StyleInV: A Temporal Style Modulated Inversion Network for Unconditional Video Generation

Supplementary Materials

Yuhan Wang, Liming Jiang, Chen Change Loy
S-Lab, Nanyang Technological University
{yuhan004, liming002, ccloy}@ntu.edu.sg

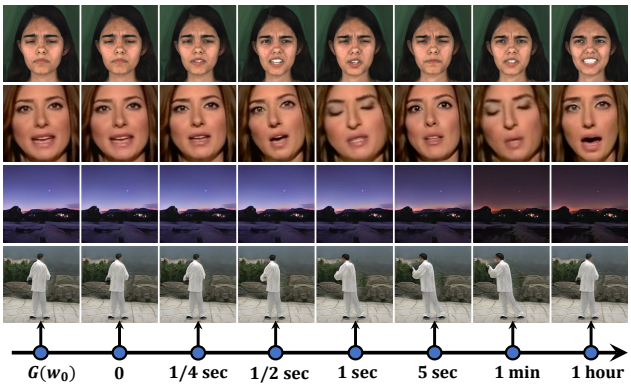


Figure 1. Our StyleInV can generate arbitrarily long videos with long-lasting content consistency.

Abstract

This document provides supplementary information that is not elaborated in our main paper. In Section A, we present some extra properties of our method that are not elaborated on in the main paper. In Section B, we discuss some limitations of our method and the broader impacts. In Section C, we compare the computational cost of the baselines and our model. In Section D, we introduce the different cropping strategies we applied to the DeeperForensics and FaceForensics datasets. In Section E, we show the effect of noise injection in StyleGAN models for video generation on different datasets. In Section F, we list the details of our model architecture and training setting. In Section G, we give a brief introduction to each dataset we use.

A. Other Properties

Here we provide examples of other intriguing properties that our method has.

Long video generation. Similar to [17], our network can also generate arbitrarily long videos with decent quality. The result is shown in Fig. 1 by extending the input timestamps to as large as one hour. Notably, our method can well preserve the content consistency of the generated videos without the motion collapse effect. Video examples are provided in the supplementary video and additional samples.

Temporal interpolation. Our method also supports temporal interpolation to arbitrarily increase the frame rate of generated videos. Fig. 2 shows the result of increasing the FPS of a video from 30 to 60, by doubling the density of timestamp sampling. More specifically, for a 128-frame, 30-FPS video, we input

$$t = 0, 1, 2, \cdots, 127$$

to the StyleInV network, via Eq. (2) in the main paper. To increase the FPS to 60, we only need to input

$$t = 0, 0.5, 1, 1.5, 2, \cdots, 126.5, 127, 127.5$$

, and our model can generate smooth interpolations.

B. Limitations and Broader Impacts

B.1. Limitations

Inferior motion semantics on SkyTimelapse. As is mentioned in the main paper Section 4.1, the motion semantics of our generated videos on SkyTimelapse are inferior to those generated on other datasets. The reason of this may be that the characteristics of the dataset are different.

For DeeperForensics [5], FaceForensics [13], and TaiChi [16], the first frame largely determines the content of all frames in a video, and a video is composed of the animation process of the subject. This is consistent with the characteristics of the inversion encoder's focus on the subject. But for SkyTimelapse, two frames that are far apart often have little relation in content and the video is driven by global motions. As our network is conditioned on the first frame and predicts residuals w.r.t. the initial latent, the sky

Figure 2. Temporal interpolation. All the frames with red borders form a 128-frame, 30FPS video (∼4.3 seconds). The frames without borders are the interpolated ones that increase the FPS to 60 (still ∼4.3 seconds). View the first row first from left to right, then view the second row from left to right, then the third row, and so on.

videos generated by StyleInV conform to our model nature. Please refer to the supplementary videos for visual results.

This nature makes our model outstanding in identity preservation and can be better applied to applications like animation. Addressing more dynamics and global motions is an interesting improvement and future work for StyleInV.

The impact of dataset identity richness. The second limitation of our model is that, when the identity scale of the face video dataset is too small, it is difficult for us to fully inherit all the excellent properties of an FFHQ pre-trained StyleGAN2. This is why we develop our style transfer model on a recently released large-scale face video dataset CelebV-HQ [22], as it has identity diversity on the same scale as FFHQ. Our video generation performance on CelebV-HQ demonstrates the ability of our model to generalize to larger face video datasets.

