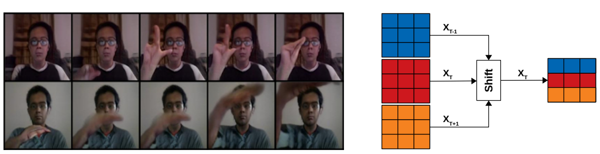
Temporal Shift GAN for Large Scale Video Generation

Andres Munoz*, Mohammadreza Zolfaghari*, Max Argus, Thomas Brox
University of Freiburg
{amunoz, zolfagha, argusm, brox}@informatik.uni-freiburg.de

Abstract

Video generation models have become increasingly popular in the last few years, however the standard 2D architectures used today lack natural spatio-temporal modelling capabilities. In this paper, we present a network architecture for video generation that models spatio-temporal consistency without resorting to costly 3D architectures. The architecture facilitates information exchange between neighboring time points, which improves the temporal consistency of both the high level structure as well as the low-level details of the generated frames. The approach achieves state-of-the-art quantitative performance, as measured by the inception score on the UCF-101 dataset as well as better qualitative results. We also introduce a new quantitative measure (S3) that uses downstream tasks for evaluation. Moreover, we present a new multi-label dataset MaisToy, which enables us to evaluate the generalization of the model.

1. Introduction



(a) Jester Samples. (b) Temporal Shift.

Figure 1: (a) Selected frames from videos generated by TSB trained on Jester at 192×192. (b) The shift operation replaces a subset of features in time step T with features from frames $T-1$ and $T+1$ to facilitate information exchange between neighboring frames.

Generative Adversarial Networks (GANs) [20] are a powerful way to generate high-resolution images [42, 3]. Video generation adds further complexity, as the resulting content should be both spatially and temporally coherent. This is particularly true for the aspect of motion, which does not exist in still images.

3D Convolutional Neural Network (CNN) architectures appear well-suited to trivially lift the progress made in single images to videos [13, 7, 33, 17], yet their usefulness for video generation is still a matter of debate [38, 30]. A argument against 3D CNNs is that the temporal dimension behaves differently from the spatial dimensions. The authors of MoCoGAN [38] showed that equal treatment of space and time results in fixed-length videos, whereas the length of real-world videos varies. Moreover 3D CNNs have more parameters, which according to studies in literature [21, 15] make them more susceptible to overfitting [18].

We share the view of TGAN [30] and MoCoGAN [38], where instead of mapping a single point in the latent space to a video, a video is assumed to be a smooth sequence of points in a latent space in which each point corresponds to a single video frame. As a result, our video generator consists of two submodules: a *sequence generator* that generates a sequence of points in the latent space, and an *image generator* that maps these points into image space.

For the image generator, we propose a Temporal Shift Self-Attention Generator, which introduces a temporal shifting mechanism [21] into residual blocks of the generator. Temporal shifting mechanism enables the model to exchange information between neighbor frames. The temporal shifting module is complementary to 2D convolutions in the image generator and allows us to efficiently model the temporal dynamics of a video by facilitating the information exchange between neighbor frames.

The growing interest in video generation methods gives rise to challenges in comparing the quality of generated samples. There are two types of approaches to evaluation: qualitative and quantitative. On one side, qualitative measures (e.g. human rating) are not good at detecting memorization or low diversity. On the other side, quantitative measures are not robust nor consistent [4, 5, 28]. Although IS [45] has gained popularity in evaluating the quality of generated images, it has several drawbacks; particularly failing to detect mode collapse and memorization. FID [29] assumes that features are from Gaussian distribution, which is not always a valid assumption.

Therefore, we propose a new evaluation measure named

*Equal Contribution

Symmetric-Similarity-Score (S3) to measure the quality of generated videos. S3 measures the domain gap of an action classifier when trained on synthesized videos and tested on real ones, and vice-versa. Consequently, it penalizes missing intra-class diversity, and is also sensitive to both structural deficits and low-level artifacts in the generated data. Hence, it is robust to over-confident classifier predictions, and it is less dependent on model parameters or pre-processing.

Currently video generation models have relied on action recognition datasets for benchmarking. However, as these datasets typically only assign one label per video, they do not allow for an easy analysis of the generalization capabilities of a model. By formulating the conditional generative modelling problem as a multi-label one, we can easily analyze generalization by forcing the model to try to generate samples from label combinations that are not in the dataset.

Experiments on the UCF101, Jester, and Weizmann datasets show substantial improvements in the quality of the generated videos for the proposed design compared to previous work. At the same time, experiments on the newly introduced MaisToy dataset show that TS-GAN is able to generalize to unseen data.

Our paper makes three contributions: (1) it introduces a new 2D video generator design with an ability to model spatio-temporal content by facilitating the information exchange between neighboring frames. (2) It introduces a new evaluation metric based on the domain gap between synthesized and real videos in terms of video classification performance. (3) It introduces a new dataset which allows a fast and more in-depth analysis of the generalization and semantic modelling capabilities of video generation models.

2. Related Work

Image generation has recently seen leaps in performance [49, 43, 41, 42, 16, 3, 39], thanks to recently introduced frameworks such as SN-GAN (Miyato *etal* [41]), introduced the concept of spectral normalization of the discriminator's weights. Zhang *et al* [16], designed a self-attention module that allowed the network to create non-local spatial relationships. Then, BigGAN [3] build upon this work by establishing some architectural and training guidelines by which GANs can be stable, converge faster and produce better quality samples. In this study we propose TS-GAN , which builds upon BigGAN and extends it to video. TS-GAN generates videos in a per-frame basis, it can thus exploit further developments on image generation.

