# Generating Long Videos of Dynamic Scenes

**Tim Brooks** NVIDIA, UC Berkeley
**Janne Hellsten** NVIDIA
**Miika Aittala** NVIDIA
**Ting-Chun Wang** NVIDIA
**Timo Aila** NVIDIA

**Jaakko Lehtinen** NVIDIA, Aalto University
**Ming-Yu Liu** NVIDIA
**Alexei A. Efros** UC Berkeley
**Tero Karras** NVIDIA

## Abstract

We present a video generation model that accurately reproduces object motion, changes in camera viewpoint, and new content that arises over time. Existing video generation methods often fail to produce new content as a function of time while maintaining consistencies expected in real environments, such as plausible dynamics and object persistence. A common failure case is for content to never change due to over-reliance on inductive biases to provide temporal consistency, such as a single latent code that dictates content for the entire video. On the other extreme, without long-term consistency, generated videos may morph unrealistically between different scenes. To address these limitations, we prioritize the time axis by redesigning the temporal latent representation and learning long-term consistency from data by training on longer videos. We leverage a two-phase training strategy, where we separately train using longer videos at a low resolution and shorter videos at a high resolution. To evaluate the capabilities of our model, we introduce two new benchmark datasets with explicit focus on long-term temporal dynamics.

## 1 Introduction

Videos are data that change over time, with complex patterns of camera viewpoint, motion, deformation and occlusion. In certain respects, videos are unbounded — they may last arbitrarily long and there is no limit to the amount of new content that may become visible over time. Yet videos that depict the real world must also remain consistent with physical laws that dictate which changes over time are feasible. For example, the camera may only move through 3D space along a smooth path, objects cannot morph between each other, and time cannot go backward. Generating long videos thus requires the ability to produce endless new content while maintaining appropriate consistencies.

In this work, we focus on generating long videos with rich dynamics and new content that arises over time. While existing video generation models can produce "infinite" videos, the type and amount of change along the time axis is highly limited. For example, a synthesized infinite video of a person talking will only include small motions of the mouth and head. Moreover, common video generation datasets often contain short clips with little new content over time, which may inadvertently bias the design choices toward training on short segments or pairs of frames, forcing content in videos to stay fixed, or using architectures with small temporal receptive fields.

We make the time axis a first-class citizen for video generation. To this end, we introduce two new datasets that contain motion, changing camera viewpoints, and entrances/exits of objects and scenery over time. We learn long-term consistencies by training on long videos and design a temporal latent representation that enables modeling complex temporal changes. Figure 1 illustrates the rich motion and scenery changes that our model is capable of generating. See our webpage[1] for video results, code, data and pretrained models.

[1] `https://www.timothybrooks.com/tech/long-videos`

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

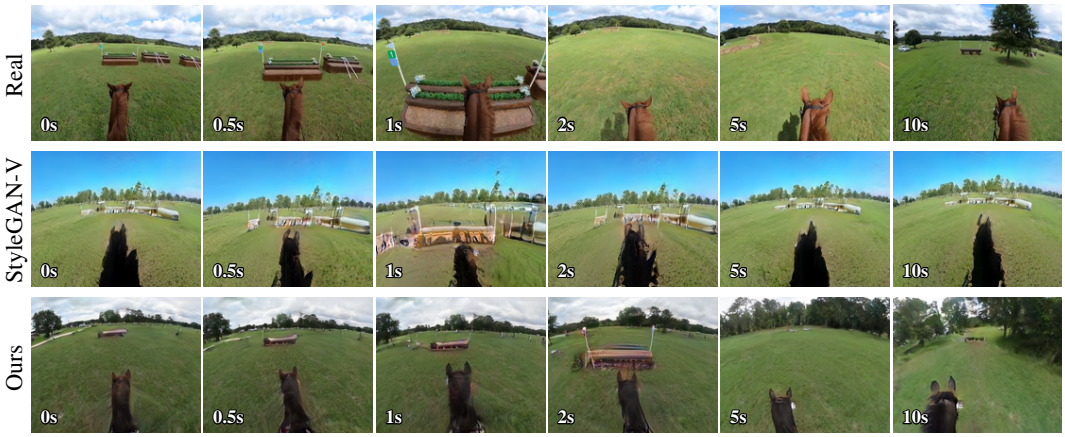



Figure 1: We aim to generate videos that accurately portray motion, changing camera viewpoint, and new content that arises over time. **Top:** Our horseback riding dataset exhibits these types of changes as the horse moves forward in the environment. **Middle:** StyleGAN-V, a state-of-the-art video generation baseline, is incapable of generating new content over time; the horse fails to move forward past the obstacle, the scene does not change, and the video morphs back and forth within a short window of motion. **Bottom:** Our novel video generation model prioritizes the time axis and generates realistic motion and scenery changes over long durations. The same videos can be viewed on the supplemental webpage.

Our main contribution is a hierarchical generator architecture that employs a vast temporal receptive field and a novel temporal embedding. We employ a multi-resolution strategy, where we first generate videos at low resolution and then refine them using a separate super-resolution network. Naively training on long videos at high spatial resolution is prohibitively expensive, but we find that the main aspects of a video persist at a low spatial resolution. This observation allows us to train with long videos at low resolution and short videos at high resolution, enabling us to prioritize the time axis and ensure that long-term changes are accurately portrayed. The low-resolution and super-resolution networks are trained independently with an RGB bottleneck in between. This modular design allows iterating on each network independently and leveraging the same super-resolution network for different low-resolution network ablations.

We compare our results to several recent video generative models and demonstrate state-of-the-art performance in producing long videos with realistic motion and changes in content. Code, new datasets, and pre-trained models on these datasets will be made available.

## 2 Prior work

Video generation is a challenging problem with a long history. The classic early works, Video Textures [50] and Dynamic Textures [10], model videos as textures by analogy with image textures. That is, they explicitly assume the content to be stationary over time, e.g., fire burning, smoke rising, foliage falling, pendulum swinging, etc., and use non-parametric [50] or parametric [10] approaches to model that stationary distribution. Although subsequent video synthesis works have dropped the "texture" moniker, much of the limitations remain similar — short training videos and models which produce little or no new objects entering the frame during the video. Below we summarize some of the more recent efforts on video generation.

**Unconditional video generation.** Many video generation works are based on GANs [14], including early models that output fixed-length videos [1, 47, 60] and approaches that use recurrent networks to produce a sequence of latent codes used to generate frames [9, 12, 55, 56]. MoCoGAN [56] explicitly disentangles "motion" from "content" and keeps the latter fixed over the entire generated video. StyleGAN-V [52] is a recent state-of-the-art model we use as a primary baseline. Similar to MoCoGAN, StyleGAN-V employs a global latent code that controls content of an entire video. MoCoGAN-HD [55], which we also compare with, and StyleVideoGAN [12] attempt to generate videos by navigating the latent space of a pretrained StyleGAN2 model [29], but struggle to produce

realistic motion. Unlike previous StyleGAN-based [28] video models, we prioritize the time axis in our generator through a new temporal latent representation, temporal upsampling, and spatiotemporal modulated convolutions. We also compare with DIGAN [66] that employs an implicit representation to generate the video pixel by pixel.

Transformers are another class of models used for video generation [13, 42, 61, 65]. We compare with TATS [13] that generates long unconditional videos with transformers, improving upon VideoGPT [65]. Both TATS and VideoGPT employ a GPT-like autoregressive transformer [4] that represents videos as sequences of tokens. However, the resulting videos tend to accumulate error over time and often diverge or change too rapidly. The models are also expensive to train and deploy due to their autoregressive nature over time and space. In concurrent work, promising results in generating diverse videos have also been demonstrated using diffusion-based models [20].

**Conditional video prediction.** A separate line of research focuses on predicting future video frames conditioned on one or more real video frames [3, 23, 34, 36, 39, 41] or past frames accompanied by an action label [6, 15, 30, 31]. Some video prediction methods focus specifically on generating infinite scenery by conditioning on camera trajectory [37, 44] and/or explicitly predicting depth [2, 37] to then simulate a virtual camera flying through a 3D scene. Our goal, on the other hand, is to support camera movement as well as moving objects by having the scene structure emerge implicitly.

**Multi-resolution training.** Training at multiple scales is a common strategy for image generation models [7, 25, 43, 46, 58]. Transformer-based video generators also employ a related two-phase setup [65, 13]. Saito *et al.* [48] subsample frames at higher resolutions in their video generator architecture to improve efficiency. A similar idea is also used in SlowFast [11] networks where different network pathways are used for high and low frame rate video streams. Acharya *et al.* [1] propose a multi-scale GAN for video generation that increases both spatial resolution and sequence length during training to produce a fixed-length video. In contrast, our multi-resolution approach is designed to enable generating arbitrarily long videos with rich long-term dynamics by leveraging training of long sequences at low resolution.

