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

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

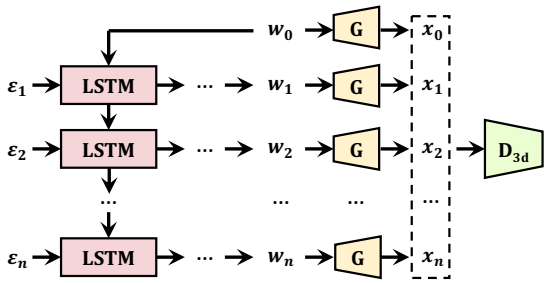
Abstract

Unconditional video generation is a challenging task that involves synthesizing high-quality videos that are both coherent and of extended duration. To address this challenge, researchers have used pretrained StyleGAN image generators for high-quality frame synthesis and focused on motion generator design. The motion generator is trained in an autoregressive manner using heavy 3D convolutional discriminators to ensure motion coherence during video generation. In this paper, we introduce a novel motion generator design that uses a learning-based inversion network for GAN. The encoder in our method captures rich and smooth priors from encoding images to latents, and given the latent of an initially generated frame as guidance, our method can generate smooth future latent by modulating the inversion encoder temporally. Our method enjoys the advantage of sparse training and naturally constrains the generation space of our motion generator with the inversion network guided by the initial frame, eliminating the need for heavy discriminators. Moreover, our method supports style transfer with simple fine-tuning when the encoder is paired with a pretrained StyleGAN generator. Extensive experiments conducted on various benchmarks demonstrate the superiority of our method in generating long and high-resolution videos with decent single-frame quality and temporal consistency. Code is available at https://github.com/johannwyh/StyleInV.

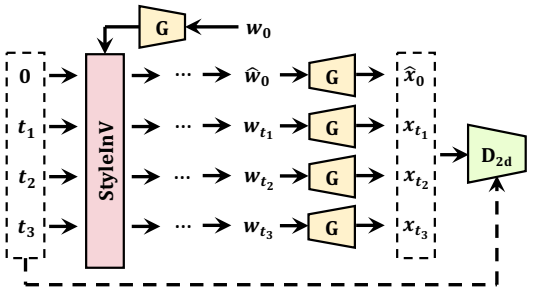
1. Introduction

Unconditional video generation aims at learning a generative model to create novel videos from latent vectors. Despite extensive studies [47, 36, 37, 43, 12, 55] in addressing this problem, it remains challenging to generate high-resolution videos with both favorable quality and motion coherence over a long-term duration. The core difficulties in this task lie in modeling consistent motion and managing the high memory consumption introduced by the addition of the temporal dimension.

To ensure high single-frame resolution and quality, many existing studies, such as MoCoGAN-HD [41], employ a powerful image generator such as StyleGAN [25] as a backbone to serve as a strong generative prior. This approach shifts the focus towards developing a robust motion generator that can capture temporally coherent motion. Most of these methods model motion in an auto-regressive manner, where the next latent is sampled conditioned on the previous one (see Fig. 1). However, this design has two main drawbacks. First, while good performance requires seeing



(a) Autoregressive Framework (e.g., MoCoGAN-HD)



(b) Non-autoregressive Framework (Ours)

Figure 1: A comparison between autoregressive and non-autoregressive pipeline: (a) Previous autoregressive motion generators require generating the whole clip for a 3D-convolution-based discriminator. (b) Our non-autoregressive motion generator, **StyleInV**, is an inversion network modulated by temporal style (as a random function of t), which enjoys sparse training using a 2D-convolution-based discriminator.

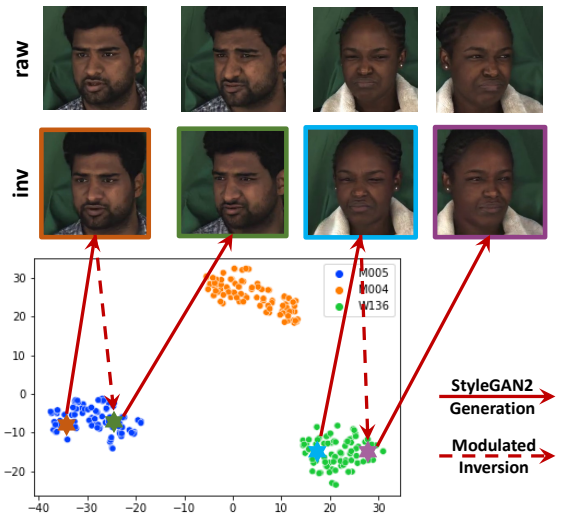


Figure 2: Inverted latent space visualization and modulated inversion process: When the StyleGAN generator is trained with video frame data, $\mathcal{W}$ space is well clustered by human identities and provides promising inversion results. Thus, the modulated inversion process can easily find the target latent corresponding to the same identity (shifted to the next motion) as the source one.

a long sequence of images, the use of heavy 3D discriminators limits its ability to be trained with longer videos. Second, the autoregressive motion generator can lead to motion collapse when extrapolating to generate longer videos.

In this study, we present an effective framework for non-autoregressive motion generation that is capable of generating long and high-resolution videos. Our approach leverages learning-based Generative Adversarial Network (GAN) inversion, which learns the inverse mapping of GANs via an inversion network that consists of an encoder and a decoder[1]. To generate long and coherent videos, we exploit the unique characteristic of the inversion encoder, which captures a rich and smooth manifold between the mapping of images and latent. As illustrated in Fig. 1, to generate a sequence of smooth motion latents, we just need to provide the initial latent code and modulate the inversion encoder with temporal style codes, which are encodings of timestamps with randomness. The motion latents can then be mapped by a StyleGAN decoder to generate a video.

The proposed framework offers several advantages in a single unified framework. **First**, the use of an inversion network naturally constrains the generation space to stay consistent with the desired appearance, which is defined by the initial latent code. As demonstrated in Fig. 2, this leads to a significant benefit. **Second**, thanks to the flexibility of the inversion network in accepting temporal styles of arbitrary timestamps, the framework allows non-autoregressive generation and sparse training [57, 40]. These merits help alleviate the need for heavy discriminators to ensure temporal consistency, as is required in existing approaches. In our implementation, we only need to use a 2D convolutional discriminator instead of a 3D discriminator like MoCoGAN-HD. **Third**, Unlike existing state-of-the-art methods [57, 40, 7] that couple content and motion decoding in one synthesis network, our framework can naturally support content decoder fine-tuning on different image datasets. Specifically, after fine-tuning the decoder (*e.g.*, StyleGAN2) on another image dataset with the mapping layers and low-resolution synthesis layers fixed, given the same sequence of synthesized motion latents, the generated video can possess the new style of the fine-tuning dataset while preserving the motion patterns of the video generated by the parent content decoder.

