One Style is All You Need to Generate a Video

Sandeep Manandhar and Auguste Genovesio
IBENS, Ecole Normale Supérieure
75005 Paris, France
sandeep.manandhar@bio.ens.psl.eu, auguste.genovesio@ens.psl.eu

Abstract

In this paper, we propose a style-based conditional video generative model. We introduce a novel temporal generator based on a set of learned sinusoidal bases. Our method learns dynamic representations of various actions that are independent of image content and can be transferred between different actors. Beyond the significant enhancement of video quality compared to prevalent methods, we demonstrate that the disentangled dynamic and content permit their independent manipulation, as well as temporal GAN-inversion to retrieve and transfer a video motion from one content or identity to another without further preprocessing such as landmark points.

Keywords *– conditional video generation, temporal style, dynamics transfer*

1. Introduction

Image synthesis has seen significant advancements with the development of generative models. However, generative models of videos have not been as successful, and controlling the dynamic generation process has been a major challenge. This is largely due to the complex spatio-temporal relationships between content/actors and dynamic/actions, which makes it difficult to synthesize and control the dynamics independently. Several methods have been proposed to address this challenge, each with their own design principles. Broadly speaking, there are two primary classes of video generative models: 3D models that learn from 2D+time volumetric data by employing 3D convolutional neural networks (CNNs), and 2D models that generate a sequence of 2D frames while disentangling the spatio-temporal components of a given video distribution. Many of the earlier methods took the former approach treating each video clip as a point in latent space, thus making the manipulation in such space hardly possible. The latter approach is not only more resource-efficient, but also allows for greater control over the generation process, as demonstrated by [42, 45, 47]. However, these methods require some pre processing (optical flow, pose information) to manipulate the generated videos.

In their work, [12] introduced a variational encoder for visual learning, which assumes that higher-level semantic information within a short video clip can be decomposed into two independent sets: static and dynamic. With similar notion, [7] employed two separate encoders to produce content and pose feature representations. Pose features are processed by an LSTM to predict future pose information which is then used along with the current content information to generate the next frame. The idea of treating content and motion information independently has laid a foundation for many works in video generation.

Instead of considering a video as a rigid 3D volume, one can model it as a sequence of 2D video frames $x(t) \in \mathbf{R}^{3\times H\times W}$, where t is the temporal point, (H, W) are the height and the width of the video frame. An image generator $G(z)$ can be trained to produce an image $x' \sim x(t)$ from a vector z coming from a latent space $Z \in \mathbf{R}^d$, where $d < H \times W$. However, the problem at hand is to come up with a sequence of $z(t)$ that can be fed into $G(z)$ to produce a realistic video frame sequence. And, if such $z(t)$ can be obtained, how can we manipulate the video generation process?

The authors of [26] proposed to first map a latent vector to a series of latent codes using a temporal generator. An image generator would then use the set of codes to output video frames. MOCOGAN, [33], on the other hand proposed to decompose the latent space Z into two independent subspaces of content Z_c and motion Z_m. Z_c is modeled by the standard Gaussian distribution, whereas Z_m is modeled by a recurrent neural network (RNN). The content code remains the same for a generated video, while motion codes varies for each generated frames. MOCOGAN-HD [32] and StyleVideoGAN [10] took advantage of a pretrained StyleGAN2 [17] image latent space and proposed to traverse in the latent space using RNNs to produce video frames.

Interestingly, in the context of a pretrained StyleGAN2 network, one can perform GAN inversion [42] on a image sequence to obtain its latent representation. StyleGAN2

produces a continuous and consistent latent space, where close by latent vectors map to similar realistic images. Taking advantage of this property, the latent vector obtained by optimization from the previous frame can be used as the starting point to search for the latent vector of the next frame, thus optimizing for minor changes. Upon simple linear projection (such as PCA) of the latent trajectory of a movie optimized in such manner, we can observe that the higher components are similar to cosine waves (see Appendix Section 1). The author in [13] also made this observation in the context of protein trajectory simulation, where he finds that the cosine content of the principal components are negatively related to the randomness of the simulation. In the case of optimized vectors corresponding to the inverted images, they are correlated. Hence, the waves are obvious and visible. This hints us that sinusoidal bases could naturally facilitate training of a StyleGAN generator to produce image sequences.

To this end, we propose a temporal style generator in order to generate videos using StyleGAN2's sythesis network. We use a *time2vec* [18] network to introduce a temporal embedding from where the temporal styles will be generated. *time2vec* network provides a learnable Fourier bases. By scaling the Fourier bases using a single motion style vector, we propose to produce diverse and arbitrary length videos. Main contributions of our work are as follow:

- We integrate a novel temporal latent space in StyleGAN's generator network using a sinusoid-based temporal embedding.
- We evaluate our method against prevalent methods in an unconditional setting, demonstrating a significant enhancement of video quality.
- We propose several approaches to rigorously evaluate conditional video generation through contexts such as talking faces and human activities.
- We recover motion from real input videos and map it to our learned latent motion style space via GAN-inversion. This further facilitates the manipulation of temporal style of the generated videos.

