StyleCineGAN: Landscape Cinemagraph Generation using a Pre-trained StyleGAN

Jongwoo Choi Kwanggyoon Seo Amirsaman Ashtari Junyong Noh
Visual Media Lab, KAIST
{jeolpyeoni0,skg1023,a.s.ashtari,junyongnoh}@kaist.ac.kr

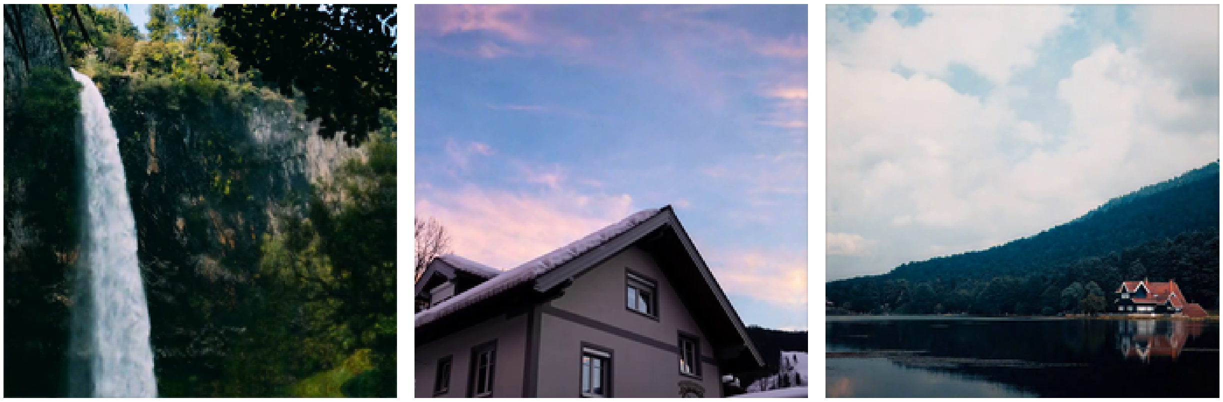

Figure 1. Given a landscape image, StyleCineGAN generates a seamless cinemagraph at 1024×1024 resolution. **This figure contains video clips, thus consider viewing it using Adobe Reader**. The same results are also included in the supplementary video.

Abstract

We propose a method that can generate cinemagraphs automatically from a still landscape image using a pre-trained StyleGAN. Inspired by the success of recent unconditional video generation, we leverage a powerful pre-trained image generator to synthesize high-quality cinemagraphs. Unlike previous approaches that mainly utilize the latent space of a pre-trained StyleGAN, our approach utilizes its deep feature space for both GAN inversion and cinemagraph generation. Specifically, we propose multi-scale deep feature warping (MSDFW), which warps the intermediate features of a pre-trained StyleGAN at different resolutions. By using MSDFW, the generated cinemagraphs are of high resolution and exhibit plausible looping animation. We demonstrate the superiority of our method through user studies and quantitative comparisons with state-of-the-art cinemagraph generation methods and a video generation method that uses a pre-trained StyleGAN.

1. Introduction

Cinemagraph is a unique form of media that combines a still image and video. While most of the scene remains still, subtle and repeated movements in only a small region effectively highlight a particular object or an important event that occurred at the time of capture. Because this mixture of still and moving elements in one scene creates an interesting eye-catching effect, cinemagraphs have recently attracted increasing attention in social media.

Despite the growing popularity of cinemagraph, its creation heavily relies on a manual process. To create a motion in part of the input image, one typically utilizes an image manipulation tool to stretch, scale, rotate, or shift the part of the image in a physically plausible and aesthetically pleasing way. This process is often time-consuming and requires a high level of skill in the use of image editing tools. Therefore, creating a cinemagraph has generally been considered to be a personal project of professionals so far.

To allow ordinary users to create a cinemagraph with a single image, automatic methods have been proposed. One line of research uses a reference video as guidance [14, 25, 33]. These methods are capable of producing realistic motions in the target scene similar to a reference video. Recent methods obviate the need for a reference video, by training deep generative models [4, 6, 9, 15, 17–19, 23]. These methods decompose the task into two processes of learning motion and spatial information individually from separate datasets. The decomposition effectively reduces the complexity of simultaneously learning both temporal and spatial information, and improves the perceptual quality of the generated videos. However, these models must be trained from scratch, which sometimes requires several days if not weeks on a modern GPU. Moreover, these models are not specifically designed to generate high-resolution of 1024×1024 cinemagraphs, because of the significant requirements in memory and processing power.

In this paper, we propose the first approach to high-quality one-shot landscape cinemagraph generation based on a pre-trained StyleGAN [12]. By using the learned image prior of StyleGAN, our method removes the need for training a large model from scratch and systematically improves the resolution of the generated cinemagraphs to 1024×1024. Moreover, our method enables a user to easily edit the style and appearance of the generated cinemagraph by leveraging the properties of StyleGAN for image stylization and editing.

Our method is inspired by recent unconditional video generation methods [7, 34] which allow to navigate the latent space of a pre-trained image generator, to synthesize a high-quality temporally coherent video. Unlike these methods that utilize the latent codes of StyleGAN, we opt to use the deep features that are generated by convolution operations in each layer of StyleGAN. We use these deep features for two reasons. First, we observed that highly detailed landscape images cannot be reconstructed accurately from the latent codes using GAN inversion methods because these latent codes are low-dimensional [30]. Second, a plausible motion that preserves the content cannot be created by only navigating the latent space, because this space is highly semantic-condensed and lacks explicit spatial prior [38].

To solve these problems, for the first time in the cinemagraph generation domain, we utilize the deep features of a pre-trained StyleGAN. These deep features preserve spatial information and encode both high-level semantic and low-level style appearances across high and low-level convolutional layers. To produce cinemagraphs from these deep features, we propose a multi-scale deep feature warping (MSDFW) to apply the motions generated using a motion generator to the deep feature space of StyleGAN.

