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.

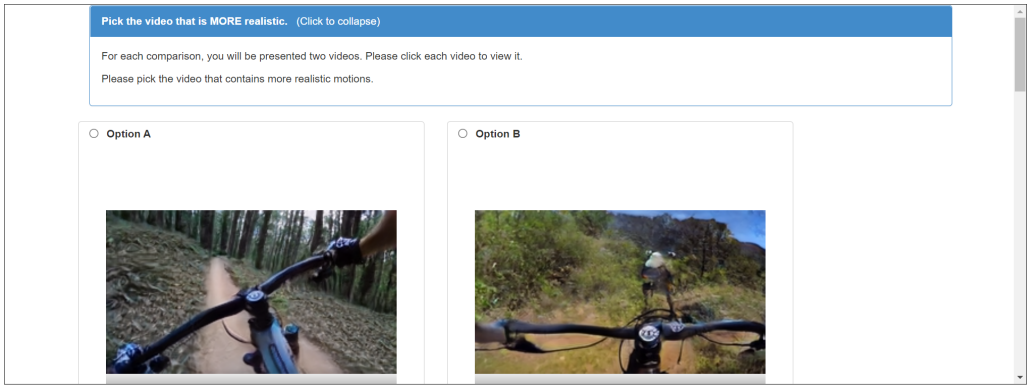


Figure 7: Screenshot of instructions provided to user study participants.

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.

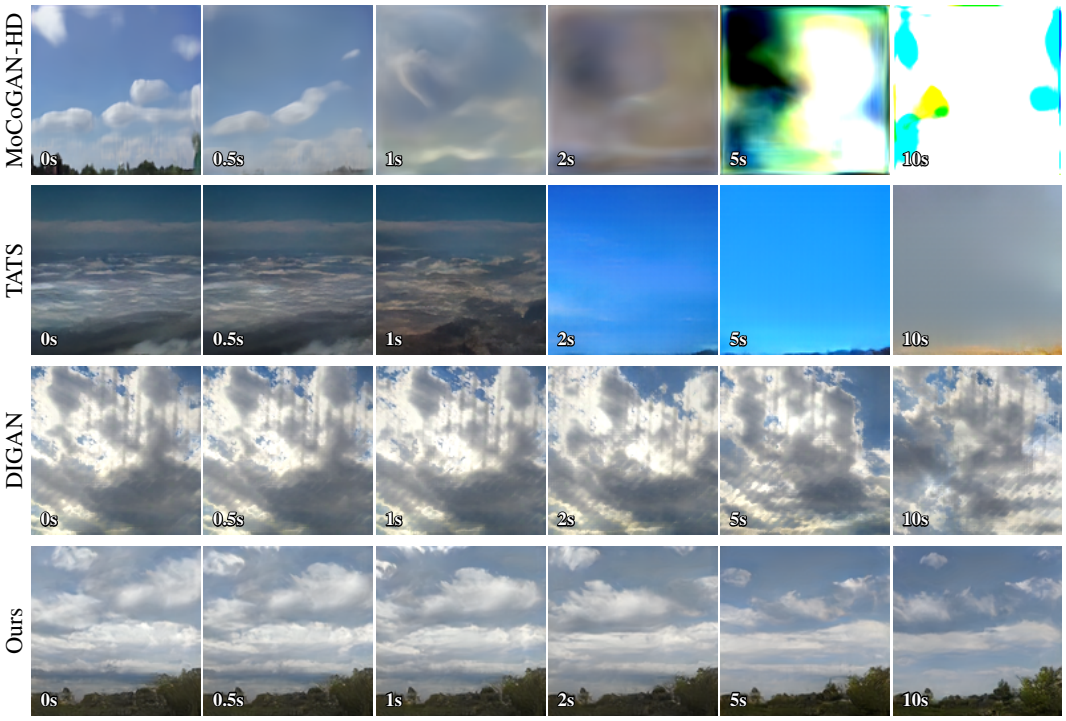
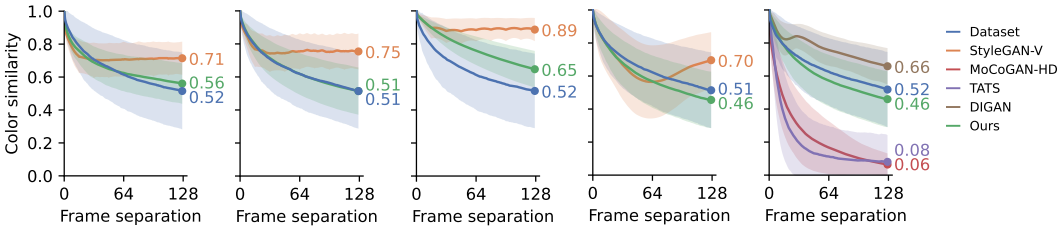
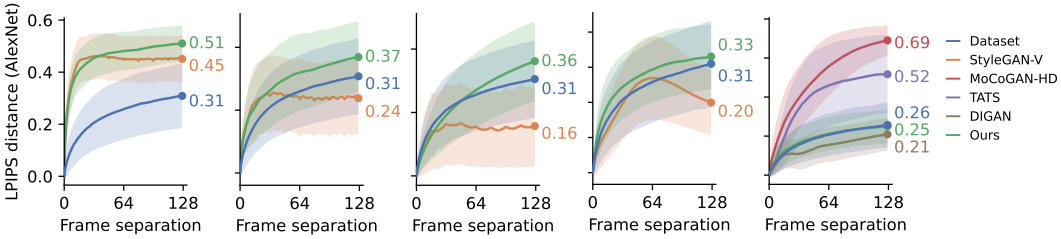