The main contribution of this work is a novel motion generator that modulates a GAN inversion network. This is the first attempt to build such a generator, and it offers several advantages in a unified framework over existing approaches. These advantages include consistent generation, sparse training, and flexibility in supporting style transfer with simple fine-tuning. We additionally contribute a reformulation to the conventional sparse training, through first-frame-aware acyclic positional encoding (FFA-APE) and first-frame-aware sparse training (FFA-ST), to ensure that our motion generator can faithfully reconstruct the initial frame and that the generated video is smooth and continuous. Extensive experiments on DeeperForensics [20], FaceForensics [35], SkyTimelapse [52] and Tai-Chi-HD [39] datasets show that our model is comparable to or even better than state-of-the-art unconditional video generation methods [41, 57, 40] both qualitatively and quantitatively.

2. Related Work

GAN inversion. The goal of GAN inversion is to find the corresponding vector in the latent space of a pretrained GAN [25, 26] to reconstruct the input image. Existing methods can be classified into three categories [51]: (1) learning-based methods [8, 42, 54, 3, 2, 49, 33], which leverage an encoder network to directly map an image into a latent vector; (2) optimization-based methods [48, 1, 50, 53, 62, 63], which iteratively find the latent vector that best reconstructs the input image using gradient descent; and (3) hybrid models [6, 5, 9, 61], which initialize the iteration process with the result of an encoder network. The design of our motion generator follows the learning-based approach. Therefore, our method is trainable, efficient for single-image inference, and suitable for hierarchical mod-

[1]In many contexts, the decoder is a StyleGAN, and the encoder learns to encode a given image to meaningful latent vectors in the StyleGAN space. There is a variety of image manipulation applications [51, 33, 4] developed based upon such an inversion framework.

ulation. We devise the motion generator on the $\mathcal{W}$ space and use the StyleGAN generated latent as the initial content code to guide the modulated inversion process (see Fig. 2).

Unconditional video generation. Unconditional video generation aims to model the distribution of real videos in a training dataset and generate videos from sampled noise vectors. Many recent studies on this topic are inspired by the success of GANs in image generation. VGAN [47] applies 3D convolutions in both the generator and discriminator, while TGAN [36] optimizes this design by decomposing the generator into an *image generator*, which is shared by the generation of each frame, and a *motion generator*[2]. This framework has been followed by most subsequent studies, such as MoCoGAN [43], which applies a content-motion decomposition. Some approaches [37, 12, 22] have focused on reducing the computational cost of the video discriminator, but the cost is still proportional to the video duration and resolution. Some recent methods have applied more advanced generative frameworks and techniques to unconditional video generation. For example, VideoGPT [55] uses VQ-VAE [32] and GPT [10] to formulate a non-GAN-based video generation approach. Recent studies have also explored unconditional video generation with higher resolution and longer duration. For example, Long-Video-GAN [7] develops a two-phase model that focuses on improving the long-term temporal dynamics of video generation. MoCoGAN-HD [41] and StyleVideoGAN [15] study the generation of latent trajectories in the latent space of a pretrained StyleGAN2 generator. Our approach is inspired by these studies, but differs in the design of the motion generator. Our motion generator is non-autoregressive, thus alleviating the use of heavy discriminators, and it is unique since it obtains the motion latent via modulating a GAN inversion network. This design allows us to attain better motion consistency and semantics.

Recent works [57, 40] explored neural representation-based generators and trained them sparsely as an image GAN. StyleSV [58] improves this framework by introducing StyleGAN3 [24] architecture and several temporal designs. In our work, we extend the idea of sparse training to first-frame-aware sparse training, allowing it to be applied to a generation pipeline conditioned on the initial latent.

Diffusion-based video generation. The diffusion models [18, 34], a new paradigm for image generation tasks, have also achieved significant progress in the task of unconditional video generation [19, 28, 46, 56]. Despite their success, temporal consistency is still an open problem for diffusion models, and GAN-based models exhibit a clear advantage in terms of inference speed.

[2]In the original paper of TGAN [36], the authors called this module *temporal generator*, which is equivalent to the *motion generator* used in subsequent studies [43, 41, 57] and in our paper.

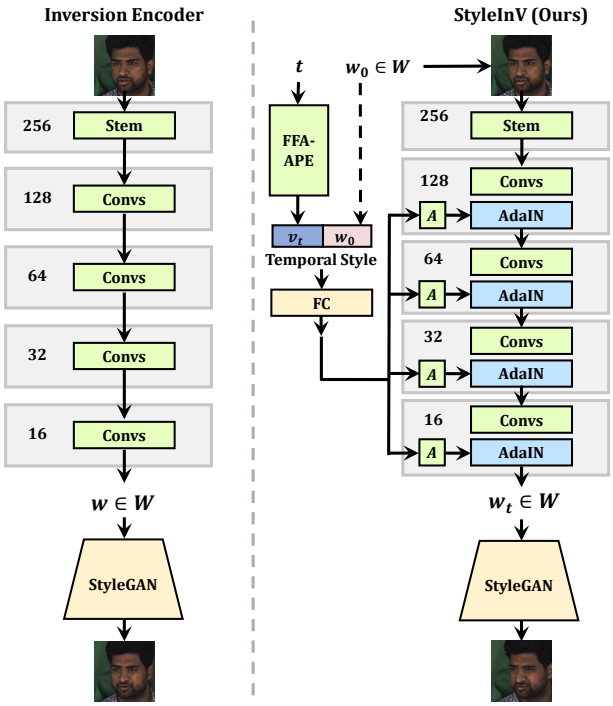


Figure 3: From a typical inversion encoder to StyleInV: We add `AdaIN` layers at the end of each residual block to inject the temporal style, which is a combination of time positional encoding and the first frame latent code. Here "A" stands for a learned affine transform [25].

3. Methodology

3.1. Preliminaries of Inversion Encoder

An inversion encoder maps an input image to a vector in the $\mathcal{W}$ or $\mathcal{W}+$ latent space of a pretrained StyleGAN2 generator. The generated image that corresponds to this vector should faithfully reconstruct the details of the input image. Therefore, when based on $\mathcal{W}$ latent space, given an input image $\mathbf{x}$, the reconstruction process can be defined on top of the inversion network Inv as:

$$\hat{\mathbf{x}} := G(\mathrm{Inv}(\mathbf{x})) := G(E(\mathbf{x}) + \overline{\mathbf{w}}). \quad (1)$$

Here E and G denote the inversion encoder and StyleGAN generator, respectively. $\overline{\mathbf{w}} \in \mathbb{R}^{512}$ denotes the average latent vector of the generator in the $\mathcal{W}$ latent space. In our implementation, the encoder E is a convolutional network backbone that outputs a 512-dimensional vector from the last layer embedding, as shown in Fig. 3(left). We build the encoder on the $\mathcal{W}$ latent space, which eases the design of temporal modulation.