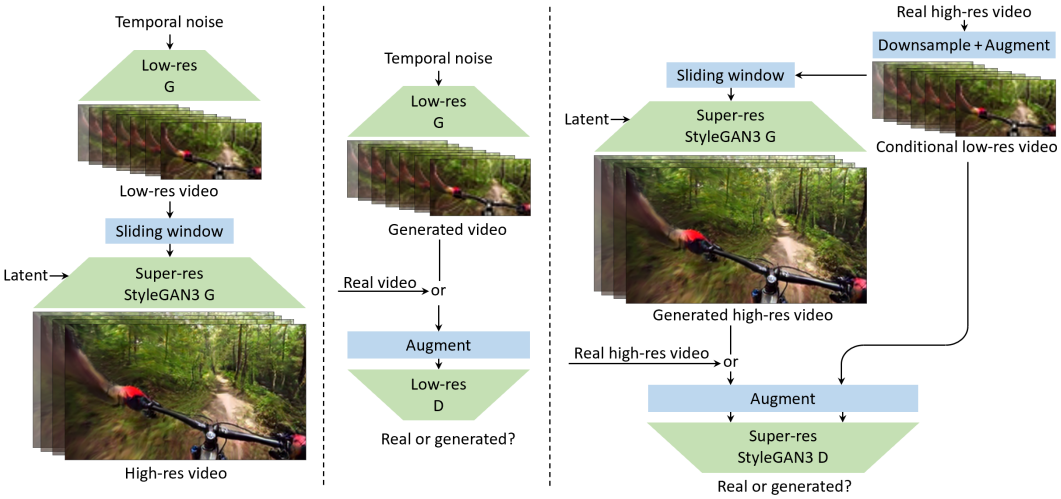
## 3 Our method

Modeling the long-term temporal behavior observed in real videos presents us with two main challenges. First, we must use long enough sequences during training to capture the relevant effects; using, e.g., pairs of consecutive frames fails to provide meaningful training signal for effects that occur over several seconds. Second, we must ensure that the networks themselves are capable of operating over long time scales; if, e.g., the receptive field of the generator spans only 8 adjacent frames, any two frames taken more than 8 frames apart will necessarily be uncorrelated with each other.

Figure 2a shows the overall design of our generator. We seed the generation process with a variable-length stream of temporal noise, consisting of 8 scalar components per frame drawn from i.i.d. Gaussian distribution. The temporal noise is first processed by a *low-resolution generator* to obtain a sequence of RGB frames at $64^2$ resolution that are then refined by a separate *super-resolution network* to produce the final frames at $256^2$ resolution.[2] The role of the low-resolution generator is to model major aspects of the motion and scene composition, which necessitates strong expressive power and a large receptive field over time, whereas the super-resolution network is responsible for the more fine-grained task of hallucinating the remaining details.

Our two-stage design provides maximum flexibility in terms of generating long videos. Specifically, the low-resolution generator is designed to be fully convolutional over time, so the duration and time offset of the generated video can be controlled by shifting and reshaping the temporal noise, respectively. The super-resolution network, on the other hand, operates on a frame-by-frame basis. It receives a short sequence of 9 consecutive low-resolution frames and outputs a single high-resolution frame; each output frame is processed independently using a sliding window. The combination of fully-convolutional and per-frame processing enables us to generate arbitrary frames in arbitrary order, which is highly desirable for, e.g., interactive editing and real-time playback.

[2] We handle datasets with non-square aspect ratio by shrinking all intermediate data accordingly. With $256 \times 144$ target resolution, for example, the low-resolution frames will have $64 \times 36$ resolution.




(a) Generator overview (b) Low-res training (c) Super-res training

Figure 2: Overview of our method. **(a)** To achieve long temporal receptive field and high spatial resolution, we split our generator into two components: a low-resolution generator, responsible for modeling major aspects of the motion and scene composition, and a super-resolution network, responsible for hallucinating fine details. **(b)** The low-resolution generator (Section 3.1) employs a wide temporal receptive field and is trained with sequences of 128 frames at $64^2$ resolution. **(c)** The super-resolution network (Section 3.2) is conditioned on short sequences of low-resolution frames and trained to produce their plausible counterparts at $256^2$ resolution.

The low-resolution and super-resolution networks are modular with an RGB bottleneck in between. This greatly simplifies experimentation, since the networks are trained independently and can be used in different combinations during inference. We will first describe the training and architecture of the low-resolution generator in Section 3.1 and then discuss the super-resolution network in Section 3.2.

### 3.1 Low-resolution generator

Figure 2b shows our training setup for the low-resolution generator. In each iteration, we provide the generator with a fresh set of temporal noise to produce sequences of 128 frames (4.3 seconds at 30 fps). To train the discriminator, we sample corresponding sequences from the training data by choosing a random video and a random interval of 128 frames within that video.

We have observed that training with long sequences tends to exacerbate the issue of overfitting [26]. As the sequence length increases, we suspect that it becomes harder for the generator to simultaneously model temporal dynamics at multiple time scales, but at the same time, easier for the discriminator to spot any mistakes. In practice, we have found strong discriminator augmentation [26, 69] to be necessary in order to stabilize the training. We employ DiffAug [69] using the same transformation for each frame in a sequence, as well as fractional time stretching between $\frac{1}{2}\times$ and $2\times$; see Appendix C.1 for details.

**Architecture.** Figure 3 illustrates the architecture of our low-resolution generator. Our main goal is to make the time axis a first-class citizen, including careful design of a temporal latent representation, temporal style modulation, spatiotemporal convolutions, and temporal upsamples. Through these mechanisms, our generator spans a vast temporal receptive field (5k frames), allowing it to represent temporal correlations at multiple time scales.

We employ a style-based design, similar to Karras *et al.* [29, 27], that maps the input temporal noise into a sequence of *intermediate latents* $\{w_t\}$ used to modulate the behavior of each layer in the main synthesis path. Each intermediate latent is associated with a specific frame, but it can significantly influence the scene composition and temporal behavior of several frames through hierarchical 3D convolutions that appear in the main path.

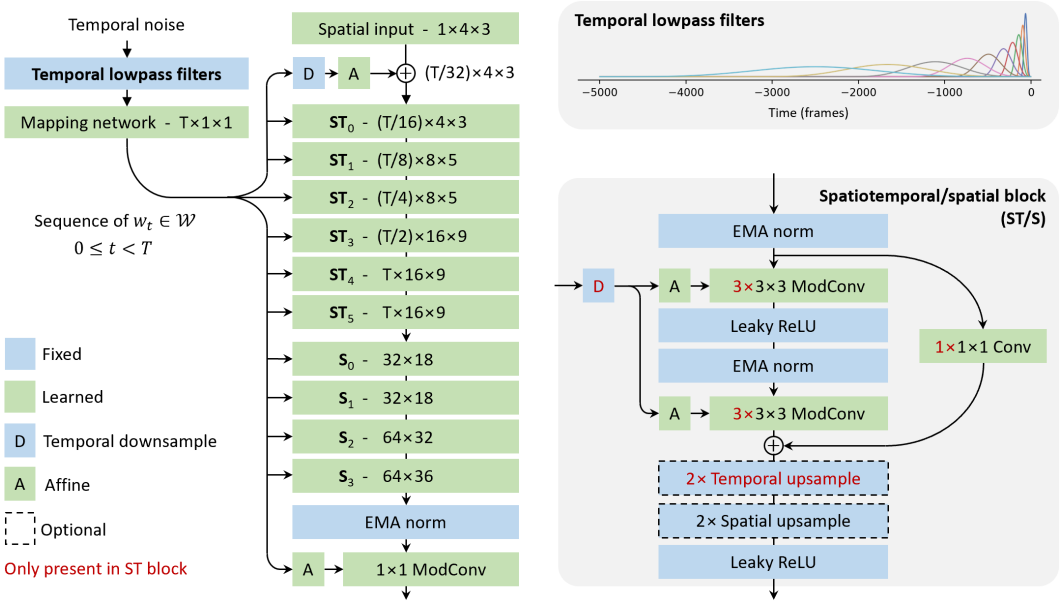



Figure 3: Low-resolution generator architecture, illustrated for 64×36 output. **Left:** The input temporal noise is mapped to a sequence of *intermediate latents* $\{w_t\}$ that modulate the intermediate activations of the main synthesis path. **Top right:** To facilitate the modeling of long-term dependencies, we enrich the temporal noise by passing it through a series of lowpass filters whose temporal footprints range all the way from 100 to 5000 frames. **Bottom right:** The main synthesis path consists of *spatiotemporal* (ST) and *spatial* (S) blocks that gradually increase the resolution over time and space.

