ADVERSARIAL VIDEO GENERATION ON COMPLEX DATASETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models of natural images have progressed towards high fidelity samples by the strong leveraging of scale. We attempt to carry this success to the field of video modeling by showing that large Generative Adversarial Networks trained on the complex Kinetics-600 dataset are able to produce video samples of substantially higher complexity and fidelity than previous work. Our proposed model, Dual Video Discriminator GAN (DVD-GAN), scales to longer and higher resolution videos by leveraging a computationally efficient decomposition of its discriminator. We evaluate on the related tasks of video synthesis and video prediction, and achieve new state-of-the-art Fréchet Inception Distance for prediction for Kinetics-600, as well as state-of-the-art Inception Score for synthesis on the UCF-101 dataset, alongside establishing a strong baseline for synthesis on Kinetics-600.

1 INTRODUCTION

Figure 1: Selected frames from videos generated by a DVD-GAN trained on Kinetics-600 at 256×256, 128×128, and 64×64 resolutions (top to bottom).

Modern deep generative models can produce realistic natural images when trained on high-resolution and diverse datasets (Brock et al., 2019; Karras et al., 2018; Kingma & Dhariwal, 2018; Menick & Kalchbrenner, 2019; Razavi et al., 2019). Generation of natural *video* is an obvious further challenge for generative modeling, but one that is plagued by increased data complexity and computational requirements. For this reason, much prior work on video generation has revolved around relatively simple datasets, or tasks where strong temporal conditioning information is available.

We focus on the tasks of video synthesis and video prediction (defined in Section 2.1), and aim to extend the strong results of generative image models to the video domain. Building upon the state-of-the-art BigGAN architecture (Brock et al., 2019), we introduce an efficient spatio-temporal decomposition of the discriminator which allows us to train on Kinetics-600 – a complex dataset of

Figure 2: Generated video samples with interesting behavior. In raster-scan order: **a)** On-screen generated text with further lines appearing.. **b)** Zooming in on an object. **c)** Colored detail from a pen being left on paper. **d)** A generated camera change and return.

natural videos an order of magnitude larger than other commonly used datasets. The resulting model, Dual Video Discriminator GAN (DVD-GAN), is able to generate temporally coherent, high-resolution videos of relatively high fidelity (Figure 1).

Our contributions are as follows:

- We propose DVD-GAN – a scalable generative model of natural video which produces high-quality samples at resolutions up to 256×256 and lengths up to 48 frames.
- We achieve state of the art for video synthesis on UCF-101 and prediction on Kinetics-600.
- We establish class-conditional video synthesis on Kinetics-600 as a new benchmark for generative video modeling, and report DVD-GAN results as a strong baseline.

2 Background

2.1 Video Synthesis and Prediction

The exact formulation of the video generation task can differ in the type of conditioning signal provided. At one extreme lies unconditional video synthesis where the task is to generate any video following the training distribution. Another extreme is occupied by strongly-conditioned models, including generation conditioned on another video for content transfer (Bansal et al., 2018; Zhou et al., 2019), per-frame segmentation masks (Wang et al., 2018), or pose information (Walker et al., 2017; Villegas et al., 2017; Yang et al., 2018). In the middle ground there are tasks which are more structured than unconditional generation, and yet are more challenging from a modeling perspective than strongly-conditional generation (which gets a lot of information about the generated video through its input). The objective of *class-conditional video synthesis* is to generate a video of a given category (e.g., "riding a bike") while *future video prediction* is concerned with generation of continuing video given initial frames. These problems differ in several aspects, but share a common requirement of needing to generate realistic temporal dynamics, and in this work we focus on these two problems.

2.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are a class of generative models defined by a minimax game between a *Discriminator* $\mathcal{D}$ and a *Generator* $\mathcal{G}$. The original objective was proposed by Goodfellow et al. (2014), and many improvements have since been suggested, mostly targeting improved training stability (Arjovsky et al., 2017; Zhang et al., 2018; Brock et al., 2019; Gulrajani et al., 2017; Miyato et al., 2018). We use the hinge formulation of the objective (Lim & Ye, 2017; Brock et al., 2019) which is optimized by gradient descent (ρ is the elementwise ReLU function):

$$\mathcal{D}\colon \min_{\mathcal{D}} \mathop{\mathbb{E}}_{x\sim data(x)} \Big[\rho(1-\mathcal{D}(x))\Big] + \mathop{\mathbb{E}}_{z\sim p(z)} \Big[\rho(1+\mathcal{D}(\mathcal{G}(z)))\Big], \quad \mathcal{G}\colon \max_{\mathcal{G}} \mathop{\mathbb{E}}_{z\sim p(z)} \Big[\mathcal{D}(\mathcal{G}(z))\Big].$$

GANs have well-known limitations including a tendency towards limited diversity in generated samples (a phenomenon known as mode collapse) and the difficulty of quantitative evaluation due

to the lack of an explicit likelihood measure over the data. Despite these downsides, GANs have produced some of the highest fidelity samples across many visual domains (Karras et al., 2018; Brock et al., 2019).

2.3 KINETICS-600

Kinetics is a large dataset of 10-second high-resolution YouTube clips (Kay et al., 2017; DeepMind, 2018) originally created for the task of human action recognition. We use the second iteration of the dataset, Kinetics-600 (Carreira et al., 2018), which consists of 600 classes with at least 600 videos per class for a total of around 500,000 videos.[1] Kinetics videos are diverse and unconstrained, which allows us to train large models without being concerned with the overfitting that occurs on small datasets with fixed objects interacting in specified ways (Ebert et al., 2017; Blank et al., 2005). Among prior work, the closest dataset (in terms of subject and complexity) which is consistently used is UCF-101 (Soomro et al., 2012). We focus on Kinetics-600 because of its larger size (almost 50x more videos than UCF-101) and its increased diversity (600 instead of 101 classes – not to mention increased intra-class diversity). Nevertheless for comparison with prior art we train on UCF-101 and achieve a state-of-the-art Inception Score there. Kinetics contains many artifacts expected from YouTube, including cuts (as in Figure 2d), title screens and visual effects. Except when specifically described, we choose frames with stride 2 (meaning we skip every other frame). This allows us to generate videos with more complexity without incurring higher computational cost.

To the best of our knowledge we are the first to consider generative modelling of the entirety of the Kinetics video dataset[2], although a small subset of Kinetics consisting of 4,000 selected and stabilized videos (via a SIFT + RANSAC procedure) has been used in at least two prior papers (Li et al., 2018; Balaji et al., 2018). Due to the heavy pre-processing and stabilization present, as well as the sizable reduction in dataset size (two orders of magnitude) we do not consider these datasets comparable to the full Kinetics-600 dataset.

2.4 EVALUATION METRICS

Designing metrics for measuring the quality of generative models (GANs in particular) is an active area of research (Sajjadi et al., 2018; Barratt & Sharma, 2018). In this work we report the two most commonly used metrics, Inception Score (IS) (Salimans et al. (2016)) and Fréchet Inception Distance (FID) (Heusel et al., 2017). The standard instantiation of these metrics is intended for generative image models, and uses an Inception model (Szegedy et al., 2016) for image classification or feature extraction. For videos, we use the publicly available Inflated 3D Convnet (I3D) network trained on Kinetics-600 (Carreira & Zisserman, 2017). Our Fréchet Inception Distance is therefore very similar to the Fréchet Video Distance (FVD) (Unterthiner et al., 2018), although our implementation is different and more aligned with the original FID metric.[3]

3 DUAL VIDEO DISCRIMINATOR GAN

Our primary contribution is Dual Video Discriminator GAN (DVD-GAN), a generative video model of complex human actions built upon the state-of-the-art BigGAN architecture (Brock et al., 2019) while introducing scalable, video-specific generator and discriminator architectures. An overview of the DVD-GAN architecture is given in Figure 3 and a detailed description is in Appendix A.2. Unlike some of the prior work, our generator contains no explicit priors for foreground, background or motion (optical flow); instead, we rely on a high-capacity neural network to learn this in a data-driven manner. While DVD-GAN contains sequential components (RNNs), it is not autoregressive in time or in space. In other words, the pixels of each frame do not directly depend on other pixels in the video, as would be the case for auto-regressive models or models generating one frame at a time.

Generating long and high resolution videos is a heavy computational challenge: individual samples from Kinetics-600 (just 10 seconds long) contain upwards of 16 million pixels which need to be

[1] Kinetics is occasionally pruned and so we cannot give an exact size.

[2] In parallel with the concurrent work of Weissenborn et al. (2019).