Video generation is a highly challenging task as a result of needing to ensure a smooth transition across video frames. Most works in video generation have been on the closely related frame prediction task [11, 48, 50, 23, 44, 8, 31, 34]. The main difference between video generation and frame prediction, is that in frame prediction the network is trying to generate a set of T frames given a set of N previously seen frames. Conversely, video generation only uses the latent code, and in some occasions a label, to generate a set of frames.

Several frameworks for video generation using GANs have been proposed in the past. Vondrick *et al* [6] proposed a two-stream network which explicitly separated the generation of the foreground and the background. They assumed that background in the entire video is static, which is not true in real-world video datasets. Saito *etal* [30] introduced a temporal generator to transform a single latent variable into a sequence of latent variables, to be able to just utilize a 2D network as a generator. As a matter of fact, they showed that a 2D generator can outperform a 3D one. MoCoGAN [38] separated motion and appearance features by dividing the latent code in two smaller sub-codes, one per-each set of features.

Acharya *etal* [10] used a coarse to fine approach to improve quality and convergence speed. TGANv2 [36], efficiently trains models that generate high dimensional samples by subsampling features and videos from the batch. DVD-GAN [2], leveraged a high capacity model to synthesize high quality samples from complex datasets. These models showed that GANs can be effective at generating videos. Nevertheless, the previously proposed models either suffer from lack of quality, mode collapse, memorization or require an excessive amount of computational power and data to train properly. Our framework outperforms the state-of-the art on UCF-101, based on IS measure, while keeping memorization to a minimum.

Moreover, previous methods lack a complete quantitative assessment of the performance of the respective methods. They rely on metrics such as IS [37] and FID [19] which don't tell the full story about sample quality. These metrics are dependent on availability of models and are also sensitive to changes in the pipeline. Here we introduce a metric, called Symmetric Similarity Score (S3), which aims to represent both quality and diversity in a single scalar value. In addition, S3 is robust to changes in pre-processing and model parameters.

3. Temporal Shift GAN

3.1. Preliminaries

GANs [20] are a class of generative models consisting of a generator and a discriminator networks. The discriminator is a binary classifier that outputs the probability a sample is either real or synthesized. The generator is a function that generates synthetic samples x that look similar to real samples.

GAN training is a minimax game, in which the discriminator D tries to minimize the probability of making a mistake, while the generator G seeks to maximize this probability:

$$\min_{G} \max_{D} V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[log\ D(x)] + \mathbb{E}_{z \sim p_z(z)}[log\ (1 - D(G(z)))] \quad (1)$$

where p_{data} is the distribution of the empirical data and p_z represents the chosen prior distribution of the latent codes z.

Although GANs tend to have problems generating diverse samples (mode collapse), the recent BigGAN method [3] demonstrated state-of-the-art performance in image synthesis by leveraging the best practice of previous methods, such as spectral normalization and projection.

The proposed video generation architecture TS-GAN consists of a sequence generator, an image generator and a video discriminator; an overview of which is shown in Figure 2. It is a projection based conditional GAN approach as proposed by Miyato & Koyama [32] using the hinge formulation of the GAN objective (Lim & Ye [27]; Tran et al.[14]):

$$L_D = \mathbb{E}_{(x,y) \sim p_{data}}[\min(0, -1 + D(x,y))] - \mathbb{E}_{z \sim p_z, y \sim p_{data}}[\min(0, -1 - D(G(z), y)] \quad (2)$$

$$L_G = -\,\mathbb{E}_{z \sim p_z, y \sim p_{data}}[D(G(z), y)] \quad (3)$$

where y is the video label. We introduce several improvements to different aspects of the video generating framework including sequence generator and the image generator.

3.2. Generator

The generator is divided into two parts. First we generate a sequence of latent codes, then in the second step the image generator maps these latent codes to a sequence of T frames.

3.2.1 Sequence Generator

We construct the latent space $Z_{ABC} \in \mathbb{R}^d$ as three independent multi-variate Gaussian distributions $Z_A \in \mathbb{R}^{d_A}$, $Z_B \in \mathbb{R}^{d_B}$ and $Z_C \in \mathbb{R}^{d_C}$ with their diagonal covariance matrices Σ_A, Σ_B and Σ_C respectively. We construct our latent code, Z_{ABC}[1], by concatenation of Z_A, Z_B, Z_C as $Z_{ABC} = \left[Z_A, Z_B, Z_C\right]^T$. The final distribution Z_{ABC} is a multi-variate Gaussian distribution with diagonal covariance matrix. By using an independent parametrization of the subspaces, the network is able to learn more nuanced distributions, thus a better modelling of the features. Subspaces have no prior meaning - the network learns to interpret each part of the code as it sees fit.

The latent code Z_{ABC} does not have a temporal dimension. Since our generator is image based, we first have to create a progression of correlated latent codes that extends through the intended duration of the video. This is done by the sequence generator (See Fig. 2). We first transform the latent code with a fully connected layer as $Z_{fc} = FC(Z_{ABC})$.