3.2. Temporal Style Modulated Inversion Encoder

We observe that the latent space of a StyleGAN trained on a video dataset is typically well-clustered by its content

subject. Figure 2 shows an example of human face videos, where we depict the results of inverting video clips of different identities into the $\mathcal{W}$ space and visualizing them with t-SNE [45]. It can be observed that the latent space is grouped by human identities. We also observe the same property in video datasets that follow other distributions. This phenomenon suggests that the inversion network inherits some important temporal priors that we could leverage to maintain motion consistency in generated videos.

Motivated by this observation, we propose **StyleInV**, in which the motion latent is generated by modulating a GAN inversion network with temporal styles. Figure 3(right) illustrates the pipeline of our framework. The temporal style $\boldsymbol{s}_t$ of a timestamp t consists of two parts: the motion code $\boldsymbol{v}_t$ and the latent code of the initial frame $\boldsymbol{w}_0$. Inspired by [40], we use an acyclic positional encoding module to compute a dynamic embedding of the timestamp t. However, unlike [40], we make the embedding of the zero timestamp fixed, so this module becomes first-frame-aware. We provide more details in Section 3.3. The latent code $\boldsymbol{w}_0$ of the initial frame is concatenated with the motion code for content-adaptive affine transform.

The temporal style is injected into the inversion encoder through `AdaIN` layers at the end of each convolution block. With this design, the encoder E of StyleInV becomes a function of the initial latent code w_0 and timestamp t. The modulated inversion process can be defined as:

$$\hat{\mathbf{x}}_t := G(\text{StyleInV}(\boldsymbol{w}_0, t)) := G(E(G(\boldsymbol{w}_0), \boldsymbol{s}_t) + \boldsymbol{w}_0). \quad (2)$$

Notably, the output of E serves as the residual w.r.t. w_0, instead of $\overline{\mathbf{w}}$. This modification provides more explicit content information guidance for the inversion encoder.

During training, we first train a raw inversion encoder following Eq. (1) on all video frames. Then, we use this network to initialize the weights of all convolution layers in the StyleInV encoder. Other parameters (*e.g.*, FFA-APE and Affine Transforms) are randomly initialized. Finally, the entire StyleInV encoder is trained end-to-end.

3.3. FFA-APE

The original implementation of acyclic positional encoding (APE) [40] samples a series of noise vectors $\boldsymbol{z}^{\mathsf{m}}_{t_0}, \cdots, \boldsymbol{z}^{\mathsf{m}}_{t_n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ where $t_i = i \cdot \delta^z$. We call these temporal points *anchor points*. Here δ^z is a set constant distance between adjacent anchor points. Then, the noise vectors are mapped to tokens $\boldsymbol{u}_{t_0}, \cdots, \boldsymbol{u}_{t_n}$ by a padding-less `conv1d`-based motion mapping network. The computation of the acyclic positional encoding $\boldsymbol{v}_t$ of arbitrary timestamp t is achieved by a scalable and learnable interpolation between the tokens of two adjacent *anchor points* that cover t. The computation pipeline is shown in Fig. 4.

In our non-autoregressive generation pipeline, the modulated inversion encoder needs to faithfully reconstruct the initial frame when the input timestamp is zero, making it necessary to fix the computation of the APE for the zero timestamp $\boldsymbol{v}_0$. The original APE computation for $\boldsymbol{v}_0$ is dynamic and depends on randomly sampled noise vectors, which can lead to dynamic output that is not desired. To address this, we devise a first-frame-aware acyclic positional encoding (FFA-APE) method that fixes $\boldsymbol{v}_0$ while maintaining the smoothness of APE (see Fig. 4). We achieve this by replacing the noise vector for the first anchor point with a learnable constant vector $\boldsymbol{c}^{\mathsf{m}}_0$, and using left-sided `conv1d` layers with constant padding instead of the padding-less `conv1d` layers. This way, the value of $\boldsymbol{v}_0$ only depends on the constant vector $\boldsymbol{c}^{\mathsf{m}}_0$ and the left-padded vectors, which are also constant. As a result, $\boldsymbol{v}_0$ is naturally fixed without affecting the continuity of positional encoding.

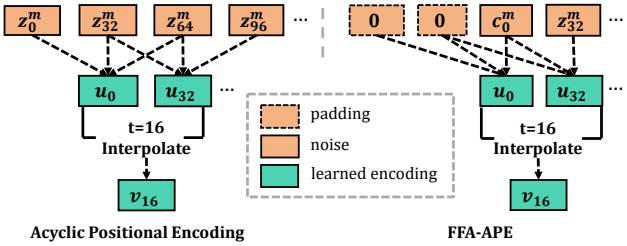


Figure 4: FFA-APE: A simplified case to compute v_{16} for demonstrating the original acyclic positional encoding and our FFA-APE when $\delta^z = 32$ and `conv1d` kernel size is 3. In FFA-APE, the encoding of zero timestamp (u_0) only depends on constant paddings and a constant noise vector, thus is fixed for any sampled noise vector sequence.

3.4. FFA-ST

In this section, we introduce the first-frame-aware sparse training specially designed for our framework. Recent non-autoregressive video generation approaches [57, 40] use a discriminator design that only considers k frames $\boldsymbol{x}_{t_1}, \cdots, \boldsymbol{x}_{t_k}$ for each video, distinguishing the realness of the input conditioned on the time difference of input frames $\delta_i = t_{i+1} - t_i$. This training scheme is called sparse training. StyleGAN-V [40] has analyzed the choice of k and found that $k = 3$ is ideal for most datasets. The discriminator is defined as $D(\boldsymbol{x}_{t_{1,2,3}}, \delta_{1,2})$.

We follow this training scheme to make full use of our non-autoregressive framework. Nonetheless, using only three randomly sampled timestamps to train the generator and discriminator can result in sharp transitions at the beginning of the generated video, where the generated $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ usually diverge too much, and sometimes even switch to another identity and never return. This happens because although we define the generation process of a video as a modulated inversion process of the start frame, the discriminator is unaware of it. The discriminator only focuses on

the smoothness of generated latent trajectories, failing to ensure the motion generator produces frames that share the identity with the start frame.

