DeepFaceVideoEditing: Sketch-based Deep Editing of Face Videos

FENGLIN LIU, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences
SHU-YU CHEN, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences
YU-KUN LAI, Cardiff University
CHUNPENG LI, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences
YUEREN JIANG, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences
HONGBO FU, School of Creative Media, City University of Hong Kong
LIN GAO*, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences

Fig. 1. Our method, named DeepFaceVideoEditing, allows users to intuitively edit face video by sketches and masks. Given an input face video, users can select multiple frames and draw sketches within selected mask regions to apply diverse editing operations. Our system supports two types of manipulations, namely, Temporally Consistent Editing, which has significant influence on the entire video (blue boxes), and Temporally Variant Editing, which dynamically changes in the timeline (orange boxes). The editing effects of these two types are propagated to all the video frames in different manners. The output video fuses all input sketch editing effects and shows stable temporal consistency. Please refer to the accompanying video for various editing results with our technique. Original videos courtesy of Vanessa Garcia.

Sketches, which are simple and concise, have been used in recent deep image synthesis methods to allow intuitive generation and editing of facial images. However, it is nontrivial to extend such methods to video editing due to various challenges, ranging from appropriate manipulation propagation and fusion of multiple editing operations to ensure temporal coherence and visual quality. To address these issues, we propose a novel sketch-based facial video editing framework, in which we represent editing manipulations in latent space and propose specific propagation and fusion modules to generate high-quality video editing results based on StyleGAN3. Specifically, we first design an optimization approach to represent sketch editing manipulations by editing vectors, which are propagated to the whole video sequence using a proper strategy to cope with different editing needs. Specifically, input editing operations are classified into two categories: temporally consistent editing and temporally variant editing. The former (e.g., change of face shape) is applied to the whole video sequence directly, while the latter (e.g., change of facial expression or dynamics) is propagated with the guidance of expression or only affects adjacent frames in a given time window. Since users often perform different editing operations in multiple frames, we further present a region-aware fusion approach to fuse diverse editing effects. Our method supports video editing on facial structure and expression movement by sketch, which cannot be achieved by previous works. Both qualitative and quantitative evaluations show the superior editing ability of our system to existing and alternative solutions.

CCS Concepts: • **Human-centered computing** → *Graphical user interfaces*; • **Computing methodologies** → *Image processing*.

Additional Key Words and Phrases: Video Editing, Video Propagation, Sketch-based Interaction

*Corresponding author.

Authors' addresses: Fenglin Liu, Shu-Yu Chen, Chunpeng Li, Yueren Jiang, and Lin Gao are with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences. Yu-Kun Lai is with the School of Computer Science and Informatics, Cardiff University. Hongbo Fu is with the School of Creative Media, City University of Hong Kong. Authors' e-mails: liufenglin21s@ict.ac.cn, chenshuyu@ict.ac.cn, LaiY4@cardiff.ac.uk, cpli@ict.ac.cn, jiangyueren15@mails.ucas.ac.cn, hongbofu@cityu.edu.hk, gaolin@ict.ac.cn.

0730-0301/2022/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Fenglin Liu, Shu-Yu Chen, Yu-Kun Lai, Chunpeng Li, Yueren Jiang, Hongbo Fu, and Lin Gao. 2022. DeepFaceVideoEditing: Sketch-based Deep Editing of Face Videos. *ACM Trans. Graph.* 1, 1 (May 2022), 16 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

While portrait image editing and manipulation have been researched extensively and achieved impressive editing results with different interaction forms, facial video editing still remains a difficult task. This is because video editing poses several key challenges. First, although existing image editing methods generate good results on individual key frames, it is hard to infer the manipulations of other frames corresponding to input editing operations on key frames. Second, certain editing effects are temporally variant and have a close relationship with dynamic expressions, making the propagation of such editing effects more difficult. Third, in order to edit a video with detailed control, users often apply different editing operations at multiple key frames. Fusing these diverse editing manipulations appropriately is nontrivial since they might cause undesired interference. Lastly, generating temporally coherent video results following editing requirements also remains challenging, since human eyes are sensitive to flickering artifacts.