[1] This paper uses Z_{ABC} to refer to the latent space, a vector of latent codes or a single latent code.

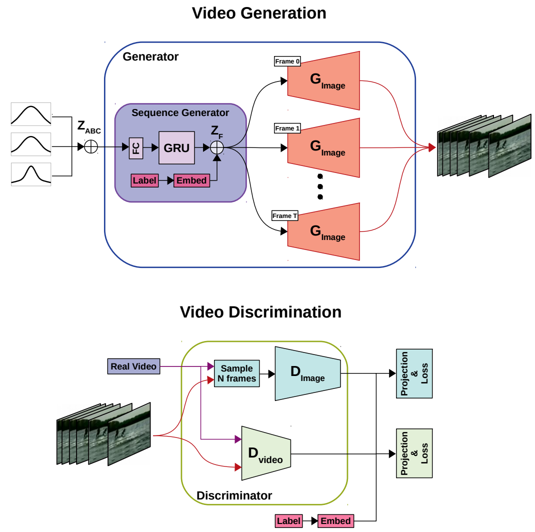


Figure 2: TS-GAN framework. Sampled Gaussian noise is with several different variances is concatenated ($\oplus$) into Z_{ABC}. A sequence generator using gated recurrent units is then used to generate a vector Z_F. The image generator then transforms this into video frames. Video discrimination is done by a 2D discriminator (D_{Image}) judging a subset of the video frames and a 3D discriminator (D_{Video}) judging all frames and assessing the motion consistency of the video.

Then we feed Z_{fc} into a Gated Recurrent Unit (GRU) to generate a sequence of T correlated codes as $Z_{gru} = [z^1_{gru}, \dots, z^T_{gru}]^T$, where each z^i_{gru} corresponds to the i-th frame in the video sequence. In total this results in an input of size $[T, d]$, where T is the number of frames to generate.

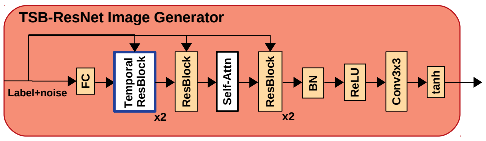
We concatenate these latent codes with per-class embeddings $e(y)$ of size 120, where y is a randomly sampled class label. This results in a sequence of T codes as

$$Z_F = \left[\begin{bmatrix} z^1_{gru} \\ e(y) \end{bmatrix}, \dots, \begin{bmatrix} z^T_{gru} \\ e(y) \end{bmatrix} \right] \in \mathbb{R}^{(d+120)} \quad (4)$$

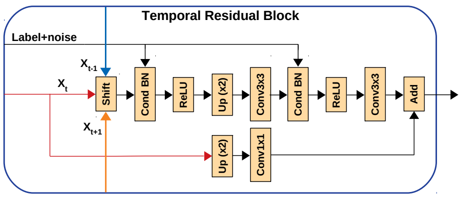
We feed Z_F into the image generator to generate a sequence of T frames (Figure 2).

3.2.2 TSB Image Generator

To synthesize “realistic“ images, some approaches [2, 12] utilized BigGAN [3] image generator as their backbone architecture. However, in this architecture each image is generated independent of others. Therefore, the networks are not able to enforce temporal consistency between frames. To alleviate this problem, we introduce the temporal shift mechanism [21] to BigGAN image generator architecture to facilitate information exchange between neighboring frames. We call the proposed generator Temporally Shifted BigGAN (TSB) image generator, illustrated in Figure 3a, because of it’s feature shifting mechanism (Figure 1b). This design not only facilitates the information exchange in temporal dimension but also equipped with a self-attention layer which enables the generator to model the relationships between spatial regions [16]. Unlike full 3D convolutions it only shares a small subset of features between neighboring frames. This allows faster inference and uses less parameters than 3D models.



(a) TSB Generator.



(b) Temporal Residual Block.

Figure 3: (a) The Temporally Shifted (TSB) image generator architecture. Note that the temporal residual blocks are only used at the beginning of the generator to minimize loss of spatial information. (b) Temporal residual up-sampling block used in TSB. The operation Up(x2) means up sampling via interpolation by a factor of 2.

In our proposed image generator TSB, the temporal shift module can simply be added to the beginning of the residual blocks, as shown in Figure 3b. This is in contrast to the non-temporal (NT) variant of our architecture, which uses normal residual blocks. We only vary the first two residual blocks of our network, and call these *temporal residual blocks* to distinguish them from the latter residual blocks which are always of the normal variant. This is shown in Figure 3a. All residual blocks use conditional batch normalization and receive as input the vector Z_F.

3.3. Discriminator

We use two independent discriminators, an image discriminator, D_{Image}, and a video discriminator named D_{Video}.

Image Discriminator D_{Image} gives a frame-wise assessment of content and structure. D_{Image} is a ResNet based architecture [24], similar to BigGAN [3], it is applied to a subset of N frames of the video. D_{Image} is doing the heavy lifting with respect to image quality. N remains a hyperparamter that allows a trade-off between memory efficiency and frame quality.

Video Discriminator D_{Video} examines the temporal consistency of videos and provides the generator with a learning signal to generate a consistent motion throughout all T frames. TS-GAN 's D_{Video} is inspired by MoCoGAN's [38] video discriminator. We chose this architecture to keep the network efficient. The factorized design allows for smaller D_{Video} networks as it can focus on the temporal aspect.