To solve this problem, we introduce the initial frame into the discriminator to enhance content consistency and motion smoothness. The adversarial loss for the first-frame-aware discriminator (FFA-D) can be written as:

$$\begin{aligned} \boldsymbol{y}_{t_{0,1,2,3}} &= G(\text{StyleInV}(w_0, t_{0,1,2,3})), \\ \mathcal{L}_{adv} &= \mathbb{E}_{\boldsymbol{x}\sim p_v}\left[\log D(\boldsymbol{x}_{t_{0,1,2,3}}, \delta_{0,1,2})\right] \\ &\quad + \mathbb{E}_{w_0\sim p_{\mathcal{W}}}\left[\log(1 - D(\boldsymbol{y}_{t_{0,1,2,3}}, \delta_{0,1,2})\right], \end{aligned} \tag{3}$$

where we specify $t_0 = 0$. Here, p_v and $p_{\mathcal{W}}$ denote the real data distribution and $\mathcal{W}$ latent space distribution, respectively. To explicitly enforce initial frame reconstruction, we use a L_2 loss for the generated $\boldsymbol{y}_{t_0}$:

$$\mathcal{L}_{L_2} = ||G(w_0) - G(\text{StyleInV}(w_0, 0))||_2. \tag{4}$$

Finally, we apply latent regularization [33, 30] to the encoder's output, so as to enhance content consistency:

$$\mathcal{L}_{reg} = \sum\nolimits_{i=0}^{3} ||E(G(w_0), t_i)||_2. \tag{5}$$

The overall loss function for training our motion generator and the discriminator is defined as:

$$\min_E \max_D \mathcal{L}_{adv} + \min_E(\lambda_{L_2}\mathcal{L}_{L_2} + \lambda_{reg}\mathcal{L}_{reg}). \tag{6}$$

Here λ_{L_2} and λ_{reg} are the loss hyperparameters. We also apply discriminator adaptive augmentation [23, 40] and $r1$ regularization [25, 40] to further improve the training stability and generation quality.

3.5. Finetuning-based Style Transfer

Our 'inversion encoder+decoder' framework can naturally take a pretrained StyleGAN model as the generator. And such a configuration allows the generator to be fine-tuned for different styles, and yet still able to use the motion generator for generating new video with styles. The capability is not possible with existing non-autoregressive video generation methods [57, 40] because they cannot be finetuned under an image GAN training scheme.

To achieve style transfer, as illustrated in Fig. 5, we fine-tune the pre-trained StyleGAN model using an image dataset, such as MegaCartoon [31], while keeping the mapping network and low-resolution ($\leqslant 32^2$, coarse and middle layers in [25]) synthesis blocks fixed. This configuration maintains the distribution of the $\mathcal{W}$ space during fine-tuning. To improve identity preservation and reduce artifacts, we apply both a perceptual loss [21] and an identity loss [14] between the images generated by the original and fine-tuned StyleGAN. We show some visual results in Fig. 8. The style-transferred video maintains the same motion pattern as the video generated by the parent model, while adopting a new style from the fine-tuning image dataset. It is noteworthy that the finetuning process is independent of the video generation training. It means that the finetuning-based style transfer is "plug-and-play" as the fine-tuned image generator can be used on any StyleInV models. It does not introduce inference latency either.

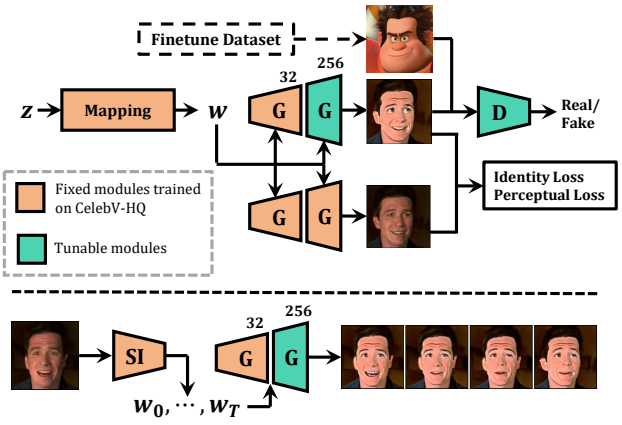


Figure 5: Finetuning-based style transfer: our framework allows easy fine-tuning of decoder (a pretrained StyleGAN generator) to a new domain by freezing mapping and low-resolution ($\leqslant 32^2$) layers. Standard identity loss and perceptual loss are applied to improve identity preservation and reduce artifacts. The style-transferred videos can then be generated by incorporating the StyleInV motion generator.

4. Experiments

Datasets. We use four video datasets in our main experiments: DeeperForensics 256^2 [20], FaceForensics 256^2 [35], SkyTimelapse 256^2 [52] and TaiChi 256^2 [39]. The cropping strategy for DeeperForensics [20] and FaceForensics [35] is different. For DeeperForensics, we use a stabilized FFHQ [25] cropping strategy [29], while we follow the strategy firstly adopted by TGAN-V2 [37] for FaceForensics. Please refer to Appx. D and Appx. G for a detailed discussion.

Baselines. We explore four state-of-the-art methods for comparison: MoCoGAN-HD [41], DIGAN [57], StyleGAN-V [40] and Long-Video-GAN [7]. Among these methods, MoCoGAN-HD and DIGAN require an explicit setting of the training clip length. We follow the default setting of their paper to set the clip length as 16 for both methods. This setting is identical to StyleGAN-V [40].

In addition, on DeeperForensics, we explore an optimized setting on DIGAN and MoCoGAN-HD for a more fair comparison. For DIGAN, we directly increase the clip length to 128 frames. For MoCoGAN-HD, we apply the first-frame-aware sparse training to train its motion genera-

tor, so as to avoid using a heavy 3D discriminator, allowing it to be trained with 128-frame clips as other methods.

Evaluation. We use Fréchet Inception Distance (FID) [17] and Fréchet Video Distance (FVD) [44] to evaluate all models quantitatively. In practice, we follow the metric calculation framework provided by StyleGAN-V [40] to first generate a fake video dataset with 2,048 synthesized clips, each of 128 frames. For FID, we sample 50k frames from real and fake video datasets to compute the result. For FVD, we compute FVD_{16} and FVD_{128} with the first 16 frames and all 128 frames of each clip, respectively. We use FID results to show the single-frame image quality of each method.

To ensure a fair comparison, we re-benchmark the quantitative results of every method on every dataset. We retrain all the baselines using the official paper setting, except for MoCoGAN-HD on SkyTimelapse, where an officially released checkpoint is available. For more implementation details, please refer to Appx. F.

4.1. Main Results

Quantitative results. Table 1 summarizes the quantitative results of our method compared to other baselines. Our method achieves competitive quantitative results on all the benchmarks. Notably, although MoCoGAN-HD and DIGAN are trained with clips of 16 frames, we still outperform them in terms of FVD_{16} metrics on all four datasets.

Qualitative results. Figure 6 shows the qualitative comparison between our method and the baselines on all four datasets. MoCoGAN-HD and DIGAN both suffer from motion collapse, resulting in a degraded generation quality over time. StyleGAN-V shows an impressive visual performance on FaceForensics and SkyTimelapse, but it sometimes fails to maintain the identity and accessories on DeeperForensics and lacks diversity and magnitude of motion over a long time span on TaiChi (the subject gradually fixes at one state). Long-Video-GAN is exceptionally good at SkyTimelapse, but it cannot achieve similar performance on other datasets. It fails to maintain the identity on DeeperForensics, and its single-frame content on TaiChi lacks details and is inferior to other methods. The generated videos by Long-Video-GAN collapse on FaceForensics.

