DOI: 10.1111/cgf.14666
Pacific Graphics 2022
N. Umetani, E. Vouga, and C. Wojtan
(Guest Editors)

Volume 41 (2022), Number 7

StylePortraitVideo: Editing Portrait Videos with Expression Optimization

Kwanggyoon Seo[1] Seoung Wug Oh[2] Jingwan Lu [2] Joon-Young Lee[2] Seonghyeon Kim[1] Junyong Noh[1]

[1]KAIST, Visual Media Lab [2] Adobe Research

Figure 1: ***Portrait videos edited by changing various attributes****. Using our approach, we can edit portrait videos while preserving the original identity of the face. An additional optimization ensures the result follows closely the original motion of the lips while preserving the edited expression. All videos are in 4K resolution except for the video on the bottom right.*

Abstract

High-quality portrait image editing has been made easier by recent advances in GANs (e.g., StyleGAN) and GAN inversion methods that project images onto a pre-trained GAN's latent space. However, extending the existing image editing methods, it is hard to edit videos to produce temporally coherent and natural-looking videos. We find challenges in reproducing diverse video frames and preserving the natural motion after editing. In this work, we propose solutions for these challenges. First, we propose a video adaptation method that enables the generator to reconstruct the original input identity, unusual poses, and expressions in the video. Second, we propose an expression dynamics optimization that tweaks the latent codes to maintain the meaningful motion in the original video. Based on these methods, we build a StyleGAN-based high-quality portrait video editing system that can edit videos in the wild in a temporally coherent way at up to 4K resolution.

CCS Concepts

• ***Computing methodologies*** → *Computer vision;* ***Image manipulation;***

1. Introduction

Portrait video editing is a task for manipulating a person's face attribute in a video while preserving other details in a temporally coherent manner. In the film industry, it is often necessary to change an actor's facial attributes to digitally age or de-age, to add or remove makeup, and to exaggerate or suppress facial expressions. However, completing such tasks in the traditional graphics pipeline requires a lot of effort and resources, since artists have to create a 3D model of the actor, manually edit the desired attributes, and compose the rendering back to the original frame seamlessly.

In recent years, high-quality portrait image editing has been made easier by advances in Generative Adversarial Networks (GAN). Most notably StyleGAN [KLA19, KLA*20] can generate high-resolution photo-realistic images. Additionally, GAN inversion, a technique to project images onto the latent space of a pre-trained unconditional GAN model, has been invented making unconditional GAN useful for image editing [AQW19, AQW20, RAP*21, TAN*21]. After projecting the input image to the latent space of a pre-trained StyleGAN, one can manipulate its high-level facial attributes by navigating the StyleGAN's latent space.

Nonetheless, it is still hard to use the current GAN inversion and editing method to produce natural-looking edited portrait video results. Two important issues need to be addressed in high-quality video editing: maintaining (1) temporal consistency and (2) natural expression dynamics after editing (Fig. 2). Undoubtedly, maintaining temporal consistency is one of the most important competencies of video editing methods. Otherwise, various temporal artifacts can appear due to various factors. In portrait video editing, we find identity and expression preservation after editing and handling naturally-occurring motion blur in the video essential to avoid temporal artifacts. StyleGAN pre-trained on images is not expressive enough to account for every possible identity, expression and head pose in video frames. This is because video frames are more diverse capturing every moment of subjects which are different from images that are usually taken on well-posed subjects. Yet, human eyes are sensitive to such details on faces and can easily pick up even the slightest change in the identity and expression after the editing. Handling motion blur in the video is another challenge. Motion blur is caused by moving cameras and subjects, and a moderate amount of motion blur is natural and makes videos realistic. Therefore, if motion blur is not properly handled and preserved, the output video will look unnatural.

As our first contribution, we propose a solution for the aforementioned issues that can hurt temporal consistency. To bridge the domain gap between videos and the learnt image manifold of StyleGAN, we adapt the pre-trained StyleGAN weights to the characteristics of the input target video. First, every video frame is projected onto the latent space of StyleGAN. We find processing each frame independently using an image inversion encoder [TAN*21] leads to a better projection of the latent space for temporal consistency compare to the optimization-based method. Then, inspired by PTI [RMBCO21], we further fine-tune the StyleGAN generator using self-supervised reconstruction losses to adapt the generator to the input video. After this video adaptation, the generator can render temporally-coherent video and reproduce the identity, poses, expressions, and even motion blur in the input video as shown in Fig. 2 (b). While adapting the generator is not new, we extend the idea to address video-specific problems and we make an important design decision that allows the method to produce temporally consistent video results.

In addition to temporally satisfying video reconstruction, manipulating facial expressions in portrait videos requires extra care. In previous latent-based editing methods [SGTZ20, HHLP20, YNGH21], images are usually manipulated by walking the latent code in a certain direction. However, naïvely applying the same amount of editing for every frame leads to unnatural results. The facial expression can be unnaturally exaggerated, and it can lose original natural dynamics. For example, the lip-sync in a talking-head video can be sabotaged resulting in a semantic mismatch between the visual and the audio.

As our second contribution, we propose an expression dynamics optimization method that modifies the globally manipulated latent codes to maintain the meaningful facial motion after editing. Our optimization is designed based on the Facial Action Coding System (FACS) [EF78] which correlates facial emotion with the shape of the mouth and eyes. To be specific, we constrain the movement of lips to follow the original video and the appearance of the eyes to follow the initially edited video. This allows the final generated video to reflect the desired expression while maintaining the original expression dynamics. This optimization is crucial for editing a talking-head video, because it can effectively reverse undesirable changes in the facial expression (e.g. mouth wide open) as shown in Fig. 2 (d).

With our video adaptation and expression dynamics optimization, we develop a StyleGAN-based high-quality portrait video editing system. Our system can edit in-the-wild portrait videos and a diverse set of identities at up to 4K resolution as shown in Fig. 1. We also provide extensive analysis for each system component to validate their effectiveness. Video results can be found at our project webpage style-portrait-video.github.io.

2. Related Work

GAN Inversion. GAN inversion methods are typically either optimization-based [AQW19, KLA*20, AQW20, GSZ20, XDX*21] or encoder-based [RAP*21, TAN*21, APCO21]. Optimization-based methods can reconstruct the input image accurately but require a few minutes per image to complete. On the other hand, feed-forward image encoders are much faster, but they come at a cost of reconstruction inaccuracy. Instead of inverting images into the native latent space $\mathcal{W} \in \mathbb{R}^{1\times512}$ or the extended space $\mathcal{W}^+ \in \mathbb{R}^{18\times512}$ [AQW20], other studies have investigated different latent spaces for better reconstruction accuracy and editability [WLS21, KKC21, ZLW*21, ZAFW21]. We employ a pre-trained image encoder, e4e [TAN*21], which projects images into $\mathcal{W}^+$ space. Then, we fine-tune the generator to represent the input portrait video better given the e4e-projected latent codes. This allows the adapted StyleGAN to reconstruct all the video frames accurately without time-consuming per-image optimization.