Image generation quality. The third limitation is that the generation quality of StyleGAN determines the performance upper bound of our method. In this work, the images generated by the StyleGAN2 models trained on SkyTimelapse and TaiChi [16] have certain artifacts in the background. Especially for the TaiChi dataset, although our

Table 1. GPU memory consumption of different methods for one video to be added into the batch. "A" means autoregressive while "N-A" means non-autoregressive. "pSG" means employing a pretrained StyleGAN2. "mp" stands for mixed precision. "FpV" stands for frames per video. "MpV" stands for memory per video, reported in GB. "GPU Days" shows the total training time, aligned on V100 GPU.

Method	Type	pSG	mp	FpV	MpV	GPU Days
MoCoGAN-HD	A	✓		16	5.37	$(7.5+9)\times 2$
MoCoGAN-HD	A	✓		32	11.37	$(7.5+18)\times 2$
DIGAN	N-A			2	1.32	16
StyleGAN-V	N-A		✓	3	1.20	8
Long-Video-GAN	N-A		✓	-	-	$16\uparrow+16\uparrow$
StyleInV	N-A	✓	✓	4	2.85	$7.5+1+9$

approach has greatly surpassed state-of-the-art methods in terms of quantitative metrics, the visual quality can be further improved. The generated background and human body both lack fine details and a sense of structure. That is to say, for video generation on non-face video datasets, it remains improvement space to develop a high-quality image generator.

Model training. Finally, our approach is two-stage and thus requires more training time compared to StyleGAN-V. Our method requires 7.5 and 9 GPU days for each stage, respectively, while StyleGAN-V is one stage and only requires 8 GPU days to train. Despite this, when finetuning hyperparameters on a dataset, our StyleInV is actually as efficient as StyleGAN-V, because the image generator only needs to be trained once and can be used for all StyleInV networks. The two stages of our method are well separated. Besides, our method has some unique properties, such as finetuning-based style transfer.

B.2. Broader Impacts

We believe that the potential of StyleInV can be further exploited. Our method can provide a natural solution towards mega-pixel level video generation and StyleGAN-based editing, and it might in return promote the research of learning-based GAN inversion methods.

As for the negative side, StyleInV may ease the synthesis of better-quality fake videos that might have potential threats. We believe that this issue can be alleviated by developing more advanced falsified media detection methods or contributing larger-scale and higher-quality forgery detection datasets.

C. Computational Cost

The advantage of our method in computational cost over autoregressive approaches is mainly reflected in the GPU memory consumption during training. Table 1 shows the comparison result. Our approach is the only non-autoregressive method that employs a pretrained StyleGAN generator. Our FpV is fixed and thus StyleInV can be trained on arbitrarily long videos.

For the autoregressive MoCoGAN-HD, its memory consumption for one video in the batch is proportional to the clip length, making it difficult to be trained on long videos. Meanwhile, its codebase is ≈ 2 times slower than ours as it does not support mixed precision training.

Compared to other non-autoregressive methods, our network consumes a bit more memory due to an extra encoder network and the initial frame included in sparse training.

For Long-Video-GAN, its model is split into two parts, each of which requires finely setting the clip length according to the output resolution. It is also the most expensive model to train. Following its default setting, it takes 64 GPU Days to train the low-resolution model and 32 GPU days to train the high-resolution model. Due to the limitation of computing resources, we can only reduce the batch size to have each part trained in 16 GPU days, with negligible performance degradation.

D. Cropping Strategies

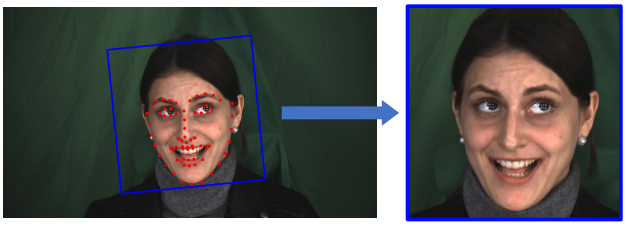
In this section, we introduce the cropping strategies of the FaceForensics dataset and the DeeperForensics dataset, then explain the difference between them.

Algorithm 1 FaceForensics dataset cropping.

