Generating Long Videos of Dynamic Scenes

Anonymous Author(s)
Affiliation
Address
email

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 bias 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. To this end, 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 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 realistic long videos thus requires the ability to produce endless new content while simultaneously incorporating the 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 shows an example of the rich motion and scenery changes that our model is capable of generating.

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

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

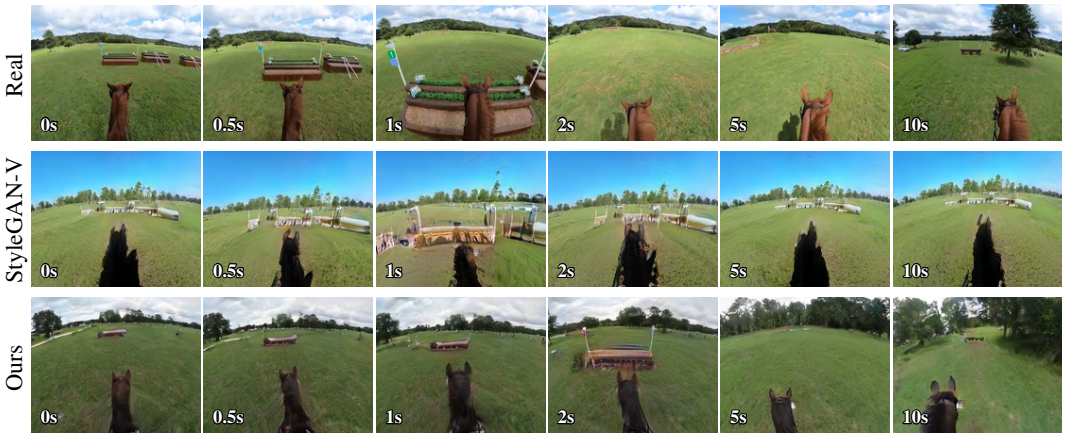


Figure 1: We aim to generate videos that accurately portray motion, changing camera viewpoint, and new content that arises over time. **Top:** Our horseback riding dataset exhibits these types of changes as the horse moves forward in the environment. **Middle:** StyleGAN-V, a state-of-the-art video generation 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.

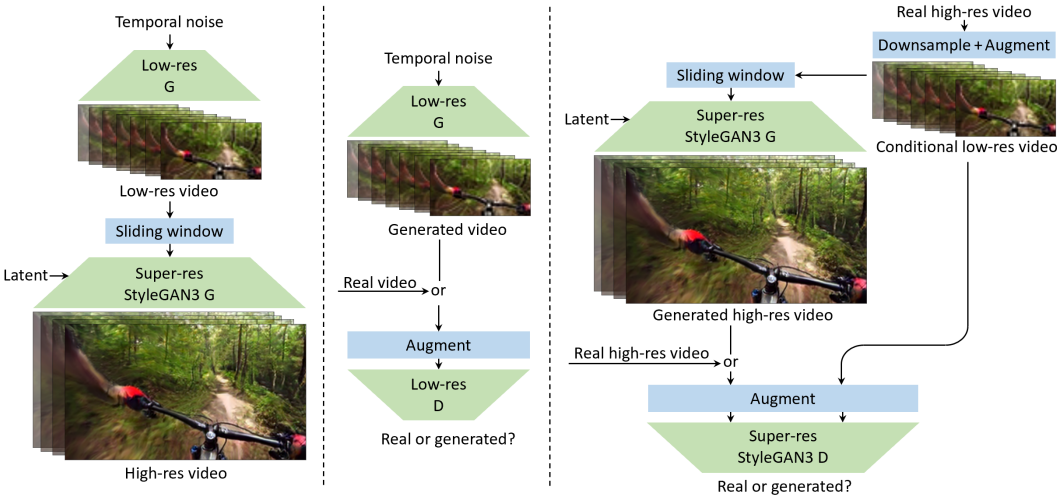
videos at low resolution and then refine them using a separate super-resolution network. Naively training on long videos at high spatial resolution is prohibitively expensive, but we observe that the main aspects of a video persist at a low spatial resolution. This observation allows us to train with 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.