In contrast to existing methods, our method demonstrates stable results on all four datasets, particularly with superior identity preservation on human-face video and long-term generation quality on TaiChi. Although our method outperforms existing methods in terms of content quality, continuity, and quantitative results, the motion semantics of our generated videos on SkyTimelapse are inferior to those on other datasets. This could be one of the limitations of our work and an area for future improvement.

Extended experiments. We present more in-depth comparisons in Table 2 and Fig. 7 by introducing training improvements to baselines. Increasing the clip length generally improves the results of MoCoGAN-HD and DIGAN, but they are still inferior to our method. Notably, training with longer clips harms the short-term FVD_{16} result of DI-

Table 1: FID, FVD_{16} and FVD_{128} results of video generation methods on (a) DeeperForensics 256^2, (b) FaceForensics 256^2, (c) TaiChi 256^2, and (d) SkyTimelapse 256^2. **Bolds** indicate best and underlines indicate the second best.

(a) DeeperForensics 256^2

Method	FID (↓)	FVD_{16} (↓)	FVD_{128} (↓)
MoCoGAN-HD	135.30	101.07	610.30
DIGAN	191.99	46.69	1060.27
StyleGAN-V	59.59	**39.33**	68.81
Long-Video-GAN	56.54	74.77	169.45
StyleInV (ours)	**54.05**	41.58	**53.93**

(b) FaceForensics 256^2

Method	FID (↓)	FVD_{16} (↓)	FVD_{128} (↓)
MoCoGAN-HD	24.45	112.67	486.69
DIGAN	151.53	146.62	1993.20
StyleGAN-V	**8.64**	52.92	108.86
Long-Video-GAN	40.40	233.26	567.78
StyleInV (ours)	12.06	**47.88**	**103.63**

(c) TaiChi 256^2

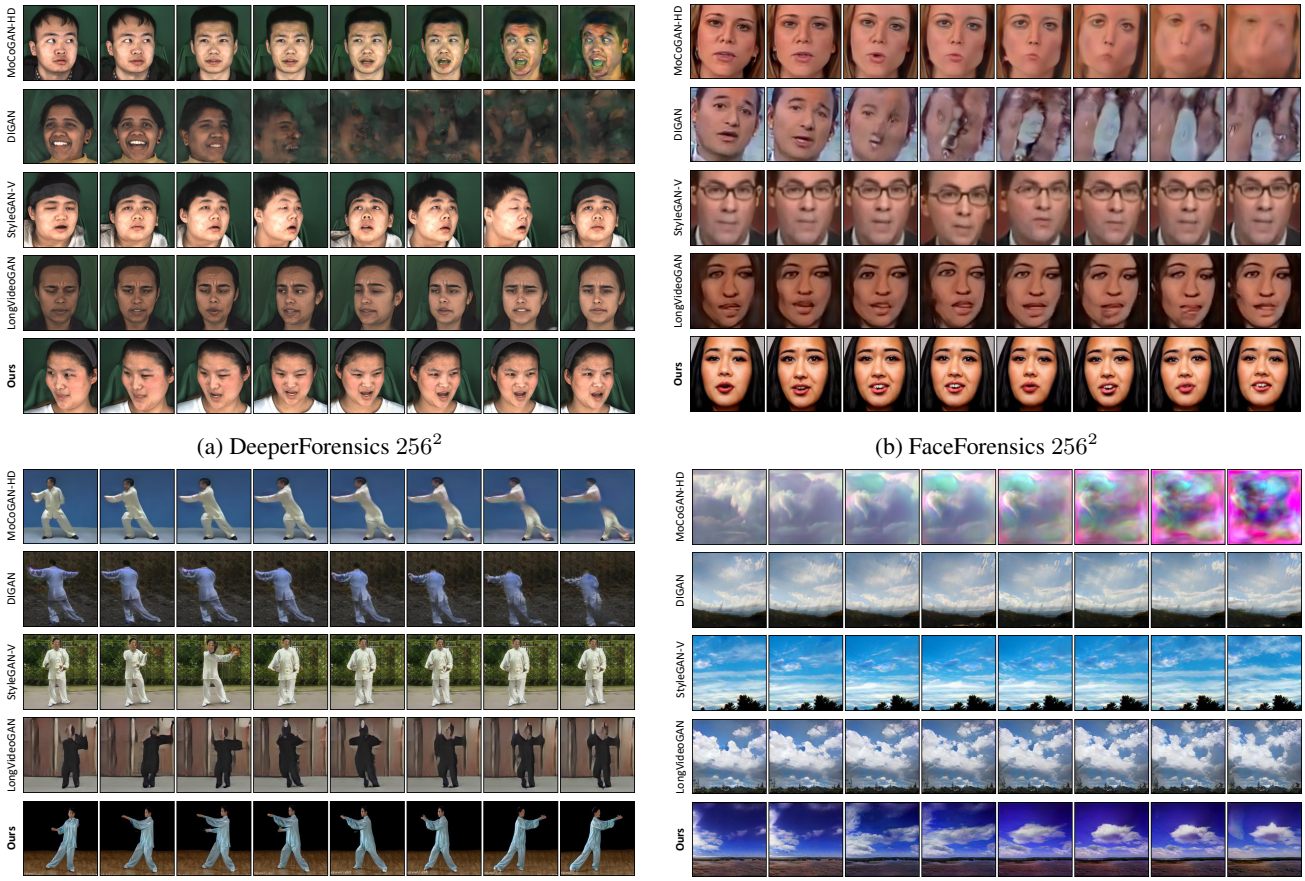
Method	FID (↓)	FVD_{16} (↓)	FVD_{128} (↓)
MoCoGAN-HD	73.61	315.03	622.95
DIGAN	67.24	196.77	954.93
StyleGAN-V	**35.68**	254.74	477.78
Long-Video-GAN	43.90	248.55	502.65
StyleInV (ours)	41.55	**185.72**	**328.90**

(d) SkyTimelapse 256^2

Method	FID (↓)	FVD_{16} (↓)	FVD_{128} (↓)
MoCoGAN-HD	251.81	696.58	4116.03
DIGAN	32.83	148.08	269.43
StyleGAN-V	16.95	81.32	197.83
Long-Video-GAN	25.41	116.50	**152.70**
StyleInV (ours)	**14.32**	**77.04**	194.25

Table 2: FID, FVD_{16} and FVD_{128} results of extended experiments on DeeperForensics 256^2. We apply sparse training to MoCoGAN-HD [41] (#1) and change the preset clip length of DIGAN [57] to 128 (#2). **Bolds** indicate best. (-) indicates a smaller (better) quantitative result, while (+) indicates a larger (worse) one, compared with Table 1a.