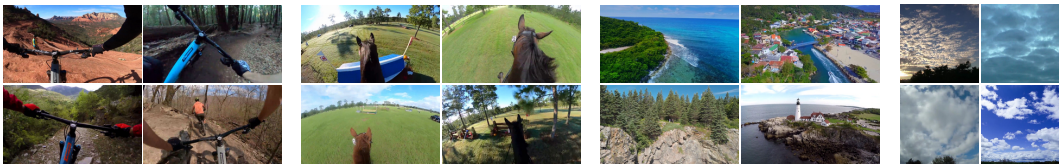
In order to reap the full benefits of the style-based design, it is crucial for the intermediate latents to capture long-term temporal correlations, such as weather changes or persistent objects. To this end, we adopt a scheme where we first enrich the input temporal noise using a series of temporal lowpass filters and then pass it through a fully-connected *mapping network* on a frame-by-frame basis. The goal of the lowpass filtering is to provide the mapping network with sufficient long-term context across a wide range of different time scales. Specifically, given a stream of temporal noise $z(t) \in \mathbb{R}^8$, we compute the corresponding enriched representation $z'(t) \in \mathbb{R}^{128\times 8}$ as $z'_{i,j} = f_i * z_j$, where $\{f_i\}$ is a set of 128 lowpass filters whose temporal footprint ranges from 100 to 5000 frames, and $*$ denotes convolution over time; see Appendix C.2 for details.

The main synthesis path starts by downsampling the temporal resolution of $\{w_t\}$ by 32× and concatenating it with a learned constant at $4^2$ resolution. It then gradually increases the temporal and spatial resolutions through a series of processing blocks, illustrated in Figure 3 (bottom right), focusing first on the time dimension (ST) and then the spatial dimensions (S). The first four blocks have 512 channels, followed by two blocks with 256, two with 128 and two with 64 channels. The processing blocks consist of the same basic building blocks as StyleGAN2 [29] and StyleGAN3 [27] with the addition of a skip connection; the intermediate activations are normalized before each convolution [27] and modulated [29] according to an appropriately downsampled copy of $\{w_t\}$. In practice, we employ bilinear upsampling [28] and use padding [27] for the time axis to eliminate boundary effects. Through the combination of our temporal latent representation and spatiotemporal processing blocks, our architecture is able to model complex and long-term patterns across time.

For the discriminator, we employ an architecture that prioritizes the time axis via wide temporal receptive field, 3D spatiotemporal and 1D temporal convolutions, and spatial and temporal downsamples; see Appendix C.3 for details.

### 3.2 Super-resolution network

Figure 2c shows our training setup for the super-resolution network. Our video super-resolution network is a straightforward extension of StyleGAN3 [27] for conditional frame generation. Unlike



(a) Mountain biking (b) Horseback riding (c) ACID (d) SkyTimelapse

Figure 4: Example real frames from training datasets. We introduce first-person datasets of **(a)** mountain biking and **(b)** horseback riding videos that contain complex motion and new content over time. We also evaluate on existing datasets of **(c)** nature drone footage and **(d)** sky timelapse videos.

the low-resolution network that outputs a sequence of frames and includes explicit temporal operations, the super-resolution generator outputs a single frame and only utilizes temporal information at the input, where the real low-resolution frame and $4$ neighboring real low-resolution frames before and after in time are concatenated along the channel dimension to provide context. We remove the spatial Fourier feature inputs and resize and concatenate the stack of low-resolution frames to each layer throughout the generator. The generator architecture is otherwise unchanged from StyleGAN3, including the use of an intermediate latent code that is sampled per video. Low-resolution frames undergo augmentation prior to conditioning as part of the data pipeline, which helps ensure generalization to *generated* low-resolution images.

The super-res discriminator is a similar straightforward extension of the StyleGAN discriminator, with $4$ low and high-resolution frames concatenated at the input. The only other change is the removal of the minibatch standard deviation layer that we found unnecessary in practice. Both low- and high-resolution segments of 4 frames undergo adaptive augmentation [26] where the same augmentation is applied to all frames at both resolutions. Low-resolution segments also undergo aggressive dropout ($p = 0.9$ probability of zeroing out the entire segment), which prevents the discriminator from relying too heavily on the conditioning signal; see Appendix D.1 for details.

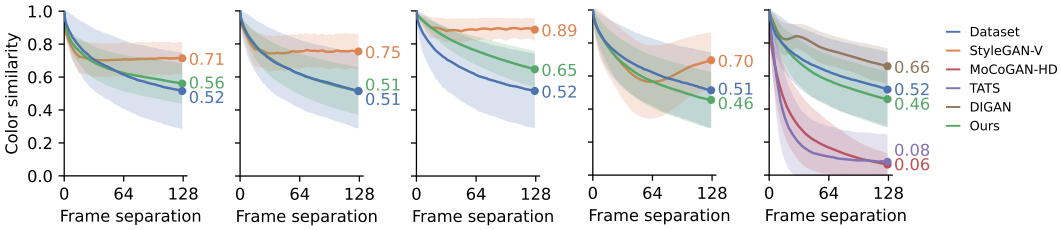
We find it remarkable that such a simple video super-resolution model appears sufficient for producing reasonably good high-resolution videos. We focus primarily on the low-resolution generator in our experiments, utilizing a single super-resolution network trained per dataset. We feel that replacing this simple network with a more advanced model from the video super-resolution literature [16, 24, 49, 54] is a promising avenue for future work.

## 4 Datasets

Most of the existing video datasets introduce little or no new content over time. For example, talking head datasets [8, 45, 62, 63] show the same person for the duration of each video. UCF101 [53] portrays diverse human actions, but the videos are short and contain limited camera motion and little or no new objects that enter the videos over time.

To best evaluate our model, we introduce two new video datasets of first-person mountain biking and horseback riding (Figure 4a,b) that exhibit complex changes over time. Our new datasets include subject motion of the horse or biker, a first-person camera viewpoint that moves through space, and new scenery and objects over time. The videos are available in high definition and were manually trimmed to remove problematic segments, scene cuts, text overlays, obstructed views, etc. The mountain biking dataset has 1202 videos with a median duration of 330 frames at 30 fps, and the horseback dataset has 66 videos with a median duration of 6504 frames also at 30fps. We have permission from the content owners to publicly release the datasets for research purposes. We believe our new datasets will serve as important benchmarks for future work.

We also evaluate our model on the ACID dataset [38] (Figure 4c) that contains significant camera motion but lacks other types of motion, as well as the commonly used SkyTimelapse dataset [67] (Figure 4d) that exhibits new content over time as the clouds pass by, but the videos are relatively homogeneous and the camera remains fixed.




(a) Biking (b) Horseback (c) ACID (d) Sky $256^2$ (e) Sky $128^2$

Figure 5: Color similarity (Eq. 1) of real and generated videos as a function of frame separation, reported as the mean (solid lines) and standard deviation (shaded regions) over 1000 random clips.

# 5 Results

We evaluate our model through qualitative examination of the generated videos (Section 5.1), analyzing color change over time (Section 5.2), computing the FVD metric (Section 5.3), and ablating the key design choices (Section 5.4). We compare with StyleGAN-V [52] on all datasets. Mountain biking, horseback riding and ACID [37] datasets contain videos with a 16×9 widescreen aspect ratio. We train at 256×144 resolution on these datasets to preserve the aspect ratio. Since StyleGAN-V is based on StyleGAN2 [29], we can easily extend it to support non-square aspect ratios by masking real and generated frames during training. We found it necessary to increase the R1 $\gamma$ hyperparameter by $10\times$ to produce good results with StyleGAN-V on our new datasets that exhibit complex changes over time. We compare with MoCoGAN-HD [56], TATS [13] and DIGAN [66] using pre-trained models for the SkyTimelapse dataset at $128^2$ resolution. For these comparisons, we train a separate super-resolution network to output the frames at $128^2$ resolution, but use the same low-resolution generator as in the $256^2$ comparison.

## 5.1 Qualitative results

The major qualitative difference in results is that our model generates realistic new content over time, whereas StyleGAN-V continually repeats the same content. The effect is best observed by watching videos on the supplemental webpage and is additionally illustrated in Figure 1. Scenery changes over time in real videos and our results as the horse moves forward through space. However, the videos generated by StyleGAN-V tend to morph back to the same scene at regular intervals. Similar repeated content from StyleGAN-V is apparent on all datasets. For example, results on the webpage for the SkyTimelapse dataset show that clouds generated by StyleGAN-V repeatedly move back and forth. MoCoGAN-HD and TATS suffer from unrealistic rapid changes over time that diverge, and DIGAN results contain periodic patterns visible in both space and time. Our model is capable of generating a constant stream of new clouds.

As a further validation of our observations, we conducted a preliminary user study on Amazon Mechanical Turk. We created 50 pairs of videos for each of the 4 datasets. Each pair contained a random video generated by StyleGAN-V and one generated by our method, and we asked the participants which of them exhibited more realistic motion in a forced-choice response. Each pair was shown to 10 participants, resulting in a total of 50×4×10 responses. Our method was preferred over 80% of the time for every dataset. Please see Appendix A.1 for details.