We trained our model on videos of talking head (MEAD [37], RAVDESS [20]) and human activities (UTD-MHAD [3]). Besides the Fréchet video distance (FVD) [34] metric, we conducted human evaluation focused on the realism of the generated videos using the MEAD dataset. Additionally we proposed LiA (Lips Area) metric to evaluate the videos generated from the MEAD dataset. We also benchmarked our results using publicly available method for human action recognition with UTD-MHAD dataset.

2. Related work

The domain of video synthesis consists of tasks such as future frame prediction [7, 9, 21, 36], frame interpolation [15, 24, 41] and in our context, video generation from scratch [35]. Video generation follows the success of image generative adversarial models which can produce highly controllable images of remarkable quality [11]. Much focus has been given to temporal extension of such GANs. [23, 26, 27, 33] have adopted the strategy to use content and motion codes by leveraging on 2D image generator. MOCOGAN-HD [33] used a pretrained StyleGAN2's network [17] and trained a RNN model to simply explore along the principal components of the latent space. Recently, [2] also proposed a style-based temporal encoding for a 3D version of StyleGAN3's synthesis network [16] where temporal codes are generated by a noise vector filtered by a fixed set of temporal low pass-filters. [44] used implicit neural representation (INR) [4, 5] to model videos as continuous signal. Finally, StyleGAN-V [30], relied on training a modified StyleGAN2 generator with an INR-inspired positional embedding for the successive video frames. Both of these methods produce videos with arbitrary frame rates. Our method is related to StyleGAN-V as it uses StyleGAN2 synthesis network. However, while StyleGAN-V requires multiple random input vectors to obtain a single trajectory, our approach requires only one such input vector. The latter allows us to manipulate the temporal aspect of the generated videos [25, 42].

Conditional generative models are another exciting field of research. Besides explicit vector based labels, text, audio and images have been used in conditioning for frame generation. [39] proposes a simple and efficient 3D CNN based generator that takes a single image and a conditioning label as an input to generate videos. [40] takes a source frame with one human face and generates video that has pose and expression of another face in a driving video. [38] conditioned their video generation on semantic maps where objects present in the frame are labelled with colors. The network can also take information like optical flow and pose information during the training. [31] generated videos of talking face using sequence of facial landmarks of target face. [47] is yet another image-conditioned video generation model, which has dedicated networks for motion prediction and keypoint detection. However, it is not straightforward to generated videos with arbitrary frame rates with image-conditioned models. Recently diffusion based models have been employed to generate videos as they can output high quality images and demonstrated great flexibility while used with language based prompts. [14] proposed a 3D U-Net based diffusion model for text-to-video generation. Following this [29] proposed another text-to-video generation method that makes use of efficient 3D convolutions and temporal attention modules. They also added an

embedding in order to specify the frame rates. The authors of [1] introduced a temporal dimension to the latent space of a pretrained text-to-image diffusion model to generate videos. The work in [43] introduces a video encoder that projects a video clip to a 2D latent representation, which is further processed by a diffusion model to synthesize videos. However, diffusion models are notorious for being resource hungry and slow due to their gradual iterative denoising process at training and inference times. In our study, we have limited our comparative study to GAN-based models only.

3. Method

Our method contains two main components: (1) a temporal style generator that drives StyleGAN2's synthesis network to produce frames in time-conditioned manner, (2) two discriminators to impose content consistency and temporal consistency. Our generator is further conditioned on actor identity and action classes, though it can be used in unconditional setting.

3.1. Generator

Our generator consists of three distinct networks: a synthesis network $\mathbf{G}$, a temporal style generator $\mathbf{F_t}$, and a conditional embedding $\mathbf{F_c}$ as shown in Fig. 1. The synthesis network $\mathbf{G}$, which is based on StyleGAN2, is inherently agnostic to temporal cues when generating images. To ensure the generation of temporally coherent video frames, we introduce a specific temporal embedding, which interfaces with $\mathbf{G}$ to guide the synthesis process using a latent trajectory. This trajectory is derived from the network $\mathbf{F_t}$, which comprises a 4-layer Perceptron (MLP) that maps a random vector z_m to a k-dimensional motion style vector m. At this stage, the vector m does not encompass any temporal context. An auxiliary network *time2vec* produces sinusoidal bases that are scaled by m to finally output the temporal style vectors w_m^t.