Fine-tuning Generators. Training high-quality GANs requires a considerable amount of computation and data [KLA19, BDS19]. Thus, it is preferred to fine-tune a pre-trained model to adapt

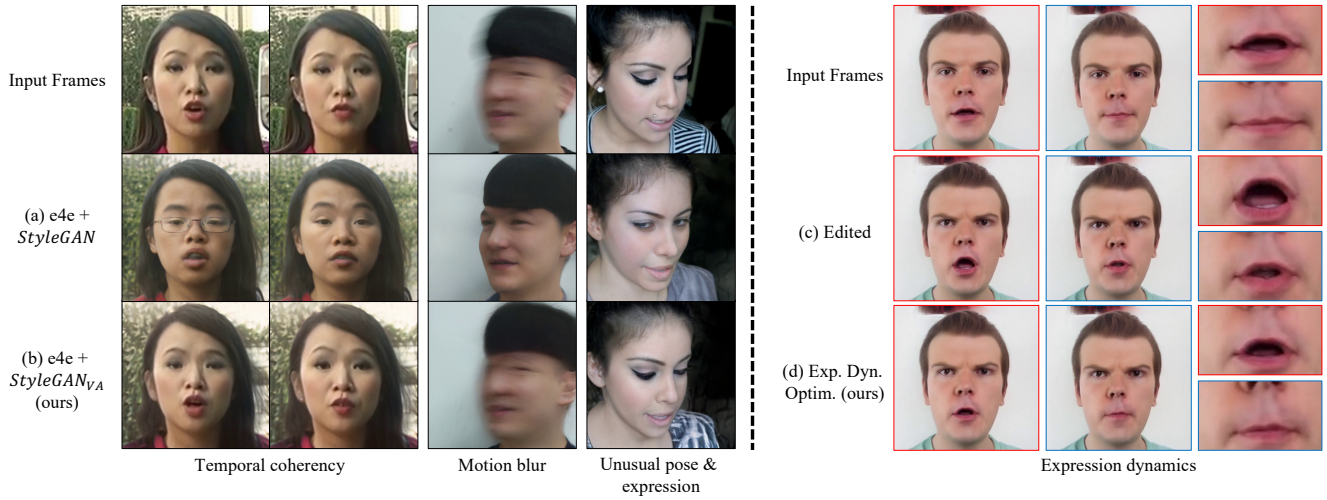


Figure 2: ***Challenges in videos.*** *Unlike images, video has much more diversity including unusual poses, expressions, and motion blur. In addition, natural expression dynamics need to be considered when editing the expression.*

to a target data distribution that is similar to the original domain. This fine-tuning often accompanies various issues such as mode collapse. Some methods place constraints on the trainable weights [MCS20, RCKH20, PA20] or use regularization terms [LZLS20, OLL*21] to handle such issues. Recently, StyleGAN-NADA proposes a method to shift the generator's domain with a powerful text embedding model [RKH*21] in a zero-shot manner [GPM*21]. Different from existing domain adaptation methods which focus on translating a source domain (e.g., photo-realistic faces) to a target domain (e.g., painterly portraits) for the generation of diverse outputs, we focus on fine-tuning the source domain model to better reproduce the identity of human faces used for the training of the target domain. Pan *et al.* proposed a method to fine-tune both generator and latent codes for downstream image restoration and manipulation tasks [PZD*21]. The method proposed in PTI [RMBCO21] fine-tunes StyleGAN with test-time images resulting in a model that better represents the target identities. During fine-tuning, they minimize changes to the latent space in order to maintain its editability. We adopt a similar approach so that the fine-tuned model can better reconstruct the input video at test-time with temporal coherence.

Face Attribute Editing. Several methods have been proposed to discover semantic editing directions in the latent space of StyleGAN. Some methods annotate images or latent codes with semantic labels for supervision [SGTZ20, AZMW21, YNGH21] while other methods use an unsupervised approach to find semantic directions [HHLP20, WLS21]. StyleCLIP [PWS*21] uses a text embedding model [RKH*21] for a text-based editing. For intuitive control, Tewari *et al.* use a 3D parametric face model to learn semantic directions in the latent space [TEB*20b, TEB*20a]. More recently, a dialog-driven approach has been introduced to manipulate a given image [JHP*21]. These editing methods are designed specifically for images and do not produce temporally coherent results when applied to videos. Our approach uses these editing methods to produce intermediate latent codes followed by a refinement of the edited codes in a way that preserves the expression dynamics of the original video.

High-Resolution Video Generation. While StyleGAN has shown incredible results for image generation, high-quality video generation remains a challenge [CDS19]. Instead of training a model specifically for videos, Tian *et al.* and Fox *et al.* have investigated generating videos using a pre-trained image generator [TRC*21, FTET21]. These methods take advantage of the fact that walking in a latent space produces temporally coherent image morphing effects [JCI20, PSN20], which meets the criteria of a video. StyleGAN-based architecture has also been adopted for temporally coherent face-swapping in videos [NHSW20]. We also use a pre-trained StyleGAN to perform portrait video editing while focusing on adapting the pre-trained StyleGAN to test-time videos in order to produce temporally coherent natural expressions. Concurrent works on video editing have been proposed similar to our idea for video GAN inversion [TMG*22, APW*22]. Unlike the methods, we additionally propose a method to maintain the meaningful motion in the original video after manipulating the expressions.

3. Method

Our video editing pipeline consists of six steps. Given an input full-frame video, (1) we align and crop the video using the alignment method adopted by the FFHQ dataset. With the cropped frames, (2) we project them into the latent space of StyleGAN using a pre-trained network [TAN*21]. Because the image-based GAN inversion method cannot exactly reproduce the identity of a human subject in the video, (3) we fine-tune the StyleGAN using the latent codes and their corresponding original frames. (4) We edit the video in the latent space using a known latent direction. (5) Then, the edited latent codes are optimized to follow the natural expression dynamics of the original video while preserving the semantic changes. Finally, (6) the edited and rendered frames are composed