Sketch, as a simple and effective interaction intermediary, has been widely used in single facial image editing. Compared with methods [Alaluf et al. 2021a; Bhat et al. 2004; Härkönen et al. 2020; Shen et al. 2020; Wu et al. 2021] that control specific predefined attributes (pose, age, expression, etc) with slider bars, sketching provides users with more editing freedom and achieves more detailed spatial control. Multiple facial editing methods [Jo and Park 2019; Portenier et al. 2018; Yang et al. 2020] have used sketch-based interfaces to guide the generation of edited local parts via image completion. DeepFaceEditing [Chen et al. 2021] further disentangles the facial geometry and appearance with the help of sketch, achieving detailed editing of local and global components. Although these methods generate attractive results for single face images, utilizing sketches for video editing is highly nontrivial, since hand-drawn sketches are sophisticated and it is difficult to propagate them reasonably to an entire video sequence. Besides, when editing a single facial image, users can flexibly modify both its expression and the shape of facial components. Some editing operations (e.g., change of face shape) influence the whole video sequence, while others imply a one-off movement (e.g., blinking) or are only associated with specific facial expressions (e.g., forehead wrinkles while smiling). These differences require progressive propagation to specific frames, making video editing with sketches more challenging.

To propagate sketch-based editing effects to a video sequence, one possible solution is to first extract edge maps from individual video frames, and then warp user-specified sketches according to optical flow fields and paste them back into the edge map sequence. Image translation methods [Isola et al. 2017; Yang et al. 2017] and video translation methods [Wang et al. 2019] could be further utilized to generate the edited video from the edge maps with the warped sketches. However, this naïve sketch warping approach often predicts unreasonable sketches, causing obvious artifacts on edited regions, and it is hard to generate high-quality temporally coherent video editing effects because of the complex processing steps. Another possible solution is to first generate edited images by applying sketch-based image editing to individual frames, and then utilize an image animation approach (e.g., [Siarohin et al. 2019]) to animate the edited frames with the motion driven by the original frames. This method generates robust video results, but it is hard to reconstruct the details only using single edited frames. Besides, neither of the above possible solutions could generate temporally variant editing results, such as adding blink in selected frames or making eyes small when smiling.

In this work, we propose a sketch-based framework for editing face videos, taking the temporal effects of single-frame editing into consideration and enabling consistent fusion of multiple editing operations. We utilize a StyleGAN-based generator to synthesize high-resolution, temporally coherent video editing results. The original video frames are first projected into the latent space by E4E [Tov et al. 2021] and the generator is fine-tuned as done in [Roich et al. 2021]. Then, we add a new branch to the original StyleGAN generator to synthesize sketches, which are utilized in our carefully designed optimization approach to find semantically meaningful editing vectors representing input sketch-based edits. Observed that a video sequence can be disentangled into facial identity and driving motion, we classify input sketches for two editing purposes: temporally consistent editing and temporally variant editing. The former includes editing operations on the facial base shape and is directly applied to all frames. Temporally variant editing is further propagated in two different ways: time window propagation that generates a movement emerging and disappearing, and expression guidance propagation that exhibits editing effects with respect to specific expressions. Furthermore, since users often input editing operations in multiple key frames of their choice, we further design a fusion technique to combine multiple editing operations in different regions.

We perform extensive qualitative and quantitative experiments and the results show effective sketch editing performance and reasonable editing propagation results of our method. Compared with the state-of-the-art methods, our approach generates better video editing results and achieves novel temporally variant editing manipulations by sketch.

The main contributions of this work are summarized as follows.