4. Symmetric Similarity Score (S3)

The Inception Score [45] (IS) and Frechet Inception Distance [29] (FID) are the most common metrics used to evaluate GANs. On one hand, IS ($\exp(D_{KL}(P(y \mid x) \mid P(y)))$) is based on two criteria: the distribution of predicted labels $P(y \mid x)$ should have a low entropy and the marginal distribution $P(y)$ should have a high entropy. On the other hand, FID measures performance of a generator by using features produced by an intermediate layer to parameterize a multivariate normal distribution of real and fake features respectively. FID rates the fake samples by calculating the distance between distributions, the closer the better.

Although high IS correlates with subjective quality and a low FID with both quality and intra-class diversity of samples, they both have drawbacks. IS cannot capture intra-class diversity. Yushchenko *etal* [46] showed that small changes to the data pre-processing leads to a change between 7% and 17% in IS score and adversarial samples may lead the classifier to be overconfident about samples leading to a higher score [40]. FID assumes features follow a normal distribution, which is not true for real world datasets. Thus, two completely different distributions might lead to a good score, while not being actually similar. At the same time, FID is also vulnerable to pre-processing and model changes. Neither IS nor FID are able to account for memorization of the dataset.

Symmetric-Similarity-Score (S3) uses generalization of classifiers between real and synthetic samples to measure quality of generated videos. The performance of model is measured by "quality of samples" and "diversity of generated samples".

The performance of a classifier trained on synthetic data and evaluated on real data (SeR) should increase, if synthetic samples are **diverse** and **realistic**. A classifier trained on real data being evaluated on synthetic (ReS) data should only perform well, if synthetic samples are **realistic**.

We normalize these values by comparing to the real performance (ReR). Since SeR has more information about the overall performance, S3 has an exponential relationship to it, thus rewarding models with good diversity and sample quality and harshly penalizing them otherwise (Equation 5). S3 has the advantages of capturing intra-class diversity, being more robust to over-confident predictions and small changes in the model's parameters, while still being easier to interpret than IS or FID.

$$S3 = \sqrt{\left(\frac{SeR}{ReR}\right)^2 \cdot \left(\frac{ReS}{ReR}\right)} \tag{5}$$

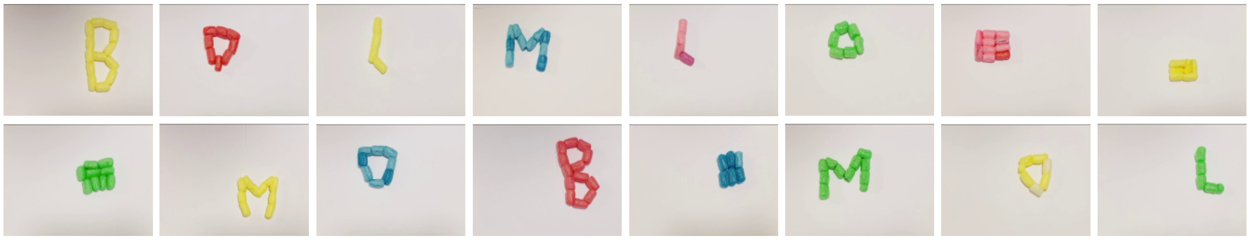


Figure 4: Samples taken from MaisToy dataset. The dataset includes shapes of varying difficulty, the dataset is balanced in terms of colors and motions.

This approach is similar to Classification Accuracy Score (CAS) [35], which used a classifier's SeR to evaluate generative models and lacks ReS evaluation. However, just using SeR to evaluate a model does not tell the full story. Since SeR is dependent on both quality of samples and intra-class diversity, we need ReS to know if the SeR performance is being driven by sample quality or diversity.

Generative models must create fake samples that comes from the same distribution as the dataset. To generate samples which are not included in the dataset it needs to be able to generalize to unseen data. However, existing datasets used for video generation are not truly equipped for testing a model's generalization due to the fact that we can only directly control the action semantic on the dataset. By only having control over the action, we can't force the network to generate certain features within the class, therefore it is hard to corroborate the model's ability to generalize. Thus, there's a need for a dataset that allows a higher degree of control over the semantics of the samples.

5. MaisToy Dataset

We introduce MaisToy, a dataset composed of 238 videos of clay figures of 5 different shapes, 4 colors performing 4 distinct motions. The videos recorded have 55 frames on average, with a size of 320×240 px. The dataset is balanced and compact. This allows for faster evaluation of design choices without requiring large computational resources, this addresses a big challenge in video GAN research. The balanced nature of the dataset facilitates testing of generalization by holding out some combinations of semantics during training and trying to generate them during testing. At the same time, the three distinct semantics (shape, color and motion) support a more in-depth analysis of the semantic modelling capabilities of model designs.

6. Experiments

We evaluated our model both qualitatively and quantitatively on the quality of frames, the realism of whole videos, diversity and memorization using several different datasets. FID and further qualitative evaluation will be found in the supplementary material.

6.1. Datasets

We use four datasets UCF101[25], Weizmann[26], Jester[22] and MaisToy for our experiments.