## 5.2 Analyzing color change over time

To gain insight into how well different methods produce new content at appropriate rates, we analyze how the overall color scheme changes as a function of time. We measure color similarity as the intersection between RGB color histograms; this serves as a simple proxy for actual content changes and helps reveal the biases of different models. Let $H(x, i)$ denote a 3D color histogram function that computes the value of histogram bin $i \in [1, \ldots, N^3]$ for the given image $x$, normalized so that $\sum_i H(x, i) = 1$. Given video clip $\boldsymbol{x} = \{x_t\}$ and frame separation $t$, we define the color similarity as

$$S(\boldsymbol{x}, t) = \sum\nolimits_i \min\big(H(x_0, i),\ H(x_t, i)\big), \tag{1}$$

| | Biking | | Horseback | | ACID | | Sky $256^2$ | |
|---|---|---|---|---|---|---|---|---|
| | $FVD_{128}$ | $FVD_{16}$ | $FVD_{128}$ | $FVD_{16}$ | $FVD_{128}$ | $FVD_{16}$ | $FVD_{128}$ | $FVD_{16}$ |
| StyleGAN-V | 533.3 | 353.7 | 427.0 | 319.2 | 112.4 | 91.5 | 151.2 | 48.4 |
| with 10× R1 $\gamma$ | 224.6 | 99.2 | 196.2 | 159.0 | – | – | – | – |
| Ours | 113.7 | 83.8 | 95.9 | 113.5 | 166.6 | 127.3 | 152.7 | 116.5 |

| | Sky $128^2$ | |
|---|---|---|
| | $FVD_{128}$ | $FVD_{16}$ |
| MoCoGAN-HD | 635.6 | 224.9 |
| TATS | 435.0 | 97.0 |
| DIGAN | 228.6 | 153.4 |
| Ours | 142.6 | 107.5 |

Table 1: We compute FVD on segments of 128 and 16 frames ($FVD_{128}$ and $FVD_{16}$ respectively), where lower is better. **Left:** Our model outperforms StyleGAN-V on horseback riding and mountain biking datasets – both of which contain complex motion and new content over time. Our model underperforms StyleGAN-V on ACID and SkyTimelapse despite qualitative improvements and favorable user study ratings in Section 5.1. **Right:** Our model outperforms MoCoGAN-HD, TATS and DIGAN baselines on SkyTimelapse at $128^2$ resolution on $FVD_{128}$.

where $S(\boldsymbol{x}, t) = 1$ indicates that the color histograms are identical between $x_0$ and $x_t$. In practice, we set $N = 20$ and report the mean and standard deviation of $S(\cdot, t)$, measured on 1000 random video clips containing 128 frames each.

Figure 5 shows $S(\cdot, t)$ as a function of $t$ for real and generated videos on each dataset. The curves trend downward over time for real videos as content and scenery gradually change. StyleGAN-V and DIGAN are biased toward colors changing too slowly — both of these models include a global latent code that is fixed over the entire video. On the other extreme, MoCoGAN-HD and TATS are biased toward colors changing too quickly. These models use recurrent and autoregressive networks, respectively, both of which suffer from accumulating errors. Our model closely matches the shape of the target curve, indicating that colors in our generated videos change at appropriate rates.

Color change is a crude approximation of the complex changes over time in videos. In Appendix A.3 we also consider LPIPS [68] perceptual distance instead of color similarly and observe the same trends in most cases.

### 5.3 Fréchet video distance (FVD)

The commonly used Fréchet video distance (FVD) [57] attempts to measure similarity between real and generated video distributions. We find that FVD is sensitive to the realism of individual frames and motion over short segments, but that it does not capture long-term realism. For example, FVD is essentially blind to unrealistic repetition of content over time, which is prominent in StyleGAN-V videos on all of our datasets. We found FVD to be most useful in ablations, i.e., when comparing slightly different variants of the same architecture.

FVD [57] computes the Wasserstein-2 distance [59] between sets of real and generated features extracted from a pre-trained I3D action classification model [5]. Skorokhodov *et al.* [52] note that FVD is highly sensitive to small implementation differences, down to the level of image compression settings, and that the reported results are not necessarily comparable between papers (Appendix C in [52]). We report all FVD results using consistent evaluation protocol, ensuring apples-to-apples comparison. We separately measure FVD using 128- and 16-frame segments, denoted by $FVD_{128}$ and $FVD_{16}$, and sample 2048 random segments from both the dataset and generator in each case.

Table 1 (left) reports FVD on all datasets for StyleGAN-V and our model. We outperform StyleGAN-V on horseback riding and mountain biking datasets that contain more complex changes over time, but underperform on ACID and slightly underperform on SkyTimelapse in terms of $FVD_{128}$. However, this underperformance strongly disagrees with the conclusions from the qualitative user study in Section 5.1. We believe this discrepancy comes from StyleGAN-V producing better individual frames, and possibly better small-scale motion, but falling seriously short in recreating believable long-term realism – and the FVD being sensitive primarily to the former aspects. Table 1 (right) reports FVD metrics on MoCoGAN-HD, TATS, DIGAN and our model for SkyTimelapse at $128^2$; we outperform all baselines in terms of $FVD_{128}$ on this comparison.

| | $FVD_{128}$ | $FVD_{16}$ |
|---|---|---|
| Ours (128 frames) | 113.7 | 83.8 |
| 16 frames | 163.6 | 108.5 |
| 2 frames | 396.8 | 169.4 |

(a) Ablation of training sequence length

| | $FVD_{128}$ | $FVD_{16}$ |
|---|---|---|
| Ours | 113.7 | 83.8 |
| 0.1× lowpass width | 153.1 | 113.2 |
| 10× lowpass width | 217.9 | 126.5 |

(b) Ablation of temporal lowpass filter footprint

Table 2: **(a)** Our model learns to generate realistic long videos by training on long videos; decreasing the sequence length used during training is consistently harmful. **(b)** The footprint of the temporal lowpass filters plays an important role in producing inputs to the low-resolution mapping network at appropriate temporal frequencies; changing the footprint by an order of magnitude hurts performance.

(a) Mountain biking

(b) ACID

| | | $FVD_{128}$ |
|---|---|---|
| Biking | Ours | 113.7 |
| | SR on reals | 58.3 |
| ACID | Ours | 166.6 |
| | SR on reals | 68.8 |

(c) Ablation

Figure 6: Evaluation of the super-resolution network. **(a,b)** Generated low-resolution frames and the corresponding high-resolution frames produced by the super-resolution network. **(c)** The super-resolution network yields remarkably good FVD when provided with real low-resolution videos as input; the overall quality of our results is largely dictated by the low-resolution generator.

### 5.4 Ablations

**Training on long videos improves generation of long videos.** Observing long videos during training helps our model learn long-term consistency, which is illustrated in Table 2a that ablates the sequence length used during training of the low-resolution generator. We found that the benefits of training with long videos only became evident after designing a generator architecture with appropriate temporal receptive field to utilize the rich training signal. Note that even though we ablate aspects of the low-resolution generator, we still compute FVD using the final high-resolution videos produced by the super-resolution network.

**Footprint of the temporal lowpass filters.** Our temporal latent representation serves a vital role in expanding the receptive field of our generator, modeling patterns over different time scales, and enabling the generation of new content over time. While we primarily leverage long training videos to learn long-term consistencies from data, the size of our temporal lowpass filters plays a role in encouraging the low-resolution mapping network to learn correlations at appropriate time scales. Table 2b demonstrates the negative impact of using inappropriately sized filters. We find that our model performs well with the same filter configuration for all datasets, although it is possible that the ideal settings may vary slightly between datasets.

**Effectiveness of the super-resolution network.** Figure 6a,b shows examples of low-resolution frames generated by our model along with the corresponding high-resolution frames produced by our super-resolution network; we find that the super-resolution network generally performs well. To ensure that the quality of our results is not disproportionately limited by the super-resolution network, we further measure FVD when providing the super-resolution network with *real* low-resolution videos as input in Figure 6c. Indeed, FVD greatly improves in this case, which indicates that there are still significant gains to be realized by further improving the low-resolution generator.

### 5.5 Failure cases

Separate low- and super-resolution networks makes the problem computationally feasible, but it may somewhat compromise the quality of the final high-resolution frames. We observed that "swirly" artifacts are most prominent in the super-resolution output and not in the low-resolution output. Our model also struggles with long-term consistency of small details (e.g., distant jumps in generated horseback riding videos) that begin to appear before quickly fading out. We believe these issues are due to limitations of our super-resolution network, and that improving the super-resolution network

would benefit the model in this regard. Another failure case we observed is difficulty preserving 3D consistency for scenes with very little motion, such as in the ACID dataset. In cases where there is little motion, one may consider using an explicit 3D representation.

## 6 Conclusions

Video generation has historically focused on relatively short clips with little new content over time. We consider longer videos with complex temporal changes, and uncover several open questions and video generation practices worth reassessing — the temporal latent representation and generator architecture, the training sequence length and recipes for using long videos, and the right evaluation metrics for long-term dynamics.

We have shown that representations over many time scales serve as useful building blocks for modeling complex motions and the introduction of new content over time. We feel that the form of the latent space most suitable for video remains an open, almost philosophical question, leaving a large design space to explore. For example, what is the right latent representation to model persistent objects that exit from a video and re-enter later in the video while maintaining a consistent identity?

The benefits we find from training on longer sequences open up further questions. Would video generation benefit from even longer training sequences? Currently we train using segments of adjacent frames, but it might be beneficial to use larger frame spacings to cover longer time spans.