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. We will release our implementation, new datasets, and pre-trained models on these datasets upon publication.

2 Prior work

Video generation is a challenging problem with a long history. The classic early works, Video Textures [39] 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 [39] 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 [13], including early models that output fixed-length videos [1, 37, 48] and approaches that use recurrent networks to produce a sequence of latent codes used to generate frames [9, 11, 43, 44]. MoCoGAN [44] explicitly disentangles "motion" from "content" and keeps the latter fixed over the entire generated video. StyleGAN-V [40] 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 [43], which we also compare with, and StyleVideoGAN [11] attempt to generate videos by navigating the latent space of a pretrained StyleGAN2 model [23], but struggle to produce realistic motion. Unlike previous StyleGAN-based [22] 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 [53] that employs an implicit representation to generate the video pixel by pixel.



(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.

Transformers are another class of models used for video generation [12, 32, 49, 52]. We compare with TATS [12] that generates long unconditional videos with transformers, improving upon VideoGPT [52]. 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 [16].

Conditional video prediction. A separate line of research focuses on predicting future video frames conditioned on one or more real video frames [3, 17, 26, 27, 30, 31] or past frames accompanied by an action label [6, 14, 24, 25]. Some video prediction methods focus specifically on generating infinite scenery by conditioning on camera trajectory [29, 34] and/or explicitly predicting depth [2, 29] 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, 19, 33, 36, 46], and transformer-based video generators also employ a related two-phase setup [52, 12]. 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 explicitly designed to enable generating arbitrarily long videos with rich long-term dynamics by utilizing the ability to train with 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 several seconds apart. 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 30 frames, any two frames taken more than one second 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[1]. 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.

We will first describe the training and architecture of the low-resolution generator in Section 3.1 and then proceed to 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 [20]. 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 [20, 55] to be necessary in order to stabilize the training. We employ DiffAug [55] 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 the supplemental 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, allowing it to represent temporal correlations at multiple time scales.

We employ a style-based design, similar to Karras *et al.* [23, 21], 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 have a vast influence on the scene composition and temporal behavior of several frames through hierarchical 3D convolutions that appear in the main path.

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 the supplemental for details.

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

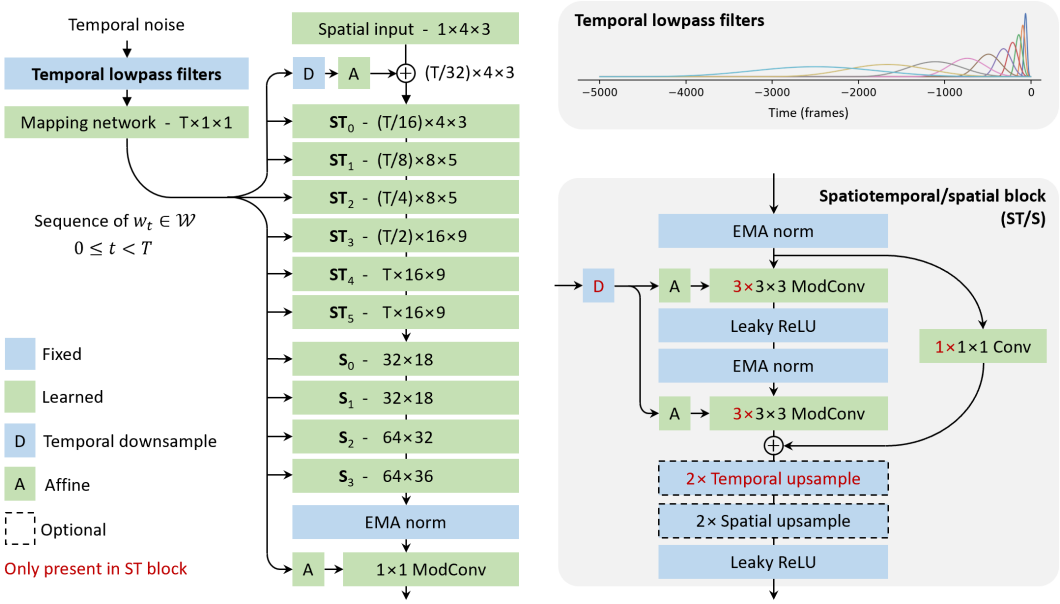


Figure 3: Low-resolution generator architecture, illustrated for 64×36 output. **Left:** The input temporal noise is mapped to a sequence of *intermediate latents* $\{w_t\}$ that modulate the intermediate activations of the main synthesis path. **Top right:** To facilitate the modeling of long-term dependencies, 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.