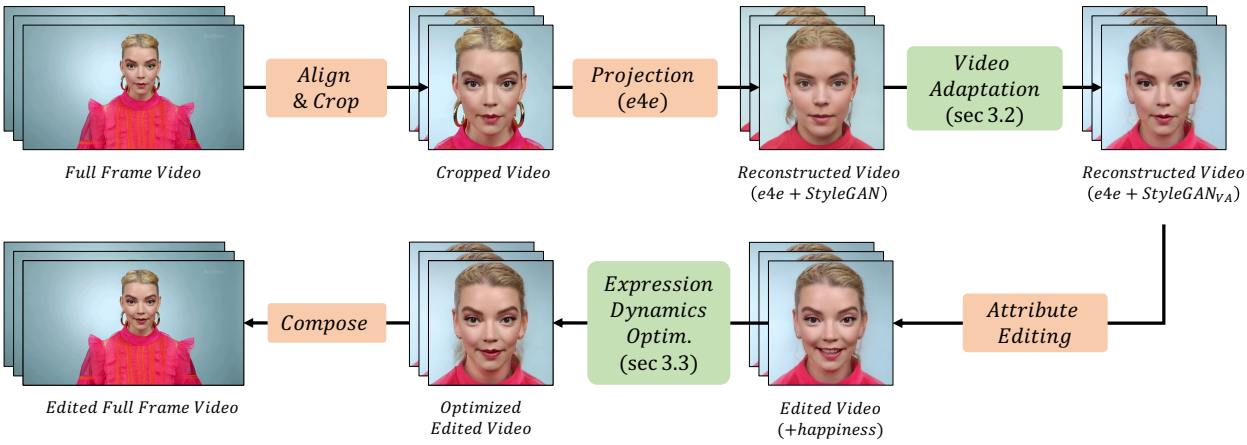


Figure 3: ***Video editing pipeline.*** *Given a full-frame video as input, the video is aligned and cropped following the method of FFHQ dataset [KLA19]. Then, the cropped video is projected into the latent space of StyleGAN using a pre-trained image encoder. The generator is fine-tuned using our video adaptation method so that the generator fits to the target input video. To edit the original video, we linearly combine the latent codes with a known latent direction. We further optimize the edited latent codes for the natural expression dynamics of the original video. The optimized codes and the adapted StyleGAN are used to render a desired edited portrait video, and the result is composed back to the original full-frame video.*

back to the original full-frame video. The overall pipeline is shown in Fig. 3.

3.1. Preprocessing

To perform attribute editing in StyleGAN's latent space, we first crop and align the portrait video so that it can be aligned the way the FFHQ dataset [KLA19] is aligned. This processing is essential because StyleGAN is trained on an aligned face dataset and cannot generate portrait images that deviate from the aligned space. We use a face landmark detection method [BT17] to extract the landmark for every frame. Because the face landmark detection method does not consider temporal information, we use the iterative Lucas-Kanade method with pyramids to compute an optical flow between two consecutive frames. The predicted landmark positions from the optical flow and current positions from the landmark detection method are blended to incorporate motion information. Finally, 1D Gaussian filtering is applied to the blended landmark position along the whole video sequence. We then use the processed landmarks to align the frames such that the eyes are centered, and the frames are resized to 1024×1024. The transformation parameters are saved to be later used when composing the edited video to the original input video (Sec. 3.4).

3.2. Video Inversion via Video Adaptation

After preprocessing all the frames, we project all the frames of the video into $\mathcal{W}^+$ space. Given a portrait video, $V = \{I^f | f = 1,...,N\}$ where N denotes a total number of frames, we invert I^f into a latent code w^f in $\mathcal{W}^+$ space using a pre-trained encoder e4e [TAN*21]. After projection, we can generate a reconstructed video $\hat{V} = \{\hat{I}^f | f = 1,...,N\}$, where $\hat{I}^f = G(w^f;\theta)$ is a generated image from StyleGAN generator G and its network parameters θ.

While $\hat{V}$ and V are similar, they can be different in terms of expressions and the identity of a human subject. In order to bridge the gap between $\hat{V}$ and V, we fine-tune G such that the reconstructed video frames $\hat{V}$ are almost identical to V. For this video adaptation task, we follow the approach of domain adaptation methods [RMBCO21, YL20]. Specifically, given V and $W = \{w^f | f = 1,...,N\}$ as anchor points, we update G's network parameter θ using the following loss terms:

$$\lambda_{LPIPS}\mathcal{L}_{LPIPS}(I^f, G(w^f;\theta^*)) + \lambda_{L2}\mathcal{L}_{L2}(I^f, G(w^f;\theta^*)), \quad (1)$$

where $\mathcal{L}_{LPIPS}$ measures the perceptual distance between two images [ZIE*18], $\mathcal{L}_{L2}$ computes the L^2 distance between two images, and θ^* is the tuned parameters of G. After the generator is adapted to the target video, we can still edit the original video using off-the-shelf StyleGAN editing operations [SGTZ20, HHLP20, PWS*21, YNGH21] as our video adaptation does not change the behavior of the latent space. We provide extensive analysis of the video adaptation process and the editability of the adapted StyleGAN after adaptation in Sec. 4.1.

Implementation Details. We fine-tune G using the Adam optimizer [KB15] with a learning rate of $1e^{-3}$. Both λ_{LPIPS} and λ_{L2} are set to 1. The generator is tuned for about two minutes performing 1,000 iterations on a single NVIDIA Tesla V100 GPU.

3.3. Expression Dynamics Optimization

Once the input video is projected onto the latent space and the generator is adapated to the video, we are ready for editing. Given a known latent direction Δw_{attr}, for a certain face attribute, we can edit a video frame I by

$$I_{edit} = G(w_{edit};\theta^*), \quad (2)$$

where $w_{edit} = w + s\Delta w_{attr}$, and s denotes a scalar value. For simplicity, we omit the frame index unless noted otherwise.

In the case of video, we can apply this operation for every single frame equally to get the edited portrait video V_{edit}. This gives temporally smooth edited results when editing texture styles such as putting makeups, aging, and deaging. Upon the change of the expression state of the original video, however, the expression dynamics of the V_{edit} often becomes unnatural. This happens because the predefined editing does not consider the expression dynamics. As mentioned in Sec. 1, naïvely using the pre-defined editing direction changes the lip position resulting in possibly very different expression dynamics from that observed in V.

To produce a plausible video that closely follows the original expression dynamics, we propose to optimize the individual latent codes. According to the FACS, which assumes many of the facial expressions are related to the movements of lips and eyes, we design two optimization objectives tailored for each area. The first objective constrains the inner points of the lips to be similar to the original video using the following equation:

$$\mathcal{L}_{lip} = \sum_i \|\phi_i(I) - \phi_i(G(w_{edit} + \Delta w_{opt}; \theta^*))\|_2, \quad (3)$$

where ϕ is a pre-trained landmark detection network [WSC*21] that outputs heatmaps of the landmark. i is the channel index of the inner lip landmarks, and Δw_{opt} is the latent direction to be optimized. This term helps to preserve the original mouth motion of a talking-head video after editing.