- To the best of our knowledge, DeepFaceVideoEditing is the first sketch-based human portrait video editing system, in which the user-specified edits on key frames are represented by latent editing vectors with optimization and propagated into an entire video.
- Our system distinguishes the editing sketches into two categories: temporally consistent editing and temporally variant editing. The latter is propagated with the guidance of expression, or transformed in a temporal window.
- We develop an editing fusion technique to combine the edits of different regions in multiple frames, providing more freedom to add diverse editing operations in arbitrary selected frames.

2 RELATED WORK

2.1 Facial Image Editing

2.1.1 Editing via Conditional GANs. Generative Adversarial Networks (GANs) [Goodfellow et al. 2014] and conditional GANs [Mirza and Osindero 2014] have enabled splendid approaches for users to edit facial images via diverse user input forms, including sketches, color strokes, semantic masks, and attribute-based sliders. Based on general image-to-image translation [Isola et al. 2017; Yang et al. 2017] methods, Gu et al. [2019] learn facial features from a component level by employing semantic masks, thereby achieving convincing local editing results. Lee et al. [2020] further improve the editing performance by modeling the user editing behavior with geometry structure prior and present a more robust editing system. As semantic masks contain no style information, Zhu et al. [2020] design semantic region-adaptive normalization blocks and support style control for local regions. On the other hand, sketch is yet another user-friendly interaction intermediary and provides more detailed control compared with label maps. FaceShop [Portenier et al. 2018] and SC-FEGAN [Jo and Park 2019] both take mask, color strokes, and sketches as inputs and generate local facial editing results based on image completion frameworks. In order to control hand-drawn sketches more robustly, Yang et al. [2020] present a sketch refinement network, which generates realistic results even for coarse input sketches. DeepFaceEditing [Chen et al. 2021] further disentangles the facial geometry and appearance, allowing both local geometry editing by sketches and global appearance editing with reference images. Although previous works generate excellent editing results for single images, it is hard to directly transfer them into video editing for the following reasons. First, since the user-specified editing is only imposed on a single frame, it is a nontrivial task to infer corresponding edits for other frames. Second, some editing effects are temporally dependent and entangled with expression and pose, bringing about additional challenges for the editing propagation process.

2.1.2 Editing via GAN Latent Space Exploration. Pioneering GANs, typically StyleGAN and its follow-up works [Karras et al. 2021, 2019, 2020], could generate high-resolution photo-realistic facial images from a Gaussian distribution, by first projecting random noise to an intermediate space $\mathcal{W}$. The disentangled nature of $\mathcal{W}$ (or its extension $\mathcal{W}+$) further induces awesome discoveries that real images could be projected back into the latent space for editing.

Therefore, the first problem is how to properly project real images into the latent space, and this has caught extensive attention in the field. For example, pSp [Richardson et al. 2021] designs an encoder to map real images, sketches and/or semantic masks to an extended $\mathcal{W}+$ space, defined by the concatenation of 18 different 512-dimensional vectors, and solves an image-to-image translation problem. To make the inverse latent codes more robust for editing, E4E [Tov et al. 2021] further constrains different style vectors in $\mathcal{W}+$ to have low variance and utilizes an adversarial training strategy to enforce the style vectors lying in the distribution of $\mathcal{W}$. Although these methods could generate results that maintain main visual characteristics of input images, there still exist an identity gap and they cannot reconstruct the input images precisely in detail by using the pretrained generator. Considering the performance of reconstruction is limited by the generator's latent space, [Alaluf et al. 2021b; Roich et al. 2021] utilize the latent codes (editable but having construction distortion) of real images as pivot and fine-tune parameters of the StyleGAN generator. The slight modification of the generator is able to compensate for the discrepancy of inverting real images and achieve unprecedented identity preservation quality as a result. A few attempts have been made to extend the projection idea to videos. For example, Tian et al. [2021] present a motion generator to predict a latent code sequence, which is fed into a pretrained image generator to synthesize videos. To process real videos, Yao et al. [2021] utilize pSp [Richardson et al. 2021] to encode frames and apply editing effects with a latent Transformer. Compared with the above video generation methods, ours leverages a more robust E4E [Tov et al. 2021] encoder to project all frames, and then uniformly samples several frames to fine-tune the generator as [Roich et al. 2021] to reconstruct the input video. The video projection approach further supports novel sketch-based editing.