Quantitative evaluation of the results continues to be challenging. As we observed, FVD goes only a part of the way, being essentially blind to repetitive, even very implausible results. Our tests with how the colors and LPIPS distance change as a function of time partially bridge this gap, but we feel that this area deserves a thorough, targeted investigation of its own. We hope our work encourages further research into video generation that focuses on more complex and longer-term changes over time.

**Negative societal impacts** Our work falls within data-driven generative modeling, which, as a field, has well known potential for misuse with increasing quality improvements. The training of video generators is even more intensive computationally than training still image generators, increasing energy usage. Our project consumed 300MWh on an in-house cluster of V100 and A100 GPUs.

**Acknowledgements** We thank William Peebles, Samuli Laine, Axel Sauer and David Luebke for helpful discussion and feedback; Ivan Skorokhodov for providing additional results and insight into the StyleGAN-V baseline; Tero Kuosmanen for maintaining compute infrastructure; Elisa Wallace Eventing (`https://www.youtube.com/c/WallaceEventing`) and Brian Kennedy (`https://www.youtube.com/c/bkxc`) for videos used to make the horseback riding and mountain biking datasets. Tim Brooks is supported by the National Science Foundation Graduate Research Fellowship under Grant No. 2020306087.

## References

[1] Dinesh Acharya, Zhiwu Huang, Danda Pani Paudel, and Luc Van Gool. Towards high resolution video generation with progressive growing of sliced wasserstein gans. *CoRR*, abs/1810.02419, 2018.

[2] Adil Kaan Akan, Sadra Safadoust, Erkut Erdem, Aykut Erdem, and Fatma Güney. Stochastic video prediction with structure and motion. *CoRR*, abs/2203.10528, 2022.

[3] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. In *Proc. ICLR*, 2018.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Proc. NeurIPS*, 33:1877–1901, 2020.

[5] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, pages 4724–4733, 2017.

[6] Silvia Chiappa, Sébastien Racaniere, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. In *Proc. ICLR*, 2017.

[7] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

[8] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *Interspeech*, 2018.

[9] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *CoRR*, abs/1907.06571, 2019.

[10] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[12] Gereon Fox, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. Stylevideogan: A temporal generative model using a pretrained stylegan. *CoRR*, abs/2107.07224, 2021.

[13] 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. *CoRR*, abs/2204.03638, 2022.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Proc. NIPS*, 27, 2014.

[15] David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018.

[16] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *Proc. CVPR*, pages 3897–3906, 2019.

[17] 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 I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[18] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[19] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *CoRR*, abs/2204.03458, 2022.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[22] James F Kaiser. Nonrecursive digital filter design using the i_0-sinh window function. In *Proc. 1974 IEEE International Symposium on Circuits & Systems, San Francisco DA, April*, pages 20–23, 1974.

[23] Nal Kalchbrenner, Aäron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *Proc. ICML*, pages 1771–1779, 2017.

[24] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE transactions on computational imaging*, 2(2):109–122, 2016.

[25] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proc. ICLR*, 2018.

[26] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020.

[27] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021.

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4401–4410, 2019.

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proc. CVPR*, pages 8110–8119, 2020.

[30] Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. Drivegan: Towards a controllable high-quality neural simulation. In *Proc. CVPR*, pages 5820–5829, 2021.

[31] Seung Wook Kim, Yuhao Zhou, Jonah Philion, Antonio Torralba, and Sanja Fidler. Learning to Simulate Dynamic Environments with GameGAN. In *Proc. CVPR*, Jun. 2020.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[34] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. In *Proc. ICLR*, 2020.

[35] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance. *arXiv preprint arXiv:2203.06026*, 2022.

[36] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *CoRR*, abs/1804.01523, 2018.

[37] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proc. ICCV*, 2021.

[38] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proc. CVPR*, 2021.

[39] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *CoRR*, abs/2003.04035, 2020.

[40] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, 2018.

[41] Charlie Nash, João Carreira, Jacob Walker, Iain Barr, Andrew Jaegle, Mateusz Malinowski, and Peter Battaglia. Transframer: Arbitrary frame prediction with generative models. *CoRR*, abs/2203.09494, 2022.

[42] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. *CoRR*, abs/2006.10704, 2020.

[43] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Proc. NeurIPS*, 32, 2019.

[44] Xuanchi Ren and Xiaolong Wang. Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image. In *Proc. CVPR*, 2022.

[45] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*, 2018.

[46] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv:2104.07636*, 2021.

[47] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proc. ICCV*, pages 2830–2839, 2017.

[48] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision*, 128(10):2586–2606, 2020.

[49] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *Proc. CVPR*, 2018.

[50] Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. In *Proc. SIGGRAPH*, page 489–498, 2000.

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[52] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. *CoRR*, abs/2112.14683, 2021.

[53] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. In *Proc. ICCV*, 2013.

[54] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *Proc. ICCV*, 2017.

[55] 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 *Proc. ICLR*, 2021.

[56] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proc. CVPR*, pages 1526–1535, 2018.

[57] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *CoRR*, abs/1812.01717, 2018.

[58] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Proc. NeurIPS*, 2020.

[59] Leonid Nisonovich Vaserstein. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 5(3):64–72, 1969.

[60] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Proc. NIPS*, 2016.

[61] Jacob Walker, Ali Razavi, and Aäron van den Oord. Predicting video with vqvae. *CoRR*, abs/2103.01950, 2021.

[62] Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen Qian, Ran He, Yu Qiao, and Chen Change Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *Proc. ECCV*, 2020.

[63] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proc. CVPR*, 2021.

[64] Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *Proc. CVPR*, pages 2364–2373, 2018.

[65] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *CoRR*, abs/2104.10157, 2021.

[66] 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 *Proc. ICLR*, 2022.

[67] Jiangning Zhang, Chao Xu, Liang Liu, Mengmeng Wang, Xia Wu, Yong Liu, and Yunliang Jiang. Dtvnet: Dynamic time-lapse video generation via single still image. In *Proc. ECCV*, 2020.

[68] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[69] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *Proc. NeurIPS*, 2020.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] `https://github.com/NVlabs/long-video-gan`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Training video generators is expensive and we did not have capacity to repeat experiments with multiple seeds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 6
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See appendix.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] `https://github.com/NVlabs/long-video-gan`
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Section 4
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See appendix.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] See appendix.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] See appendix.

# A Additional results

## A.1 User study

We conducted a user study on Amazon Mechanical Turk to gauge realism of motion generated by our method in comparison to StyleGAN-V, as discussed in Section 5.1 of the main paper. While the user study is on a relatively small scale and does not measure all aspects of video quality, it provides an important signal about realism that is not captured by the Fréchet video distance (FVD) [57] metric. FVD does not favor our method on all datasets, but we observe a substantial qualitative improvement regarding generation of motion and introduction of new content over time. The user study shows preference for videos generated by our method on all datasets, corroborating this observation.

For our user study we create 50 pairs of videos for each of the four datasets, where each pair has one random video from our method and one random video from StyleGAN-V. We instruct participants to select the favorable video in a forced-choice response: "Pick the video that is MORE realistic. For each comparison, you will be presented two videos. Please click each video to view it. Please pick the video that contains more realistic motions." See Figure 7 for a screenshot of instructions provided to participants and Table 3 for the portion of responses that favor our method compared to StyleGAN-V. Our method was preferred over 80% of the time for every dataset.

Each video pair was shown to 10 participants resulting in 500 responses per dataset. Each participant gave responses for 5 different video pairs. We select workers who have a past approval rating over 95% and who have completed over 1000 jobs. Our user study uses participants to complete a labeling task to measure video realism; humans are not the subjects and we do not study the participants themselves. IRB review is not applicable. Based on the average completion time, the hourly wage per participant ranged from $6 to $9.

| | Mountain biking | Horseback riding | ACID | SkyTimelapse |
|---|---|---|---|---|
| StyleGAN-V | 16.4% | 13.4% | 19.4% | 18.4% |
| Ours | 83.6% | 86.6% | 80.6% | 81.6% |

Table 3: Percent of responses that label motions more realistic in videos generated with our method compared with StyleGAN-V in a forced-choice user study with 500 responses per dataset.

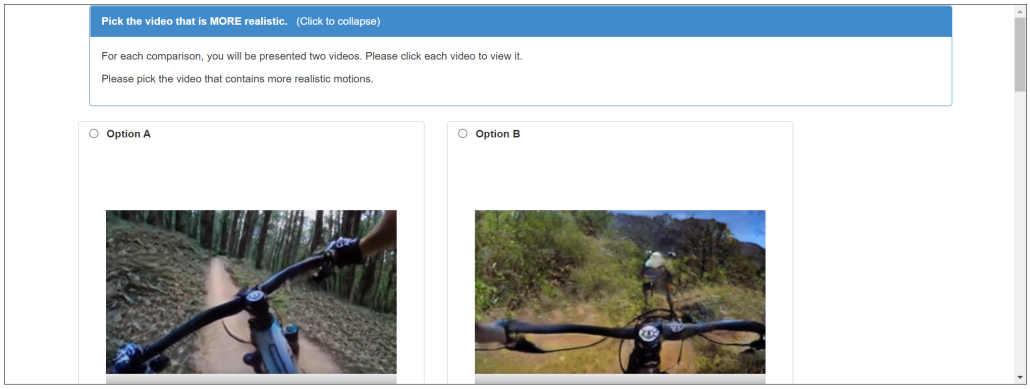



Figure 7: Screenshot of instructions provided to user study participants.