Another objective is designed for eyes due to their importance in conveying meanings and emotions. We design an objective to keep the eye shape in the edited face as follows:

$$\mathcal{L}_{eye} = \mathcal{L}_{LPIPS}(I_{edit} \odot M_{eye}, G(w_{edit} + \Delta w_{opt}; \theta^*) \odot M_{eye}), \quad (4)$$

where M_{eye} denotes a predefined eye mask, and $\odot$ indicates element-wise multiplication. The objective measures the emotions conveyed by the eyes as well as the shape of the eyes.

We also use a regularization term to stabilize the optimization. To ensure that the optimized latent codes does not deviate much from the original edited latent codes, we enforce Δw_{opt} to be as small as possible:

$$\mathcal{L}_{reg} = \|\Delta w_{opt}\|_2. \quad (5)$$

Additionally, to preserve temporal smoothness of the latent codes we use the following objective:

$$\mathcal{L}_{temp} = \sum_f^{N-1} \|\Delta w_{opt}^f - \Delta w_{opt}^{f+1}\|_2, \quad (6)$$

where f indicates the frame index. This enforces Δw_{opt} to not change abruptly from frame to frame.

Our final optimization is expressed as

$$\arg\min_{\Delta w_{opt}} \lambda_{lip}\mathcal{L}_{lip} + \lambda_{eye}\mathcal{L}_{eye} + \lambda_{reg}\mathcal{L}_{reg} + \lambda_{temp}\mathcal{L}_{temp}. \quad (7)$$

Implementation Details. We use the Adam optimizer [KB15] with a learning rate of $3e^{-4}$ for 200 iterations for a video with 48 frames. $\lambda_{lip}, \lambda_{eye}, \lambda_{reg}, \lambda_{temp}$ are set to 5000, 0.5, 8000, 5000, respectively. Optimization requires about 10 minutes on 8 NVIDIA Tesla V100 GPUs.

3.4. Postprocessing

After all the editing is completed, we apply the same transformation parameters used in the preprocessing step (Sec. 3.1) to composite the edited portrait video back to the original full-frame video. When composing, we only use the face region which is extracted from a face segmentation network [YWP*18] trained on the CelebAMask-HQ dataset [LLWL20].

4. Experiment

In the following sections, we show that our video adaptation method can successfully reconstruct original videos and support semantic edits using different types of editing methods [SGTZ20, YNGH21, HHLP20]. In addition, we show that by using expression dynamic optimization we can preserve the original expression dynamics after editing. We further show that our method can be used in different application scenarios. Refer to the supplementary materials for video results.

Datasets. To illustrate our approach is general, we perform experiments on videos from various sources: FaceForensics++ [RCV*19], RAVDESS [LR18], Pexels [pex], and Youtube. Videos from Pexels are cinematic and high-resolution. The subjects in the videos are moving their heads slowly with minimal changes in expression. On the other hand, subjects in the videos from RAVDESS, FaceForensics++, and Youtube are talking constantly and the expression dynamics become important when editing. We also shot some videos to validate that our method can handle videos with motion blur. Videos with motion blur are recorded by adjusting the shutter speed of a camera.

4.1. Video Adaptation and Editing Quality

Full Frame Edited Results. Our method can edit high-resolution portrait video from Pexels [pex] as shown in Fig. 1. We have cropped 4K video results to visualize the editing results closely. While StyleGAN can only generate 1024×1024 resolution faces, the edited face can be upsampled as needed and composed back to the original video without much perceived loss of details. Our method is compatible with various editing methods including InterFaceGAN [SGTZ20], Latent Transformer [YNGH21], and GANSpace [HHLP20]. Our method shows temporally coherent results through video adaptation and expression dynamics optimization.

Comparison to Image-based GAN Inversion. We compare our video adaptation method to image-based state-of-the-arts GAN inversion method [RAP*21, TAN*21]. As shown in Fig. 4, we tested out with diverse ethnicity, and our method achieves superior reconstruction quality compared to the other competing methods. In addition, our adapted StyleGAN can perform high-level semantic editing while preserving the original identity. Pre-trained image encoders [RAP*21, TAN*21] can project video in a frame-by-frame manner, but using a pre-trained StyleGAN, it cannot accurately reconstruct the input identity and small expression details on lips and eyes. We additionally experimented with our video adaptation on videos with motion blur. As shown in Fig. 5, our method successfully reconstructs the input frames with motion blur. On the other

