

[44] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan, 2020. 2

[45] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022. 4, 7, 12

[46] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 7

[47] Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Mostgan-v: Video generation with temporal motion styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[48] 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. *arXiv*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[49] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[50] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015. 3

[51] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3

[52] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[53] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models, 2021. 3

[54] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. 3

[55] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv*, 2012. 2, 6, 7, 8, 3, 11

[56] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning (ICML)*, 2015. 2

[57] Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. *arXiv*, 2023. 3

[58] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[59] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[60] 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*, 2018. 6, 7

[61] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual description. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[62] Wenjing Wang, Huan Yang, Zixi Tuo, Huiguo He, Junchen Zhu, Jianlong Fu, and Jiaying Liu. Videofactory: Swap attention in spatiotemporal diffusions for text-to-video generation. *arXiv*, 2023. 1, 2, 5, 6, 7, 15

[63] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *ArXiv*, 2021. 2, 6, 7

[64] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pretraining for neural visual world creation. In *Proceedings of the European Conference of Computer Vision (ECCV)*, 2022. 2, 7

[65] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 2, 6, 7, 8

[66] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv*, 2021. 2

[67] Sheng-Siang 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. In *Annual Meeting of the Association for Computational Linguistics*, 2023. 2

[68] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations (ICLR)*, 2020. 6, 1

[69] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. 1

[70] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G. Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, and Lu Jiang. Magvit: Masked generative video transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[71] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2022. 2

[72] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv*, 2023. 2, 3, 6, 7