We demonstrate the effectiveness and advantages of our method by comparing it to state-of-the-art methods in cinemagraph generation [6, 9, 17, 19] and to an unconditional video generation method that uses a pre-trained StyleGAN [34], using various metrics. We also performed a user study to verify the effectiveness of our method for creating visually pleasing cinemagraph effects. Both qualitative and quantitative results confirm that our method substantially outperforms all existing baselines.

2. Related Work

2.1. Cinemagraph Generation

One of the early studies for cinemagraph generation used a procedural approach to decompose each object into layers and apply time-varying stochastic motion [5]. This method can handle a diverse range of images such as photos and paintings by relying on manual interaction from the user. Another approach is to use a reference video to animate the given image [20–22, 25]. These methods rely on a statistical motion analysis of videos to transfer their periodic motion property to the desired image. Another method distills a dynamic NeRF to render looping 3D video textures from the representative motion found in the reference video [14]. Without any reference video as guidance, Halperin et al. [8] proposed a framework to animate arbitrary objects with periodic patterns in a simple motion direction.

Recent approaches use the capacity of deep learning to automatically create cinemagraphs from a single image. One study trained an image-based renderer to generate water animation, utilizing the water simulation results [32]. Another research first predicts a sequence of normal maps, then generates the corresponding RGB images that show a looping animation of a garment as if it is blown in the wind [4]. Another line of research trains a generator to produce a motion field [6, 9, 15, 18, 19]. Our work is similar to these deep learning-based approaches in that a motion generator is utilized to synthesize the cinemagraphs. The difference is that our method leverages the deep features of a pre-trained image generator and thus can generate plausible high-quality results without fine-tuning or training a separate synthesis network. At the same time, our method systematically improves the resolution of the generated cinemagraphs to 1024×1024.

2.2. Unconditional Video Generation

The task of video generation is known to be difficult because the generation process has to take into account both spatial and temporal information. To ease the problem, previous approaches [27, 36] decompose video generation into content and motion generation. These methods first predict the latent trajectory for motion, then generate a video from the set of predicted latent codes using the image generator. Instead of training the generation model from scratch, MoCoGAN-HD [34] and StyleVideoGAN [7] leverage a pre-trained image generator model, StyleGAN [12]. These methods use the property of well-constructed latent space to morph one frame toward the next frames [10]. Utilizing the image generation capability of a pre-trained generator, MoCoGAN-HD and StyleVideoGAN only need to train a motion trajectory generator, which greatly reduces the training time. These methods can generate high-quality videos in multiple domains such as faces, cars, and even sky. However, the content details are not well preserved, which has hindered the methods' application for cinemagraph generation. Our method builds upon a similar concept of using a pre-trained StyleGAN. We leverage the deep features instead of the latent codes of StyleGAN to generate looping and content-preserving cinemagraphs.

2.3. Video Synthesis using pre-trained StyleGAN

A pre-trained generative model, StyleGAN [11–13], has been actively employed for downstream image and video editing applications. For GAN-based applications, one of the essential steps is to project the desired images or videos to the latent space of the pre-trained GAN model, which is known as GAN inversion. The latent codes are obtained using either an optimization technique [1, 2], a trained encoder [26, 35], or a hybrid of both approaches [16]. For video editing applications, previous methods [3, 29, 37, 42] have utilized a pre-trained image encoder to project all of the frames to the latent space. While most of the methods operate in $\mathcal{W}+$ space [2], we opt to use both w^+ and deep features [24, 43, 44] of the pre-trained StyleGAN to accurately reconstruct the original image and synthesize motion.

3. Methods

An overview of our method is shown in Figure 2. Given a landscape image I, our method generates a seamlessly looping cinemagraph $V = \{\hat{I}_0, ..., \hat{I}_N\}$ using a pre-trained StyleGAN. At the core of our method, we use a pre-trained generator without additional fine-tuning. The overall process is described below.

(A) First, we project the landscape image into both the latent space and the feature space of StyleGAN. To this end, we train an encoder that outputs both latent codes w^+ and intermediate features D^{10} of StyleGAN (Sec. 3.1).

(B) In addition, we predict a mask S to divide the image into static and dynamic regions (Sec. 3.2).

(C) Next, we use a motion generator that accepts the landscape image I to synthesize a motion field M, which defines the position of each pixel in the future frames (Sec. 3.3).

(D) Lastly, we generate the final cinemagraph frames using the pre-trained StyleGAN with deep feature warping (DFW) operation added in between the original layers. (Sec. 3.4).

In the following sections, we will describe the details of each process.

3.1. GAN Inversion

The first step for generating a cinemagraph is to project the input image to the latent space of a pre-trained StyleGAN. This process is necessary because StyleGAN is an unconditional generator that cannot take a conditioning input. Many previous methods have used $\mathcal{W}+$ [2] latent space, which is an extended space of native latent space $\mathcal{W}$. However, we observed that $w^+ \in \mathcal{W}+$ is not expressive enough to reconstruct the original high-frequency details of the landscape images. Therefore, we chose to use the generated deep features D^{10} of StyleGAN as well as w^+. The use of D^{10} enables us to recover details in the original input as shown in Figure 8.

To project an image I to both the latent space and feature space and obtain w^+ and D^{10}, we train an encoder E similarly to Yao et al. [43], in which the encoder takes I as input and predicts (w^+, D^{10}). Additional training details can be found in the supplementary material.

3.2. Mask Prediction

We improve the quality of GAN inversion described in Sec. 3.1 by training an additional classifier that predicts a mask to separate the static and dynamic regions. Using the mask, the structure of the static regions can be preserved. To this end, we train a multi-layer-perceptron (MLP) classifier that accepts the deep features of StyleGAN as its input and outputs the mask that specifies each dynamic region in the image, as performed in DatasetGAN [46].