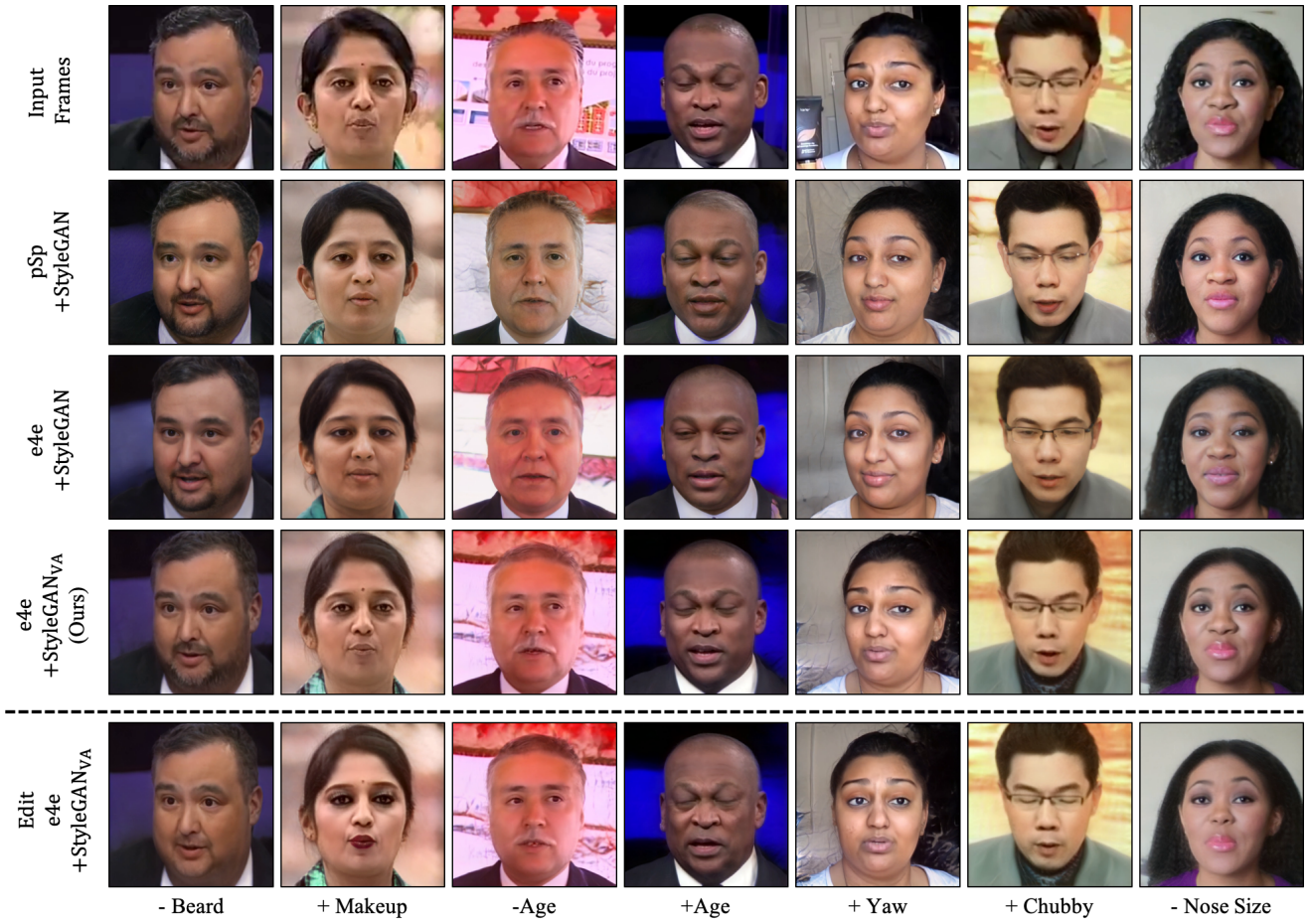


Figure 4: ***Reconstruction of input videos.*** *For pSp [RAP*21] and e4e [TAN*21], we used the original pre-trained StyleGAN weight trained on FFHQ. Our video adaptation uses StyleGAN which is fine-tuned to the input video. The last row shows the edited results from our method. StyleGAN$_{VA}$ denotes the video adapted StyleGAN.*

hand, using e4e and the original weight of StyleGAN, it is only able to generate faces that are much sharper than the original.

We quantitatively evaluate our results using several metrics: MSE, PSNR, MS-SSIM [WSB03], LPIPS [ZIE*18], and ID similarity using a pre-traind face recognition network [DGXZ19]. We measure these metrics using the first 50 videos from FaceForensics++ [RCV*19] yielding 20,991 frames. As shown in Tab. 1, our method outperforms the other image encoder methods. Optimization-based methods [AQW19, AQW20] take several minutes per frame, making it difficult to compare quantitatively against our method. We will discuss and show some visual results for the optimization method in our study on design choices.

Design Choices for Video Adaptation We validate our design choices for the video adaptation pipeline and discuss the differences between our approach and PTI [RMBCO21]. Our method shares the test-time generator fine-tuning idea but adapts it to the video domain. We argue that PTI is not suitable for video applications in two aspects. First, it is intractably inefficient due to the time-consuming latent code optimization step. Given an image, a latent code in $\mathcal{W}$ is obtained from optimization-based GAN inversion [KLA*20] which takes up to several minutes. The processing time linearly increases with the number of frames in a video. Second, PTI uses only a few iterations to optimize for the latent code, which leads to a sub-optimal latent code when rendered with the original pre-trained StyleGAN. Also, optimizing the latent codes separately without enforcing any temporal constraints might lead to temporally incoherent reconstruction. We overcome both issues

Table 1: ***Quantitative comparison on video reconstruction.*** *We compare our video adaptation method to GAN inversion methods, e4e [TAN*21] and pSp [RAP*21]. ↓ and ↑ denote the lower the better and the higher the better, respectively. The best results are marked in* ***bold****.*

Method	LPIPS ↓	MSE ↓ ($\times e$-4)	MS-SSIM ↑	PSNR ↑	ID Similarity ↑
pSp [RAP*21]	0.5025	9.474	0.735	20.564	0.756
e4e [TAN*21]	0.3548	10.90	0.730	19.973	0.659
Ours	**0.2790**	**7.810**	**0.777**	**21.71**	**0.832**

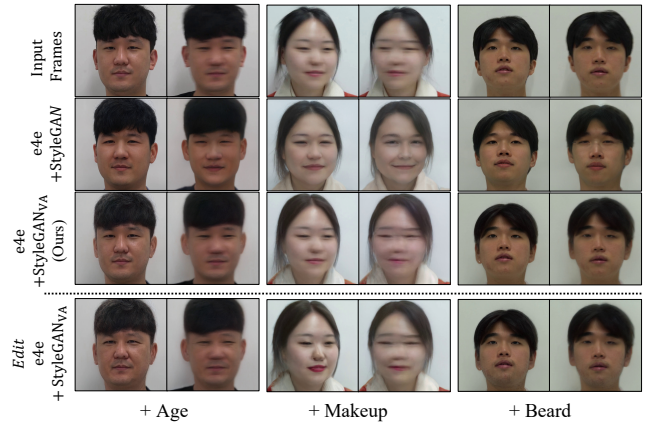


Figure 5: ***Motion blurred frames.*** *Unlike the original pre-trained StyleGAN, after video adaptation, it can reconstruct the blurred frame while maintaining editability.*

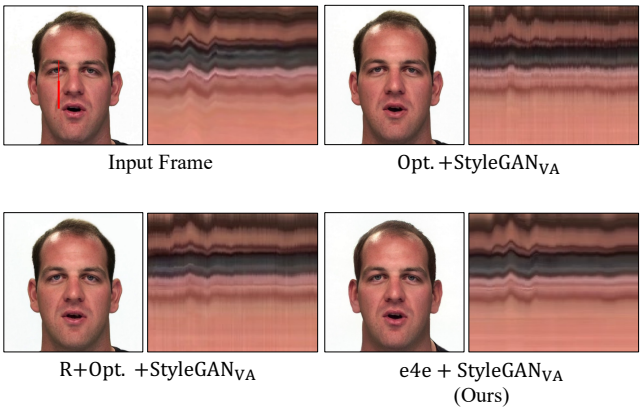


Figure 6: ***Temporal profile of reconstructed video.*** *Given an input video, we visualize the temporal profile along the red scan line. The temporal profile is horizontally stacked.*

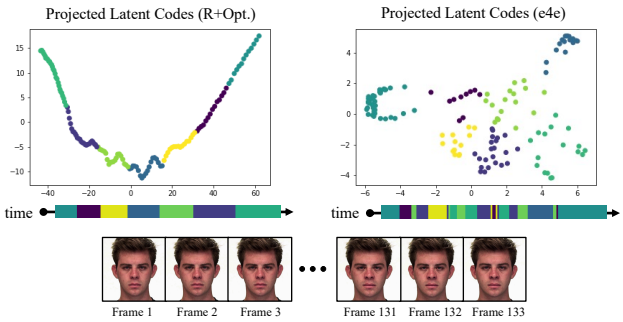


Figure 7: ***Latent codes from R+Opt. and e4e [TAN*21].*** *We project the latent codes into two-dimensional space obtained by applying PCA on the latent codes and clustered using K-means clustering. Each color denotes a cluster. The time bar shows the cluster label for each frame.*

by using a pre-trained image encoder and video adaptation method. These changes make our method not only orders of magnitude faster than any optimization-based methods but also more stable during our video adaptation.

