Temporal Shift GAN for Large Scale Video Generation –Supplementary Material–

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

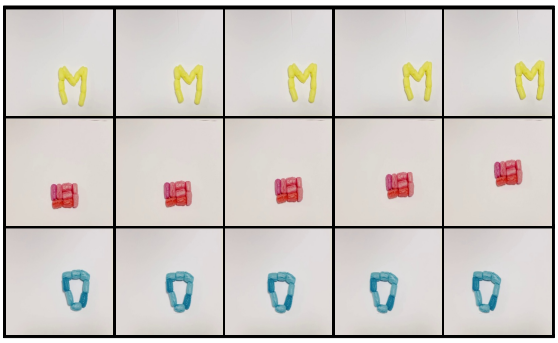
1. Additional Experiments

1.1. Generalization test

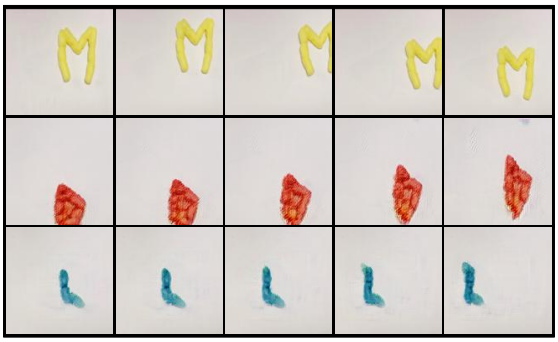
In order to evaluate the generalization of model, we held out the following label combinations when training on $\text{MaisToy}_{\text{Multi}}$ dataset:

- Square, red, right
- Square, red, left
- Square, red, upwards
- Square, red, downwards
- Triangle, blue, right
- Triangle, blue, left
- Triangle, blue, upwards
- Triangle, blue, downwards
- Letter M, yellow, right
- Letter M, green, right
- Letter M, blue, right
- Letter M, red, right

Then, we used the network to generate all the unseen combinations above to corroborate if it is actually learning the meaning of each label. Qualitative results in Figure 1 show that the network is, for the most part, able to generalize to unseen combinations. However, it appears that the triangle shape is difficult for the network.



(a) Real sequence.



(b) Generated sequence.

Figure 1: (a) Samples from held combinations (from top to bottom): Letter M, yellow, right; Square, red, upwards; Triangle, blue, left. (b) Generated sequences for the same combinations presented in (a).

1.2. S3

To calculate S3 we train a classifier on real samples and on fake samples as explained on Section 5. We calculated the S3 metric, for UCF-101 [9], based on two different classifiers, a TSN [15] action recognition network and a 3D ResNet-18 [8]. The TSN was trained on a batch size of 14 and an initial learning rate of 0.01. We trained the 3D ResNet-18 using a batch size of 32, an initial learning rate of 0.001 and a dropout probability of 0.6. To get the S3 score for Jester [5]

*Equal Contribution

Classifier Architecture	Method	Train on: Synth.	Real		S3
		eval. on: Real	Synth.	Real	
TSN	NT	45.5	46.8	85.9	0.39
	TSB	48.55	54.91	85.9	0.45
3D ResNet18	NT	36.63	28.83	76.82	0.29
	TSB	44.36	29.61	76.82	0.36

Table 1: UCF-101 results of S3 on two different architectures.

Method	Dataset	FID
NT	UCF-101	**3108.77 ± 0.04**
TSB	UCF-101	3110.29 ± 0.10
	Jester	841.08 ± 0.005

Table 2: FID scores on Jester and UCF-101.

we decided to use the ECO [13] action recognition network. We trained it using a batch size of 14, an initial learning rate if 0.001 and a dropout probability of 0.6. All networks were trained on 16 frames and their respective learning rates were scheduled to drop by an order of magnitude after failing to beat the best recorded test accuracy for 4 straight epochs.

We calculated the S3 with two different architectures for the UCF-101 dataset to provide a reference for the comparisons in the future works. Table 1 shows that the change of architecture does alter the relative performances of ReS and SeR to ReR significantly enough to produce important changes in the score. Therefore, S3 scores obtained from different classification architectures does not provide a fair comparison.

1.3. FID

FID [4] calculations were done using the features from the second-to-last layer of a TSN pretrained on Imagenet [2] and finetuned on the respective dataset it is going to be tested on. The network was trained as explained above. We calculated FID using 4000 samples, we repeated the process 5 times to get the standard deviation. Table 2 seems to suggest NT is better than TSB, but this could be due to the fact that FID cannot separate image quality from diversity. If we take into account IS and S3 we can deduce that although FID points to NT being better than TSB this is most likely due to better sample diversity, not sample quality.

1.4. Motion Constraint

Prior work has used optical flow to generate videos by warping the images [11, 16] or using it as a prior to generate spatial features [7, 6]. We rather introduce a intra-class constraint on similarity between optical flows produced by real videos and by generated videos. An illustration is provided in Figure 2.

To estimate this constraint, first we generate synthetic samples and sample real videos from the dataset. Optical

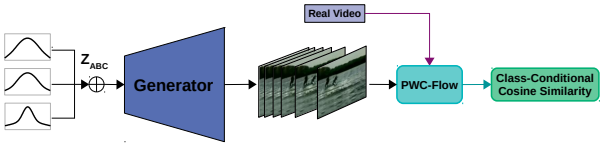


Figure 2: Illustration of the motion constraint calculation.

flow is calculated using the PWC Flow network [3]. Moreover, we calculate the cosine similarity between flows resulted from real videos and flows from generated videos. We only do this for pairs of real and synthetic videos with matching labels:

$$L_M = \frac{1}{C} \Sigma_i^B \Sigma_j^B \text{Sim}(f_{r_j}, f_{f_i}) = \begin{cases} \frac{f_{r_j} \cdot f_{f_i}}{||f_{r_j}|| \cdot ||f_{f_i}||} & \text{if } y_{f_i} = y_{r_i} \\ \varnothing & \text{otherwise} \end{cases} \quad (1)$$

where f_r and f_f stand for real and generated flows, respectively, B is batch size and C is the number of matching real and generated flow pairs. This similarity measure enforces the similarity of motion between samples from the same class. Finally, we add the constraint only to the generator loss:

$$L_{G_M} = L_G + (\alpha \cdot (1 - L_M)) \quad (2)$$

where α is a hyperparameter that controls the importance of the motion constraint L_M.