Input: $x_{min}, y_{min}, x_{max}, y_{max}$
Output: $\hat{x}_{min}, \hat{y}_{min}, \hat{x}_{max}, \hat{y}_{max}$

$w = x_{max} - x_{min}$
$h = y_{max} - y_{min}$
if $w < h$ **then**
  $\Delta = h - w$
  $\hat{x}_{min} = x_{min} - \Delta/2$
  $\hat{x}_{max} = x_{max} + \Delta/2$
  $\hat{y}_{min} = y_{min}$
  $\hat{y}_{max} = y_{max}$
else
  $\Delta = w - h$
  $\hat{x}_{min} = x_{min}$
  $\hat{x}_{max} = x_{max}$
  $\hat{y}_{min} = y_{min} - \Delta/2$
  $\hat{y}_{max} = y_{max} + \Delta/2$
end if

FaceForensics cropping. The FaceForensics [13] dataset is composed of news broadcasting videos. Apart from raw videos, it also releases labeled face masks for each frame. TGAN-V2 [14] proposes to crop the dataset based on these masks. For each frame, it first computes the minimum and maximum values of the coordinates of the face region to get, $x_{min}, y_{min}, x_{max}, y_{max}$. Then this rectangle region is padded to be a square, as is stated in Algorithm 1. Finally,



(a) raw frame (b) FFHQ cropped

Figure 3. FFHQ cropping strategy on DeeperForensics dataset. The landmarks are detected.

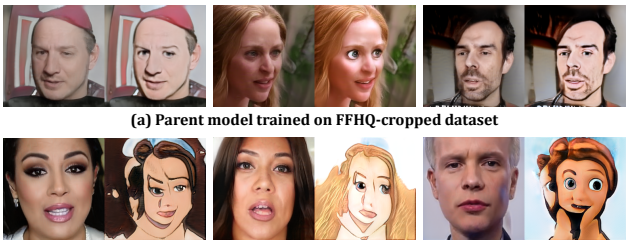
the selected square region is cropped and resized to the target resolution to become the cropped frame. This pipeline is followed by all recent works [18, 17]. We also apply it for the FaceForensics dataset pre-processing.

DeeperForensics cropping. The DeeperForensics [5] dataset is composed of humans expressing given emotions. As this dataset does not release the labeled face masks, we turn to the unsupervised cropping strategy applied in FFHQ dataset [7]. The cropping pipeline is shown in Fig. 3, where the square region is determined by the detected landmarks, then the square region is resized to the target resolution.

As this cropping strategy is based on the detected landmarks, the stability of the landmark detection will greatly affect the stability of the cropped videos. In the implementation, if each frame is simply detected by a landmark detector and cropped, the cropped video will shake violently. We first replace the landmark detector with a state-of-the-art RetinaFace [3], then follow a *stabilizing approach* proposed by [10]. We find that the *stabilizing approach* significantly reduces the shaking effect. Here we briefly describe it.

The state-of-the-art landmark detectors input a bounding box of the detected face and output the landmarks. We shift the bounding box at a random distance and a random angle multiple times. Then we use these bounding boxes to detect the landmarks and average the results. This approach statistically reduces the variance of the detected landmarks.

Difference. The FFHQ cropping strategy aligns the human facial features in a fixed position. This property improves the effect of finetuning-based style transfer. As the common datasets adopted for style transfer (*e.g.*, Cartoon [11] and Metfaces [6]) are also aligned by the FFHQ cropping strategy, when the datasets are well aligned in structure, the finetuning process can more naturally adjust the weights of high-resolution layers upon fixed low-resolution layers. Fig. 4 compares the finetuning-based style transfer result of the parent model trained on CelebV-HQ [22] (where we also apply the stabilized FFHQ cropping) and FaceForensics. When the parent model is trained on a dataset (*e.g.*, FaceForensics) which does not share the alignment of finetuning dataset (*e.g.*, Cartoon), the style transfer fails due to the structure collapse.



(a) Parent model trained on FFHQ-cropped dataset

(b) Parent model trained on non-FFHQ-cropped dataset

Figure 4. When the parent model is trained on an FFHQ-cropped dataset (*e.g.*, CelebV-HQ), finetuning-based style transfer produces promising results. Otherwise, a severe structure collapse occurs.

E. Effect of Noise Injection

StyleGAN series [7, 8, 6] proposes to inject noise vectors at all layers of the generator for finer details in the background, hair, skin, etc. As reported by [7], the omission of noise will lead to a "featureless painterly look". However, though designed on top of StyleGAN2, StyleGAN-V [17] turns off the noise injection by default for training and inference on all datasets. It also makes sense as the totally randomized noise will bring content inconsistency among frames.