Time2vec: Our proposed k dimensional time embedding consists of $k-1$ sinusoidal bases and a linear term as seen in Eq. 1, where the parameters w_j and ϕ_j are trainable.

$$v_j(t) = \mathcal{F}(\omega_j t + \phi_j), \tag{1}$$

where $\mathcal{F}$ is the identity function when $j = 0$ and the sine function for $1 \leq j \leq k-1$. The linear term $v_0(t)$ represents the time direction. The time t does not need to be discrete as the *time2vec* embedding is continuous. This allows us to generate videos with arbitrary frame rates. However, during the training we use integer valued time-points. We note that StyleGAN-V's time representation lacks the linear term, which might explain why its generation is plagued by unnatural repetitive motion despite its elaborate interpolation scheme. By restricting the dynamics to a fixed set of sinusoidal functions, we avoid over-fitting to the training data, and make the model more robust and generalizable to unseen data. Moreover, since sinusoidal functions are periodic, they can naturally capture cyclic patterns in the data (e.g. lip movement, hand waving). We obtain a temporal style vector as an input to $\mathbf{G}$ using the following set of equations:

$$m = \mathbf{F_t}(z_m), \tag{2}$$

$$w_m^t = m * v(t), \tag{3}$$

$$w_m^{t+1} = m * v(t+1). \tag{4}$$

The product of motion style m and temporal embedding vector $v(t)$ is a temporal style vector w_m^t. Note that a single m is used to compute temporal styles for consecutive frames. Additionally, $\mathbf{F_c}$ encodes the action and actor embeddings and outputs a content style vector w_c. It defines the general appearance of the actor along with the nature of the action. To generate a frame at time t, both temporal and content styles $[w_c, w_m^t]$ are concatenated and injected to the synthesis blocks. During the training, we generate three consecutive frames for each video element of the batch. The triplets share the same vector m while their temporal embeddings are generated from their respective time points. During the inference, a single vector m is enough to generate a long duration video. We leverage this ability of m to encapsulate the entire dynamics of a sequence to compute a temporal GAN inversion, as described in Section 5.4. A basic structure of the generator network is shown in Figure 1. To ensure the smooth integration of action-id embeddings, we employ a ramp function [28] that linearly scales the vectors derived from the action-id embedding with a factor ranging from 0 to 1, in a scheduled manner.

Unlike StyleGAN-V, we choose to stay closer to the StyleGAN's original principle, which is to allow variations in input only through the style vectors. Furthermore, our time embedding fundamentally differs from StyleGAN-V's in its design. StyleGAN-V requires multiple randomly sampled vectors to compute wave parameters which ultimately defines the motion of a generated video. In contrast, our wave parameters are independently learned and are fixed during inference. Our latent vector m interacts with the waves only as an amplitude scaling factor. Hence, our time representation is simpler and manipulable. We leverage these advantages in Section 5.4 to perform GAN-inversion of the motion style using off-the-self methods, which cannot be achieved with StyleGAN-V.

3.2. Discriminators

Shuffle discriminator: Consistency in content over time is a crucial aspect of video generation. Although the *time2vec* module in $\mathbf{G}$ provides temporal bases to guide motion learning, it does not ensure consistency in content across the sequence. In order to address this, we design