To train the classifier, we manually annotate 32 segmentation masks S of the selected images. We then follow the same procedure as DatasetGAN but using the deep features D^i where $i \in \{10, 11, ..., 18\}$. We resize all these deep features followed by concatenating them in the channel dimension to construct the input feature D^*. In the end, paired data (D^*, S) is constructed. An ensemble of 10 MLP classifiers is trained with a cross-entropy loss. To further improve the performance of the previous work, at inference, we also refine the predicted mask. For additional details, please refer to the supplementary material.

3.3. Motion Generation

To generate motion from a static landscape image I, we use an Eulerian motion field M which represents the immediate velocity of each pixel. To predict M given I, an image-to-image translation network [39] is trained as a motion generator. We train the network with paired data (I, M) as performed in the previous method [9]. Adding controls in motion generation is also possible, either by using text prompts [19] or drawing arrows [18].

While the trained network works well to a certain degree, we observed that the predicted motion field M does not align with the boundaries between the static and dynamic regions and often contains motion in static regions. Thus, we further refine M by multiplying it with the segmentation mask S, which effectively removes most errors. The refined M is then used to simulate the motion of a pixel at time t using the Euler integration as follows:

$$\begin{aligned} &x_{t+1} = x_t + M(x_t), \quad M(x_t) = F_{t\to t+1}(x_t), \\ &F_{0\to t}(x_0) = F_{0\to t-1}(x_0) + M(x_0 + F_{0\to t-1}(x_0)), \end{aligned} \tag{1}$$

where x_0 is the source pixel position at time 0, $F_{0\to t}$ is the displacement field that defines the displacement of the pixel position x_0 to the pixel position x_t at time t.

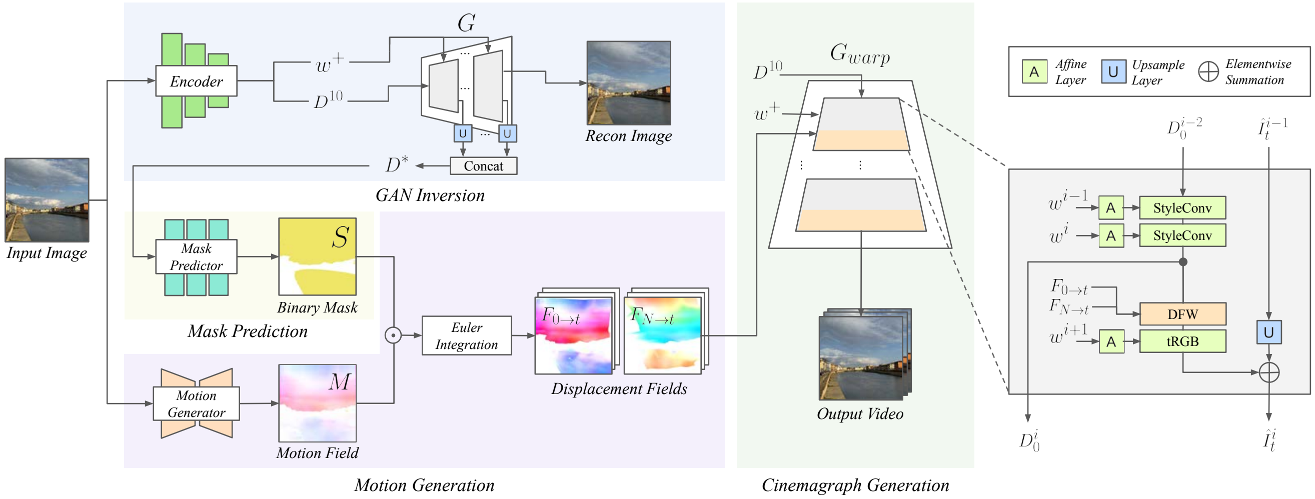

Figure 2. **Overview of StyleCineGAN.** Given an input landscape image I, our goal is to generate a cienemagraph using a fixed pre-trained StyleGAN G. We project the image into both latent codes w^+ and deep features D^{10} of G. Using the deep features D^*, a mask predictor predicts a segmentation mask S. To animate the input image, we use a motion generator to predict the motion field M from I. M is refined using S. Through Euler integration, M produces the future and past displacement fields $F_{0\to t}$ and $F_{N\to t}$. To synthesize cinemagraph frames, we add a DFW layer in between the layers of G. DFW refers to Eqns. 2 and 3. This modification enables the intermediate features of G to be warped according to $F_{0\to t}$ and $F_{N\to t}$ using a joint splatting method at different resolutions, specifically for the StyleGAN layers indexed with $i \in [10, 12, 14, 16, 18]$. The warped deep features are used to synthesize frames $\hat{I}_t$ resulting in the final cinemagraph video.

3.4. Cinemagraph Generation

After acquiring the displacement field $F_{0\to t}$ (Sec. 3.3), latent codes w^+, and deep features D^{10} of StyleGAN (Sec. 3.1), we feed these data into the pre-trained StyleGAN with our DFW layer added. We apply forward warping (or splatting) to the original content to generate a cinemagraph. Here, the application of forward warping directly to an RGB image often results in images with tearing artifacts. To reduce the artifacts, we warp the deep features D_0^i of StyleGAN in different resolutions or scales:

$$D_t^i = Warp(D_0^i, F_{0\to t}), \tag{2}$$

where $Warp$ is the forward warping function and D_t^i is the warped deep feature at time t and scale i. We observed that warping a single deep feature (e.g., only D_0^{10}) results in blurry textures in the generated cinemagraphs. Therefore, we opt to warp the multi-scale deep features; we call this operation a MSDFW. Specifically, we apply warping to the deep features D_0^i where $i \in [10, 12, 14, 16, 18]$. The deep feature D_0^{10} is acquired using GAN inversion, and the consequent deep features $D_0^{12}, D_0^{14}, ...,$ and D_0^{18} are subsequently generated using both D_0^{10} and w^+, as shown in the rightmost column of Figure 2.