UCF-101. $13,220$ videos of 101 different sports action classes. We trained models to generate samples both at 96×96 and at 128×128, we resize to 127×127 and 170×170 respectively and crop to its corresponding final size. We set N to 8. [25]

Weizmann. 93 videos of 9 people performing 10 different actions. To train we cropped the videos to 96×96. For all experiments, we randomly extract 16 subsequent frames and set N to 4. [26]

Jester. $118,562$ videos and 27 different hand gesture classes. Due to the small frame size, we first re-size 96×136 before cropping to 96×96 to preserve the aspect ration. As in Weizmann, we extract 16 subsequent frames and set N to 4. [22]

MaisToy$_{\text{Multi}}$. Multi label variant of MaisToy, we trained a model to generate at 128×128, we resize as for UCF-101. We set N to 4. For generalization testing, the dataset was split into train and test sets.

MaisToy$_{\text{Single}}$. Single label variant of MaisToy, we use only the motion labels to generate videos. We trained a model to generate at 96×96, we resize as for UCF-101. We set N to 4.

6.2. Model Configurations

Three different variation of our method were tested in order to find the strongest configuration.

NT: A non-temporal model using a BigGAN generator and all latent variables' distribution were $\mathcal{N}(\mu = 0, \sigma = 1)$.

NT-VAR: Same generator as above, but we change the latent variables' distribution deviations to $\sigma_A = 0.5, \sigma_B = 1$ and $\sigma_C = 2$.

TSB: Same as in NT-VAR, however we change the generator to the temporally shifted BigGAN generator.

For all trained models we set d to 120 by setting d_A, d_B and d_C to 20, 20 and 80 respectively. We employ a learning rate of 5×10^{-5} for the generator and 2×10^{-4} for the discriminator. The video length T is fixed to 16. All

Dataset	Method	IS (↑)
UCF-101	VGAN [6]	8.18 ± 0.09
	TGAN [30]	11.85 ± 0.07
	MDP [46]	11.86 ± 0.11
	MoCoGAN [38]	12.42 ± 0.03
	Prog. VGAN [10]	14.56 ± 0.05
	TGANv2 [36]	24.34 ± 0.35
	DVD-GAN [2]	32.97 ± 1.7
	NT	31.91 ± 0.14
	TSB	**42.79 ± 0.63**
Jester	**TSB**	**11.90 ± 0.12**

(a) Comparison of Inception Score between models on UCF-101.

Dataset	Method	Train on: Synth.	Real		**S3**
		eval. on: Real	Synth.	Real	
Weizmann	MoCoGAN	61.11	32.83	95.83	0.37
	NT	80.31	62.53	95.83	0.67
	NT-VAR	**88.30**	64.29	95.83	0.75
	TSB	88.10	**71.43**	95.83	**0.79**
MaisToy$_{\text{Multi}}$	TSB	**82.74**	**45.11**	67.70	**0.99**
MaisToy$_{\text{Single}}$	TSB	**45.83**	**52.98**	66.07	**0.62**
UCF101	NT	45.5	46.8	85.9	0.39
	TSB	**48.55**	**54.91**	85.9	**0.45**
Jester	TSB	**15.85**	**13.86**	91.4	**0.07**

(b) Evaluation S3 measure based on action classification generalization between generated and real samples.

Table 1: **Comparison with previous methods on four different datasets.**

Low Quality	Collapsed	Normal	Train on: Synth.	Real		**S3**	**IS** (↑)
			eval. on: Real	Synth.	Real		
✗			41.4	50.06	85.9	0.36	34.01 ± 0.22
	✗		28.3	59.3	85.9	0.27	41.04 ± 0.82
		✗	**48.55**	**54.91**	85.9	**0.45**	**42.79** ± **0.63**

Table 2: Comparison between TSB's UCF-101 S3 and IS results at different levels of model quality.

experiments performed on Weizmann and Jester are done with a batch size of 40. We trained both NT and TSB to generate 96×96 and 128×128 sized samples respectively. NT was trained on a batch size of 56, while TSB was trained on a batch size of 64.

All models were trained on full precision, 96×96 models were trained on two Nvidia RTX2080Ti's, 128×128 models were trained on one 32GB Nvidia V100. Jester and UCF-101 models took 4 weeks to reach the performance reported here, while models trained on Weizmann and MaisToy took 7 and 10 days respectively. Training times for Jester and UCF-101's could be cut short by using larger computational resources.

6.3. Quantitative Evaluation

A thorough evaluation of quality of samples simply by qualitative experiments is not possible due to the sheer number of samples that need to be evaluated in order to do a proper assessment.

IS: We evaluate the IS as comparative benchmark on the UCF-101 dataset. The IS is calculated using the last layer of a C3D[2] [13] model which was pre-trained on Sports-1M [1] and fine-tuned on the UCF-101 dataset as per [30]. The model receives frames of size 128×128, we resized when necessary. We use 10,000 samples to calculate the score. The standard deviation is calculated by repeating this procedure 10 times. Values for VGAN, MoCoGAN, Progressive VGAN, TGANv2, and DVD-GAN are shown as reported in [2], TGAN and MDP's values are reported as they appear in the original works [30, 46]. On Jester, we use a TSN [47] action recognition network pre-trained on ImageNet [9] and fine-tuned on Jester, otherwise te same procedure as for UCF-101 is used. TSB produces samples that beat the state of the art (see Table 1a). Although, IS scores might suggest an overwhelming improvement over all existing methods, when qualitatively comparing samples from NT and TSB (Figure 6) we don't see a vast improvement as the score suggests. This could be because our samples might be exploiting C3D [13] in a way that it is over confident about its prediction, thus a higher score.