To validate our observation, we design two variants of our method to use optimized $\mathcal{W}$ latent codes. Our first variant, **Opt.+StyleGAN$_{VA}$**, strictly follows how PTI encodes latent codes (e.g., running optimization per video frame). However, this variant takes hours to finish for a video with 100 frames. Our second variant, **R+Opt.+StyleGAN$_{VA}$**, uses a more efficient way to find the latent codes. We recurrently use the previous frame's latent code to initialize the current frame's latent code for optimization and expect the optimized sequence of latent codes to follow a smoother temporal trajectory and faster convergence. We visualize the generated results for both variants and our method in Fig 6. When looking at individual frames, both variants produce comparable results to our method. However, when looking at the video as a whole, temporal inconsistency problems caused by the projected latent codes are quite visible.

To understand why this happens, we project and visualize the latent codes of **R+Opt.** and e4e [TAN*21]. We use Principal Component Analysis (PCA) to reduce the dimension and project the latent codes into 2-D by using the first two components of the PCA. Additionally, we group the latent codes using K-means clustering with $k = 7$ to see how the latent codes are clustered. As shown in Fig. 7, in case of **R+Opt.**, we can observe that the frames with similar expressions are not labeled in the same cluster. For example, frame 1 and frame 133 are very similar in terms of both facial expression and head pose, but labeled as a different cluster. For e4e, latent codes for frame 1 and frame 133 are located in a similar position in the latent space and labeled in the same cluster. Unlike R+Opt., e4e is able to project the similar pose and expression to be close to the latent space of pre-trained StyleGAN.

4.2. Expression Dynamics Optimization Quality

We show that expression dynamics optimization enables the generated frames to follow the original frames' lip motion while maintaining the originally edited semantics. As shown in Fig 8, the lip openness is similar to the original. Also, even when the lip motion is similar to the original, the corners of the lips still follow the edited results. For happiness, we observe that the corners of the lips point upward in the optimized frames but do not do so in the original neutral frames. Cheekbones are also elevated with the mouth. For surprise and anger, lips are gathered to show their expressions. Eyes are conveying the correct intended emotions. Our method also works well in the case of motion blur is present as shown in the sixth column of Fig 8. Additionally, we visualize the temporal profiles as shown in the last column of Fig. 8. Without the optimization, we can see that the mouth is not closing when the original frame is (green box). After optimization, we see that the lip motions are much more similar to the original. In addition, we observe that the corner of the mouth is slightly elevated from the original to follow the intended editing (blue box).

Figure 8: ***Qualitative results on expression editing.*** *Before optimization, the mouth opens for all expressions. After optimization, the mouth follows the original expression while maintaining the edited expression. See the eyes and corners of lips. Additionally, we visualize temporal profiles along the scan lines. The temporal profiles are horizontally stacked. See the dotted white box for the differences.*

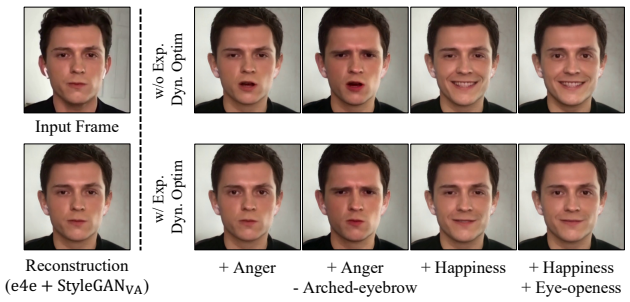


Figure 9: ***Sequential editing.*** *Expression dynamics optimization can be performed after sequential editing for both eye and eyebrow shapes.*

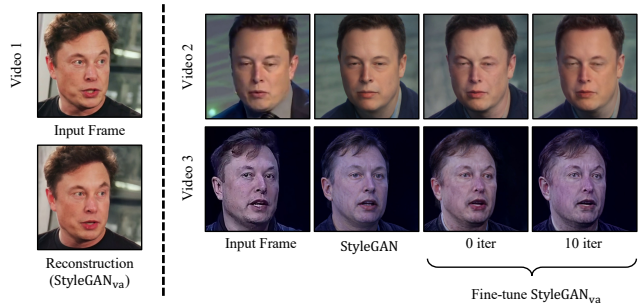


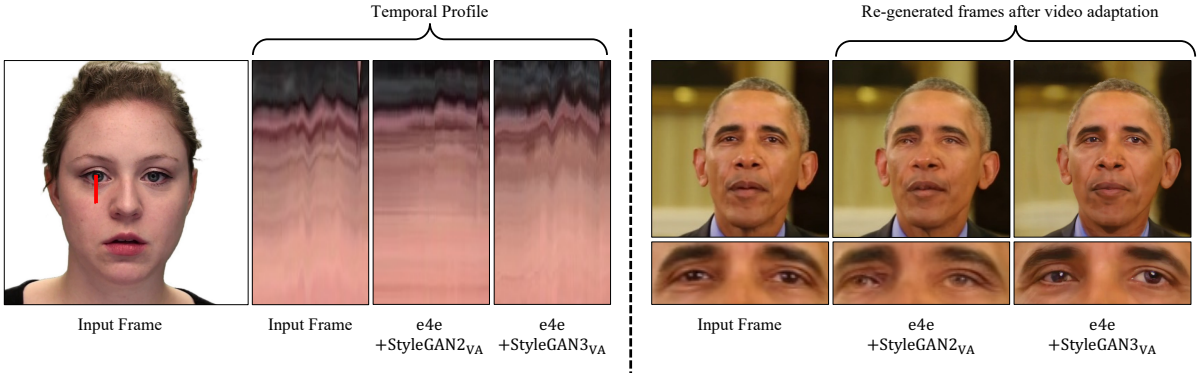
Figure 10: ***Continual learning.*** *After adapting StyleGAN to video 1, we continuously adapt to different videos with the same identity. By continually learning the same identity, it can adapt within a few iterations.*

Figure 11: ***Frame interpolation.*** ***Red*** *boxes indicate the original frames. Frames in-between them are the interpolated results.*

As done in GAN-based image editing methods [SGTZ20, HHLP20, YNGH21], multiple attributes can be manipulated sequentially in the latent space of StyleGAN using our method. When doing sequential editing, we simply change I in Eq. 3 to I_{edit}^{prev} which is the edited frame before expression editing. After all the desired editing including the expression editing is done, expression dynamics optimization modifies the latent code to follow the original dynamics of the video while preserving the previously applied editing. As presented in Fig 9, edited eye and eyebrow shapes are preserved after the optimization. This shows that the method is capable of changing any facial features that are related to the eyes and mouth while preserving the expression dynamics of the original video.