[3] We use 'avgpool' features (rather than logits) by default, our I3D model is trained on Kinetics-600 (rather than Kinetics-400), and we pre-calculate ground-truth statistics on the entire training set.

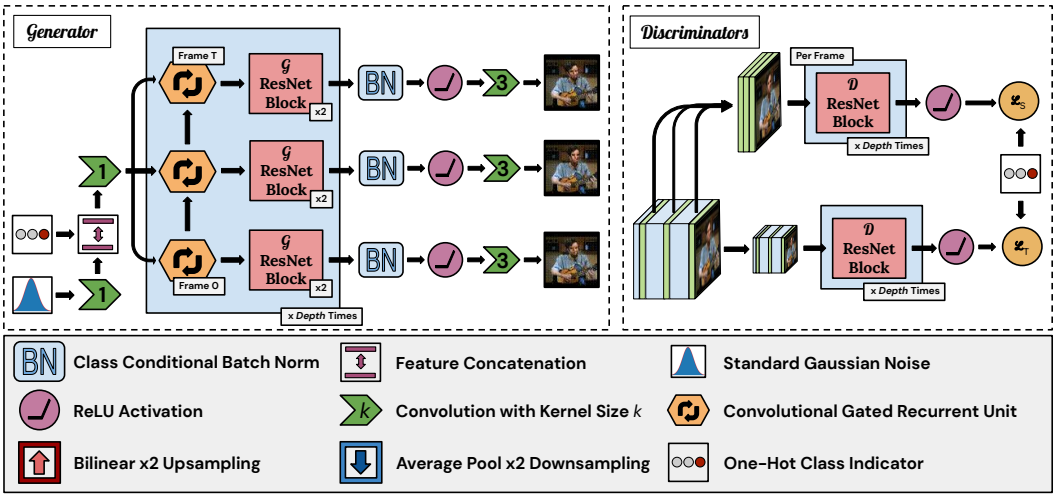

Figure 3: Simplified architecture diagram of $\mathcal{G}$ (left) and $\mathcal{D}_S$/$\mathcal{D}_T$ (right). More details in A.2.

generated in a consistent fashion. This is a particular challenge to the discriminator. For example, a generated video might contain an object which leaves the field of view and incorrectly returns with a different color. Here, the ability to determine this video is generated is only possible by comparing two different spatial locations across two (potentially distant) frames. Given a video with length T, height H, and width W, discriminators that process the entire video would have to process all $H \times W \times T$ pixels – limiting the size of the model and the size of the videos being generated.

3.1 Dual Discriminators

DVD-GAN tackles this scale problem by using two discriminators: a *Spatial Discriminator* $\mathcal{D}_S$ and a *Temporal Discriminator* $\mathcal{D}_T$. $\mathcal{D}_S$ critiques single frame content and structure by randomly sampling k full-resolution frames and judging them individually. We use $k = 8$ and discuss this choice in Section 4.3. $\mathcal{D}_S$'s final score is the sum of the per-frame scores. The temporal discriminator $\mathcal{D}_T$ must provide $\mathcal{G}$ with the learning signal to generate movement (something not evaluated by $\mathcal{D}_S$). To make the model scalable, we apply a spatial downsampling function $\phi(\cdot)$ to the whole video and feed its output to $\mathcal{D}_T$. We choose ϕ to be 2×2 average pooling, and discuss alternatives in Section 4.3. This results in an architecture where the discriminators do not process the entire video's worth of pixels, since $\mathcal{D}_S$ processes only $k \times H \times W$ pixels and $\mathcal{D}_T$ only $T \times \frac{H}{2} \times \frac{W}{2}$. For a 48 frame video at 128×128 resolution, this reduces the number of pixels to process per video from 786432 to 327680: a 58% reduction. Despite this decomposition, the discriminator objective is still able to penalize almost all inconsistencies which would be penalized by a discriminator judging the entire video. $\mathcal{D}_T$ judges any temporal discrepancies across the entire length of the video, and $\mathcal{D}_S$ can judge any high resolution details. The only detail the DVD-GAN discriminator objective is unable to reflect is the temporal evolution of pixels within a 2×2 window. We have however not noticed this affecting the generated samples in practice. DVD-GAN's $\mathcal{D}_S$ is similar to the per-frame discriminator $\mathcal{D}_I$ in MoCoGAN (Tulyakov et al., 2018). However MoCoGAN's analog of $\mathcal{D}_T$ looks at full resolution videos, whereas $\mathcal{D}_S$ is the only source of learning signal for high-resolution details in DVD-GAN. For this reason, $\mathcal{D}_S$ is essential when ϕ is not the identity, unlike in MoCoGAN where the additional per-frame discriminator is less crucial.

3.2 Related Work

Generative video modeling is a widely explored problem which includes work on VAEs (Babaeizadeh et al., 2018; Denton & Fergus, 2018; Lee et al., 2018; Hsieh et al., 2018), auto-regressive models (Ranzato et al., 2014; Srivastava et al., 2015; Kalchbrenner et al., 2017; Weissenborn et al., 2019), normalizing flows (Kumar et al., 2019), and GANs (Mathieu et al., 2015; Vondrick et al., 2016; Saito et al., 2017; Saito & Saito, 2018). Much prior work considers decompositions which model the texture and spatial consistency of objects separately from their temporal dynamics. One approach is

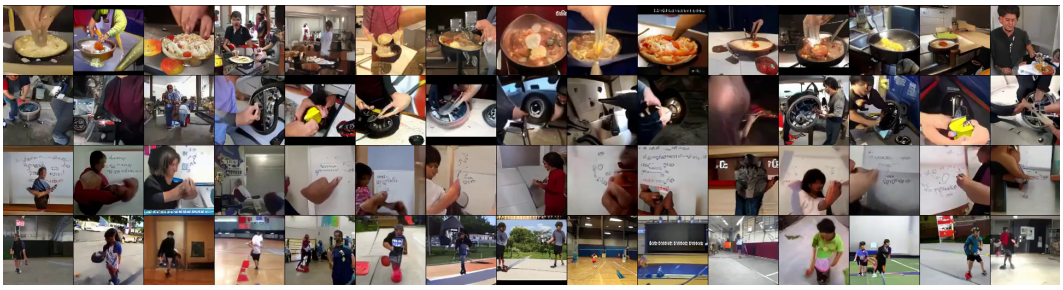

Figure 4: Each row is the first frame of 15 videos from a random class, all from the same checkpoint. The classes are: **cooking scallops**, **changing wheel (not on bike)**, **calculating**, **dribbling basketball**.

Figure 5: All 48 frames (in raster-scan order) from a 64×64 sample from watermelon cutting class.

to split $\mathcal{G}$ into foreground and background models (Vondrick et al., 2016; Spampinato et al., 2018), while another considers explicit or implicit optical flow in either $\mathcal{G}$ or $\mathcal{D}$ (Saito et al., 2017; Ohnishi et al., 2018). Similar to DVD-GAN, MoCoGAN (Tulyakov et al., 2018) discriminates individual frames in addition to a discriminator which operates on fixed-length K-frame slices of the whole video (where $K < T$). Though this potentially reduces the number of pixels to discriminate to $(H \times W) + (K \times H \times W)$, Tulyakov et al. (2018) describes discriminating sliding windows, which increases the total number of pixels. Other models follow this approach by discriminating groups of frames (Xie et al., 2018; Sun et al., 2018; Balaji et al., 2018).

TGANv2 (Saito & Saito, 2018) proposes "adaptive batch reduction" for efficient training, an operation which randomly samples subsets of videos within a batch and temporal subwindows within each video. This operation is applied throughout TGANv2's $\mathcal{G}$, with heads projecting intermediate feature maps directly to pixel space before applying batch reduction, and corresponding discriminators evaluating these lower resolution intermediate outputs. An effect of this choice is that TGANv2 discriminators only evaluate full-length videos at very low resolution. We show in Figure 6 that a similar reduction in DVD-GAN's resolution when judging full videos leads to a loss in performance. We expect further reduction (towards the resolution at which TGANv2 evaluates the entire length of video) to lead to further degradation of DVD-GAN's quality. Furthermore, this method is not easily adapted towards models with large batch sizes divided across a number of accelerators, with only a small batch size per replica.

4 Experiments and Analysis

A detailed description of our training setup is in Appendix A.3. Each DVD-GAN was trained on TPU pods (Google, 2018) using between 32 and 512 replicas with an Adam (Kingma & Ba, 2014) optimizer. Video Synthesis models are trained for around 300,000 learning steps, whilst Video Prediction models are trained for up to 1,000,000 steps. Most models took between 12 and 96 hours to train.