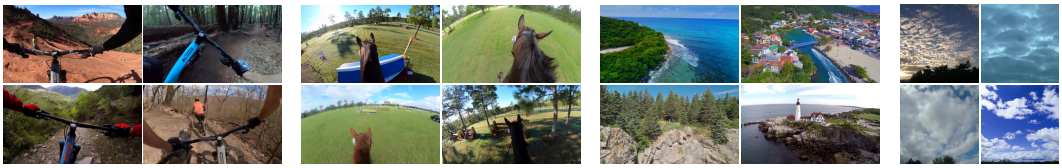
The main synthesis path starts by downsampling the temporal resolution of $\{w_t\}$ by $32\times$ and concatenating it with a learned constant at 4^2 resolution. It then gradually increases the temporal and spatial resolutions through a series of processing blocks, illustrated in Figure 3 (bottom right), focusing first on the time dimension (ST) and then the spatial dimensions (S). The processing blocks consist of the same basic building blocks as StyleGAN2 [23] and StyleGAN3 [21] with the addition of a skip connection; the intermediate activations are normalized before each convolution [21] and modulated [23] according to an appropriately downsampled copy of $\{w_t\}$. In practice, we employ bilinear upsampling [22] and use padding [21] for the time axis to eliminate boundary effects.

For the discriminator, we employ a similar architecture that prioritizes the time axis via wide temporal receptive field, 3D spatiotemporal and 1D temporal convolutions, and spatial and temporal downsamples; see the supplemental 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 [21] for conditional frame generation. Unlike 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 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, which we found unnecessary in practice. Both



(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.

low- and high-resolution segments of 4 frames undergo adaptive augmentation [20] 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), which prevents the discriminator from relying too heavily on the conditioning signal; see the supplemental for details.

We find it remarkable that 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 [15, 18, 38, 42] is a promising avenue for future work.

4 Datasets

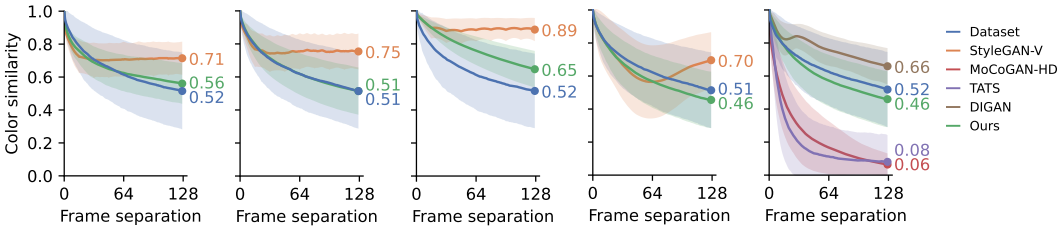
Most of the existing video datasets introduce little or no new content over time. For example, talking head datasets [8, 35, 50, 51] show the same person for the duration of each video. UCF101 [41] portrays diverse human actions, but the videos are short and contain limited camera motion and little or 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 [28] (Figure 4c) that contains significant camera motion but lacks other types of motion, as well as the commonly used SkyTimelapse dataset [54] (Figure 4d) that exhibits new content over time as the clouds pass by, but the videos are relatively homogeneous and the camera remains fixed.

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 [40] on all datasets. Mountain biking, horseback riding and ACID [29] datasets contain videos with a $16{\times}9$ widescreen aspect ratio. We train at $256{\times}144$ resolution on these datasets to preserve the aspect ratio. Since StyleGAN-V is based on StyleGAN2 [23], 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 γ 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 [44], TATS [12] and DIGAN [53] 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.



(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.