4.3. Application

Continual Learning. In VFX industry, artists often need to edit the same person in different videos. Since our video adaptation method adapts StyleGAN to a specific person, after the adaption using the first video, we can use it to edit another video of the same subject directly to get reasonable results. As shown in Fig. 10, the previously adapted StyleGAN reconstructs the new target video of the same person better than the pre-trained StyleGAN before adaptation. Given a new video of the same person, we can adapt our model quickly (under 3 seconds) for 10 iterations, and then the newly adapted model will reconstruct the input identity even better. However, we observe that the final adapted StyleGAN is biased toward the first video, e.g., the eye gaze direction in the reconstructed video 3 look similar to that of video 1.



(a) Texture Sticking (b) Face Misalignment

Figure 12: ***Limitations and capacity of StyleGAN3.*** *Using StyleGAN2 as a generator, (a) the texture sticking problem cannot be solved as shown in the temporal profile, and (b) also our method cannot invert misaligned faces well. However, when using StyleGAN3 as a generator, we can alleviate the above-mentioned problems.*

Frame Interpolation. We use the adapted StyleGAN to perform frame interpolation as shown in Fig. 11. By linearly interpolating the two latent codes, we can generate infinite in-between frames. The interpolated results look natural and smoothly transition from one expression to another. Therefore, we are confident that our adapted StyleGAN is not locally overfitted to every frame.

5. Discussion

Limitation and Future Work. While our method is able to generate high-resolution, temporally coherent, and natural-looking edited portrait video, there are several limitations to be addressed in the future. Using StyleGAN2, we observe severe texture sticking artifacts [KAL*21] where high-frequency texture details are glued to image coordinates. In Fig. 12 (a), this artifact is noticeable as there are straight horizontal lines around the eyes in the temporal profile. This problem is known to be originated from the aliasing artifact caused by convolutions [KAL*21].

Another limitation is in handling misaligned faces as shown in Fig. 12 (b), which leads to small drifting or floating of the face when compositing. This artifact is caused by the combination of errors in our pre-processing and the pre-trained StyleGAN2. When pre-processing the original video, a fast head or camera motion results in inaccuracy in the alignment of the subject's face in some frames. This misalignment cannot be recovered from the pre-trained StyleGAN2 after the video adaptation, because the StyleGAN2 was trained only with aligned faces and thus is strongly biased towards well-aligned faces.

As a preliminary attempt to remedy the above issues, we have tried to leverage the pre-trained StyleGAN3 [KAL*21] rotation model on FFHQU as the generator. We train an e4e image encoder [TAN*21] to project unaligned faces onto StyleGAN3's latent space and test our video adaption method. Results in Fig. 12 show the potential of StyleGAN3 in solving both texture sticking and face misalignment issues. Our method is not restricted to a specific GAN backbone, and thus the performance can be improved by using a better GAN model if the model is invertible.

6. Conclusion

We present a method for portrait video editing using a pre-trained StyleGAN. Our method is able to accurately invert the faces in video frames with challenging poses, expressions, and motion blur. We further optimize for natural expression dynamics in case of editing the expression state of a talking-head video. Our method works well on in-the-wild videos with subjects from different ethnic groups, and the edited results can be composited back to the original 4k resolution videos without much loss in quality. Despite the potential positive impacts on the creative industry, our method is susceptible to misuse. While detecting edited faces is out of the scope of this paper, many works in forensics research focus on detecting generated and manipulated faces [WWO*19, RCV*19, YL20]. We believe the methods we develop can further push the performance of the media forensics technologies.

Acknowledgements

We thank the anonymous reviewers for their insightful comments and feedback. This work was supported by Institute of Information & communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-00450, A Deep Learning Based Immersive AR Content Creation Platform for Generating Interactive, Context and Geometry Aware Movement from a Single Image)

References

[APCO21] ALALUF Y., PATASHNIK O., COHEN-OR D.: Restyle: A residual-based stylegan encoder via iterative refinement. *arXiv preprint arXiv:2104.02699* (2021). 2

[APW*22] ALALUF Y., PATASHNIK O., WU Z., ZAMIR A., SHECHTMAN E., LISCHINSKI D., COHEN-OR D.: Third time's the charm? image and video editing with stylegan3, 2022. arXiv:2201.13433. 3

[AQW19] ABDAL R., QIN Y., WONKA P.: Image2stylegan: How to embed images into the stylegan latent space? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019). 2, 6

[AQW20] ABDAL R., QIN Y., WONKA P.: Image2stylegan++: How to edit the embedded images? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020). 2, 6

[AZMW21] ABDAL R., ZHU P., MITRA N. J., WONKA P.: Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics 40*, 3 (2021), 1–21. 3

[BDS19] BROCK A., DONAHUE J., SIMONYAN K.: Large scale GAN training for high fidelity natural image synthesis. In *Int. Conf. Learn. Represent.* (2019). 2

[BT17] BULAT A., TZIMIROPOULOS G.: How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017). 4

[CDS19] CLARK A., DONAHUE J., SIMONYAN K.: Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571* (2019). 3

[DGXZ19] DENG J., GUO J., XUE N., ZAFEIRIOU S.: Arcface: Additive angular margin loss for deep face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019). 6

[EF78] EKMAN P., FRIESEN W. V.: *Facial action coding system: Investigator's guide*. Consulting Psychologists Press, 1978. 2

[FTET21] FOX G., TEWARI A., ELGHARIB M., THEOBALT C.: StyleVideoGAN: A Temporal Generative Model using a Pretrained StyleGAN. *arXiv* (2021). 3

[GPM*21] GAL R., PATASHNIK O., MARON H., CHECHIK G., COHEN-OR D.: Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv* (2021). 3

[GSZ20] GU J., SHEN Y., ZHOU B.: Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 3012–3021. 2

[HHLP20] HÄRKÖNEN, HERTZMANN A., LEHTINEN J., PARIS S.: Ganspace: Discovering interpretable gan controls. In *Advances in Neural Information Processing Systems* (2020). 2, 3, 4, 5, 8

[JCI20] JAHANIAN A., CHAI L., ISOLA P.: On the "steerability" of generative adversarial networks. In *ICLR* (2020). 3

[JHP*21] JIANG Y., HUANG Z., PAN X., LOY C. C., LIU Z.: Talk-to-edit: Fine-grained facial editing via dialog. In *IEEE/CVF International Conference on Computer Vision* (2021). 3