4.1 CLASS-CONDITIONAL VIDEO SYNTHESIS

Our primary results concern the problem of *Class-Conditional Video Synthesis*. We provide our results for the UCF-101 and Kinetics-600 datasets. With Kinetics-600 emerging as a new benchmark for generative video modelling, our results establish a strong baseline for future work.

4.1.1 KINETICS-600 RESULTS

Table 1: FID/IS for DVD-GAN on Kinetics-600 Video Synthesis. We present the scores of the model taken at the point in training when the best FID was attained. The "No Truncation" columns contain the scores obtained without the truncation trick. The "With Truncation" columns contain the scores obtained at the truncation level which results in the best Inception Score.

(# Frames / Resolution)	No Truncation		With Truncation	
	FID (↓)	IS (↑)	FID (↓)	IS (↑)
$12/64 \times 64$	0.85	53.81	7.13	187.23
$12/128 \times 128$	1.16	77.45	13.04	246.18
$12/256 \times 256$	2.05	62.78	10.17	162.44
$48/64 \times 64$	13.75	104.09	47.86	264.12
$48/128 \times 128$	28.44	81.41	45.79	188.32

In Table 1 we show the main result of this paper: benchmarks for Video Synthesis on Kinetics-600. We consider a range of resolutions and video lengths, and measure Inception Score and Fréchet Inception Distance (FID) for each (as described in Section 2.4). We further measure each model along a truncation curve, which we carry out by calculating FID and IS statistics while varying the standard deviation of the latent vectors between 0 and 1. There is no prior work with which to quantitatively compare these results (for comparative experiments see Section 4.1.2 and Section 4.2.1), but we believe these samples to show a level of fidelity not yet achieved in datasets as complex as Kinetics-600 (see samples from each row in Appendix D.1). Because all videos are resized for the I3D network (to 224×224), it is meaningful to compare metrics across equal length videos at different resolutions. Neither IS nor FID are comparable across videos of different lengths, and should be treated as separate metrics.

Generating longer and larger videos is a more challenging modeling problem, which is conveyed by the metrics (in particular, comparing 12-frame videos across 64×64, 128×128 and 256×256 resolutions). Nevertheless, DVD-GAN is able to generate plausible videos at all resolutions and with length spanning up to 4 seconds (48 frames). As can be seen in Appendix D.1, smaller videos display high quality textures, object composition and movement. At higher resolutions, generating coherent objects becomes more difficult (movement consists of a much larger number of pixels), but high-level details of the generated scenes are still extremely coherent, and textures (even complicated ones like a forest backdrop in Figure 1a) are generated well. It is further worth noting that the 48-frame models do not see more high resolution frames than the 12-frame model (due to the fixed choice of $k = 8$ described in Section 3.1), yet nevertheless learn to generate high resolution images.

4.1.2 VIDEO SYNTHESIS ON UCF-101

We further verify our results by testing the same model on UCF-101 (Soomro et al., 2012), a smaller dataset of 13,320 videos of human actions across 101 classes that has previously been used for video synthesis and prediction (Saito et al., 2017; Saito & Saito, 2018; Tulyakov et al., 2018). Our model produces samples with an IS of 32.97, significantly outperforming the state of the art (see Table 2 for quantitative comparison and Appendix B.1 for more details).

4.2 FUTURE VIDEO PREDICTION

Future Video Prediction is the problem of generating a sequence of frames which directly follow from one (or a number) of initial conditioning frames. Both this and video synthesis require $\mathcal{G}$ to

Table 2: IS on UCF-101 with C3D.

Method	IS (↑)
VGAN (Vondrick et al., 2016)	8.31 ± .09
TGAN (Saito et al., 2017)	11.85 ± .07
MoCoGAN (Tulyakov et al., 2018)	12.42 ± .03
ProgressiveVGAN (Acharya et al., 2018)	14.56 ± .05
TGANv2 (Saito & Saito, 2018)	24.34 ± .35
DVD-GAN (ours)	**32.97** ± 1.7

Table 3: FVD on BAIR.

Method	FVD (↓)
SVP-FP	315.5
CDNA	296.5
SV2P	262.5
SAVP	116.4
DVD-GAN-FP (ours)	109.8
Video Transformer	**94** ± **2**

Table 4: DVD-GAN-FP's FVD scores on Video Prediction for 16 frames of Kinetics-600 without frame skipping. The final row represents a *Video Synthesis* model generating 16 frames.

Method	Training Set FVD (↓)	Test Set FVD (↓)
Video Transformer (Weissenborn et al., 2019)	-	170 ± 5
DVD-GAN-*FP*	**68.66** ± **0.78**	**69.15** ± **1.16**
DVD-GAN	32.3 ± 0.82	31.1 ± 0.56

learn to produce realistic scenes and temporal dynamics, however video prediction further requires $\mathcal{G}$ to analyze the conditioning frames and discover elements in the scene which will evolve over time. In this section, we use the Fréchet Video Distance exactly as Unterthiner et al. (2018): using the logits of an I3D network trained on Kinetics-400 as features. This allows for direct comparison to prior work. Our model, DVD-GAN-FP (Frame Prediction), is slightly modified to facilitate the changed problem, and details of these changes are given in Appendix A.4.

4.2.1 Frame-Conditional Kinetics

For direct comparison with concurrent work on autoregressive video models (Weissenborn et al., 2019) we consider the generation of 11 frames of Kinetics-600 at 64×64 resolution conditioned on 5 frames, where the videos for training are not taken with any frame skipping. We show results for all these cases in Table 4. Our frame-conditional model **DVD-GAN-*FP*** outperforms the prior work on frame-conditional prediction for Kinetics. The final row labeled DVD-GAN corresponds to 16-frame class-conditional Video Synthesis samples, generated without frame conditioning and without frame skipping. The FVD of this video synthesis model is notably better.

On the one hand, we hypothesize that the synthesis model has an easier generative task: it can choose to generate (relatively) simple samples for each class, rather than be forced to continue frames taken from videos which are class outliers, or contain more complicated details. On the other hand, a certain portion of the FID/FVD metric undoubtedly comes from the distribution of objects and backgrounds present in the dataset, and so it seems that the prediction model should have a handicap in the metric by being given the ground truth distribution of backgrounds and objects with which to continue videos. The synthesis model's improved performance on this task seems to indicate that the advantage of being able to select videos to generate is greater than the advantage of having a ground truth distribution of starting frames.

4.2.2 BAIR Robot Pushing

We further test future video prediction on the single-class BAIR Robot Pushing Dataset (Ebert et al., 2017), a dataset of stationary videos of a robot arm moving around a set of changing objects. In order for direct comparison with previous results reported in Unterthiner et al. (2018), we consider generating 15 frames conditioned on a single starting frame. Like on prediction with Kinetics, we report FVD exactly as in Unterthiner et al. (2018), with ground truth statistics and conditioning frames taken from the 256-video dev set. Results are reported in Table 3. Scores are taken from Unterthiner et al. (2018). DVD-GAN-FP outperforms all prior adversarial models trained on this dataset, but performs slightly worse than Video Transformer, a concurrently developed autoregressive model Weissenborn et al. (2019).

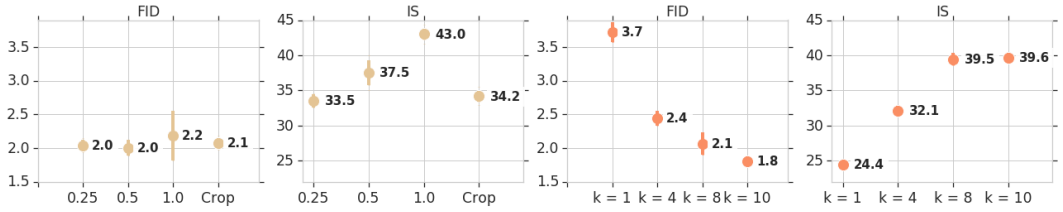

Figure 6: The effect of ϕ in $\mathcal{D}_T$ (left two) and k in $\mathcal{D}_S$ (right two). FID is similar for any choice of ϕ, while IS declines as downsampling increases. Increasing k improves both with diminishing returns.