Looping video V with frame length $N+1$ is synthesized using the joint splatting technique [9] using the future and past displacement fields $F_{0\to t}$ and $F_{N\to t}$, generated through Euler integration. The displacement fields are computed with t times of integration on M for $F_{0\to t}$, and $N-t$ times of integration on $-M$ for $F_{N\to t}$. The multi-scale feature D_0^i is warped in both directions and is composited to form D_t^i. Specifically, D_t^i is computed as a weighted sum of $Warp(D_0^i, F_{0\to t})$ and $Warp(D_0^i, F_{N\to t})$ as follows:

$$\begin{aligned} D_t^i(x') = & \frac{\sum_{x\in\chi} \alpha_t \cdot Warp(D_0^i(x), F_{0\to t})}{\sum_{x\in\chi} \alpha_t} \\ & + \frac{\sum_{x\in\chi} (1-\alpha_t) \cdot Warp(D_0^i(x), F_{N\to t})}{\sum_{x\in\chi} (1-\alpha_t)}, \end{aligned} \tag{3}$$

where $x \in \chi$ is a set of pixels being mapped to the same destination pixel x', and α_t is the looping weight defined as $(1 - \frac{t}{N})$.

With the above DFW module, we can generate a video frame given the predicted deep features D^{10}, latent code w^+, and mask S:

$$\begin{aligned} \hat{I}_t = S \odot (G_{warp}(D_0^{10}, w^+, F_{0\to t}, F_{N\to t})) \\ + (\mathbf{1} - S) \odot I, \end{aligned} \tag{4}$$

where $\odot$ is an element-wise multiplication and G_{warp} is a fixed pre-trained StyleGAN that incorporates our DFW module.

Style Interpolation In addition to motion generation, the change of appearance style is an additional feature often observed in a landscape cinemagraph. This can be achieved by interpolating the latent code with the target latent code as follows:

$$w_s^+ = w^+ \cdot (1-\beta) + w_t^+ \cdot \beta, \tag{5}$$

Figure 3. Generated cinemagraph results. **This figure contains video clips, thus consider viewing it using Adobe Reader.** The first two are cinemagraphs without appearance change, and the last two are cinemagraphs with appearance change.

where w_s^+ is the interpolated latent code, w_t^+ is the latent code of the target style, and β is the interpolation weight. This is possible because the later layers of the StyleGAN only modify the color of the synthesized image while the original structure is preserved.

With modified Equation 4, we prevent the visual mismatch between the static and dynamic regions by also reflecting the changes to the static regions as follows:

$$\begin{aligned}\hat{I}_t = S \odot (G_{warp}(D_0^{10}, w_s^+, F_{0\to t}, F_{N\to t})) \\ + (\mathbf{1} - S) \odot (I + \Delta I_s),\end{aligned} \tag{6}$$

$$\Delta I_s = G(D_0^{10}, w_s^+) - G(D_0^{10}, w^+), \tag{7}$$

where ΔI_s is the color difference between the two images generated from latent codes w_s^+ and w^+.

4. Experiments

In this section, we compare our method with state-of-the-art landscape cinemagraph generation methods [6, 9, 17, 19] (Sec. 4.1) and with an unconditional video generation method [34] (Sec. 4.2). In addition, we show the importance of the components used in our method through ablation studies (Sec. 4.3). To observe more various results, we recommend readers see the supplementary video. Example results are presented in Figures 1 and 3.

4.1. Comparisons with Cinemagraph Generation Methods

We compared our method with state-of-the-art cinemagraph generation methods: Animating Landscape (AL) [6], Deep Landscape (DL) [17], Eulerian Motion Field (EMF) [9], and Text2Cinemagraph (T2C) [19]. AL is a learning-based method that uses backward warping to generate cinemagraph frames. DL trains a style-based generator to synthesize images and dynamic motion in a scene. EMF and T2C train an encoder-decoder architecture paired with a motion generator to produce looping cinemagraphs. We used official implementations of AL, DL, and T2C, and faithfully reproduced EMF based on the provided training details and hyper-parameters. In the following, we will examine the qualitative difference, evaluate the results using two metrics, and report the results of human perceptual studies.

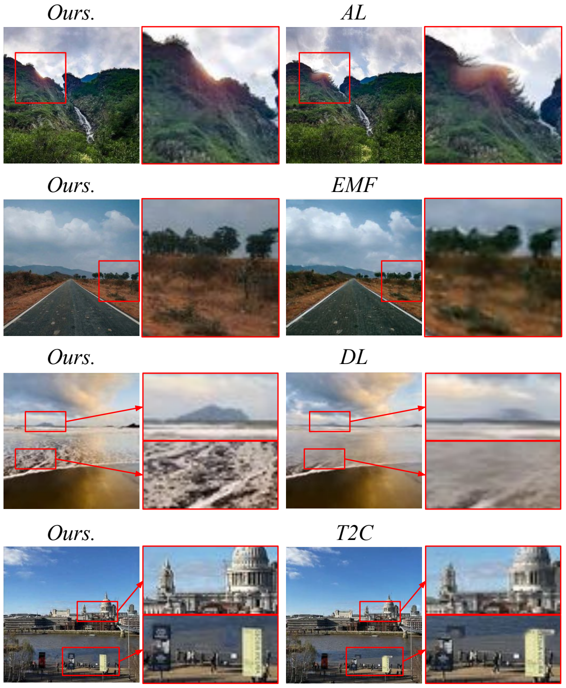

Figure 4. Qualitative comparison with state-of-the-art cinemagraph generation methods. Please refer to the supplementary video for more examples.

Table 1. Quantitative comparison with the state-of-the-art cinemagraph generation methods. We compared our method with AL, EMF, DL, and T2C. The best scores are bolded.