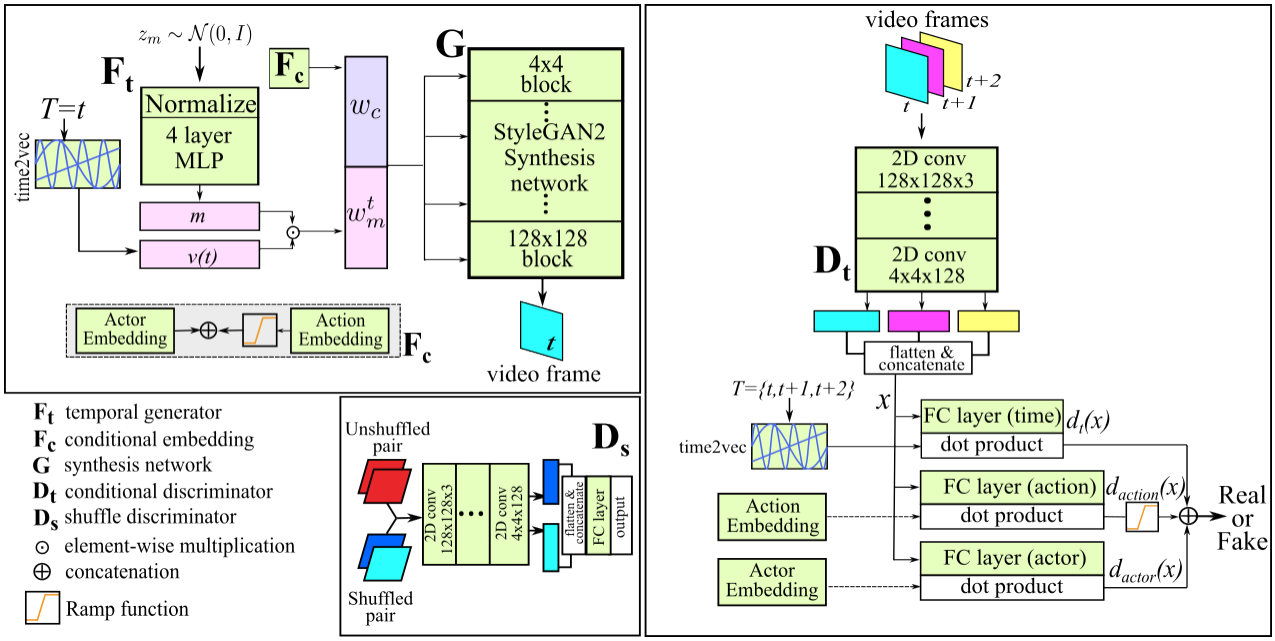


Figure 1. Proposed model: a temporal style generator F_t equipped with a *time2vec* module generates the motion code. F_c outputs a vector formed by concatenation of actor and action embeddings. Similar embeddings are activated in $\mathbf{D_t}$'s final layer. A ramp function [28], which gradually increases from 0 to 1, is used to scale the action embedding vectors in both F_c and $\mathbf{D_t}$. Here, G is the StyleGAN2's synthesis block.

a 2D-CNN based discriminator $\mathbf{D_s}$ (as seen in Figure 1) that evaluates whether the frame features are consistent or not. During the training of $\mathbf{D_s}$, each batch element consists of two frames. For the fake adversarial example, pairs of frames are shuffled among the batch to contain two different contents. In contrast, for the real example, the pairs are consecutive frames drawn from real videos. The feature maps of the pairs undergo a series of 2D convolutions, are flattened, and then concatenated into a single vector before passing through a fully connected layer. During the training of $\mathbf{G}$, a batch of unshuffled fake pairs is input to $\mathbf{D_s}$.

Conditional discriminator: To ensure temporal consistency in the generated videos, we adopt a time-conditioned discriminator, inspired by prior works of [22, 44]. The discriminator, denoted as $\mathbf{D_t}$, takes in a batch of video triplets (three consecutive frames per video) along with their respective time information, and learns to distinguish real videos from fake ones based on their temporal coherence. Then the video frames are processed by a set of 2D CNNs and a linear layer $d_t(.)$ to produce frame features. These features are then concatenated following the temporal order. $\mathbf{D_t}$ is equipped with another *time2vec* module which enforces learning of a time representation. The temporal encoding for the three input time points are also concatenated. The dot product of these concatenated vectors is computed to generate the final score [22]. Design-wise, $\mathbf{D_t}$ is similar to StyleGAN-V's discriminator as it was also inspired by the aforementioned works. However, our $\mathbf{D_t}$ learns the temporal order using absolute time information via *time2vec* in contrast to time difference conditioning used in StyleGAN-V. The use of absolute time information increases flexibility as it allows $\mathbf{D_t}$ to evaluate an arbitrary number of frames. We demonstrate this in our ablation studies where we use a single frame instead of three time frames.

As shown in Figure 1, two additional linear layers ($d_{action}(.), d_{actor}(.)$) are present at the level of $d_t(.)$, which produce actor and action representations. Dot products are computed between the corresponding embedded vector and the feature vector. The final output of the discriminator is the weighted sum of the three dot products. We use the same ramp-up function to scale $d_{action}(.)$ as in the generator [28].

4. Experimental settings

We focus on the conditional generation of videos. However, we also present the results of unconditional generation for comparative studies.

4.1. Datasets

We have used three publicly available video datasets with their labels: MEAD [37], RAVDESS [20] and UTD-MHAD [3]. Our MEAD training set contains 30 individuals talking while expressing 8 different emotions (18883

Figure 2. Here, we show few samples of generated frames by different methods. Please refer to the accompanying supplementary videos for more examples. Note that the generation of StyleGAN-V lacks natural facial motion. MOCOGAN-HD and ImaGINator's generated videos are ridden with artefacts. Meanwhile, our generation does not have these issues.

videos). We train our network only with the sequences where generic sentences are being recited. We set aside the emotion specific dialogues as unseen test sequences. For the training, we chose 128×128 image dimension and between $60 - 170$ frames as the dataset contains videos of variable length.

The RAVDESS dataset contains 24 talking faces also with 8 different emotions (not same categories as MEAD). To create a test set, we exclude sequences of 7 different emotions for four individuals. Though the dataset set contains only two dialogues, compared to over 20 dialogues in MEAD, RAVDESS contains more variation in head movements of the actors.

UTD-MHAD contains 754 videos of 8 individuals performing 27 different actions. The video frame size is 128×128 with variable video length $(33 - 81)$ as provided in the dataset. We created a test set by excluding videos of each action sequence performed by few selected target actor from the training set. Thus, we train the network to learn motion and content independently.

4.2. Baseline Methods

For the conditional video generation, we choose ImaGINator [39] as our baseline. Though it requires a conditional input image to generate videos, it is free of any additional representation like pose or motion maps. We adapted its network to output $128 \times 128 \times 32$ size image (originally $64 \times 64 \times 32$). We trained it on MEAD and UTD-MHAD datasets for up to 5K epochs.