4.3 Dual Discriminator Input

We analyze several choices for k (the number of frames per sample in the input to $\mathcal{D}_S$) and ϕ (the downsampling function for $\mathcal{D}_T$). We expect setting ϕ to the identity or $k = T$ to result in the best model, but we are interested in the maximally compressive k and ϕ that reduce discriminator input size (and the amount of computation), while still producing a high quality generator. For ϕ, we consider: 2×2 and 4×4 average pooling, the identity (no downsampling), as well as a ϕ which takes a random half-sized crop of the input video (as in Saito & Saito (2018)). Results can be seen in Figure 6. For each ablation, we train three identical DVD-GANs with different random initializations on 12-frame clips of Kinetics-600 at 64×64 resolution for 100,000 steps. We report mean and standard deviation (via the error bars) across each group for the whole training period. For k, we consider 1, 2, 8 and 10 frames. We see diminishing effect as k increases, so settle on $k = 8$. We note the substantially reduced IS of 4×4 downsampling as opposed to 2×2, and further note that taking half-sized crops (which results in the same number of pixels input to $\mathcal{D}_T$ as 2×2 pooling) is also notably worse.

5 Conclusion

We approached the challenging problem of modeling natural video by introducing a GAN capable of capturing the complexity of a large video dataset. We showed that on UCF-101 and frame-conditional Kinetics-600 it quantitatively achieves the new state of the art, alongside qualitatively producing sample videos with high complexity and diversity. We further wish to emphasize the benefit of training generative models on large and complex video datasets, such as Kinetics-600, and envisage the strong baselines we established on this dataset with DVD-GAN will be used as a reference point by the generative modeling community moving forward. While much remains to be done before realistic videos can be consistently generated in an unconstrained setting, we believe DVD-GAN is a step in that direction.

References

Dinesh Acharya, Zhiwu Huang, Danda Pani Paudel, and Luc Van Gool. Towards high resolution video generation with progressive growing of sliced Wasserstein GANs. *arXiv:1810.02419*, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv:1701.07875*, 2017.

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

Yogesh Balaji, Martin Renqiang Min, Bing Bai, Rama Chellappa, and Hans Peter Graf. Tfgan: Improving conditioning for text-to-video synthesis. 2018.

Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *arXiv:1511.06432*, 2015.

Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-GAN: Unsupervised video retargeting. In *ECCV*, September 2018.

Shane Barratt and Rishi Sharma. A note on the Inception Score. *arXiv:1801.01973*, 2018.

Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *ICCV*, 2005.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *ICLR*, 2019.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *ICCV*, 2017.

Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about Kinetics-600. *arXiv:1808.01340*, 2018.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *NeurIPS*, 2017.

DeepMind, 2018. Kinetics. `https://deepmind.com/research/open-source/open-source-datasets/kinetics/`. Accessed: 2019.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *ICML*, 2018.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *ICLR*, 2017.

Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *arXiv:1710.05268*, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

Google, 2018. Cloud tpu. `https://cloud.google.com/tpu/`. Accessed: 2019.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *NeurIPS*, 2017.

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 *NeurIPS*, 2017.

Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *NeurIPS*. 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*, 2015.

Nal Kalchbrenner, Aäron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *ICML*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv:1812.04948*, 2018.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics human action video dataset. *arXiv:1705.06950*, 2017.

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

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.

Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. VideoFlow: A flow-based generative model for video. *arXiv:1903.01434*, 2019.

Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv:1804.01523*, 2018.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016.

Yitong Li, Martin Renqiang Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Jae Hyun Lim and Jong Chul Ye. Geometric GAN. *arXiv:1705.02894*, 2017.

Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv:1511.05440*, 2015.

Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In *ICLR*, 2019.

Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. *arXiv:1802.05637*, 2018.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv:1802.05957*, 2018.

Katsunori Ohnishi, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Hierarchical video generation from orthogonal information: Optical flow and texture. In *AAAI*, 2018.

Marc'Aurelio Ranzato, Arthur Szlam, Joan Bruna, Michaël Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv:1412.6604*, 2014.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. *arXiv preprint arXiv:1906.00446*, 2019.

Masaki Saito and Shunta Saito. TGANv2: Efficient training of large models for video generation with multiple subsampling layers. *arXiv:1811.09245*, 2018.

Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *ICCV*, 2017.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *NeurIPS*, 2018.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NeurIPS*, 2016.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:1312.6120*, 2013.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012.

Concetto Spampinato, Sergio Palazzo, P D'Oro, Francesca Murabito, Daniela Giordano, and M Shah. VOS-GAN: Adversarial learning of visual-temporal dynamics for unsupervised dense prediction in videos. *arXiv:1803.09092*, 2018.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015.

Ximeng Sun, Huijuan Xu, and Kate Saenko. A two-stream variational adversarial network for video generation. *arXiv:1812.01037*, 2018.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *ICML*, 2011.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *CVPR*, 2016.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Workshop on Systems for ML and Open Source Software at NeurIPS*, 2015.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 2018.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv:1607.08022*, 2016.

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

Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *ICML*, 2017.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016.

Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *ICCV*, 2017.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv:1808.06601*, 2018.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *arXiv:1906.02634*, 2019.

Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow. *TOG*, 2018.

Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *ECCV*, 2018.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv:1805.08318*, 2018.

Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L Berg. Dance dance generation: Motion transfer for internet videos. *arXiv:1904.00129*, 2019.

A Experiment Methodology

A.1 Dataset Processing

For all datasets we randomly shuffle the training set for each model replica independently. Experiments on the BAIR Robot Pushing dataset are conducted in the native resolution of 64×64, where for UCF-101 we operate at a (downsampled) 128×128 resolution. This is done by a bilinear resize such that the video's smallest dimension is mapped to 128 pixels (maintaining aspect ratio). From this we take a random 128-pixel crop along the other dimension. We use the same procedure to construct datasets of different resolutions for Kinetics-600. All three datasets contain videos with more frames than we generate, so we take a random sequence of consecutive frames from the resized output.

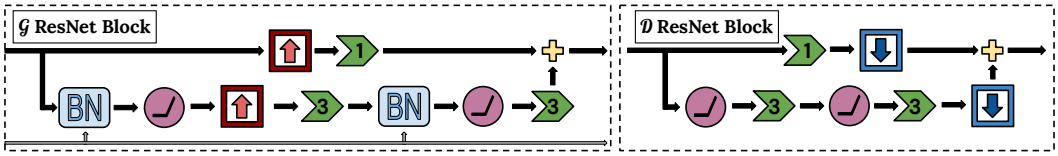

Figure 7: The residual blocks for $\mathcal{G}$ and $\mathcal{D}_S$/$\mathcal{D}_T$. See Figure 3 for the icons and A.2 for more detail.

A.2 Architecture Description

Our model adopts many architectural choices from Brock et al. (2019) including our nomenclature for describing network width, which is determined by the product of a channel multiplier ch with a constant for each layer in the network. The layer-wise constants for $\mathcal{G}$ are $[8, 8, 8, 4, 2]$ for 64×64 videos and $[8, 8, 8, 4, 2, 1]$ for 128×128. The width of the i-th layer is given by the product of ch and the i-th constant and all layers prior to the residual network in $\mathcal{G}$ use the initial layer's multiplier and we refer to the product of that and ch as ch_0. ch in DVD-GAN is 128 for videos with 64×64 resolution and 96 otherwise. The corresponding ch lists for both $\mathcal{D}_T$ and $\mathcal{D}_S$ are $[2, 4, 8, 16, 16]$ for 64×64 resolution and $[1, 2, 4, 8, 16, 16]$ for 128×128.

The input to $\mathcal{G}$ consists of a Gaussian latent noise $z \sim \mathcal{N}(0, I)$ and a learned linear embedding $e(y)$ of the desired class y. Both inputs are 120-dimensional vectors. $\mathcal{G}$ starts by computing an affine transformation of $[z; e(y)]$ to a $[4, 4, ch_0]$-shaped tensor (in Figure 3 this is represented as a 1×1 convolution). $[z; e(y)]$ is used as the input to all class-conditional Batch Normalization layers throughout $\mathcal{G}$ (the gray line in Figure 7).