After projecting real images into the latent space, the second problem comes as how to manipulate the latent codes to achieve desired editing effects. [Abdal et al. 2021; Härkönen et al. 2020; Shen et al. 2020; Wu et al. 2021] utilize different methods to get linear editing directions in $\mathcal{W}/\mathcal{W}+$ for controlling global attributes in a disentangled manner. With the help of an age regression network and a cycle-consistency strategy, [Alaluf et al. 2021a] further learns a more disentangled, nonlinear path to solve the age transformation problem. The underlying 3D property of human faces triggered works on how to control the facial generation process with 3D parameters. For example, StyleRig [Tewari et al. 2020b] designs a rigging network to manipulate latent codes controlled by 3DMM parameters, and trains it in a self-supervised way by utilizing a facial reconstruction network with a differentiable render. PIE [Tewari et al. 2020a] extends StyleRig to real facial images by adding an identity preservation optimization approach to embed given images. Instead of training with unrealistic 3D rendering results in a self-supervised way, PhotoApp [Mallikarjun et al. 2021] directly utilizes collected paired data to train a latent code manipulation network and allows detailed control for lighting and view pose. Although these works achieve interesting editing results, they can only edit pre-defined attributes. In contrast, our method utilizes sketch as an interaction tool and provides users more freedom for image/video editing.

Lastly, the incorporation of multiple modalities spurs more interesting editing methods operating on the latent space, such as utilizing text as the interface via a large pretrained language encoder of CLIP [Radford et al. 2021] and combining semantic masks for latent optimization. One of the ground-breaking works in this direction, StyleCLIP [Patashnik et al. 2021], leverages CLIP [Radford et al. 2021] image/text encoders to measure the distance between text specification and edited results, by assuming co-linearity between the image latent space and the text latent space. It then iteratively optimizes or trains a mapping network to generate text-guided editing results. With the guidance of semantic masks, Barbershop [Zhu et al. 2021] uses a novel latent space to better encode spatial information, and then aligns the spatial structure of the source image to the target image to generate realistic image blending results for

hair transfer and face swapping. Ling et al. [2021] further utilize semantic masks as an interaction tool. Their approach models the joint distribution of semantic masks and real images similar to ours, and then performs latent code optimization with the semantic constraints in edited regions to synthesize edited images. Pandey et al. [2021] also share a similar latent code optimization idea but project a target sample into the manifold of source images for classification. Although previous works achieve excellent single image editing performance, it is nontrivial to apply them to video editing, where temporal variance and editing fusion need to be heavily considered. Besides, we leverage another simple but user-friendly interaction form, sketch, which has never been used in previous video editing works.

2.2 Video Editing

Video editing is a more complicated problem compared with single image editing, since the editing should be propagated reasonably from one or multiple frames to all the rest frames, and the synthesized results should be temporally coherent. Some previous works [Meyer et al. 2018; Vondrick et al. 2018; Zhang et al. 2019] have paid attention to the problem of video colorization, where they propagate the reference colors to the entire gray-scale video sequence. Lei and Chen [2019] further present an automatic video colorization approach to generate diverse colorized videos without any color reference. Although these works generate interesting video-to-video translation results, their editing effects are limited to color transformation, rather than editing propagation. To generate editing results beyond colorization, [Jamriska et al. 2019; Ruder et al. 2018; Texler et al. 2020] extend style transfer methods [Gatys et al. 2016] to videos by adding temporally consistent constraints and propagating stylization manipulations into the video sequence. Despite the success in video appearance/style modification, how to effectively edit the content of target videos is yet another important topic, and this problem is the main focus of this paper. Bhat et al. [2004] present a traditional method to edit videoes of waterfalls, rivers, flames, and smoke by analyzing and synthesizing texture motion and utilizing user-specified flow lines. The approach in [Huang et al. 2016] generates object removal results by utilizing a temporally coherent video completion approach. Kasten et al. [2021] represent videos into a novel atlas space, composed of multiple semantically interpretable layered 2D atlases, to edit videos more intuitively and creatively. Then, diverse editing operations introduced to the atlases can be mapped back into the original video and propagated reasonably. Compared with previous works, we present the first sketch-based facial video editing framework. Our method not only supports the editing of facial base shape, but also allows the manipulation of facial expression, therefore being more interactive while providing users with more editing freedom.