To demonstrate that our generator does not falter in video quality, we choose MOCOGAN-HD [33] and StyleGAN-V [30] as our baselines in unconditional setting as they both use StyleGAN2's image synthesizer. For MOCOGAN-HD, we first trained a StyleGAN2 network on MEAD dataset with 256^2 image size for upto 150K iterations. Then the MOCOGAN-HD network was trained with the hyperparameters set as suggested in the author's implementation. For StyleGAN-V, we trained on with image of dimension 256^2, with a batch size of 64 and with up to 25000K images according to the author's implementation.

4.3. Training

We trained our method on a single Nvidia's A100 GPU with 80GB VRAM. The training image size was 128^2 with a batch size of 16 triplet frames. The hyperparameters for the generator, discriminators and the optimizers were kept the same as suggested in [17]. The transition factor λ of action-id vectors in both generator and discriminator started at 4000 iterations and ended at 6000 iterations, which was set empirically. We trained our model on all datasets for up to $120k$ iterations which took about 2 weeks. Our method can generate longer videos with diverse motion types and arbitrary frame rates.

5. Results

5.1. Video quality is improved

Table 1 reports the FVD scores of the generated videos by all the methods. Our conditional method (Ours(C)) scores the best which is in agreement with the videos provided in the supplementary data. Few frames of the generated video samples are depicted in Figure 2. Motion artifacts are strongly present in MOCOGAN-HD and ImaGINator's output. Though StyleGAN-V generates long duration videos, it suffers from erratic, repeated motion. Our methods (both conditional Ours(C) and unconditional Ours(UC)) produce far better results. We have reported FVD score computed over 64 frames only for StyleGAN-V and our method as other baselines are incapable of long duration video generation.

Method	$FVD_{16/64}\downarrow$	$\bar{r}_t\uparrow$	ArcFace ↑
ImaGINator	319	0.041	0.93±0.03
MOCOGAN-HD	272	0.52	0.80±0.13
StyleGANV	191/920	0.77	0.92±0.05
Ours(UC)	140	**0.79**	**0.97**±0.018
Ours(C)	**115/655**	0.7	0.96±0.02

Table 1. All the scores pertain to the training with MEAD dataset. $FVD_{16/64}$ is computed with 16 and 64 (only for StyleGAN-V and ours(C)) frames. $\bar{r}_t$ is the average correlation coefficient of the LiA signals. ArcFace is the average of the cosine similarity between the features of the first frame and the successive frames.

To assess the preservation of the actor's identity, we computed the ArcFace [6] similarity between the frames of the generated videos. ArcFace computes the cosine similarity between the feature vector of the first and the successive frames obtained from face recognition network. As seen in Table 1, our methods preserve the appearance of the actor throughout the sequence while MOCOGAN-HD is not consistent generating the same face over the sequence. The authors of [30] also made this observation.

The FVD score is widely used to evaluate video quality. However, as it is a comparison of distributions of representations in a high dimensional space, it may not accurately characterize the true quality of the video. The same can be said about the ArcFace score. Furthermore, these metrics can be influenced by factors such as spatial resolution, video length, etc. To complement these metrics, a human evaluation was conducted to assess the realism of the generated videos. To conduct the human evaluation, we generated 10 sets of videos, each consisting of 6 videos with 32 frames (1 real video and 5 generated videos using the proposed methods and the baselines). We asked 25 university students and researchers to watch 3 randomly selected sets and rank the 6 videos based on their perceived realism. The ranking distributions of the survey is presented in Figure 3. Notably, videos generated with Ours(C) and Ours(UC) models consistently ranked higher than those generated using the baseline methods. This demonstrates that our method produces more realistic videos compared to existing approaches.

5.2. Temporal style encodes temporal semantic

While we demonstrated that the video quality is improved, the aforementioned metric cannot assess the preservation of temporal semantics across different sequences. We then propose a new metric named LiA (for Lips Area) to evaluate our ability to reproduce the semantic of talking-face videos while changing the content such as the actor-id or action-id (emotion). LiA value computes the polygonal area of the lips detected using face landmark detectors [19]. A LiA signal is then obtained by computing LiA value sequentially for each frames of a generated or a real video. Though there are other factors such as eye brows and head orientation that contribute to the overall dynamics of a talking face, we focus on lip motion as it appears to be the most dynamic part of the face on this dataset. We generated 100 different sequences using different content styles and the same temporal style for the baseline methods. The average correlation coefficient $\bar{r}_t$ of the LiA signals of the generated videos by all the methods are reported in Table 1. We observed that even for the same temporal style, the ImaGINator produced different motion pattern depending on the starting input frame. MOCOGAN-HD has relatively low $\bar{r}_t$ score as the face unusually distorts over time.

Methods	StyleGAN-V	ImaGINator	Ours(C)
$FVD_{16}\downarrow$	421.32	649.23	**184.55**
(top-1, top-3)% ↑	n/a	(0,25)	**(68.5, 93.5)**

Table 2. FVD score for the UTD-MHAD dataset. Though we train StyleGAN-V unconditionally, it serves as a good baseline for assessing the video quality. We also report top-(1,3) action recognition accuracies among 27 action classes.