[KAL*21] KARRAS T., AITTALA M., LAINE S., HÄRKÖNEN E., HELLSTEN J., LEHTINEN J., AILA T.: Alias-free generative adversarial networks. *arXiv preprint arXiv:2106.12423* (2021). 9

[KB15] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.* (2015), Bengio Y., LeCun Y., (Eds.). 4, 5

[KKC21] KANG K., KIM S., CHO S.: Gan inversion for out-of-range images with geometric transformations. In *IEEE/CVF International Conference on Computer Vision* (2021). 2

[KLA19] KARRAS T., LAINE S., AILA T.: A style-based generator architecture for generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019). 2, 4

[KLA*20] KARRAS T., LAINE S., AITTALA M., HELLSTEN J., LEHTINEN J., AILA T.: Analyzing and improving the image quality of StyleGAN. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020). 2, 6

[LLWL20] LEE C.-H., LIU Z., WU L., LUO P.: Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020). 5

[LR18] LIVINGSTONE S. R., RUSSO F. A.: 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*, 5 (2018), e0196391. 5

[LZLS20] LI Y., ZHANG R., LU J. C., SHECHTMAN E.: Few-shot image generation with elastic weight consolidation. In *Advances in Neural Information Processing Systems* (2020). 3

[MCS20] MO S., CHO M., SHIN J.: Freeze the discriminator: a simple baseline for fine-tuning gans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop* (2020). 3

[NHSW20] NARUNIEC J., HELMINGER L., SCHROERS C., WEBER R. M.: High-resolution neural face swapping for visual effects. In *Comput. Graph. Forum* (2020), vol. 39, pp. 173–184. 3

[OLL*21] OJHA U., LI Y., LU C., EFROS A. A., LEE Y. J., SHECHTMAN E., ZHANG R.: Few-shot image generation via cross-domain correspondence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021). 3

[PA20] PINKNEY J. N., ADLER D.: Resolution dependent gan interpolation for controllable image synthesis between domains. *arXiv preprint arXiv:2010.05334* (2020). 3

[pex] Pexels. https://www.pexels.com. 5

[PSN20] PARK S., SEO K., NOH J.: Neural crossbreed: neural based image metamorphosis. *ACM Transactions on Graphics 39*, 6 (2020), 1–15. 3

[PWS*21] PATASHNIK O., WU Z., SHECHTMAN E., COHEN-OR D., LISCHINSKI D.: Styleclip: Text-driven manipulation of stylegan imagery. In *IEEE/CVF International Conference on Computer Vision* (2021). 3, 4

[PZD*21] PAN X., ZHAN X., DAI B., LIN D., LOY C. C., LUO P.: Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021). 3

[RAP*21] RICHARDSON E., ALALUF Y., PATASHNIK O., NITZAN Y., AZAR Y., SHAPIRO S., COHEN-OR D.: Encoding in style: a stylegan encoder for image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 2287–2296. 2, 5, 6

[RCKH20] ROBB E., CHU W.-S., KUMAR A., HUANG J.-B.: Few-shot adaptation of generative adversarial networks. *arXiv preprint arXiv:2010.11943* (2020). 3

[RCV*19] ROSSLER A., COZZOLINO D., VERDOLIVA L., RIESS C., THIES J., NIESSNER M.: Faceforensics++: Learning to detect manipulated facial images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019). 5, 6, 9

[RKH*21] RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SASTRY G., ASKELL A., MISHKIN P., CLARK J., ET AL.: Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020* (2021). 3

[RMBCO21] ROICH D., MOKADY R., BERMANO A. H., COHEN-OR D.: Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744* (2021). 2, 3, 4, 6

[SGTZ20] SHEN Y., GU J., TANG X., ZHOU B.: Interpreting the latent space of gans for semantic face editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020). 2, 3, 4, 5, 8

[TAN*21] TOV O., ALALUF Y., NITZAN Y., PATASHNIK O., COHEN-OR D.: Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics 40*, 4 (2021), 1–14. 2, 3, 4, 5, 6, 7, 9

[TEB*20a] TEWARI A., ELGHARIB M., BERNARD F., SEIDEL H.-P., PÉREZ P., ZOLLHÖFER M., THEOBALT C.: Pie: Portrait image embedding for semantic control. *ACM Transactions on Graphics 39*, 6 (2020), 1–14. 3

[TEB*20b] TEWARI A., ELGHARIB M., BHARAJ G., BERNARD F., SEIDEL H.-P., PÉREZ P., ZÖLLHOFER M., THEOBALT C.: Stylerig: Rigging stylegan for 3d control over portrait images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (june 2020), IEEE. 3

[TMG*22] TZABAN R., MOKADY R., GAL R., BERMANO A. H., COHEN-OR D.: Stitch it in time: Gan-based facial editing of real videos, 2022. arXiv:2201.08361. 3

[TRC*21] TIAN Y., REN J., CHAI M., OLSZEWSKI K., PENG X., METAXAS D. N., TULYAKOV S.: A good image generator is what you need for high-resolution video synthesis. In *International Conference on Machine Learning* (2021). 3

[WLS21] WU Z., LISCHINSKI D., SHECHTMAN E.: Stylespace analysis: Disentangled controls for stylegan image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021). 2, 3

[WSB03] WANG Z., SIMONCELLI E. P., BOVIK A. C.: Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003* (2003), vol. 2, Ieee, pp. 1398–1402. 6

[WSC*21] WANG J., SUN K., CHENG T., JIANG B., DENG C., ZHAO Y., LIU D., MU Y., TAN M., WANG X., ET AL.: Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 43* (2021), 3349–3364. 5, 1

[WWO*19] WANG S.-Y., WANG O., OWENS A., ZHANG R., EFROS A. A.: Detecting photoshopped faces by scripting photoshop. In *IEEE/CVF International Conference on Computer Vision* (2019). 9

[XDX*21] XU Y., DU Y., XIAO W., XU X., HE S.: From continuity to editability: Inverting gans with consecutive images. In *IEEE/CVF International Conference on Computer Vision* (2021). 2

[YL20] YANG C., LIM S.-N.: One-shot domain adaptation for face generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (June 2020). 4, 9

[YNGH21] YAO X., NEWSON A., GOUSSEAU Y., HELLIER P.: A latent transformer for disentangled and identity-preserving face editing. In *IEEE/CVF International Conference on Computer Vision* (2021). 2, 3, 4, 5, 8

[YWP*18] YU C., WANG J., PENG C., GAO C., YU G., SANG N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision* (2018). 5

[ZAFW21] ZHU P., ABDAL R., FEMIANI J., WONKA P.: Barbershop: Gan-based image compositing using segmentation masks. *ACM Transactions on Graphics* (2021). 2

[ZIE*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018). 4, 6, 1

[ZLW*21] ZHU Y., LI Q., WANG J., XU C., SUN Z.: One shot face swapping on megapixels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021). 2