5.1 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 further validation of our observations, we conducted a preliminary user study on Amazon Mechanical Turk. We created 50 pairs of videos for each dataset, containing a random video generated by StyleGAN-V and one generated by our method, and 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 $4{\times}50{\times}10$ responses. Our method was preferred over 80% of the time for every dataset. Please see the supplemental 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

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

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 target curve, indicating that colors in our generated videos change at appropriate rates.

	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 γ	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 slightly underperforms on 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}.

5.3 Fréchet video distance (FVD)

The commonly used Fréchet video distance (FVD) [45] attempts to measure the similarity between real and generated video distributions by computing the Wasserstein-2 distance [47] between sets of features extracted from a pre-trained I3D action classification model [5]. Skorokhodov *et al.* [40] observe 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 [40]). We report all FVD results using the same evaluation protocol, ensuring apples-to-apples comparison. Skorokhodov *et al.* compute FVD using one clip per video in the training dataset. This results in too few samples for datasets with a small number of long videos (such as horseback riding), so we instead sample 2048 random clips from both the dataset and the generator. We measure FVD separately with 128 and 16 frames per clip, denoted by FVD_{128} and FVD_{16}, respectively.

Table 1 (left) reports FVD on all datasets for StyleGAN-V and our model. We outperform StyleGAN-V in terms of FVD on horseback riding and mountain biking datasets that contain more complex changes over time. We underperform StyleGAN-V in terms of FVD on ACID and slightly underperform on SkyTimelapse. This result does not agree with the observed qualitative improvements — also supported by our user study — which suggests that FVD deviates from human perception at assessing video quality in certain cases. 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.

We have found that the FVD of StyleGAN-V is particularly sensitive to the time offset used during sampling, especially on SkyTimelapse. StyleGAN-V takes an absolute timestep t as input, with a default of $t = 0$ that we use in all other experiments. While Skorokhodov *et al.* [40] highlight examples of large time offsets up to an hour, we find that FVD_{128} increases from 151.2 to 263.9 on this particular dataset when such time offsets are sampled randomly. Our model has no notion of absolute timestep, so there is no degredation in quality over long durations.

Even though FVD is useful for high-level assessment, it does not cover all the relevant aspects of generated videos. We found FVD most valuable when comparing similar training runs and ablations. However, we feel that the qualitative evaluation and analysis of color changes in Sections 5.2 and 5.1 are equally important for assessing the capabilities of the different models, especially for long videos with complex motion and new content that enters the frame over time.

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.

	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.

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 capped 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.

6 Conclusions

We have shown that representations over many time scales serve as useful building blocks for modeling both complex motions and 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. The benefits we observe from training on long sequences open up several interesting question, such as how long training sequence length can and should be pushed for video generation, and how to better combat overfitting when training with long videos. Separate low- and super-resolution networks greatly expedites exploration, but has a potential negative impact on the final high-resolution frames — we believe the "swirly" artifacts visible in some of the results are due to this RGB bottleneck. We also see several secondary factors, including appropriate metrics and visualization practices, that require attention as part of future effort. We hope our work encourages further research into video generation that focuses on complex and long-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.

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. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (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] Gereon Fox, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. Stylevideogan: A temporal generative model using a pretrained stylegan. *CoRR*, abs/2107.07224, 2021.

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

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

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

[15] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3897–3906, 2019.

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

[17] 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.

[18] 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.

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

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

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

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[23] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[24] 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.

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

[26] 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.

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

[28] 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 *CVPR*.

[29] 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.

[30] 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.

[31] 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.

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

[33] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[34] 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.

[35] 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.

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

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

[38] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

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

[40] 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.

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

[42] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[43] 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.

[44] 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.

[45] 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.

[46] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Neural Information Processing Systems (NeurIPS)*, 2020.

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

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

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

[50] 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 *ECCV*. Springer, 2020.

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

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

[53] 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 *ICLR*, 2022.

[54] 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 *ECCV*, 2020.

[55] 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 6
   (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)? [No] We will release code and datasets upon publication.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See supplemental.
   (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 supplemental.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We create new datasets as explained in Section 4 but did not include them in the submission do to size constraints.
   (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 supplemental.
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 supplemental.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [Yes] See supplemental.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] See supplemental.