	Static Consistency			Motion Quality			
Method	LPIPS↓	MS-SSIM↑	RMSE↓	LPIPS↓	MS-SSIM↑	RMSE↓	FID↓
AL	0.0477	0.9524	8.8956	0.0617	0.6819	25.1089	53.6893
EMF	0.0103	0.9723	6.0440	0.0533	0.7102	22.8932	45.8035
DL	0.0071	0.9931	2.1044	0.0504	0.7062	21.8011	41.6374
T2C	0.0063	0.9773	4.8785	0.0159	0.7224	21.2784	40.1186
Ours	**0.0062**	**0.9962**	**1.9430**	**0.0131**	**0.7237**	**20.8948**	**39.2113**

Qualitative Comparisons Figure 4 shows visual results from AL, EMF, DL, T2C, and our method. AL exhibits stretching artifacts (row 1) due to recurrent backward warping, whereas our method uses forward warping that prevents the appearance of such distortions. EMF struggles with precise reconstruction (row 2), because its encoder-decoder architecture is trained from scratch on a low-quality video dataset. In contrast, our method leverages a StyleGAN model pre-trained on high-quality images, resulting in cinemagraphs with high perceptual quality. DL often loses fine details in textures (row 3) as it performs GAN inversion using the latent codes of StyleGAN, and fine-tunes the generator. Our method, utilizing deep features for GAN inversion, retains original texture details. T2C encounters difficulties in preserving high-frequency details, and accurate motion segmentation between static and dynamic regions (row 4). In contrast, our method utilizes a pre-trained StyleGAN for cinemagraph generation, and uses its features for mask prediction, preserving both fine and structural details of the source image.

Quantitative Comparisons The evaluation was performed considering two aspects: (1) static consistency and (2) motion quality. Static consistency measures the consistency over time of non-movable regions such as buildings and mountains. Motion quality refers to both the image quality of the animated regions and the animation plausibility. For a fair comparison, we normalized the speed of motion in the generated results for all methods to match the average speed in the test dataset. AL, EMF, DL, and T2C were provided with the first frame I_0 of the test video to generate 512×512 videos. Our method generates 1024×1024 results, thus we downsampled it to the size of 512×512.

For a quantitative evaluation, we used 224 test videos from the Sky Time-Lapse dataset [41]. To measure the static consistency, we computed LPIPS [45], Root-Mean-Squared Error (RMSE), and MS-SSIM [40] between generated frame $\hat{I}_n$ and the input image I_0 with the sky masked out. To measure the motion quality, we used the same evaluation metrics along with Fréchet inception distance (FID) [28] between $\hat{I}_n$ and the ground-truth frame I_n with the static parts masked out.

Table 1 reveals the quantitative evaluation results. For both static consistency and motion quality, our method outperforms the other approaches. Our cinemagraphs represent static consistency better because use of the mask improves the accuracy in the detection of static regions. In addition, the quality of texture details are improved in the generated frames due to GAN inversion and the MSDFW based on the deep features of StyleGAN. The use of our motion generator and DFW leads to improved motion quality for each scene, compared with that of the results of previous approaches.

User Study We conducted a user study with 17 participants to subjectively evaluate our method in comparison with previous cinemagraph generation methods. We did not target any specific demographic groups, as the evaluation of cinemagraphs should not be dependent on particular demographic distributions. We conducted two evaluations: score evaluation and side-by-side comparison. Both evaluations assessed static consistency and motion quality. In the score evaluation, participants rated the video presented with a reference image on a 1-to-5 scale, with ”very bad” being 1 and ”very good” being 5. In the side-by-side comparison, participants selected the preferred cinemagraph from two videos generated using different methods, presented with a reference image. For both evaluations, we used ten samples randomly chosen from the LHQ [31] dataset. To eliminate bias, distinct samples were used for each evaluation, and the positions of the cinemagraphs were randomized in all questions.

Tables 2 and 3 summarize the statistics of the user studies. For both score evaluation and human preference, our method outperforms the previous approaches by a substantial margin. We also performed statistical analysis (Kruskal-Wallis H-test) on the resulting evaluation scores. The results showed that our method achieved significantly higher scores in all pairs of comparisons with every previous method (p<0.001 in post-hoc analysis). Both user study results reveal the advantages of our method for generating cinemagraphs with high image quality and plausible animation.

Table 2. Human perceptual study results for score evaluation. The best scores are bolded.

Method	Static Consistency	Motion Quality
AL	1.59 ± 0.20	1.76 ± 0.45
EMF	2.35 ± 0.89	2.24 ± 0.99
DL	3.35 ± 0.83	3.25 ± 0.67
T2C	3.75 ± 0.69	2.96 ± 0.66
Ours	**4.37 ± 0.18**	**3.86 ± 0.53**

Table 3. Human perceptual study results for side-by-side comparison. The percentage of ours chosen against each competing method is reported.

	Human Preference (Ours %)	
Method	Static Consistency	Motion Quality
vs AL	**99.35%**	**92.81%**
vs EMF	**96.08%**	**86.93%**
vs DL	**94.77%**	**84.97%**
vs T2C	**77.78%**	**73.86%**

4.2. Comparisons with Video Generation Methods

We compared our approach with MoCoGAN-HD [34], a video generation method that predicts the trajectory within the latent space of a pre-trained StyleGAN. For content generation, we used the same pre-trained StyleGAN as that used by MoCoGAN-HD, which was trained on the Sky Time-lapse [41] dataset to generate 128×128 images. The comparison was made in an unconditional manner using 300 randomly sampled latent codes, at a resolution of 128×128. We compared the first 60 frames of the generated videos with those of MoCoGAN-HD, as our method produces looping videos.

Qualitative Comparisons Figure 5 presents a qualitative comparison. As shown in the first and third rows, the videos generated using our method exhibit static consistencies, with the clouds moving while the ground remains static. In contrast, as shown in the second and fourth rows, the frames generated by MoCoGAN-HD deviate significantly from the original image, making the method unsuitable for cinemagraph generation.