### A.2 Qualitative results

See Figures 8,9,10,11 for qualitative results of our videos compared with baseline methods. Please also see the supplemental webpage to watch the same videos, as well as watch grids of randomly sampled videos for each dataset and method. In all videos, StyleGAN-V [52] fails to generate new content as the video progresses, and instead replays the same content repeatedly (e.g., clouds moving back and forth for the SkyTimelapse dataset).

Figure 8: **Top:** Our mountain biking dataset exhibits complex motions and changes to the environment, such as transitioning between open areas and areas with tree coverage. **Middle:** StyleGAN-V is incapable of generating new content over time and the biker fails to move forward. **Bottom:** Our video generation method produces realistic motion and scenery changes. Over a 10s interval, the biker transitions out of the woods — a natural occurrence when mountain biking.

Figure 9: **Top:** ACID [37] contains nature drone footage with large gradual changes in camera viewpoint. **Middle:** StyleGAN-V produces videos with pulsating camera motion, unable to create the illusion of a smooth camera trajectory. **Bottom:** Our model implicitly learns to generate changes in camera viewpoint over smooth trajectories, such as rotating while moving forward in 3D space.

Figure 10: **Top:** SkyTimelapse [64] ($256^2$ resolution) includes timelapse videos with a stream of new clouds and weather conditions. **Middle:** StyleGAN-V moves the same clouds back and forth. For example, compare the clouds at 1s, 2s and 5s marks: the clouds change between 1s and 2s, but then return back to the same clouds at 5s. **Bottom:** Our model generates new clouds over time.

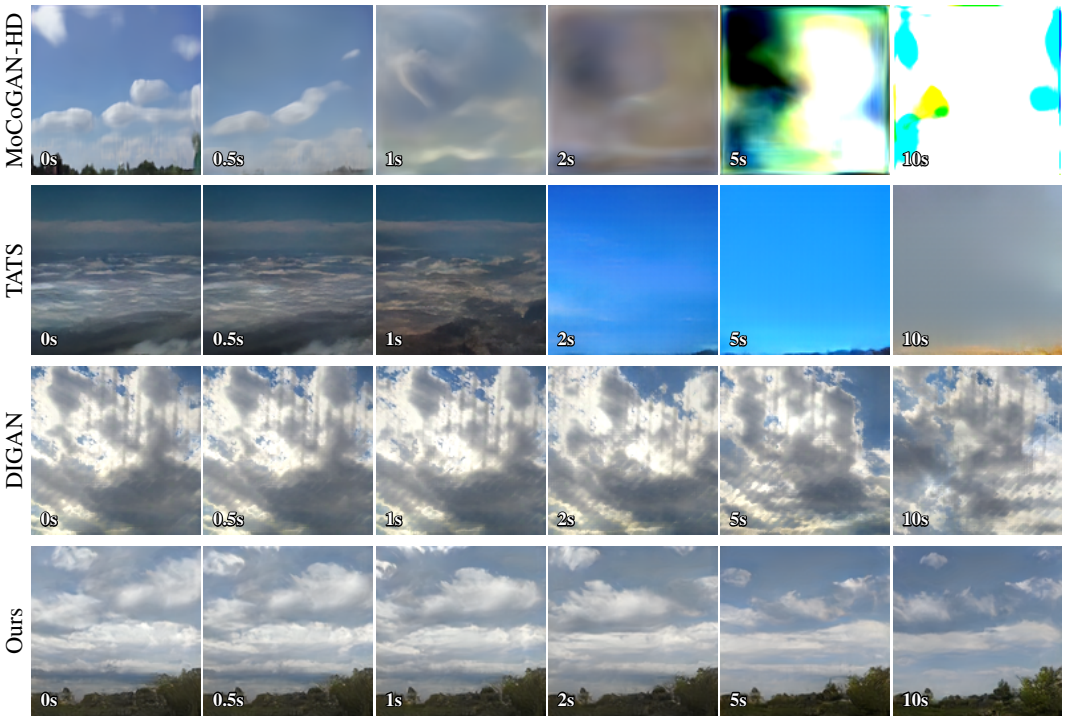
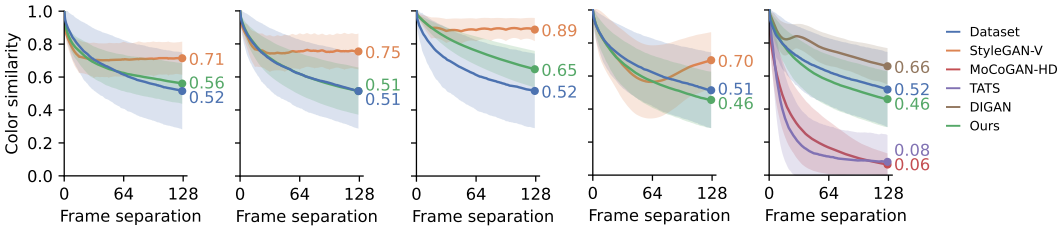
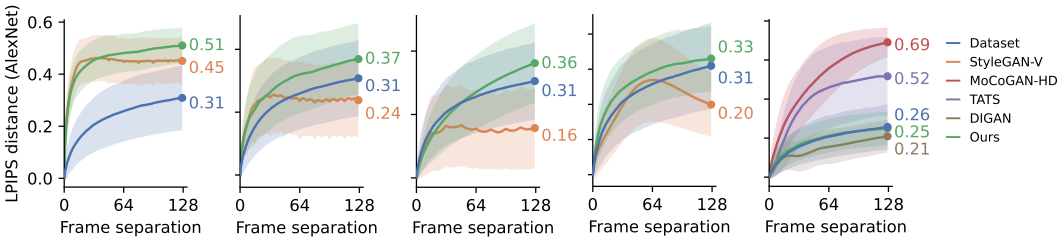



Figure 11: SkyTimelapse [64] ($128^2$ resolution). Real video omitted. **Top:** MoCoGAN-HD [55] is based on a recurrent network in latent space of a pretrained StyleGAN2 [29] model. It produces a realistic initial frame, but the video quickly explodes over a long duration. **2nd:** TATS [13] employs an autoregressive transformer to generate videos. While short segments produce plausible frames, videos change far too rapidly. **3rd:** DIGAN [66] uses an implicit representation to generate videos pixel by pixel. Strong periodic patterns are visible in space and time. **Bottom:** Our model generates videos that are consistent over time.



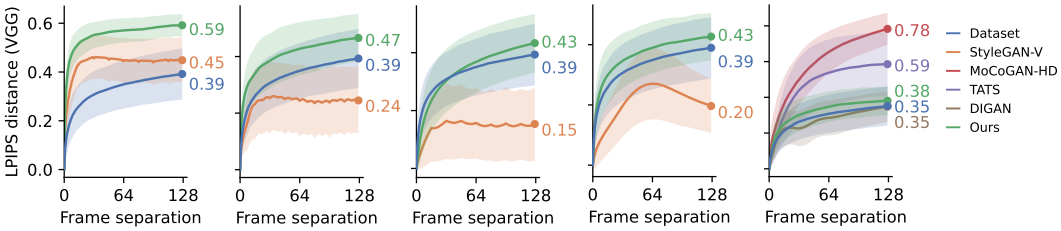

(a) Biking (b) Horseback (c) ACID (d) Sky $256^2$ (e) Sky $128^2$

Figure 12: Color similarity over time (same as Figure 5 in main paper).




(a) Biking (b) Horseback (c) ACID (d) Sky $256^2$ (e) Sky $128^2$

Figure 13: LPIPS distance (AlexNet) over time.




(a) Biking (b) Horseback (c) ACID (d) Sky $256^2$ (e) Sky $128^2$

Figure 14: LPIPS distance (VGG) over time.

### A.3 Analyzing change over time in feature spaces

In Section 5.2 of the main paper, we measure color similarity at increasing frame spacings for different datasets and methods to uncover bias in how much change occurs over time. Intersection of color histograms (Equation 1) is a simple proxy for change over time, and is entirely agnostic to spatial patterns. We include the color similarity plots in Figure 12 of the supplement as well for reference. It is reasonable to also consider other distance functions, such as perceptual similarity metrics [21, 68]. In Figure 13 and Figure 14 we show the LPIPS [68] metric based on AlexNet [33] and VGGNet [51] features respectively. (Note the opposite direction of change: color similarity decreases over time, whereas feature distance *increases* over time.)

In most cases, we observe the same trend as for color similarity — StyleGAN-V changes too slowly in horseback, ACID and SkyTimelapse, and our method does a relatively better job at matching the rate of change in real videos. The mountain biking dataset shows a different trend for perceptual similarity, where both our method and StyleGAN-V curves are shifted too high (too much change), and StyleGAN-V is closer to the dataset curve. One caveat of this use of perceptual metrics is that, even ignoring the temporal aspect, we observe substantial distributional shift of pretrained features between generated and real frames (e.g., penultimate VGG features for both our model and StyleGAN-V have over 30% larger magnitudes than for real frames on the biking dataset). It is thus unclear to what extent the difference in curves between real and generated videos is due to different rates of change over time or the distributional shift of features independent of change over time.