This architecture was dubbed NT-MC, although it did score better than the baseline with an S3 score of 0.73 on the Weizmann dataset, it fell short of NT-VAR and TSB. Among some other disadvantages of this motion constraint is the fact that it makes training significantly slower and unstable.

1.5. Ablation studies

Our TSB trained on Jester did not record a good performance on the S3 measure, hence we need a qualitative evaluation to look for a possible reason why this was the case. Figure 3 shows an acceptable level of quality in both spatial and motion feature generation. However, TSB still was not able to produce realistic enough samples in fine structures of hands and faces as a real person would have.

Impact of latent codes. We wanted to know if in fact using a multi-variate model for the latent codes had any effect on what the network learned. Specifically we wanted to see if assigning a different variance to each subspace had any effect on the features the network learned to map to each one of the subspaces. To test this we froze two out of the three subspaces and re-sampled the remaining one to produce a new sample. Every subspace will get a turn at being re-sampled. Figure 4 shows some examples

of this experiment compared to a sample produced by the originally sampled latent vector. The samples show that the network learns to assign Z_C features that result in bigger changes in the overall visual features, like gender. We can observe as well that Z_B appears to be in charge more of motion features, without affecting features such as location or person identity as much. It appears that Z_A is in charge of more infrequent features like location or small changes in appearance. This experiment points towards the variance assigned to a subspace being directly related the types of features it represents.

Ablation study on different model designs. Figure 5 shows the improvement between models in different classes of the Weizmann [10] dataset.

Figure 3: Generated samples of TSB trained to produce 192×192 samples of the Jester dataset.

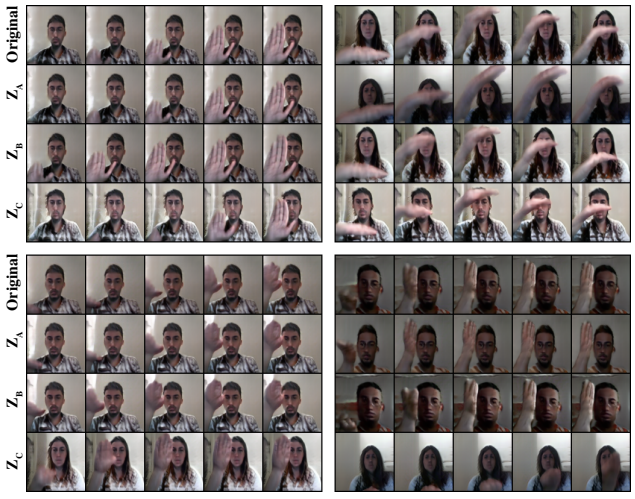


Figure 4: Latent variable experiment. We freeze two out the three subspaces and re-sample the remaining one to produce a new sample. We compare each sample to the original to see what meaning is the network assigning to that specific subspace.

2. Architectural Details

We adopted most of BigGAN's [1] architectural choices in G_{Image}, with the exception that we moved the self-attention module down one level of abstraction to save video memory. D_{Image} follows exactly the discriminator guidelines set in BigGAN, while D_{Video} adopted the exact architecture used in MoCoGAN [14], but extended for class conditional hinge loss per [12]. To describe the width of all networks we use the product of a layer-wise constant c and a per-layer constant a. In all experiments a was set to 96. We chose c to be $[16, 8, 4, 2, 1]$ for G_{Image}, $[1, 2, 4, 8, 16, 16]$ for D_{Image} and $[1, 2, 4, 8]$ on D_{Video}.

At the input of G_{Image} we have a fully connected layer which applies an affine transformation to Z_F to transform it from $[T, d + 120]$ to $[T, w \cdot h \cdot 16 \cdot a]$. When generating 96×96 sized samples we set w and h to 3 and when we generated samples of size 128×128 they were both set to 4.

The sequence generator is composed of a fully connected layer FC and a GRU cell. FC has a size of d and the GRU cell has a size of 2048.

Figure 5: Generated samples (from top to bottom) of NT , NT-MC, NT-VAR trained on Weizmann.

References

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

[2] 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.

[3] Ming-Yu Liu Deqing Sun, Xiaodong Yang and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and

cost volume. *CVPR*, 2018.

[4] 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.

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

[6] Xu Jia Jing Shao Lu Sheng Junjie Yan Xiaogang Wang. Junting Pan, Chengyu Wang. Video generation from single semantic label map. *CVPR*, 2019.

[7] Yoshitaka Ushiku Katsunori Ohnishi, Shohei Yamamoto and Tatsuya Harada. Hierarchical video generation from orthogonal information: Optical flow and texture. *AAAI*, 2017.

[8] Hirokatsu Kataoka Kensho Hara and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. *ICCV*, 2017.

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

[10] 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.

[11] Jiaming Guo Jing Shao Xiaogang Wang Chen Change Loy. Lu Sheng, Junting Pan. Unsupervised bi-directional flow-based video generation from one snapshot. *arXiv:1903.00913*, 2019.

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

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

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

[15] 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.

[16] X Huang Z Hao and S Belongie. Controllable video generation with sparse trajectories. *CVPR*, 2018.