Quantitative Comparisons We compared the content preservation ability of both methods by measuring LPIPS [45], RMSE, and MS-SSIM [40] between the first frame $\hat{I}_0$ and the subsequent frames $\hat{I}_n$. Because the main content to be preserved in cinemagraphs is the static regions, we defined content preservation as static consistency. We masked out the sky for all image frames according to the

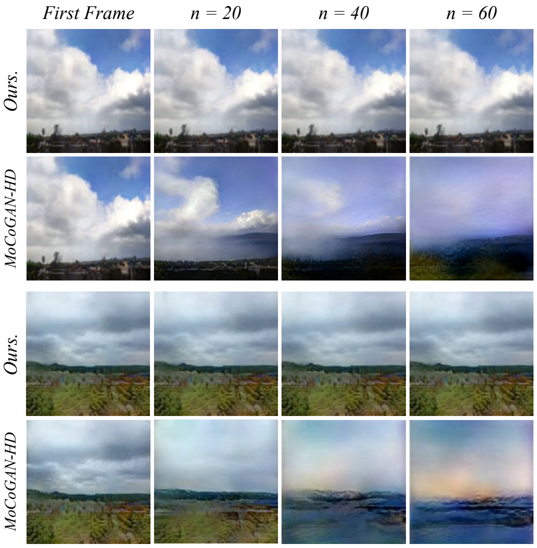

Figure 5. Qualitative comparison with the state-of-the-art video generation method, MoCoGAN-HD [34]. For more examples from this comparison, please refer to the supplementary video.

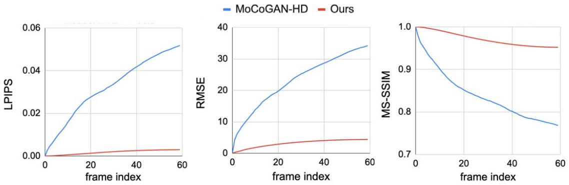

Figure 6. Quantitative comparison of content preservation with state-of-the-art video generation method MoCoGAN-HD [34].

segmentation mask and compared only the static parts. We used 219 videos in which the scene could be horizontally divided into static and animated regions. LPIPS, RMSE, and MS-SSIM were all computed for each pair of $\hat{I}_0$ and $\hat{I}_n$, and were averaged over the number of samples. Figure 6 shows that MoCoGAN-HD exhibits significant divergence in terms of LPIPS, RMSE, and MS-SSIM over time. This indicates that MoCoGAN-HD is unable to generate motion that preserves the content of the original image. In contrast, our method exhibits a small diverging trend, confirming its superiority in preserving the content of the original image over time, by utilizing deep features.

4.3. Ablation Study

To demonstrate the effectiveness of our design choices for the proposed method, we conducted a series of ablation studies. Specifically, we focused on the effectiveness of our warping and GAN inversion methods. To evaluate warping, we compared the image frames generated with and without the use of 1) forward warping, 2) DFW, 3) MSDFW, and 4) segmentation mask. The first frames of 224 test videos from the Sky Time-Lapse [41] dataset were given as input to generate 1024×1024 videos, and we used the first 60 frames for the evaluation. For the evaluation of GAN inversion, we compared images reconstructed with and without the use of the deep features. A total of 256 images, 128 from the Sky Time-Lapse and 128 from the Eulerian [9] dataset were provided to generate 1024×1024 reconstructed images.

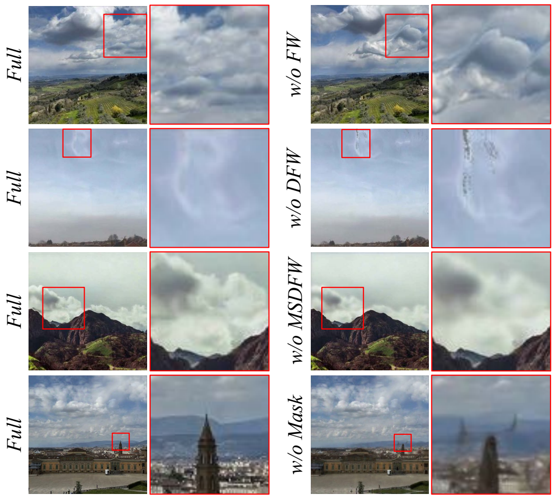

Figure 7. Results of qualitative comparison in ablation study. Please see the supplementary video for the animated results.

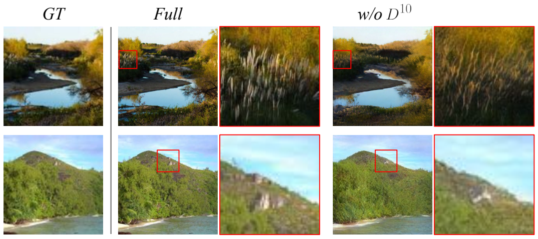

Figure 8. Qualitative comparison results on GAN inversion.

Forward Warping We compared the image frames generated with and without forward warping (FW). For the case without FW, we warped the deep features of StyleGAN using backward warping. The first row in Figure 7 shows the qualitative results of this comparison. As revealed in the figure, without FW, the results usually contain unrealistically stretched textures. For quantitative comparisons, we computed LPIPS, MS-SSIM, and RMSE between I_n and $\hat{I}_n$. Table 4 shows that excluding forward warping resulted in significant degradation in terms of MS-SSIM. This indicates that the stretched textures resulted in huge structural distortions in the generated images.

Deep Feature Warping We then compared frames generated with and without DFW. For the case without DFW, we directly warped RGB images using predicted motion fields for comparison. The second row in Figure 7 presents the qualitative comparison. As shown in the figure, the frame generated without DFW contained tearing artifacts in the animated regions. Table 4 shows an increase in the LPIPS score, which indicates that excluding DFW degraded the perceptual quality by introducing tearing artifacts.