In this work, we find that the effect of noise injection in our system can be different on different datasets, positive or negative. We first investigate its effect on the image generator in terms of the FID metric. On FaceForensics and TaiChi datasets, the FID results of models with or without noise are close. But on the SkyTimelapse dataset, the model without noise injection has a much better FID result.

Then we look into how the noise in the StyleGAN2 generator affects the video generation quality. The first intuitive observation is that we should apply **constant noise** for all frames when synthesizing a video, instead of injecting random noises for different frames. This is to avoid content inconsistency. Then we compare the results of StyleInV networks with or without noise. The results are exactly the opposite for the first two datasets and the third dataset. On FaceForensics and TaiChi datasets, injecting constant noise improves the FVD results significantly, while on the SkyTimelapse dataset, the model without noise gives a much better result.

We deduce that this is because there is no distinction between subject and background on the SkyTimelapse dataset, making it difficult to clarify the way the injected noise works. While on a dataset with clear subjects and backgrounds, the injected noise effectively handles the generation of stochastic aspects, leaving the latent space focusing on synthesizing the subject, which helps our StyleInV encoder find meaningful trajectories in the latent space.

Table 2. FID results of StyleGAN2 generator with or without noise injection.

Method	FaceForensics	TaiChi	SkyTimelapse
with noise	10.19	**38.1**	15.05
w/o noise	**9.52**	38.37	**11.80**

Table 3. FVD results of StyleInV video generator with or without noise injection in its StyleGAN2 image generator.

Method	FaceForensics		TaiChi		SkyTimelapse	
	FVD_{16}	FVD_{128}	FVD_{16}	FVD_{128}	FVD_{16}	FVD_{128}
with noise	**47.88**	**103.63**	**185.72**	**328.90**	115.68	266.67
w/o noise	106.42	238.93	326.60	583.60	**77.04**	**194.25**

F. Implementation Details

In this section, we discuss the training of baselines and our model, the architecture parameters, and the detailed training setting.

Baseline details. MoCoGAN-HD [18] designs motion generators for a pretrained StyleGAN2 as we do. DIGAN [20] and StyleGAN-V [17] train the entire framework as a whole in a non-autoregressive manner. Long-Video-GAN [1] is split into a low-resolution stage and a high-resolution stage.

All baselines are trained on 4 NVIDIA Tesla A100 GPUs. The StyleGAN2 generator for MoCoGAN-HD is pretrained with all frames of the video dataset. Then the motion generator is trained for 100 epochs following its default setting. DIGAN models are trained under its default config for approximately four days. All StyleGAN-V models are trained under its paper setting except on DeeperForensics dataset, for which we need to increase the R1 γ parameter by 10 times to avoid training collapse.

Development and training. Our StyleInV is built upon the official `PyTorch` implementation of StyleGAN2-ADA [6], with which we enable the mixed precision setting for StyleGAN2 and significantly speed up the training. The StyleGAN2 image generator is firstly trained on all frames of the video dataset with class-aware sampling [15, 21]. The noise injection is turned off for SkyTimelapse dataset only. Then we train an inversion encoder based on Fig. 3 and Eq. (1) to initialize the convolution layers of the StyleInV encoder. Finally, the entire StyleInV model is trained under the objective of Eq. (6). Three steps take roughly 7.5, 1, and 9 GPU days, respectively. All StyleInV models are trained on 8 NVIDIA Tesla A100 GPUs. We apply an unbalanced learning rate setting for the Adam optimizer [9], where the learning rate for the StyleInV encoder and the discriminator is 0.0001 and 0.002, respectively.

Model details. For the computation of temporal styles, the sampled temporal noise for each timestamp is a 512-dimensional vector. FFA-APE consists of two left-sided 1D-convolution layers with kernel size 6 and padding 5. The length of the vector sequence remains unchanged after each 1D-convolution layer. The learnable interpolation part is identical to that of StyleGAN-V [17]. The dimension of positional encoding v_t is 512. It is concatenated with the initial frame latent w_0 and goes through two fully connected layers to output the final temporal style, whose dimension is also 512.

For the modulated inversion encoder, its convolution blocks are identical to those in pSp inversion encoder [12], which compose a ResNet-50 backbone [4]. The `AdaIN` layers are adopted from StarGAN-V2 [2], with residual connection and variance normalization enabled. The `AdaIN` layers do not down-sample the feature maps. A fully connected layer is appended after the last adaptive average pooling layer to output a 512-dimensional vector, which is the residual w.r.t. w_0 by definition.