This is then treated as the input (at each frame we would like to generate) to a Convolutional Gated Recurrent Unit (Ballas et al., 2015; Sutskever et al., 2011) whose update rule for input x_t and previous output h_{t-1} is given by the following:

$$
\begin{aligned}
r &= \sigma(W_r \star_3 [h_{t-1}; x_t] + b_r) \\
u &= \sigma(W_u \star_3 [h_{t-1}; x_t] + b_u) \\
c &= \rho(W_c \star_3 [x_t; r \odot h_{t-1}] + b_c) \\
h_t &= u \odot h_{t-1} + (1 - u) \odot c
\end{aligned}
$$

In these equations σ and ρ are the elementwise sigmoid and ReLU functions respectively, the $\star_n$ operator represents a convolution with a kernel of size $n \times n$, and the $\odot$ operator is an elementwise multiplication. Brackets are used to represent a feature concatenation. This RNN is unrolled once per frame. The output of this RNN is processed by two residual blocks (whose architecture is given by Figure 7). The time dimension is combined with the batch dimension here, so each frame proceeds through the blocks independently. The output of these blocks has width and height dimensions which are doubled (we skip upsampling in the first block). This is repeated a number of times, with the output of one RNN + residual group fed as the input to the next group, until the output tensors have the desired spatial dimensions. We do not reduce over the time dimension when calculating Batch Normalization statistics. This prevents the network from utilizing the Batch Normalization layers to pass information between timesteps.

The spatial discriminator $\mathcal{D}_S$ functions almost identically to BigGAN's discriminator, though an overview of the residual blocks is given in Figure 7 for completeness. A score is calculated for each of the uniformly sampled k frames (we default to $k = 8$) and the $\mathcal{D}_S$ output is the sum over per-frame scores. The temporal discriminator $\mathcal{D}_T$ has a similar architecture, but pre-processes the real or generated video with a 2×2 average-pooling downsampling function ϕ. Furthermore, the first two residual blocks of $\mathcal{D}_T$ are 3-D, where every convolution is replaced with a 3-D convolution with a kernel size of $3 \times 3 \times 3$. The rest of the architecture follows BigGAN (Brock et al., 2019).

A.3 Training Details

Sampling from DVD-GAN is very efficient, as the core of the generator architecture is a feed-forward convolutional network: two 64×64 48-frame videos can be sampled in less than 150ms on a single

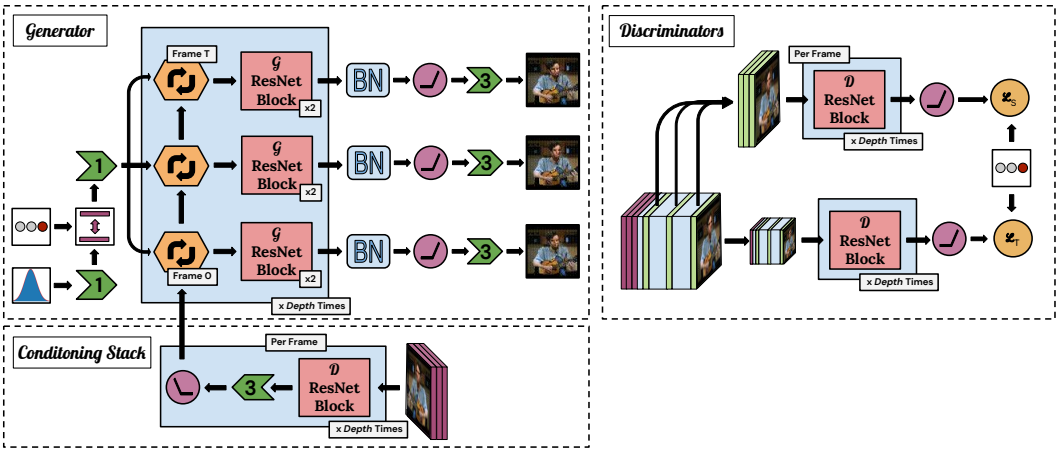

Figure 8: An architecture diagram describing the changes for the frame conditional model.

TPU core. The dual discriminator $\mathcal{D}$ is updated twice for every update of $\mathcal{G}$ (Heusel et al., 2017) and we use Spectral Normalization (Zhang et al., 2018) for all weight layers (approximated by the first singular value) and orthogonal initialization of weights (Saxe et al., 2013). Sampling is carried out using the exponential moving average of $\mathcal{G}$'s weights, which is accumulated with decay $\gamma = 0.9999$ starting after 20,000 training steps. The model is optimized using Adam (Kingma & Ba, 2014) with batch size 512 and a learning rate of $1 \cdot 10^{-4}$ and $5 \cdot 10^{-4}$ for $\mathcal{G}$ and $\mathcal{D}$ respectively. Class conditioning in $\mathcal{D}$ (Miyato & Koyama, 2018) is projection-based whereas $\mathcal{G}$ relies on class-conditional Batch Normalization (Ioffe & Szegedy, 2015; De Vries et al., 2017; Dumoulin et al., 2017): equivalent to standard Batch Normalization without a learned scale and offset, followed by an elementwise affine transformation where each parameter is a function of the noise vector and class conditioning.

A.4 Architecture Extension to Video Prediction

In order to provide results on future video prediction problems we describe a simple modification to DVD-GAN to facilitate the added conditioning. A diagram of the extended model is in Figure 8.

Given C conditioning frames, our modified DVD-GAN-*FP* passes each frame separately through a deep residual network identical to $\mathcal{D}_S$. The (near) symmetric design of $\mathcal{G}$ and $\mathcal{D}_S$'s residual blocks mean that each output from a $\mathcal{D}$-style residual block has a corresponding intermediate tensor in $\mathcal{G}$ of the same spatial resolution. After each block the resulting features for each conditioning frame are stacked in the channel dimension and passed through a 3×3 convolution and ReLU activation. The resulting tensor is used as the initial state for the Convolutional GRU in the corresponding block in $\mathcal{G}$. Note that the frame conditioning stack reduces spatial resolution while $\mathcal{G}$ increases resolution. Therefore the smallest features of the conditioning frames (which have been through the most layers) are input earliest in $\mathcal{G}$ and the larger features (which have been through less processing) are input to $\mathcal{G}$ towards the end. $\mathcal{D}_T$ operates on the concatenation of the conditioning frames and the output of $\mathcal{G}$, meaning that it does not receive any extra information detailing that the first C frames are special. However to reduce wasted computation we do not sample the first C frames for $\mathcal{D}_S$ on real or generated data. This technically means that D_S will never see the first few frames from real videos at full resolution, but this was not an issue in our experiments. Finally, our video prediction variant does not condition on any class information, allowing us to directly compare with prior art. This is achieved by settling the class id of all samples to 0.

B Further Experiments

B.1 UCF-101

UCF-101 (Soomro et al., 2012) is a dataset of 13,320 videos of human actions across 101 classes that has previously been used for video synthesis and prediction (Saito et al., 2017; Saito & Saito,

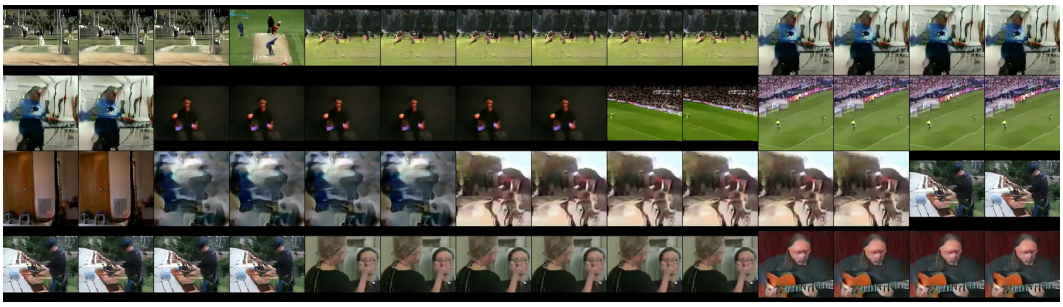

Figure 9: The first frames of interpolations between UCF-101 samples. Each row is a separate interpolation. Contrast with samples in Appendix D.2.

2018; Tulyakov et al., 2018). We report Inception Score (IS) calculated with a C3D network (Tran et al., 2015) for quantitative comparison with prior work.[4] Our model produces samples with an IS of 32.97, significantly outperforming the state of the art (see Table 2). The DVD-GAN architecture on UCF-101 is identical to the model used for Kinetics, and is trained on 16-frame 128×128 clips from UCF-101.

However, it is worth mentioning that our improved score is, at least partially, due to memorization of the training data. In Figure 9 we show interpolation samples from our best UCF-101 model. Like interpolations in Appendix D.2, we sample 2 latents (left and rightmost columns) and show samples from the linear interpolation in latent space along each row. Here we show 4 such interpolations (the first frame from each video). Unlike Kinetics-600 interpolations, which smoothly transition from one sample to the other, we see abrupt jumps in the latent space between highly distinct samples, and little intra-video diversity between samples in each group. It can be further seen that some generated samples highly correlate with samples from the training set.