Figure 11: SkyTimelapse [64] (128^2 resolution). Real video omitted. **Top:** MoCoGAN-HD [55] is based on a recurrent network in latent space of a pretrained StyleGAN2 [29] model. It produces a realistic initial frame, but the video quickly explodes over a long duration. **2nd:** TATS [13] employs an 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.



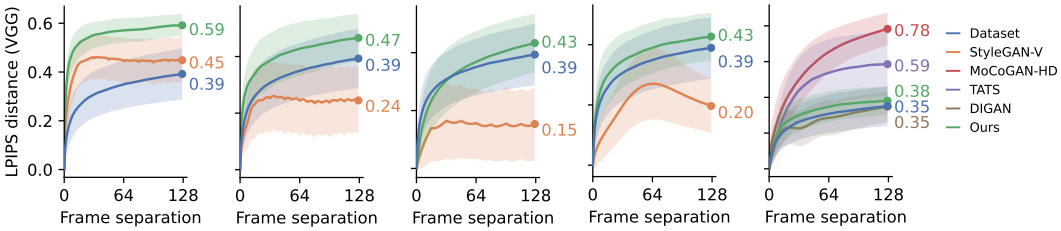
(a) Biking (b) Horseback (c) ACID (d) Sky 256^2 (e) Sky 128^2

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 γ	12.5	17.7	–	–
Ours	18.9	12.2	18.2	26.6

Table 4: Video-balanced Fréchet inception distance (FID_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. FVD_{128} and 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 (FID_V), which we describe below and report in Table 4. StyleGAN-V outperforms our model on three of the four datasets in terms of FID_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.

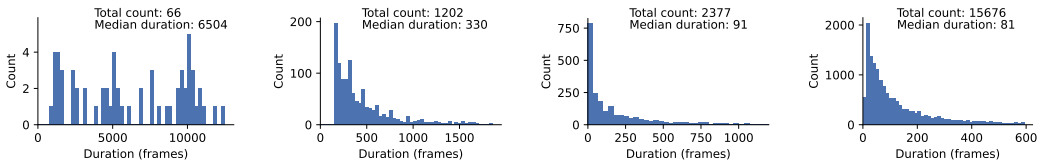
Assessing quality of generated videos is multifaceted, and we believe all of the evaluation we provide — qualitative results, user study, color change over time, FVD, and FID — help expose gaps in the abilities of existing methods and the strengths and weaknesses of our new model.

Video-balanced Fréchet inception distance (FID_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

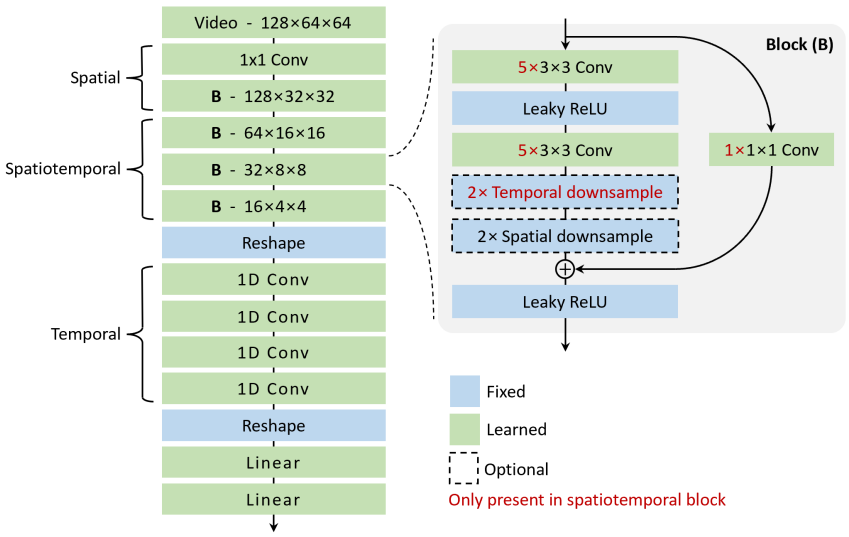


Figure 16: Low-resolution discriminator architecture. **Left:** The input video undergoes a single 1×1 convolutional layer, followed by 4 residual blocks. Features are then reshaped, combining spatial and channel dimensions, followed by 4 temporal 1D convolutional layers. Finally, features are flattened, 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 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., 64×64) 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 FVD_{16} when evaluated using real low resolution conditioning, and use the same super-resolution network for many low-resolution experiments.