For the discriminator design, we simply follow the model architecture of the StyleGAN-V discriminator. We did not delve into this part. The first frame used in the discriminator is $G(w_0)$, instead of $G(\text{StyleInV}(w_0, 0))$

Training details. For hyper-parameters of FFA-ST, we set $\lambda_{L_2} = 10$ and $\lambda_{reg} = 0.05$ for all four datasets. We apply adaptive differentiable augmentation [6], where the augmentation operation is always identical for all frames in a video. We use the `bgc` augmentation pipe. The augmentation target is 0.6. The R1 γ parameter for $r1$ regularization is 1. The learning rate for the modulated inversion encoder is 0.0001. The learning rate for the discriminator is 0.002.

For the inversion encoder training which is used for weight initialization, we follow all the training settings described in the pSp paper [12], except that the ID loss is turned off for TaiChi and SkyTimelapse datasets.

For the finetuning-based style transfer, we fix the mapping network and synthesis layers whose resolution is no larger than 32. The training setting is identical to that of the parent model. The finetuning process takes only 4-8 GPU hours.

G. Dataset Details

We provide dataset details in this section.

DeeperForensics [5]. This dataset is composed of 100 identities expressing eight emotions (angry, contempt, disgust, fear, happy, neutral, sad, and surprise). The videos are collected under nine lighting conditions and seven camera positions, among which we only select the condition where the lighting is uniform and the camera shoots from the straight front. All videos are cropped to 256 resolution following the stabilized FFHQ cropping strategy which is described in Section D. The entire dataset has 732 videos of 194,770 frames.

FaceForensics [13]. We follow the same cropping strategy of StyleGAN-V to process and organize the dataset. The entire dataset has 704 videos of 364,017 frames.

SkyTimelapse [19]. StyleGAN-V releases its SkyTimelapse 256^2 dataset [1]. We directly use it for our experiments. The entire dataset has 2,114 videos of 1,168,920 frames. Notably, some videos in SkyTimelapse are hours long. We use class-aware sampling in both training and metric calculation, following StyleGAN-V.

TaiChi [16]. We follow the link [2] provided by DIGAN to download and crop the dataset. The original dataset resolution after processing is 256, so we directly use it for all experiments. Notably, some of the video links had expired when we were processing this dataset, thus the composition of our dataset may be slightly different from previous work. The entire dataset has 3,103 videos of 951,533 frames.

CelebV-HQ [22]. We download the video dataset using the link for processed CelebV-HQ videos [3] and crop the dataset to 256 resolution with stabilized FFHQ cropping. The entire dataset has 35663 videos.

References

[1] Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei A Efros, and Tero Karras. Generating long videos of dynamic scenes. In *NeurIPS*, 2022. 5

[2] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020. 5

[3] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. RetinaFace: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020. 4

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *ICCV*, 2016. 5

[5] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. DeeperForensics-1.0: A large-scale dataset for real-world face forgery detection. In *CVPR*, 2020. 1, 4, 5

[6] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020. 4, 5

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 4

[8] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 4

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[10] Jacek Naruniec, Leonhard Helminger, Christopher Schroers, and Romann M Weber. High-resolution neural face swapping for visual effects. In *CGF*, 2020. 4

[11] Justin NM Pinkney and Doron Adler. Resolution dependent GAN interpolation for controllable image synthesis between domains. *arXiv preprint*, arXiv:2010.05334, 2020. 4

[12] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a StyleGAN encoder for image-to-image translation. In *CVPR*, 2021. 5

[13] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint*, arXiv:1803.09179, 2018. 1, 3, 6

[14] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal GAN. *IJCV*, 128:2586–2606, 2020. 3

[15] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, 2016. 5

[16] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019. 1, 2, 6

[17] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. StyleGAN-V: A continuous video generator with the price, image quality and perks of StyleGAN2. In *CVPR*, 2022. 1, 4, 5

[18] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *ICLR*, 2021. 4, 5

[19] Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *CVPR*, 2018. 6

[20] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *ICLR*, 2022. 5

[21] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, 2020. 5

[22] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. CelebV-HQ: A large-scale video facial attributes dataset. In *ECCV*, 2022. 2, 4, 6

[1] https://disk.yandex.ru/d/7JU3c5mdWQfrHw

[2] https://github.com/AliaksandrSiarohin/first-order-model

[3] https://github.com/CelebV-HQ/CelebV-HQ/issues/8