We show this both as a failure of the Inception Score metric, the commonly reported value for class-conditional video synthesis on UCF-101, but also as strong signal that UCF-101 is not a complex or diverse enough dataset to facilitate interesting video generation. Each class is relatively small, and reuse of clips from shared underlying videos means that the intra-class diversity can be restricted to just a handful of videos per class. This suggests the need for larger, more diverse and challenging datasets for generative video modelling, and we believe that Kinetics-600 provides a better benchmark for this task.

C Miscellaneous Experiments

Here we detail a number of modifications or miscellaneous results we experimented with which did not produce a conclusive result.

- We experimented with several variations of normalization which do not require calculating statistics over a batch of data. Group Normalization (Wu & He, 2018) performed best, almost on a par with (but worse than) Batch Normalization. We further tried Layer Normalization (Lei Ba et al., 2016), Instance Normalization (Ulyanov et al., 2016), and no normalization, but found that these significantly underperformed Batch Normalization.
- We found that removing the final Batch Normalization in $\mathcal{G}$, which occurs after the ResNet and before the final convolution, caused a catastrophic failure in learning. Interestingly, just removing the Batch Normalization layers within $\mathcal{G}$'s residual blocks still led to good (though slightly worse) generative models. In particular, variants without Batch Normalization in the residual blocks often achieve significantly higher IS (up to 110.05 for 64×64 12 frame samples – twice normal). But these models had substantially worse FID scores (1.22 for the aforementioned model) – and produced qualitatively worse video samples.

[4]We use the Chainer (Tokui et al., 2015) implementation of Inception Score for C3D available at https://github.com/pfnet-research/tgan.

- Early variants of DVD-GAN contained Batch Normalization which normalized over all frames of all batch elements. This gave $\mathcal{G}$ an extra channel to convey information across time. It took advantage of this, with the result being a model which required batch statistics in order to produce good samples. We found that the version which normalizes over timesteps independently worked just as well and without the dependence on statistics.
- Models based on the residual blocks of BigGAN-deep trained faster (in wall clock time) but slower with regards to metrics, and struggled to reach the accuracy of models based on BigGAN's residual blocks.

D GENERATED SAMPLES

It is difficult to accurately convey complicated generated video through still frames. Where provided, we recommend readers view the generated videos themselves via the provided links. We refer to videos within these batches by row/column number where the video in the 0th row and column is in the top left corner.

D.1 SYNTHESIS SAMPLES

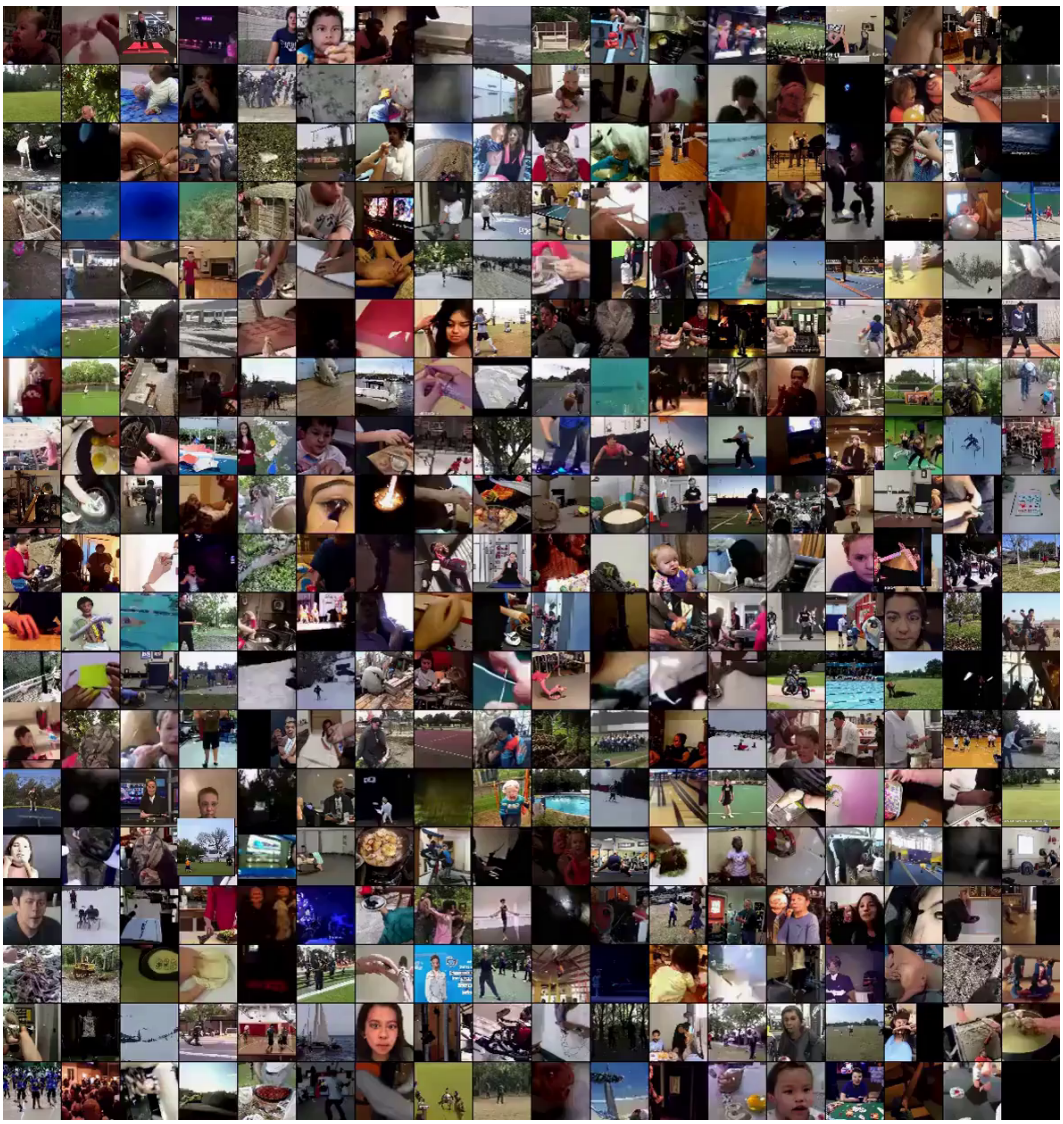

Figure 10: The first frames from a random batch of samples from DVD-GAN trained on 12 frames of 64×64 Kinetics-600. Full samples at https://drive.google.com/file/d/155F1lkHA5fMAd7k4W3CQvTsi1eKQDhGb/view?usp=sharing.