#	Method	FID (↓)	FVD_{16} (↓)	FVD_{128} (↓)
1	[41] + Sparse Training	55.84 (-)	54.58 (-)	129.13 (-)
2	[57] + Clip 128	74.80 (-)	87.42 (+)	95.80 (-)
3	StyleInV (ours)	**54.05**	**41.58**	**53.93**



(a) DeeperForensics 256^2

(b) FaceForensics 256^2

(c) TaiChi 256^2

(d) SkyTimelapse 256^2

Figure 6: *Uncurated* samples from the existing methods on DeeperForensics 256^2, FaceForensics 256^2, TaiChi 256^2 and SkyTimelapse 256^2, respectively. We sample a 128-frame video and display every 16 frames, starting from $t = 0$.

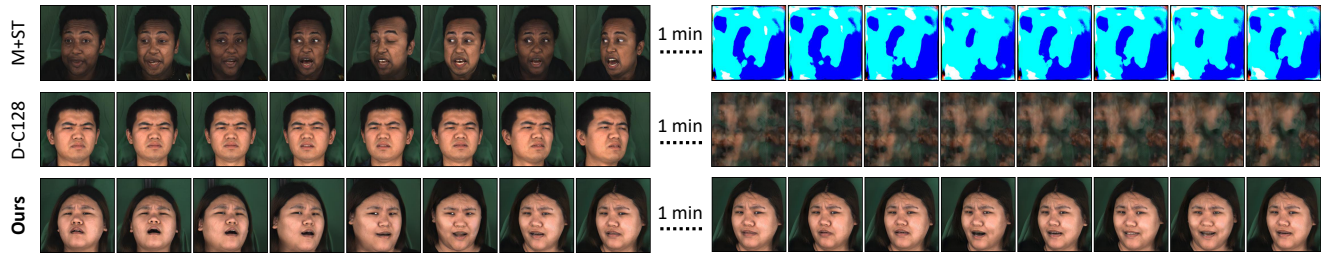


Figure 7: Qualitative comparison of extended experiments. "M+ST" and "D-C128" correspond to Table 2 (#1) and (#2), respectively. Each row shows the first and last 128 frames of a 2056-frame (68.5s) video, displayed every 16 frames.

GAN, which indicates its tradeoff between duration length and local temporal quality. Qualitatively, for both methods, the generated content is evidently improved within 128 frames, although the sparsely trained MoCoGAN-HD exhibits issues with identity switching. Motion collapse is still observed when MoCoGAN-HD and DIGAN generates long videos. In contrast, our method can stably generate extremely long videos without motion collapse. Our method outperforms the sparsely trained MoCoGAN-HD, demonstrating the superiority of our motion generator design.

4.2. Properties

As discussed in Section 3.5, our method has the unique advantage over state-of-the-art methods, such as StyleGAN-V and Long-Video-GAN, on its high compatibility with StyleGAN-based downstream techniques.

Figure 8: Finetuning-based style transfer result. The 1st row is generated by the parent model trained on CelebV-HQ [60]. The 2nd, 3rd, and 4th row uses the StyleGAN generator fintuned on Cartoon, Arcane, and MetFace, respectively.

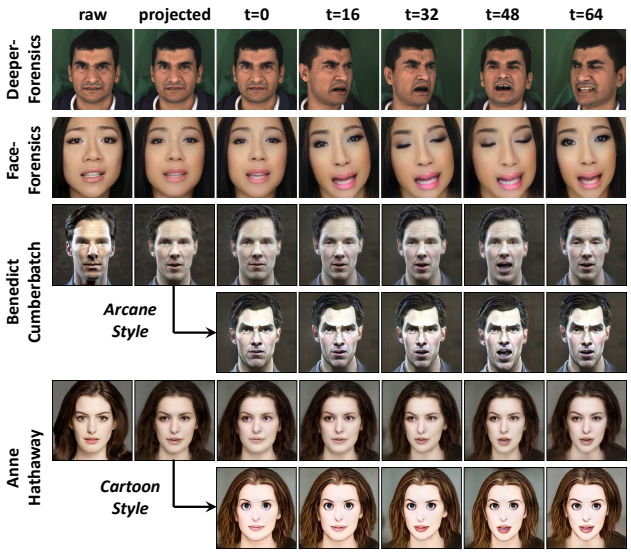


Figure 9: Initial-frame conditioned generation and style transferred results.

Finetuning-based style transfer. We train the parent model (motion generator and StyleGAN) on CelebV-HQ [60], as its rich identity makes it more suitable for transfer learning. To perform style transfer, we fine-tune the StyleGAN on the Cartoon [31], MetFace [23], and Arcane datasets following the procedure outlined in Section 3.5. In Fig. 8, we show examples where the same StyleInV-generated latent sequence is decoded by different but aligned StyleGAN generators. Our method achieves satisfactory results in terms of smooth video style transfer with well-aligned face structure, identity, and expression, demonstrating its desirable properties and potential for various applications. More results can be found on our project page.

Initial-frame conditioned generation. Our network supports generating a series of content given a real-world image as the initial frame. We first inverse the image into the StyleGAN2 latent space with a pSp [33] encoder, which is trained to initialize the weights of StyleInV. We treat it as the 512-dimensional initial frame latent w_0, then use it to generate a video with our StyleInV. The generated latent sequence can be also applied to a finetuned image generator to synthesize a style-transferred animation video. Through this pipeline, the real image is reconstructed twice, the first time is during the inversion process, while the second time is when synthesizing $G(\text{StyleInV}(w_0, 0))$.

When the real images are sampled from the training dataset (see Fig. 9 first two rows), $G(\text{StyleInV}(w_0, 0))$ can faithfully reconstruct the raw image and generate high-quality videos. We then test the generation quality for real images sampled out of the training set (see Fig. 9 last two rows, where we select Benedict Cumberbatch and Anne Hathaway). We use the StyleGAN2 generator and StyleInV model trained on CelebV-HQ [60] dataset as it is richer in its identities. The results show that our StyleInV network can still generate meaningful videos while reconstructing the initial frame decently, and the style transfer results are smooth and well-aligned. Please refer to our project page for more results.

Table 3: Ablation result on the DeeperForensics dataset.