S3: To calculate S3 on Weizmann and UCF-101 we used the TSN [47] action recognition network pretrained on ImageNet [9]. Since Jester is a motion dataset we decided to use ECO [33] because it incorporates a 3D network, to improve classification. On the Weizmann dataset we compare to MoCoGAN. All experiments on a dataset were done under the same conditions. Training details of the classifier will be included in the supplemental material. From Table 1b, we can see TSB produces a significant performance increase over all methods. It appears TSB, is able to increase the quality of the samples with a minimal loss of diversity. TSB was able synthesize test set samples of the MaisToy$_{\text{Multi}}$ dataset as implied the by SeR score being higher than the ReS, this suggests generalization. MaisToy samples will be included in the supplementary material. On UCF-101, Table 1b shows small discrepancies between SeR and ReS indicating a good diversity of samples. S3 scores produced by TSB show an improvement over NT. The performance difference between methods seems more reasonable, when visually comparing samples (Figure 6), than the ones shown in Table 1a. Additionally, Table 2 shows that S3 is able to to capture mode collapse in the generated samples, while still being equally as good as IS at measuring sample quality. This indicates

[2]Using the code provided by github.com/pfnet-research/tgan.

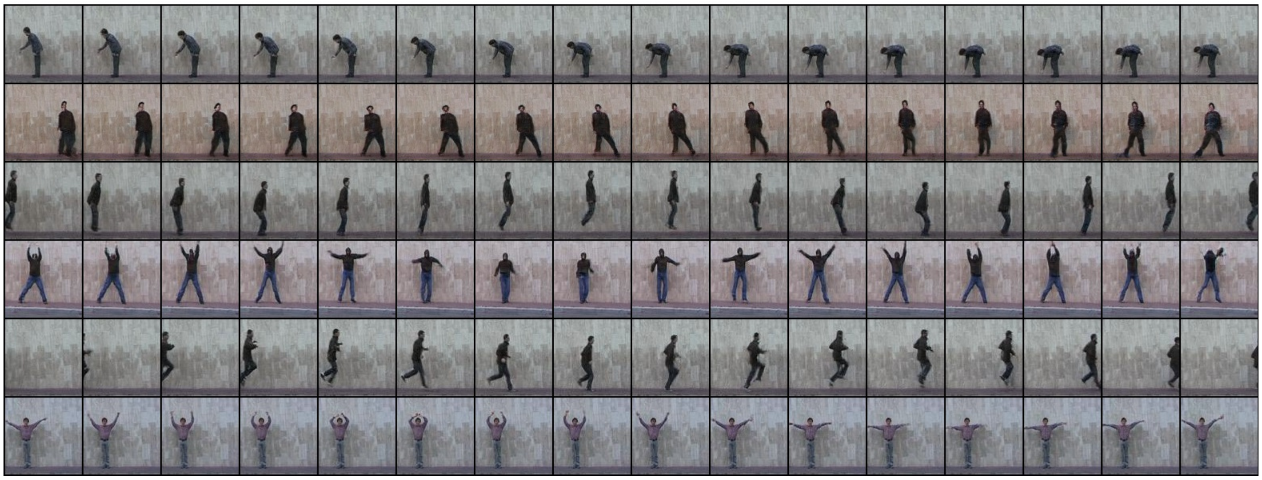
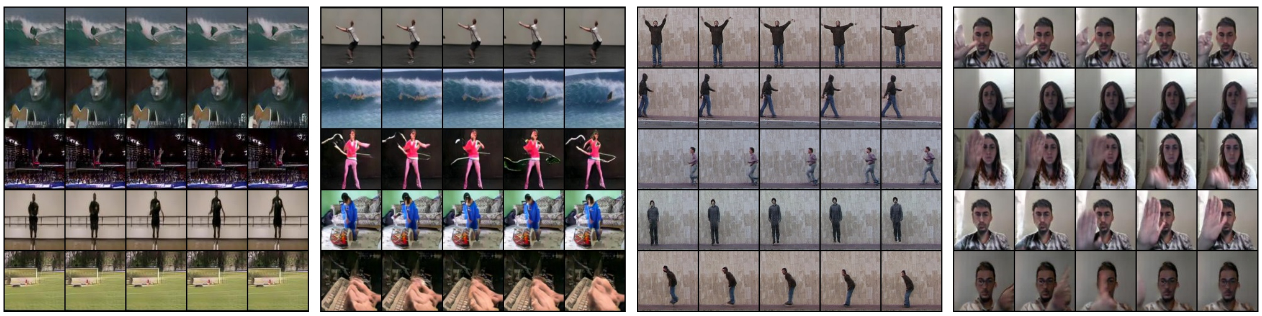


Figure 5: Generated samples on Weizmann dataset with our TSB model.



11

Figure 6: Generated samples (from left to right) of NT trained on UCF-101 and TSB trained on UCF-101, Weizmann and Jester. Videos can be found here: `https://drive.google.com/file/d/14k17fQTTztV2MPKAglOS6kP_yDKJ24n6/view?usp=sharing`

that S3 is more reliable than IS.