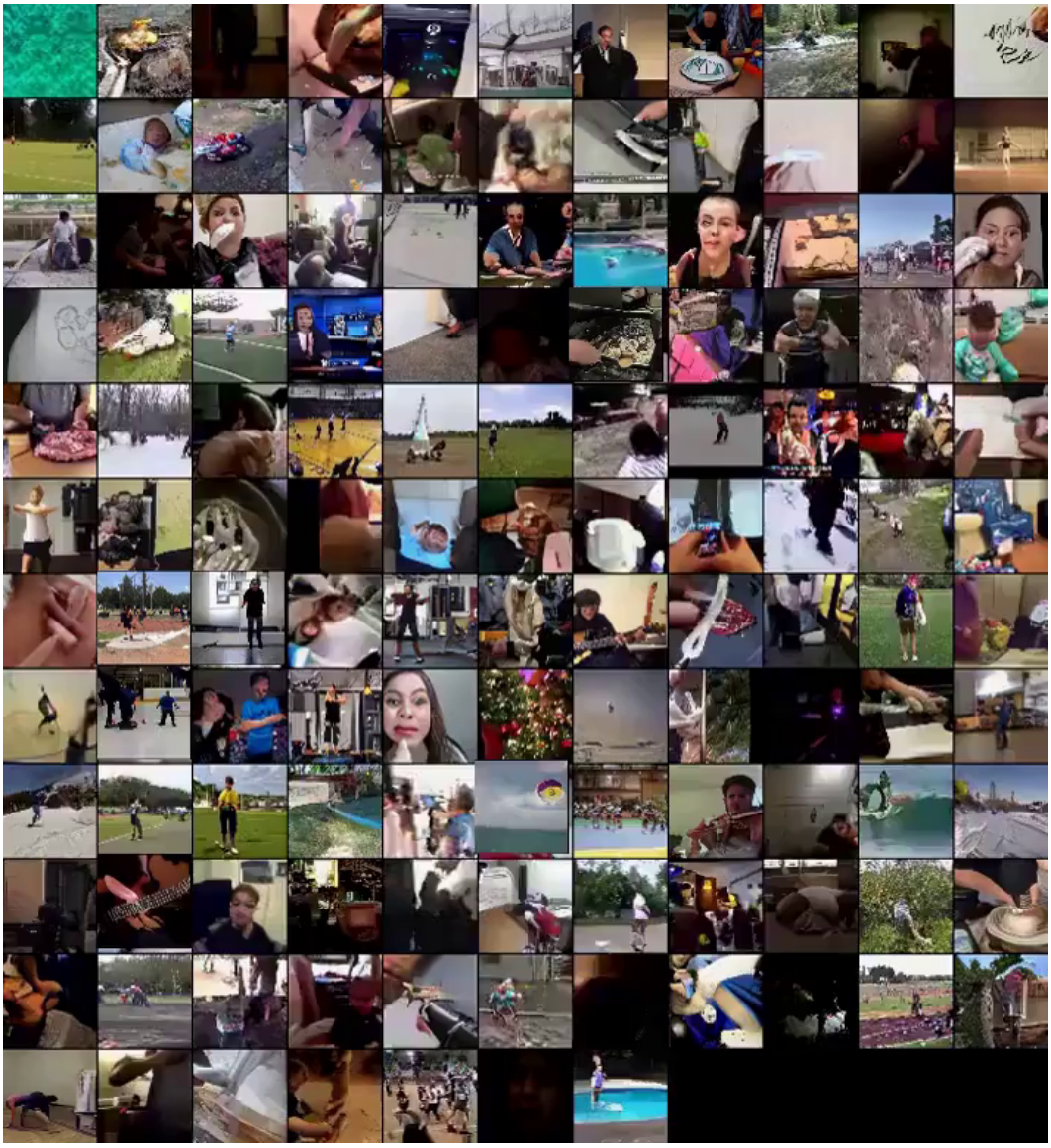

Figure 11: The first frames from a random batch of samples from DVD-GAN trained on 48 frames of 64×64 Kinetics-600. Full samples at https://drive.google.com/file/d/1FjOQYdUuxPXvS8yeOhXdPQMapUQaklLi/view?usp=sharing.

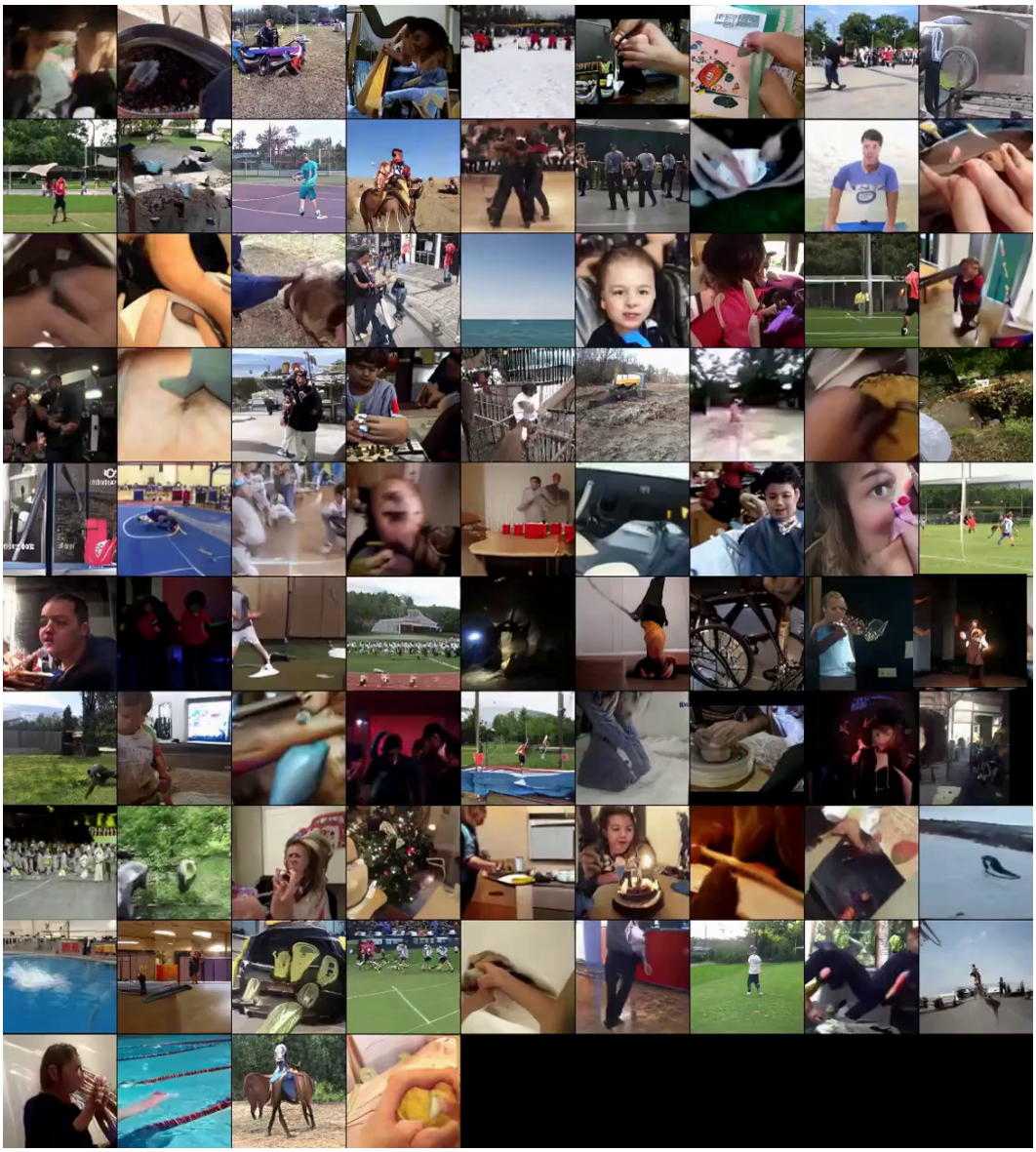

Figure 12: The first frames from a random batch of samples from DVD-GAN trained on 12 frames of 128×128 Kinetics-600. Full samples at https://drive.google.com/file/d/165Yxuvvu3viOy-39LhhSDGtczbWphj_i/view?usp=sharing