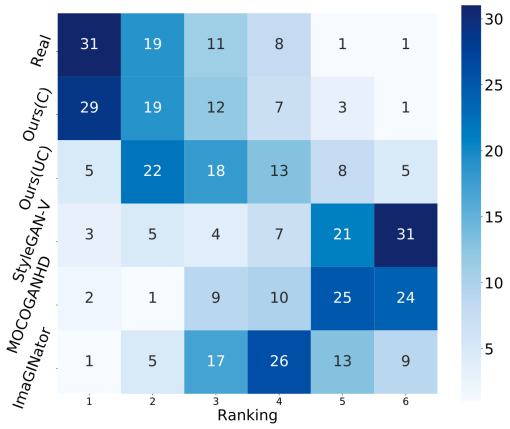
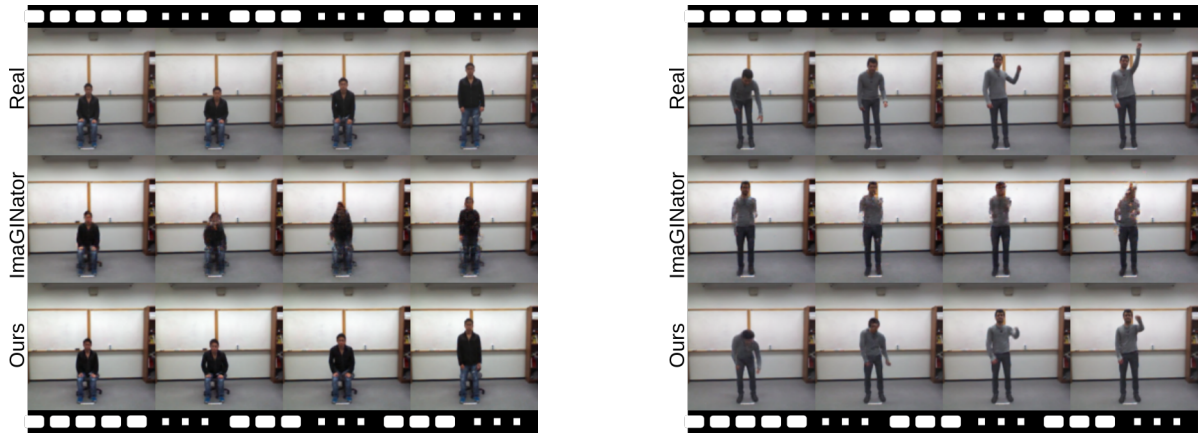


Figure 3. Human preference ranking for different videos. Videos generated by our conditional model (Ours(C)) tops the preference over other methods.

5.3. Generation of unseen coupled conditions

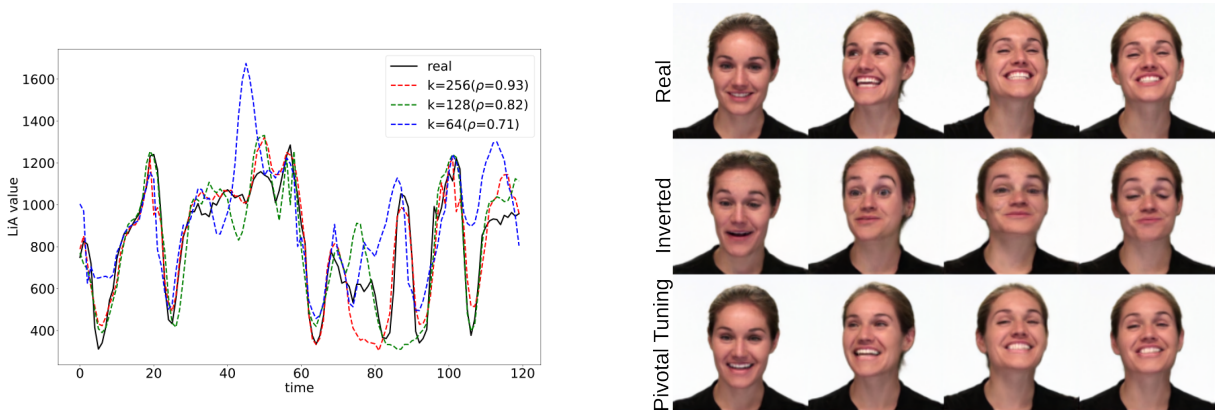
We generate videos of unseen actor-action combination only present in the test set. Figure 4 shows a few selected frames of real videos, and generated videos by ImaGINator, and our conditional method. Our method is able to successfully transfer a learnt action to an actor who was never seen performing this action in the training set. To evaluate our method in this dataset, we additionally train an action recognition model using the implementation of [8]. We train the model on skeletal key points extracted from the video frames of our training set which contains 27 different actions. The trained model was able to achieve (77%, 100%) top-1 and top-3 accuracies on the real test cases. In our generated case, it was able to achieve (68.5%, 93.5%) top-1 and top-3 accuracies. We present the confusion ma-



(a) Sit-to-Stand sequence

(b) Pick-and-throw sequence

Figure 4. Dynamics transfer in unseen conditions. Both panels display few frames sampled from videos of real sequence (top), generated sequence by ImaGINator (mid), and our method (bottom). The actors were never seen performing these actions (top) during the training.