6.4. Qualitative Results

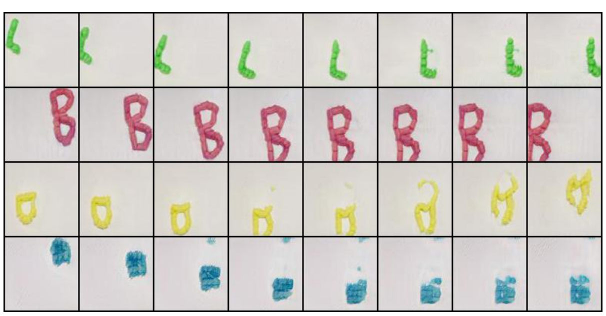


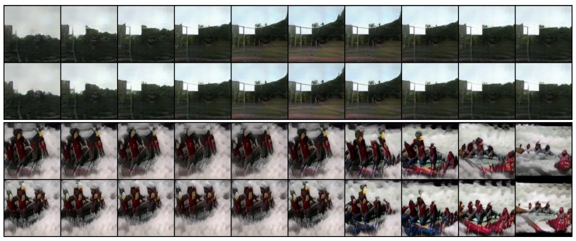
Figure 7: Samples from the $\text{MaisToy}_{\text{Single}}$ dataset, here we present samples from classes (top to bottom) right, left, up and down. The shapes represented here are Letter L, Letter B, triangle and square.

Figures 6, 7, and 5 show qualitative samples generated by training TSB to generate samples of size 96×96 on UCF101, MaisToy, and Weizmann datasets respectfully. In Figure 4, we used the motion label only variant of MaisToy called $\text{MaisToy}_{\text{Single}}$. In this figure we can appreciate that the generation quality is good for all shapes except the triangle. In Figure 4 we showed that TSB had problems generating the triangle shape as well. This might be because of having two different types of triangles in the dataset, filled triangles and empty triangles.

6.5. Memorization test

There is no quantitative measure of memorization in generative models, thus we check this via *intra-class interpolation*, *class interpolation* and k-NN retrieval. In *intra-class interpolation* we linearly interpolate between to two different latent codes Z_{ABC} while keeping the label fixed, as shown in Figure 8. In the Figure 9, we explore *class interpolation* by linearly interpolating between label embeddings, while keeping Z_{ABC} fixed. Figures 8 and 9 show smooth transition between modes and classes. If a model would suffer

(a) NT.



(b) TSB.

Figure 8: Example of intra-class interpolation on UCF-101. The vertical axis represents time, the horizontal axis represents different modes of the class. We sample two latent codes which are represented by the leftmost and right most samples and linearly interpolate between them to generate intermediate latent code samples.

from memorization, we would expect the interpolation to abruptly jump from mode to mode in *intra-class interpolation* and from label to label in *class interpolation*. Samples from the retrieval experiment (Figure 10), show that generated samples are noticeable dissimilar to real samples, this suggests that the model does not suffer from memorization. The k-NN experiment was done using the last layer of the ECO [33] architecture.

7. Conclusion

We presented a TSB architecture for video generation that enforces temporal consistency into a 2D video generator network. We show that TSB design improves the quality of generated videos in comparison to the BigGAN baseline. To validate effectiveness of our method, we conduct experiments on four different datasets including our new dataset MaisToy. Our new dataset enables us to analyze the generalization power of model and also understand which semantics are easier for model to learn. As a supplement to the well established IS score, we proposed the generalization based S3 score, which is intended to be sensitive to intra-class variation. Based on this metric our method also achieves the best performance. These quantitative results are further supported by our qualitative.

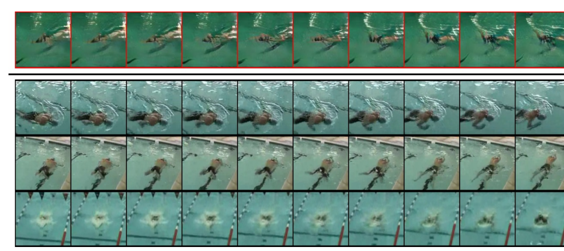
8. Acknowledgements

We acknowledge funding by the German Research Foundation (DFG) via the grant BR 3815/10-1 and the gift of a compute server by Facebook.

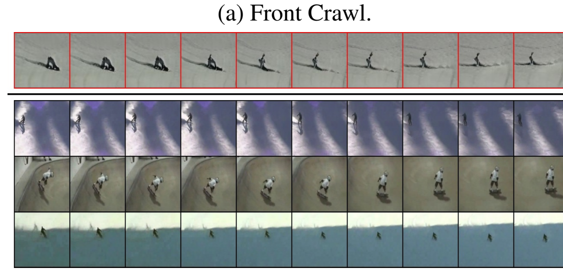
(a) NT.

(b) TSB.

Figure 9: Interpolation Performance showing smooth transitions between classes on UCF-101. Each column is a sequence.(a) The top figure is interpolating between classes writing on board (left) and pole vault (right), while the bottom one is interpolating volleyball spiking (left) and frisbee catch (right). (b) The top figure is interpolating between classes basketball (left) and frisbee catch (right), while the bottom one is interpolating between golf swing (left) and diving (right).



(a) Front Crawl.



(b) Skiing.

Figure 10: Examples of retrieval of top-3 nearest neighbors (black) of TSB generated samples (red). We can see that although the generated samples look similar to their respective 3-NNs, they are still quite visually distinct. This implies the model isn't just memorizing the real data.