We favor the color similarity measure as the simplest approximation for how quickly things change over time, and acknowledge that it is not intended as a standalone metric but a probe into the biases of videos generated with different methods.

| | Mountain biking | Horseback riding | ACID | SkyTimelapse |
|---|---|---|---|---|
| StyleGAN-V | 33.9 | 51.6 | 11.3 | 12.6 |
| with 10× R1 $\gamma$ | 12.5 | 17.7 | – | – |
| Ours | 18.9 | 12.2 | 18.2 | 26.6 |

Table 4: Video-balanced Fréchet inception distance ($\text{FID}_\text{V}$) measures per-frame image quality, where lower is better. While our emphasis is the time axis, we report image quality to gain insight on the priorities of StyleGAN-V and our model. StyleGAN-V outperforms our model in terms of per-frame image quality on three of the four datasets, which aligns with StyleGAN-V's focus on image quality and our focus on accurate change over time.

### A.4 Image quality tradeoff

In practice, there exists a tradeoff between per-frame image quality and the quality of motion and change over time. At one extreme, an image generator is optimized specifically for image quality. Image generators produce very high quality images, but have no inherent ability to produce realistic videos. Many video generation models prioritize frame quality, whereas our model prioritizes accurate changes over long durations. $\text{FVD}_{128}$ and $\text{FVD}_{16}$ metrics [57] measure unknown combinations of spatial and temporal patterns, and while they provide a useful signal, it is not clear where these metrics fall in terms of favoring per-frame image quality or accurate temporal changes.

We analyze color similarity over time in Section 5.2 of the main paper. Color similarity between frames is agnostic to spatial patterns, and provides insight on the rate of change over time in isolation from per-frame image quality. To gain a holistic picture of the priorities of our model, we also compute a per-frame image quality metric, video-balanced Fréchet inception distance ($\text{FID}_\text{V}$), which we describe below and report in Table 4. StyleGAN-V outperforms our model on three of the four datasets in terms of $\text{FID}_\text{V}$. This tradeoff is expected, since StyleGAN-V is heavily based on the StyleGAN2 [29] image generator. It produces high image quality but is unable to model complex motions or changes over time, whereas our model prioritizes the time axis.

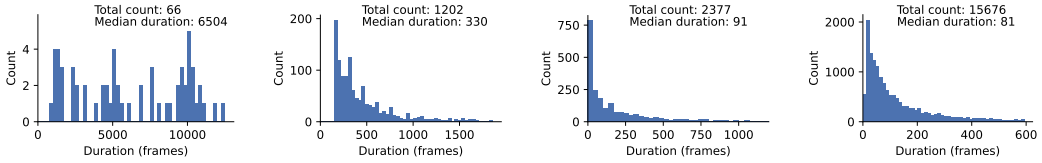
Assessing quality of generated videos is multifaceted, and we believe all of the evaluation we provide — qualitative results, user study, color change over time, FVD, and FID — help expose gaps in the abilities of existing methods and the strengths and weaknesses of our new model.

**Video-balanced Fréchet inception distance ($\text{FID}_\text{V}$)** To correctly measure per-frame image quality, it is important to balance the computation of FID [17] such that very long videos in the dataset do not overpower results. (This is particularly important for the SkyTimelapse [64] dataset, which has an outlier video that is extremely long.) Skorokhodov *et al.* [52] point out that it is undesirable for these very long videos to bias training or computing FVD [57], and the same is true for computing FID [17] per-frame on video data.

To correctly balance FID to value each training video equally, we weight calculation of the covariance and mean by the inverse of the number of frames in each clip when measuring the Wasserstein-2 distance [59] between sets of features. This has the effect of valuing each video equally, while still including contribution from all frames, which is important when there are a small number of long videos such as in our horseback riding dataset. A similar strategy to weight covariance and mean when computing FID is used by Kynkäänniemi *et al.* [35] to analyze the effect of balancing object class occurrences. When computing statistics for generated frames, we sample $50\,000$ videos of length 1 frame (at $t = 0$ for StyleGAN-V).

## B Dataset details

We evaluate our model using two existing datasets, Aerial Coastline Imagery Dataset (ACID) [37] and SkyTimelapse [64], and two new datasets: horseback riding and mountain biking. We center crop videos to the desired aspect ratio if needed (16×9 for all datasets except SkyTimelapse, for which we use a square crop to match prior work), and then resize to the target resolution using the PIL library's Lanczos resampling method. For the ACID dataset we combine both train and test splits to maximize




(a) Horseback (ours) (b) Mountain bike (ours) (c) SkyTimelapse [64] (d) ACID [37]

Figure 15: Counts and durations of training videos. Training a model to prioritize the time axis requires training on long videos. Existing video datasets, such as (c) and (d), include relatively short videos with median durations of 91 and 81 frames respectively. We introduce two new datasets of longer videos, (a) and (b), with median durations of 6504 and 330 frames. We show results on all four of these datasets.

| | Horseback riding | | Mountain biking | |
|---|---|---|---|---|
| | # Videos | Total duration | # Videos | Total duration |
| Videos considered | 194 | 27h:29m:42s | 48 | 38h:46m:56s |
| Videos selected | 44 | 7h:21m:49s | 28 | 9h:06m:50s |
| Clips extracted | 66 | 4h:01m:41s | 1202 | 5h:07m:55s |

Table 5: We manually curate horseback riding and mountain biking datasets in two phases: first by selecting source videos containing sufficient first-person footage with stable motion and a consistent camera perspective, and then by extracting clips free from scene changes, text overlays, or other unwanted content. Here we report the number of videos and total duration of video content at each phase of curation.

the amount of training data. For the SkyTimelapse dataset we use only the train split to ensure our model is comparable with prior work.

Figure 15 shows histograms of the durations and counts of training videos for all four datasets. Our new datasets both feature longer median clip lengths than the existing datasets. When training our model, we filter ACID and SkyTimelapse datasets for clips with at least 128 frames. We allow the StyleGAN-V baseline to train on all clips with at least 3 frames (the number needed by their method). Both datasets can be obtained from their respective project webpages. ACID: `https://infinite-nature.github.io/`, and SkyTimelapse: `https://sites.google.com/site/whluoimperial/mdgan`. The copyright status of both existing datasets is ambiguous, as neither specify a license or details about content ownership. We ensure to attain explicit licenses for our two new datasets below.

### B.1 Horseback riding

We introduce a new dataset of first-person horseback riding that we will release to the public for research purposes. The videos were created by Wallace Eventing and examples of the videos can be found on their YouTube channel: `https://www.youtube.com/c/WallaceEventing`. We reached out directly and received permission to create a dataset from their videos to use in our research and release as a dataset for non-commercial research purposes. We will release the filtered and processed video frames directly, which avoids inconsistent versions of the dataset when videos become unavailable or are processed differently. The dataset will be released under a custom license agreed upon with Wallace Eventing that permits use for non-commercial research purposes but does not allow redistribution of the dataset.

The videos contain first-person helmet camera footage of horseback riding events, with little or no personally identifying information visible. They are high quality (1080p) at 60fps, although we subsample frames to attain 30fps. Statistics of our dataset filtering are presented in Table 5. The dataset was sourced from 194 original videos, which we then filtered down to 44 videos with stabilized motion and a consistent camera perspective. We manually extracted 66 clips from the

selected videos, cutting out scene changes, text overlays, videos with obstructed views, and the beginnings and ends of videos.

### B.2 Mountain biking

We also introduce a new dataset of first-person mountain biking that we will release to the public. The videos were created by Brian Kennedy (BKXC) and examples of the videos can be found on their YouTube channel: `https://www.youtube.com/c/bkxc`. We reached out directly and received permission to create a dataset from their videos to use in our research and release as a dataset under a CC BY 4.0 license.

The videos contain first-person mountain biking. There is little personally identifying information visible, although there are occasional other bikers who pass by and whose faces can be seen. The videos are high quality (2160p) at 30fps. This dataset underwent much more extensive filtering and extraction of training clips since the source videos contain many cuts and abrupt changes. See Table 5 for statistics of our dataset curation. From 48 source videos we selected 28 videos with ample footage of stable mountain biking, and then manually filtered for contiguous segments of mountain biking that were at least 5 seconds long, resulting in 1202 total clips.

## C Low-resolution implementation details

### C.1 Augmentation

We find that overfitting of the discriminator network is particularly severe when training with long sequences. To alleviate the overfitting, we apply DiffAug [69] to real and generated videos prior to the discriminator. We use all categories of DiffAug augmentations — color, cutout, and translation — with default strengths for color and cutout augmentations, and maximum x- and y-translations of 32 pixels for the square SkyTimelapse dataset and 16 pixels for the non-square biking, horseback and ACID datasets. We also tried using the ADA [26] adaptive augmentation strategy, but it caused leakage of augmentations into the generated videos, even when augmentations were applied with low probability.