(a) LiA signals of real, and inverted sequences

(b) GAN-inversion and Pivotal tuning

Figure 5. (a) The correlation coefficients (ρ) between LiA signals of real and inverted videos suggest that the network with a higher number of sinusoidal bases generates more faithful videos. (b) Pivotal tuning further improves the facial structure even though most of the motion is already recovered in the first step.

trices for 27 different classes in the supplementary data (in Appendix Section 4). Our model not only generated high-quality videos, as shown in Table 2, but also accurately captured many actions. On the other hand, ImaGINator performed poorly on this dataset, with evidence of mode collapse in the type of motion despite the conditioning during inference. We have included the generated videos in the supplementary data.

5.4. Motion recovery with GAN inversion

A talking face video can be generated with a random temporal style, however it is a challenging task to map a real motion to a learned temporal representation. However, thanks to our simple temporal representation m, it is possible to extract a temporal style of real lip movement, without the need of any motion computation or landmark point detection, directly by GAN-inversion. To the best of our knowledge, it is the first time a GAN-inversion for temporal styles is proposed to recover a reusable dynamic representation.

In the following experiments, we invert the temporal style of unseen videos from test cases of MEAD and RAVDESS dataset. For the MEAD dataset, we recover the motion from real video of actors pronouncing sentences which were excluded from the training set. We assume that the excluded dialogues carry unseen lip motions, and recovering such motion should demonstrate the flexibility of our temporal representation. In these experiments, conditional labels are set to the known actor and emotion of the real input sequence and only m is recovered by optimization. To this end, we minimize the sum of the LPIPS loss [46] and the

Ablation	FVD_{16}	accuracy (top-1/top-3)%
$\mathbf{D_t}$(1 time-point)	**179.73**	59.2/79.6
$\mathbf{D_t}$(3 time-points)	184.55	**68.5/93.5**
w/o $\mathbf{D_t}$	526.21	42.6/80.5

Table 3. FVD and classification accuracy for UTD-MHAD with three different versions of our model.

MSE between N real and generated frames:

$$m^* = \arg\min_m \sum_{t=0}^{N} \mathcal{L}(I(t), G([w_c, w_m^t])), \qquad (5)$$

where, $I(t)$ is the real video frame at time t, $\mathcal{L}$ is the summation of the two losses, and m^* is the optimized motion style vector. Figure 5a shows an example of LiA signals for real and inverted videos using our model trained with $k = 64, 128, 256$. The higher the number of sinusoidal bases used in the generator, the more faithful the recovered motion is to the real video. We performed the inversion and LiA signal analysis for 39 different emotion specific sentences excluded from the training set (see Appendix Section 6 for the complete list), and report the average correlation to be 0.6, 0.79 and 0.91 for $k = 64, 128, 256$ respectively. The evaluation was done for input videos with 120 frames because of the memory limitation. This implies that a single vector m^* can faithfully represent the dynamic of at least up to 120 frames. Furthermore, in Figure 5b, the inversion is able to recover the large movement of head in the RAVDESS dataset. The facial structure is further improved using pivotal tuning [25] where we adjust generator's weight by fixing the previously optimized m^*. Thus the recovered motion in the form of m^* can then be transferred to another actor. We believe this is a novel way for re-enactment between different actors and actions.

5.5. Interpolating conditions over time

Because our content and motion spaces are highly disentangled, it is possible to edit the attributes of the videos over time. We choreograph a sequence where actors change their expression over time by a linear interpolation in the action embedding space (see supplementary videos). The interpolation does not interfere with the general motion.

6. Ablation

In our ablation studies, we investigate the impact of different components on the performance of our model. Using a higher number of sinusoidal bases improves the recovered motion with GAN-inversion as discussed in the previous section. However, higher number of k leads to small intermittent motion artefacts of eyes in MEAD dataset. For $k = 128$, most of the artefacts are unnoticeable. We trained $\mathbf{D_t}$ using only one time point instead of three time points, which resulted in a decrease in action recognition accuracy from 68% to 57% for unseen conditions. Secondly, we removed $\mathbf{D_s}$ in the training on the MEAD dataset, which led to an FVD score of 600. We report the affect of tweaking of $\mathbf{D_t}$ on UTD-MHAD dataset in Table 3. We also examined the effect of using a ramp function to schedule scaling of the action-id vectors. We found that without the ramp function, introducing the action-id at the beginning of the training caused the generator to favor one class over the other, while using the ramp function stabilized the quality of the videos for all classes.