References

[1] G. Toderici R. Sukthankar T. Le-ung A. Karpathy, S. Shetty and Li Fei-Fei. Large-scale video classification with convolutional neural networks. *CVPR*, 2014.

[2] Jeff Donahue Aidan Clark and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv:1907.06571*, 2019.

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

[4] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv:1801.01973*, 2018.

[5] Ali Borji. Pros and cons of GAN evaluation measures. *CoRR*, abs/1802.03446, 2018.

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

[7] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017.

[8] Ian Goodfellow Chelsea Finn and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *NeurIPS*, 2016.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

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

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

[12] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *CoRR*, abs/1907.02544, 2019.

[13] Rob Fergus Lorenzo Torresani Du Tran, Lubomir Bourdev and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. *ICCV*, 2015.

[14] Rajesh Ranganath Dustin Tran and David M. Blei. Hierarchical implicit models and likelihood-free variational inference. *NeurIPS*, 2017.

[15] C. Han G. Kwon and D. Kim. Generation of 3d brain mri using auto-encoding generative adversarial networks. 2019.

[16] Dimitris Metaxas Han Zhang, Ian Goodfellow and Augustus Odena. Self-attention generative adversarial networks. *PMLR*, 2019.

[17] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555, 2018.

[18] DM. Hawkins. The problem of overfitting.

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

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

[21] Chuang Gan Ji Lin and Song Han. Tsm: Temporal shift module for efficient video understanding. *ICCV*, 2019.

[22] Ingo Bax Joanna Materzynska, Guillaume Berger and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. *ICCV*, 2019.

[23] Honglak Lee Richard Lewis Junhyuk Oh, Xiaoxiao Guo and Satinder Singh. Action-conditional video prediction using deep networks in atari games. *NeurIPS*, 2015.

[24] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. *CVPR*, 2016.

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

[26] E. Shechtman M. Irani L. Gorelick, M. Blank and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern-Analysis and Machine Intelligence (TPAMI)*, 2007.

[27] J. H. Lim and J. C. Ye. Geometric gan. *arXiv:1705.02894*, 2017.

[28] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. 11 2017.

[29] Thomas Unterthiner Bernhard Nessler Martin Heusel, Hubert Ramsauer and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NIPS*, 2017.

[30] andShunta Saito. Masaki Saito, Eiichi Matsumoto. Temporal generative adversarial nets with singular value clipping. *ICCV*, 2017.

[31] Camille Couprie Michael Mathieu and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *ICLR*, 2016.

[32] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *ICLR*, 2018.

[33] Kamaljeet Singh Mohammadreza Zolfaghari and Thomas Brox. Eco: Efficient convolutional network for online video understanding. *ECCV*, 2018.

[34] Karen Simonyan Oriol Vinyal Alex Gravesa Nal Kalchbrenner, Aaaron van den Oord and Koray Kavukcuoglu. Video pixel networks. *ICML*, 2017.

[35] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models, 2019.

[36] Masaki Saito and Shunta Saito. Tganv2: Efficient training of large models for video generation with multiple subsampling layers. *arXiv:1811.09245*, 2018.

[37] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016.

[38] Xiaodong Yang Sergey Tulyakov, Ming-Yu Liu and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *CVPR*, 2018.

[39] Ravi Kiran Sarvadevabhatla Swaminathan Gurumurthy and R. Venkatesh Babu. Deligan: Generative adversarial networks for diverse and limited data. *CVPR*, 2017.

[40] Zaremba W. Sutskever I. Bruna J. Erhan D. Goodfellow I. J. Szegedy, C. and R. Fergus. Intriguing propertiesof neural networks. *ICLR*, 2014.

[41] Masanori Koyama Yuichi Yoshida Takeru Miyato, Toshiki Kataoka. Spectral normalization for generative adversarial networks. *ICLR*, 2018.

[42] Samuli Laine Tero Karras and Timo Aila. A style-based generator architecture for generative adversarial networks. *CVPR*, 2019.

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

[44] Katherine L. Bouman Tianfan Xue, Jiajun Wu and William T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *NeurIPS*, 2016.

[45] Wojciech Zaremba Vicki Cheung Alec Radford Tim Salimans, Ian Goodfellow and Xi Chen. Improved techniques for training gans. *NIPS*, 2016.

[46] Nikita Araslanov Vladyslav Yushchenko and Stefan Roth. Markov decision process for video generation. *ICCV*, 2019.

[47] Xiong Y. Wang Z. Qiao Y. Lin D. Tang X. Gool L.V. Wang, L. Temporal segment networks for action recognition in video. *TPAMI*, 41(11):2740 – 2755, 2019.

[48] Dongze Lian Wen Liu, Weixin Luo and Shenghua Gao. Future frame prediction for anomaly detection – a new baseline. *CVPR*, 2018.

[49] Rein Houthooft John Schulman Ilya Sutskever Xi Chen, Yan Duan and Pieter Abbeel. Infogan: Interpretable representation learning byinformation maximizing generative adversarial nets. *NeurIPS*, 2016.

[50] Seyed shahabeddin Nabavi Yiwei Lu, Mahesh Kumar Krishna Reddy and Yang Wang. Future frame prediction using convolutional vrnn for anomaly detection. *AVSS*, 2019.