Table 4. Results of quantitative evaluation in ablation study. The best scores are bolded.

Method	LPIPS↓	MS-SSIM↑	RMSE↓
Ours - Full	**0.0511**	**0.7165**	**21.8188**
Ours - w/o FW	0.0524	0.6853	22.9816
Ours - w/o DFW	0.0537	0.6908	22.4993
Ours - w/o MSDFW	0.0629	0.6946	22.3227
Ours - w/o Mask	0.0564	0.6980	22.6213

Table 5. Quantitative evaluation results on GAN inversion. The best scores are bolded.

w^+	D^{10}	FID↓	LPIPS↓	RMSE↓	MS-SSIM↑
✓	–	40.8276	0.0153	18.5901	0.7891
✓	✓	**11.8231**	**0.0019**	**5.6410**	**0.9907**

Multi-scale Deep Feature Warping Image frames generated with and without MSDFW were also compared. For the case without MSDFW, only a single deep feature D^{10} was warped and propagated through the next blocks of StyleGAN. The results of the qualitative comparison are shown in the third row of Figure 7. As revealed in the figure, excluding MSDFW resulted in blurry textures in the dynamic region. The quantitative comparison reported in Table 4 shows a significant increase in the LPIPS score. This indicates that excluding MSDFW degraded the perceptual quality, especially the texture details.

Segmentation Mask We compared the image frames generated with and without a segmentation mask. For the case without a mask, we used the initially predicted motion field for warping. The fourth row in Figure 7 shows the qualitative result of this comparison. As shown in the figure, excluding the mask resulted in erroneous movement in the static regions. The quantitative comparison reported in Table 4 reveals a significant degradation of MS-SSIM, which indicates the structural distortion caused by the erroneous motion.

Deep Feature Inversion To evaluate the effectiveness of GAN inversion, we compared the images reconstructed with and without the use of the deep features D^{10}. For the case without deep feature inversion, only the latent codes w^+ of a pre-trained StyleGAN were used to reconstruct the input landscape image. Figure 8 shows a qualitative result of this comparison. As revealed in the figure, using only w^+ failed to accurately reconstruct the details of the original images. For quantitative comparisons, FID, LPIPS, RMSE, and MS-SSIM were computed between I and $\hat{I}$. Table 5 reveals significant improvements in the perceptual quality of the results when reconstructed using the deep features D^{10}.

5. Limitations and Future Work

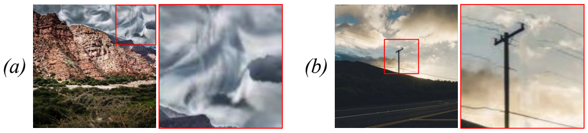

Figure 9. Limitations of StyleCineGAN: (a) automatic prediction of motion cannot be accurate for all images, and (b) the motion of a very thin structured object is hard to be isolated.

Our mask predictor and motion generator generally performed well for landscape images. However, automatic prediction cannot be accurate for all images because most landscape images contain inherent ambiguity in their motion directions except for some obvious cases (e.g., waterfalls). The failure case is illustrated in Figure 9 (a). User-defined motion hints can be used to resolve such ambiguities and provide further control capability during the generation process [18, 19]. In addition, it is hard for our method to isolate the motion of a very thin structured object placed within the animated region as shown in Figure 9 (b). This is because our method performs warping of features at multiple resolutions, in which low-resolution features cannot spatially represent the thin structures.

In this work, we mainly focused on animating the landscape images, particularly skies and fluids, while putting other types of animations outside the scope. In future developments, we would like to expand the capabilities to include other forms of motion. Investigating the rotating hands of a clock, the playing arm of a guitarist moving up and down, and a flag or the wings of a bird fluttering for cinemagraph generation would be a very interesting direction to pursue.

6. Conclusion

We proposed the first approach that leverages a pre-trained StyleGAN for high-quality one-shot landscape cinemagraph generation. In contrast to previous studies, our method does not require training a large image generator from scratch, and also systematically improves the resolution of the generated cinemagraphs to 1024×1024. At the core of our method, we utilized the deep features of a pre-trained StyleGAN, because those features can help preserve spatial information and encode both high-level semantic and low-level style appearances. Using our MSDFW approach, we applied the predicted motion to the deep feature space of StyleGAN. Both qualitative and quantitative results confirm that our method substantially outperforms existing baselines.

Acknowledgements This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2023 (Project Name: Development of Universal Fashion Creation Platform Technology for Avatar Personality Expression, Project Number: RS-2023-00228331, Contribution Rate: 90%) and Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (Project Name: The Competency Development Program for Industry Specialist, Project Number: P0012746, Contribution Rate: 10%).

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4432–4441, 2019. 3

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? CoRR, abs/1911.11544, 2019. 3

[3] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. arXiv preprint arXiv:2201.13433, 2022. 3

[4] Hugo Bertiche, Niloy J. Mitra, Kuldeep Kulkarni, Chun-Hao Paul Huang, Tuanfeng Y. Wang, Meysam Madadi, Sergio Escalera, and Duygu Ceylan. Blowing in the wind: Cyclenet for human cinemagraphs from still images, 2023. 1, 2

[5] Yung-Yu Chuang, Dan B Goldman, Ke Colin Zheng, Brian Curless, David H Salesin, and Richard Szeliski. Animating pictures with stochastic motion textures. In ACM SIGGRAPH 2005 Papers, pages 853–860. 2005. 2

[6] Yuki Endo, Yoshihiro Kanamori, and Shigeru Kuriyama. Animating landscape: self-supervised learning of decoupled motion and appearance for single-image video synthesis. ACM Transactions on Graphics (Proc. of SIGGRAPH ASIA 2019), 38(6):175:1–175:19, 2019. 1, 2, 5