#	Method	FID (↓)	FVD_{16} (↓)	FVD_{128} (↓)
1	w/o inversion encoder	54.35	59.49	152.82
2	w/o FFA-APE	55.26	88.98	144.52
3	w/o Eq.(4) & Eq.(3)	52.55	67.43	58.88
4	w/o Eq.(3)	53.95	86.32	59.76
5	Ours	54.05	41.58	53.93

4.3. Ablation Studies

Motion generator design. We explore two alternative motion generator designs. The first is the autoregressive MoCoGAN-HD design, which has been discussed in Section 4.1. For the second design, we remove all `Convs`,

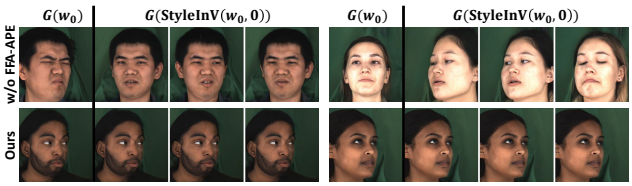


Figure 10: Ablating FFA-APE. Generate the first frame with a fixed w_0 but different temporal noise sequences. The model without FFA-APE fails to reconstruct the initial frame and generates the first frame with randomness.

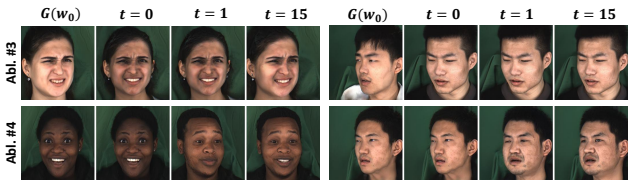


Figure 11: Ablating FFA-ST. The L_2 loss ensures the initial frame reconstruction. But without the initial frame included in the discriminator, we still cannot preserve the identity.

`AdaIN` and affine transform layers in Fig. 3 and let the mapped temporal style be the output of inversion encoder, *i.e.*, the residual w.r.t. initial frame latent w_0. It largely harms identity preservation. Both FVD_{16} and FVD_{128} degrade significantly as is shown in Table 3(#1).

FFA-APE. We evaluate the importance of our first-frame-aware acyclic positional encoding (FFA-APE) module by replacing it with the original design proposed by [40]. As shown in Fig. 10, the dynamic embedding of the zero timestamp prevents the network from faithfully reconstructing the initial frame. In contrast, our full method can stably realize reconstruction. In addition, the L2 loss in Eq. (4) fails to converge for the ablation method, and its gradient further harms the learning of the positional encoding module, leading to a much worse quantitative result shown in Table 3 (#2).

FFA-ST. We conduct two ablation experiments for the FFA-ST modules. In the first experiment, we remove the initial frame from the discriminator and remove the reconstruction loss (Eq.(4)) (Table 3(#3)). In the second experiment, we only remove the initial frame from the discriminator while keeping the reconstruction loss (Table 3(#4)). As shown in Fig. 11, without the reconstruction loss, our model cannot reconstruct the initial frame accurately. With the L_2 loss, the initial frame is reconstructed, but there is a sudden transition between the first two frames, and sometimes, the identity also changes, leading to a worse FVD_{16} result. These experiments demonstrate the importance of our first-frame-aware discriminator (FFA-D).

5. Conclusion

We have presented a novel approach for unconditional video generation by employing a pretrained StyleGAN image generator. The proposed StyleInV motion generator generates latents in the StyleGAN2 latent space by modulating a learning-based inversion network, and thus capable of inheriting its informative priors of the initial latent. Our network features non-autoregressive training and uniquely supports fine-tuning based style transfer. Extensive experiments demonstrate the superiority of our method in generating long and high-resolution videos, outperforming state-of-the-art baselines. Here we also briefly discuss our limitations and broader impacts.

5.1. Limitations

Inferior motion semantics on SkyTimelapse. Our motion semantics on SkyTimelapse [52] are inferior to those on other datasets. This could be due to different dataset characteristics, as videos in SkyTimelapse are not subject-centric and typically driven by global motions, which does not align perfectly with our model nature.

The impact of dataset identity richness. When the scale of facial identities in the video dataset is too small, the effects of inversion, editing, and style transfer are constrained.

Image generation quality. The generation quality of StyleGAN determines the performance upper bound of our method. The images generated by the StyleGAN2 models have artifacts in the background on SkyTimelapse [52], and lack fine details and a sense of structure on TaiChi [39].

Model training. Our approach is two-stage, requiring 7.5 and 9 GPU days each, which is more than the 8 GPU days of StyleGAN-V [40]. Despite this, StyleInV is as efficient when finetuning the hyperparameters of the video generator, since the image generator only needs to be trained once.

5.2. 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 with threats. We believe that it can be alleviated by developing more advanced falsified media detection methods or contributing larger-scale and higher-quality forgery detection datasets.

Acknowledgement. This work is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2022-01-030). It is also supported under the RIE2020 Industry Alignment Fund Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s). We thank Shuai Yang for his help in this work.

References

[1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *ACM TOG*, 40:1–21, 2021. 2

[2] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Only a matter of style: Age transformation using a style-based regression model. *ACM TOG*, 40:1–12, 2021. 2

[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. ReStyle: A residual-based StyleGAN encoder via iterative refinement. In *ICCV*, 2021. 2

[4] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H Bermano. HyperStyle: StyleGAN inversion with hypernetworks for real image editing. In *CVPR*, 2022. 2

[5] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM TOG*, 38:1–11, 2019. 2

[6] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a GAN cannot generate. In *ICCV*, 2019. 2

[7] 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. 2, 3, 5, 15

[8] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in GANs. In *ICLR*, 2021. 2

[9] Lucy Chai, Jun-Yan Zhu, Eli Shechtman, Phillip Isola, and Richard Zhang. Ensembling with deep generative views. In *CVPR*, 2021. 2

[10] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 3

[11] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020. 16

[12] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint*, arXiv:1907.06571, 2019. 1, 3

[13] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. RetinaFace: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020. 14

[14] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 5

[15] Gereon Fox, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. StyleVideoGAN: A temporal generative model using a pretrained StyleGAN. In *BMVC*, 2021. 3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *ICCV*, 2016. 16

[17] 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. 6

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[19] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *NeurIPS*, 2022. 3

[20] 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. 2, 5, 12, 14, 16

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 5

[22] Emmanuel Kahembwe and Subramanian Ramamoorthy. Lower dimensional kernels for video discriminators. *Neural Networks*, 132:506–520, 2020. 3

[23] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020. 5, 8, 14, 15, 16

[24] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *NeurIPS*, 2021. 3

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 2, 3, 5, 14, 15

[26] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 2, 15

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 15

[28] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. VideoFusion: Decomposed diffusion models for high-quality video generation. In *CVPR*, 2023. 3

[29] Jacek Naruniec, Leonhard Helminger, Christopher Schroers, and Romann M Weber. High-resolution neural face swapping for visual effects. In *CGF*, 2020. 5, 14

[30] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. In *ICCV*, 2021. 5

[31] Justin NM Pinkney and Doron Adler. Resolution dependent GAN interpolation for controllable image synthesis between domains. *arXiv preprint*, arXiv:2010.05334, 2020. 5, 8, 14