7. Conclusion

In this study, we proposed a video generation model which produces high quality videos in both conditional and unconditional settings. Through various experiments, we show that the temporal style can independently encode the dynamics of the training data and can be transferred to unseen targets. We demonstrated that it is possible to generate different types of action with high accuracy as seen in UTD-MHAD videos. Our generator produces videos with better fidelity than the prevalent style-based video generation methods as shown by various metrics as well as human preference score. We demonstrate that our method can recover motion of real input videos via GAN-inversion and can faithfully encode the motion of at least 120 frames with a single temporal style vector. The supplementary material includes further discussion on the limitations and future work. A PyTorch implementation of this work can be found in our project's web page at sandman002.github.io/CTSVG.

8. Acknowledgement

This work has received support under the program "Investissements d'Avenir" launched by the French Government and implemented by the ANR, with the references: ANR-10-LABX-54 MEMO LIFE ANR-11-IDEX-0001-02 PSL*. S.M. was funded by Inserm ITMO Cancer - TOTEM. This work was granted access to the HPC resources of IDRIS under the allocation 2020-AD011011495 made by GENCI.

References

[1] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 3

[2] 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] Chen Chen, Jafari Roozbeh, and Kehtarnavaz Nasser. Utd-mhad: A multimodal dataset for human action recognition

utilizing a depth camera and a wearable inertial sensor. In *ICIP*, 2015. 2, 4

[4] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *CVPR*, June 2021. 2

[5] Zeyuan Chen, Yinbo Chen, Jingwen Liu, Xingqian Xu, Vidit Goel, Zhangyang Wang, Humphrey Shi, and Xiaolong Wang. Videoinr: Learning video implicit neural representation for
continuous space-time super-resolution. *CVPR*, 2022. 2

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 6

[7] Emily L Denton and vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NeurIPS*, 2017. 1, 2

[8] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *CVPR*, 2022. 6

[9] Chelsea Finn, Ian J. Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *NeurIPS*, 2016. 2

[10] Gereon Fox, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. Stylevideogan: A temporal generative model using a pretrained stylegan, 2021. 1

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[12] Will Grathwohl and Aaron Wilson. Disentangling space and time in video with hierarchical variational auto-encoders. *CoRR*, 2016. 1

[13] Berk Hess. Convergence of sampling in protein simulations. 2002. 2

[14] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. 2

[15] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik G. Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 2018. 2

[16] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *NeurIPS*, 2021. 2

[17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 1, 2, 5

[18] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus A. Brubaker. Time2vec: Learning a vector representation of time. *CoRR*, 2019. 2

[19] Davis E. King. Dlib-ml: A machine learning toolkit. *J. Mach. Learn. Res.*, 2009. 6

[20] Steven R. Livingstone and Frank A. Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PLOS ONE*, 13:1–35, 2018. 2, 4

[21] Michael Mathieu, Camille Couprie, and Yann Lecun. Deep multi-scale video prediction beyond mean square error. 2016. 2

[22] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *ICLR*, 2018. 4

[23] Andres Munoz, Mohammadreza Zolfaghari, Max Argus, and Thomas Brox. Temporal shift gan for large scale video generation. In *WACV*, January 2021. 2

[24] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *CVPR*, 2017. 2

[25] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021. 2, 8

[26] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *ICCV*, 2017. 1, 2

[27] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan, 2020. 2

[28] Mohamad Shahbazi, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Collapse by conditioning: Training class-conditional GANs with limited data. In *ICLR*, 2022. 3, 4

[29] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023. 2

[30] 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, 5, 6

[31] Kritaphat Songsri-in and Stefanos Zafeiriou. Face video generation from a single image and landmarks. *FG*, 2019. 2

[32] 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

[33] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018. 1, 2, 5

[34] 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. 2

[35] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 2

[36] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 2

[37] Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen Qian, Ran He, Yu Qiao, and Chen Change Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *ECCV*, 2020. 2, 4

[38] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 2

[39] Yaohui WANG, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. Imaginator: Conditional spatio-temporal gan for video generation. In *WACV*, March 2020. 2, 5

[40] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *ECCV*, 2018. 2

[41] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P. Allebach, and Chenliang Xu. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In *CVPR*, 2020. 2

[42] Yangyang Xu, Yong Du, Wenpeng Xiao, Xuemiao Xu, and Shengfeng He. From continuity to editability: Inverting gans with consecutive images. In *ICCV*, 2021. 1, 2

[43] Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. In *CVPR*, 2023. 3

[44] 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, 4

[45] Polina Zablotskaia, Aliaksandr Siarohin, Bo Zhao, and Leonid Sigal. Dwnet: Dense warp-based network for pose-guided human video generation. In *BMVC*, 2019. 1

[46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[47] Jian Zhao and Hui Zhang. Thin-plate spline motion model for image animation. *CVPR*, 2022. 1, 2