In addition to DiffAug, we employ fractional time stretching augmentation, where we resize the temporal axis by a factor of $s = 2^a$ for $a \sim \mathcal{U}(-1, 1)$ with linear interpolation and zero padding. If time stretching augmentation upsamples the time axis, the video is randomly cropped to fit within the original 128-frame window. Similarly, if time stretching augmentation downsamples the time axis, the video is zero padded with random amounts before and after to fit within the original 128-frame window. Fractional time stretching augmentation is related to subsampling augmentation that is commonly used by other methods [52], but supports a greater variety of augmentations since temporal scaling amounts are fractional. Further investigation into the best augmentation policies for video generation models is an important future area for investigation.

### C.2 Temporal lowpass filters

To capture long-term temporal correlations in the intermediate latent codes, we enrich each of 8 channels of input temporal noise with a set of $N = 128$ lowpass filters $\{f_i\}$, as described in Section 3.1 of the main paper. Specifically, we use Kaiser lowpass filters [22], following the implementation of [27]. We space lowpass filter sizes exponentially, where each filter has temporal footprint $k_i = k_{\text{min}} \left(\frac{k_{\text{max}}}{k_{\text{min}}}\right)^{\frac{i}{N-1}}$ where $0 \leq i < N$, $k_{\text{min}} = 500$ and $k_{\text{max}} = 10000$.

### C.3 Discriminator architecture

Our low-resolution discriminator architecture is heavily inspired by the StyleGAN [28] discriminators, with the addition of spatiotemporal and temporal processing in order to model realistic motions and changes over time. See Figure 16 for a depiction of the discriminator architecture.

The video is first expanded from 3 RGB channels to 128 channels using a 1×1 convolutional layer. The first block only operates spatially, downsampling height and width by 2× and using 3×3 spatial convolutions. The remaining 3 blocks downsample both spatially and temporally and use 5×3×3

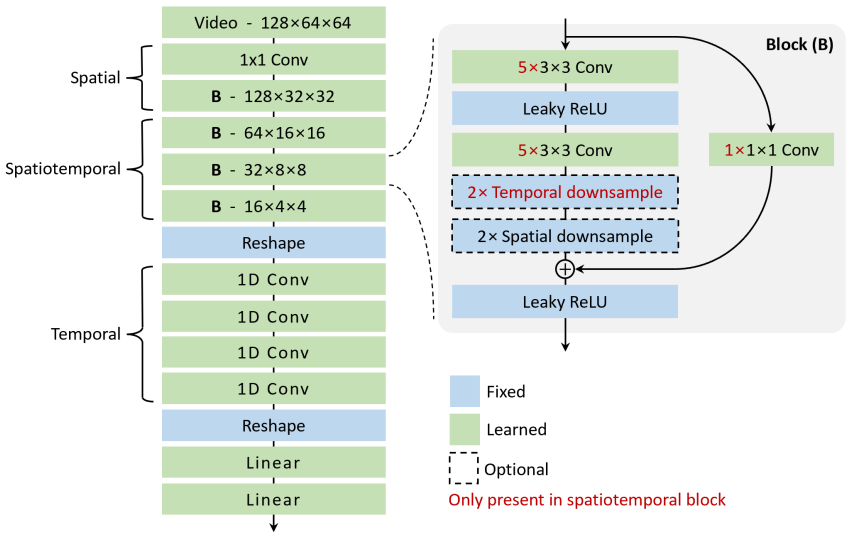



Figure 16: Low-resolution discriminator architecture. **Left:** The input video undergoes a single $1\times1$ convolutional layer, followed by 4 residual blocks. Features are then reshaped, combining spatial and channel dimensions, followed by 4 temporal 1D convolutional layers. Finally, features are flattened, followed by 2 linear layers to produce output logits. **Right:** The residual block follows the structure of discriminator blocks in StyleGAN [28] models, with optional temporal downsampling and 3D spatiotemporal convolutions used for all but the first block.

spatiotemporal convolutions. We omit temporal processing from the first block to save compute, since running 3D convolutions at the full resolution is substantially more expensive. We otherwise find the inclusion of temporal processing crucial for the model to learn temporal dynamics. In each block, the number of channels is doubled until reaching 512.

To further prioritize learning accurate motions and changes over time, we include $4\times$ 1D temporal convolutions, each with a kernel size of 5 and followed by a LeakyReLU nonlinearity. Finally, following the StyleGAN discriminator, features are flattened and passed through 2 linear layers with a LeakyReLU nonlinearity in between to produce the final logits.

### C.4 Training

We use a batch size of 64 videos, each of length 128 frames. We trained models with a variety of single- and multi-node jobs. We train each run for a maximum of $100\,000$ steps and cut training runs short if FVD begins increasing. Training the low-res generator takes $1.7$ days for the maximum $100\,000$ steps using $4\times$ nodes each containing $8\times$ NVIDIA A100 GPUs. The low-res generator has 83.2M parameters and the low-res discriminator has 46.4M parameters. We use R1 regularization [40] with $\gamma = 1$ for non-square datasets, and $\gamma = 4$ for the square SkyTimelapse dataset. We train with the Adam optimizer [32] with generator learning rate of $0.003$, discriminator learning rate of $0.002$, and $\beta_1 = 0$ and $\beta_2 = 0.99$ for both generator and discriminator. (Note: Adam with $\beta_1 = 0$ is equivalent to RMSprop [18] with the bias correction term from Adam.) We use an exponential moving average of the generator weights, with $\beta_{\text{ema}} = 0.99985$. We select the checkpoint with best $\text{FVD}_{128}$.

## D Super-resolution implementation details

### D.1 Augmentation

The super-resolution network undergoes augmentation of two forms: (1) augmentation of real and generated videos applied prior to the discriminator to prevent overfitting, and (2) augmentation of conditional real low resolution videos during training to improve generalization to *generated* low resolution videos at inference time.

**Discriminator augmentation to prevent overfitting** Augmentation to prevent discriminator overfitting uses ADA [26] with default settings, and applies the same augmentations to all frames from both high and low resolution videos. To additionally prevent overfitting and prevent the discriminator from focusing too much attention on the conditioning signal, we employ strong dropout augmentation with probability $p = 0.9$ of zeroing out the entire conditional low resolution video. This augmentation occurs before the discriminator only, and does not affect the inputs to the super-resolution network.

**Low-resolution conditioning augmentation to improve generalization** We train our super-resolution network with real low resolution videos as conditioning, but use generated low resolution videos at inference time. There exists a domain gap between the real and generated low resolution videos, and to ensure our super-resolution network is robust to the domain gap, we augment real low resolution videos during training. Similar strategies are used in image generators with super-resolution refinement [19], where corruption is added to real low resolution inputs during training. We use a modified version of the ADA [26] augmentation pipeline, only enabling additive Gaussian noise, isotropic and non-isotropic scaling, rotation, and fractional translation. Each augmentation is applied to the entire low resolution video with a fixed probability of $50\%$, and with much smaller strengths than the default pipeline (`noise_std=0.08`, `scale_std=0.08`, `aniso_std=0.08`, `rotate_max=0.016`, `xfrac_std=0.016`). This augmentation is applied in the dataset pipeline and affects conditional inputs to the discriminator and super-resolution network only during training.

### D.2 Prefiltering of low-res conditioning

The low resolution frame being upsampled is concatenated with $4$ frames before and $4$ frames after in the low resolution video sequence creating a stack of $9$ low resolution frames. The stack is then resized and concatenated with features at each layer of the StyleGAN3 generator. We experimented with different prefiltering strengths when resizing the $9$ conditioning frames, and found that strong prefiltering helps remove aliasing in the final video. This is related to the anti-aliasing properties of the StyleGAN3 generator that includes strong filtering of intermediate features [27]. Importantly, we do not prefilter the conditional frames when the input is the same resolution as the features (i.e., $64\times64$) since we found that negatively impacts the results. We only apply prefiltering when resizing, and we use the same prefiltering kernels as early layers of StyleGAN3.

### D.3 Training

We use a batch size of 32 videos. The discriminator network inputs real and generated videos of length 4 frames, and for each generated frame the super-res network is provided 9 input frames (4 neighboring frames on either side of the primary frame) to provide temporal context. The network architectures share details with StyleGAN3 [27], except the differences mentioned in Section 3.2 of the main paper. We train for a maximum of $275\,000$ steps, which takes 6.8 days using one node of $8\times$ 16GB NVIDIA V100 GPUs. The super-res network has 27.2M parameters, and the discriminator network has 24.0M parameters. We use R1 regularization with $\gamma = 1$ for all datasets. We train with the Adam optimizer with generator and discriminator learning rate of $0.003$, $\beta_1 = 0$ and $\beta_2 = 0.99$. We use an exponential moving average of the generator weights with $\beta_{\text{ema}} = 0.99985$. We select the checkpoint with best $\text{FVD}_{16}$ when evaluated using real low resolution conditioning, and use the same super-resolution network for many low-resolution experiments.