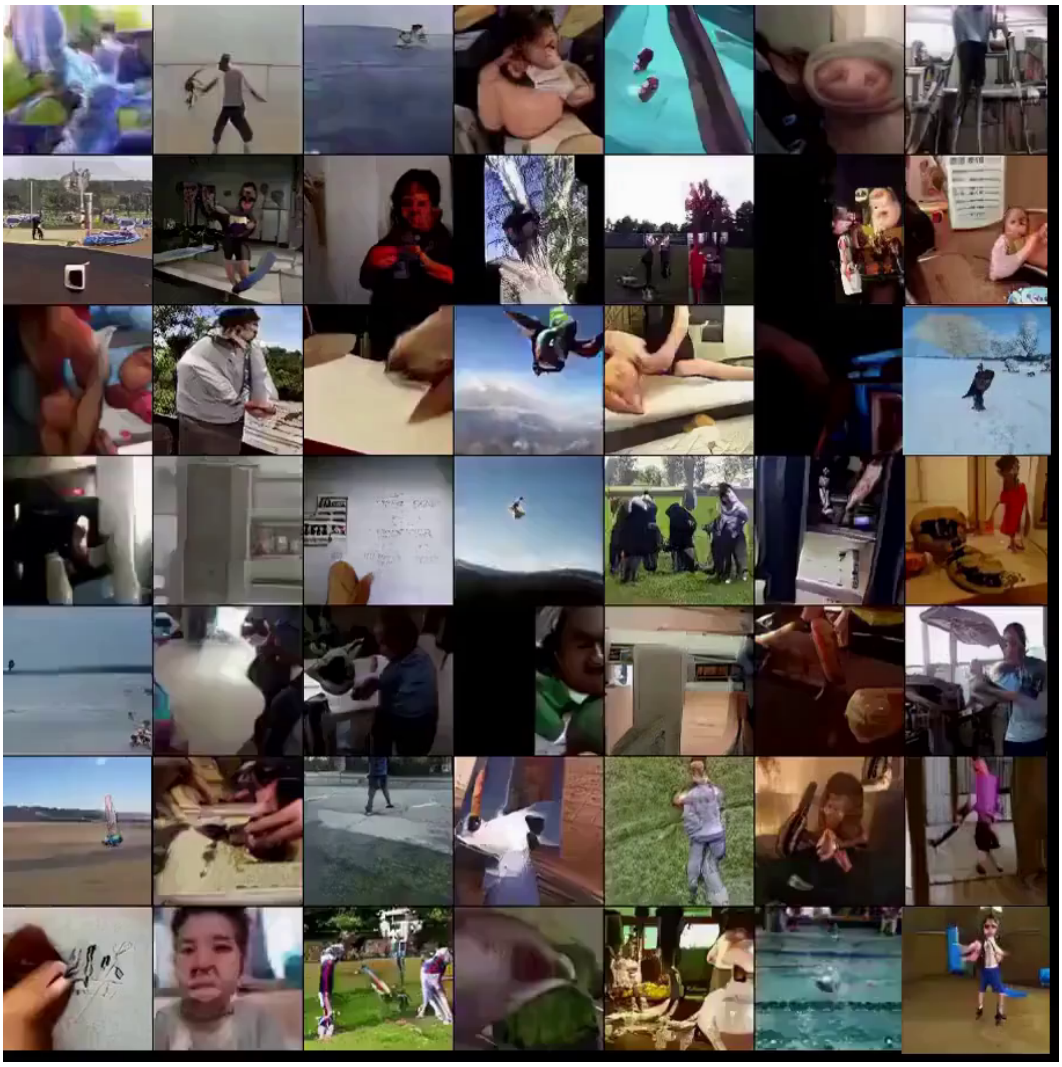

Figure 13: The first frames from a random batch of samples from DVD-GAN trained on 48 frames of 128×128 Kinetics-600. Full samples at `https://drive.google.com/file/d/1P8SsWEGP6tEGPPNPH-iVycOlN6vpIgE8/view?usp=sharing`. The sample in row 1, column 5 is a stereotypical example of a degenerate sample occasionally produced by DVD-GAN.

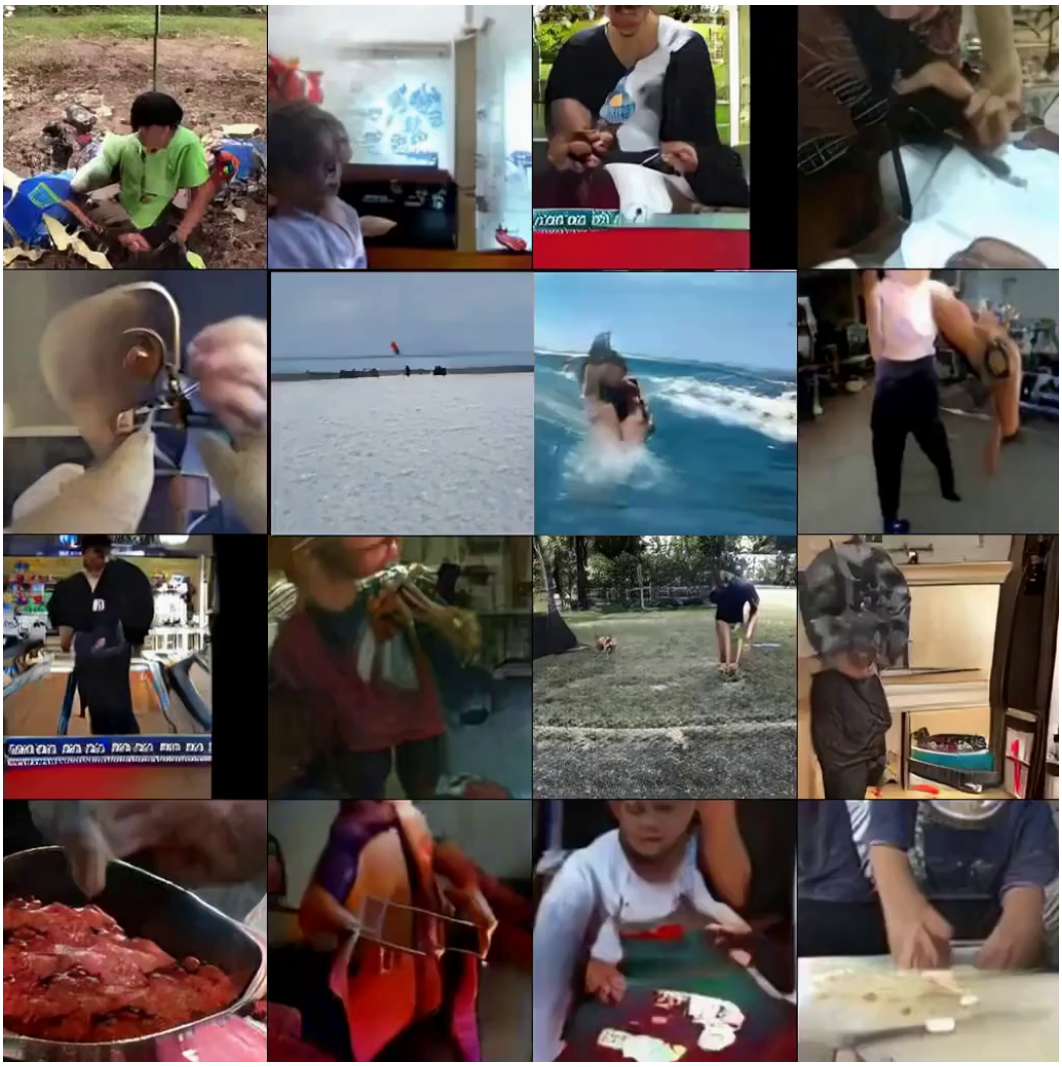

Figure 14: The first frames from a random batch of samples from DVD-GAN trained on 12 frames of 256×256 Kinetics-600. Full samples at `https://drive.google.com/file/d/1RGRVKCpVaG8z3p9GBCamRk4apiIR7jUc/view?usp=sharing`.

D.2 Interpolation Samples

We expect $\mathcal{G}$ to produce samples of higher quality from latents near the mean of the distribution (zero). This is the idea behind the Truncation Trick (Brock et al., 2019). Like BigGAN, we find that DVD-GAN is amenable to truncation. We also experiment with interpolations in the latent space and in the class embedding. In both cases, interpolations are evidence that $\mathcal{G}$ has learned a relatively smooth mapping from the latent space to real videos: this would be impossible for a network that has only memorized the training data, or which is only capable of generating a few exemplars per class. Note that while all latent vectors along an interpolation are valid (and therefore $\mathcal{G}$ should produce a reasonable sample), at no point during training is $\mathcal{G}$ asked to generate a sample halfway between two classes. Nevertheless $\mathcal{G}$ is able to interpolate between even very distinct classes.

Figure 15: An example *intra-class* interpolation. Each column is a separate video (the vertical axis is the time dimension). The left and rightmost columns are randomly sampled latent vectors and are generated under a shared class. Columns in between represent videos generated under the same class across the linear interpolation between the two random samples. Note the smooth transition between videos at all six timesteps displayed here.

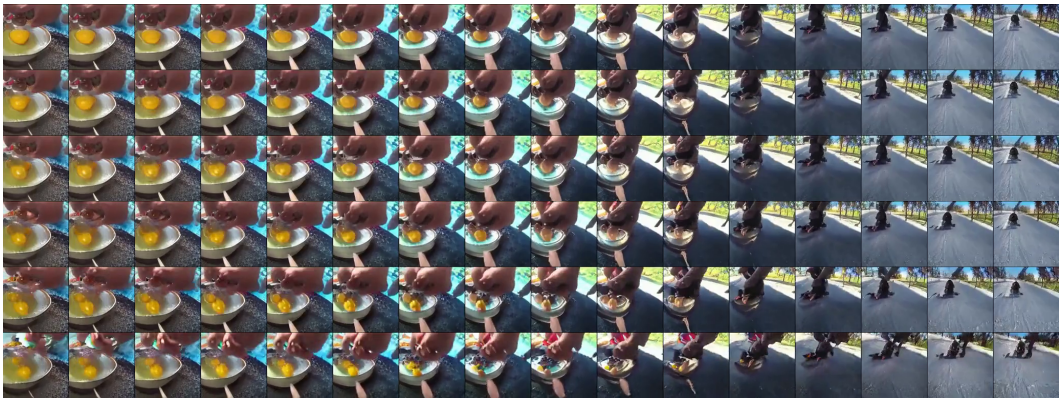

Figure 16: An example of *class* interpolation. As before, each column is a sequence of timesteps of a single video. Here, we sample a **single** latent vector, and the left and rightmost columns represent generating a video of that latent under two different classes. Columns in between represent videos of that same latent generated across an interpolation of the class embedding. Even though at no point has DVD-GAN been trained on data under an interpolated class, it nevertheless produces reasonable samples.