[32] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, 2019. 3

[33] 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. 2, 5, 8, 15, 16

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[35] 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. 2, 5, 12, 14, 16

[36] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *ICCV*, 2017. 1, 3

[37] 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. 1, 3, 5, 14

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

[39] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019. 2, 5, 9, 12, 16

[40] 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. 2, 3, 4, 5, 6, 9, 12, 14, 15

[41] 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. 1, 2, 3, 5, 6, 14, 15

[42] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. *ACM TOG*, 40:1–14, 2021. 2

[43] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 2018. 1, 3

[44] 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 preprint*, arXiv:1812.01717, 2018. 6

[45] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *JMLR*, 9(11), 2008. 4

[46] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. 2022. 3

[47] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 1, 3

[48] Hui-Po Wang, Ning Yu, and Mario Fritz. Hijack-GAN: Unintended-use of pretrained, black-box GANs. In *CVPR*, 2021. 2

[49] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Weiming Zhang, Lu Yuan, Gang Hua, and Nenghai Yu. E2Style: Improve the efficiency and effectiveness of StyleGAN inversion. *TIP*, 31:3267–3280, 2022. 2

[50] Zongze Wu, Dani Lischinski, and Eli Shechtman. StyleSpace analysis: Disentangled controls for StyleGAN image generation. In *CVPR*, 2021. 2

[51] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN inversion: A survey. *TPAMI*, 45:3121–3138, 2022. 2

[52] 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. 2, 5, 9, 16

[53] Yangyang Xu, Yong Du, Wenpeng Xiao, Xuemiao Xu, and Shengfeng He. From continuity to editability: Inverting GANs with consecutive images. In *ICCV*, pages 13910–13918, 2021. 2

[54] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images. In *CVPR*, 2021. 2

[55] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and Transformers. *arXiv preprint*, arXiv:2104.10157, 2021. 1, 3

[56] Siyuan Yang, Lu Zhang, Yu Liu, Zhizhuo Jiang, and You He. Video diffusion models with local-global context guidance. In *IJCAI*, 2023. 3

[57] 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. 2, 3, 4, 5, 6, 15

[58] Qihang Zhang, Ceyuan Yang, Yujun Shen, Yinghao Xu, and Bolei Zhou. Towards smooth video composition. In *ICLR*, 2023. 3

[59] 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. 15

[60] 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. 8, 12, 14, 16

[61] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain GAN inversion for real image editing. In *ECCV*, 2020. 2

[62] Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved StyleGAN embedding: Where are the good latents? *arXiv preprint*, arXiv:2012.09036, 2020. 2

[63] Peiye Zhuang, Oluwasanmi Koyejo, and Alexander Schwing. Enjoy your editing: Controllable GANs for image editing via latent space navigation. In *ICLR*, 2021. 2

Appendix

This document provides supplementary information that is not elaborated in our main paper. In Section A, we present some extra properties of our method. In Section B, we give a more detailed discussion of our limitations 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 and their impact on style transfer. 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 [40], our network can also generate arbitrarily long videos with decent quality. The result is shown in Fig. 12 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. 13 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 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 [20], FaceForensics [35], and TaiChi [39], 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 videos generated by StyleInV conform to our model nature. Please refer to the supplementary videos for visual results.

Figure 12: Our StyleInV can generate arbitrarily long videos with long-lasting content consistency.

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 [60], 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 [39] have certain artifacts in the background. Especially for the TaiChi dataset, although our 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 megapixel level video generation and StyleGAN-based editing, and it might in return promote the research of learning-based GAN inversion methods.

Figure 13: 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.

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 4 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 $\approx$ 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

Table 4: 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	$7.5+1+9$

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

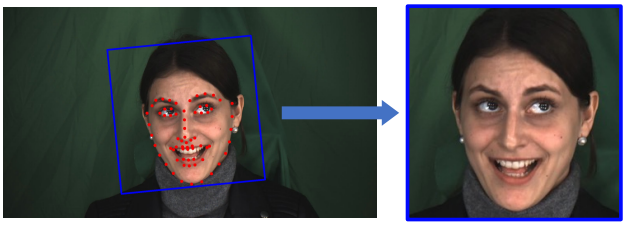
In this section, we introduce the cropping strategies of the FaceForensics dataset and the DeeperForensics dataset, then explain the difference between them.

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 [35] dataset is composed of news broadcasting videos. Apart from raw videos, it also releases labeled face masks for each frame. TGAN-V2 [37] 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, 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 [41, 40]. We also apply it for the FaceForensics dataset pre-processing.



(a) raw frame (b) FFHQ cropped

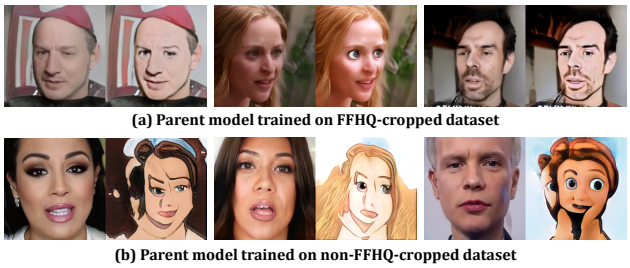
Figure 14: FFHQ cropping strategy on DeeperForensics dataset. The landmarks are detected.

DeeperForensics cropping. The DeeperForensics [20] 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 [25]. The cropping pipeline is shown in Fig. 14, 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 [13], then follow a *stabilizing approach* proposed by [29]. 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 [31] and Metfaces [23]) 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. 15 compares the finetuning-based style transfer result of the parent model trained on CelebV-HQ [60] (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 15: 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 [25, 26, 23] proposes to inject noise vectors at all layers of the generator for finer details in the background, hair, skin, etc. As reported by [25], the omission of noise will lead to a "featureless painterly look". However, though designed on top of StyleGAN2, StyleGAN-V [40] 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.

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 [41] designs motion generators for a pretrained StyleGAN2 as we do. DIGAN [57] and StyleGAN-V [40] train the entire framework as a whole in a non-autoregressive manner. Long-Video-GAN [7] is split into a low-resolution stage and a high-resolution stage.

Table 5: 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 6: 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**

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 [23], 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 [38, 59]. 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 [27], 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 [40]. 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 [33], which compose

a ResNet-50 backbone [16]. The `AdaIN` layers are adopted from StarGAN-V2 [11], 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 [23], 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 [33], 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 [20]. 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 [35]. 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 [52]. StyleGAN-V releases its SkyTimelapse 256^2 dataset [3]. 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 [39]. We follow the link [4] 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 [60]. We download the video dataset using the link for processed CelebV-HQ videos [5] and crop the dataset to 256 resolution with stabilized FFHQ cropping. The entire dataset has 35663 videos.

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

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

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