[7] Gereon Fox, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. Stylevideogan: A temporal generative model using a pretrained stylegan, 2021. 2

[8] Tavi Halperin, Hanit Hakim, Orestis Vantzos, Gershon Hochman, Netai Benaim, Lior Sassy, Michael Kupchik, Ofir Bibi, and Ohad Fried. Endless loops: detecting and animating periodic patterns in still images. ACM Transactions on Graphics (TOG), 40(4):1–12, 2021. 2

[9] Aleksander Holynski, Brian L. Curless, Steven M. Seitz, and Richard Szeliski. Animating pictures with eulerian motion fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5810–5819, 2021. 1, 2, 3, 4, 5, 7

[10] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In International Conference on Learning Representations, 2020. 2

[11] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. CoRR, abs/1710.10196, 2017. 3

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4401–4410, 2019. 2

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

[14] Alexander Kristoffersen. Loopnerf: Exploring temporal compression for 3d video textures. Master's thesis, EECS Department, University of California, Berkeley, 2023. 1, 2

[15] Xingyi Li, Zhiguo Cao, Huiqiang Sun, Jianming Zhang, Ke Xian, and Guosheng Lin. 3d cinemagraphy from a single image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4595–4605, 2023. 1, 2

[16] Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 3

[17] Elizaveta Logacheva, Roman Suvorov, Oleg Khomenko, Anton Mashikhin, and Victor Lempitsky. Deeplandscape: Adversarial modeling of landscape videos. In Proceedings of the European Conference on Computer Vision (ECCV), 2020. 1, 2, 5

[18] Aniruddha Mahapatra and Kuldeep Kulkarni. Controllable animation of fluid elements in still images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 2, 3, 8

[19] Aniruddha Mahapatra, Aliaksandr Siarohin, Hsin-Ying Lee, Sergey Tulyakov, and Jun-Yan Zhu. Text-guided synthesis of eulerian cinemagraphs. 2023. 1, 2, 3, 5, 8

[20] Makoto Okabe, Ken Anjyo, Takeo Igarashi, and Hans-Peter Seidel. Animating pictures of fluid using video examples. In Computer Graphics Forum, pages 677–686. Wiley Online Library, 2009. 2

[21] Makoto Okabe, Ken Anjyor, and Rikio Onai. Creating fluid animation from a single image using video database. In Computer Graphics Forum, pages 1973–1982. Wiley Online Library, 2011.

[22] Makoto Okabe, Yoshinori Dobashi, and Ken Anjyo. Animating pictures of water scenes using video retrieval. The Visual Computer, 34(3):347–358, 2018. 2

[23] Ehsan Pajouheshgar, Yitao Xu, Tong Zhang, and Sabine Süsstrunk. Dynca: Real-time dynamic texture synthesis using neural cellular automata. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 1

[24] Gaurav Parmar, Yijun Li, Jingwan Lu, Richard Zhang, Jun-Yan Zhu, and Krishna Kumar Singh. Spatially-adaptive multilayer selection for gan inversion and editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 3

[25] Ekta Prashnani, Maneli Noorkami, Daniel Vaquero, and Pradeep Sen. A phase-based approach for animating images using video examples. In Computer Graphics Forum, pages 303–311. Wiley Online Library, 2017. 1, 2

[26] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 3

[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] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 2020. Version 0.3.0. 6

[29] Kwanggyoon Seo, Seoung Wug Oh, Jingwan Lu, Joon-Young Lee, Seonghyeon Kim, and Junyong Noh. Styleportraitvideo: Editing portrait videos with expression optimization. 41(7), 2022. 3

[30] Claude E. Shannon. Coding Theorems for a Discrete Source With a Fidelity CriterionInstitute of Radio Engineers, International Convention Record, vol. 7, 1959., pages 325–350. 1993. 2

[31] Ivan Skorokhodov, Grigorii Sotnikov, and Mohamed Elhoseiny. Aligning latent and image spaces to connect the unconnectable. arXiv preprint arXiv:2104.06954, 2021. 6

[32] Ryusuke Sugimoto, Mingming He, Jing Liao, and Pedro V. Sander. Water simulation and rendering from a still photograph. In SIGGRAPH Asia 2022 Conference Papers, New York, NY, USA, 2022. Association for Computing Machinery. 2

[33] Matthew Tesfaldet, Marcus A. Brubaker, and Konstantinos G. Derpanis. Two-stream convolutional networks for dynamic texture synthesis. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 1

[34] 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 International Conference on Learning Representations, 2021. 2, 5, 6, 7

[35] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. ACM Transactions on Graphics (TOG), 40(4): 1–14, 2021. 3

[36] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1526–1535, 2018. 2

[37] Rotem Tzaban, Ron Mokady, Rinon Gal, Amit Bermano, and Daniel Cohen-Or. Stitch it in time: Gan-based facial editing of real videos. In SIGGRAPH Asia 2022 Conference Papers, New York, NY, USA, 2022. Association for Computing Machinery. 3

[38] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 2

[39] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 3

[40] Zhou Wang, Eero P. Simoncelli, and Alan Conrad Bovik. Multiscale structural similarity for image quality assessment. The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, 2:1398–1402 Vol.2, 2003. 6

[41] 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 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 6, 7

[42] Yiran Xu, Badour AlBahar, and Jia-Bin Huang. Temporally consistent semantic video editing. pages 357–374. Springer, 2022. 3

[43] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A style-based gan encoder for high fidelity reconstruction of images and videos. European conference on computer vision, 2022. 3

[44] Fei Yin, Yong Zhang, Xiaodong Cun, Mingdeng Cao, Yanbo Fan, Xuan Wang, Qingyan Bai, Baoyuan Wu, Jue Wang, and Yujiu Yang. Styleheat: One-shot high-resolution editable talking face generation via pre-trained stylegan. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII, pages 85–101. Springer, 2022. 3

[45] 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. 6

[46] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10145–10155, 2021. 3