3 METHOD

Figure 2 shows an overview of our novel video editing method, which supports temporally consistent editing and temporally variant editing both via sketches. Given a video sequence and corresponding latent codes (projected by E4E [Tov et al. 2021]), for efficiency, we sample an input video for every several frames and fine-tune the StyleGAN3 [Karras et al. 2021] generator to reconstruct the original video, using the training strategy of [Roich et al. 2021]. With the user's input editing mask and sketch, our optimization method (Section 3.1) generates latent editing vectors that represent the sketch structure editing. Then, we divide the sketch editing operations into two categories: temporally consistent editing and temporally variant editing, and propagate the two categories of editing in different ways (Section 3.2). Temporally consistent editing exerts modification to the facial shape, which has significant influences on the whole video sequence. Since the extracted editing directions are semantically meaningful and disentangled, we directly apply the editing vectors to all video frames. Temporally variant editing is further classified into two sub-categories: expression guidance editing and time window editing. Expression guidance editing is propagated by calculating the expression similarity between the edited key frame and the rest frames, while time window editing is applied to a specific time window and generates facial movement results within that time window. In order to combine different types of editing at multiple key frames, we further design a region-aware fusion method (Section 3.3), which warps the editing masks and replaces the feature maps of the original frames with the edited features in local regions. Finally, the synthesized faces are merged into the original video while retaining the video background.

3.1 Sketch Editing Optimization

In this section, we describe our novel optimization method to generate editing vectors from the user-drawn sketches. Note that although the proposed method is designed for editing video sequences, editing vectors are generated from a single image being edited by a user. Similar to the recent image translation method [Lu et al. 2018] and semantic segmentation editing method [Ling et al. 2021], we model a joint distribution over real facial images and sketches by an extended StyleGAN generator $\widetilde{G}$ with two branches: $\widetilde{G}_x$ that generates photo-realistic images and $\widetilde{G}_s$ that generates corresponding sketches, as shown in Figure 3. Given a single key frame x for editing, we project it into the $\mathcal{W}^+$ space to generate a latent code w, and then utilize $\widetilde{G}$ to synthesize the reconstruction image $\widetilde{G}_x(w)$ and the corresponding sketch $\widetilde{G}_s(w)$. By enforcing the sketch in the region of interest to be consistent with the user-input sketch while maintaining other regions unchanged, we optimize the latent code w to generate a new latent code w_{edit}. The editing vector is obtained by calculating the difference between the two latent codes. We describe this process in detail as follows.

The generator $\widetilde{G}$ is designed based on the StyleGAN3 [Karras et al. 2021] generator G, where the real image branch $\widetilde{G}_x$ has the same network as the original generator and shares the same weights. We add another branch $\widetilde{G}_s$ to synthesize sketches from the intermediate feature maps of $\widetilde{G}_x$. Given the latent code w, $\widetilde{G}_x$ generates a set of feature maps, denoted as $\{\mathcal{F}_i \mid i = 1, \ldots, M\}$, $M = 14$ in our network. $\mathcal{F}_1$ is utilized to generate initial sketch feature maps with the resolution of 16×16 and 32 channels. $\{\mathcal{F}_i \mid i = 2, \ldots, M\}$ are further used to generate residual maps added on the initial sketch feature maps. We leverage a progressive generation process similar to StyleGAN3 and up-sample the sketch feature maps to have the same resolution as $\mathcal{F}_i$. The sketch feature maps finally synthesize

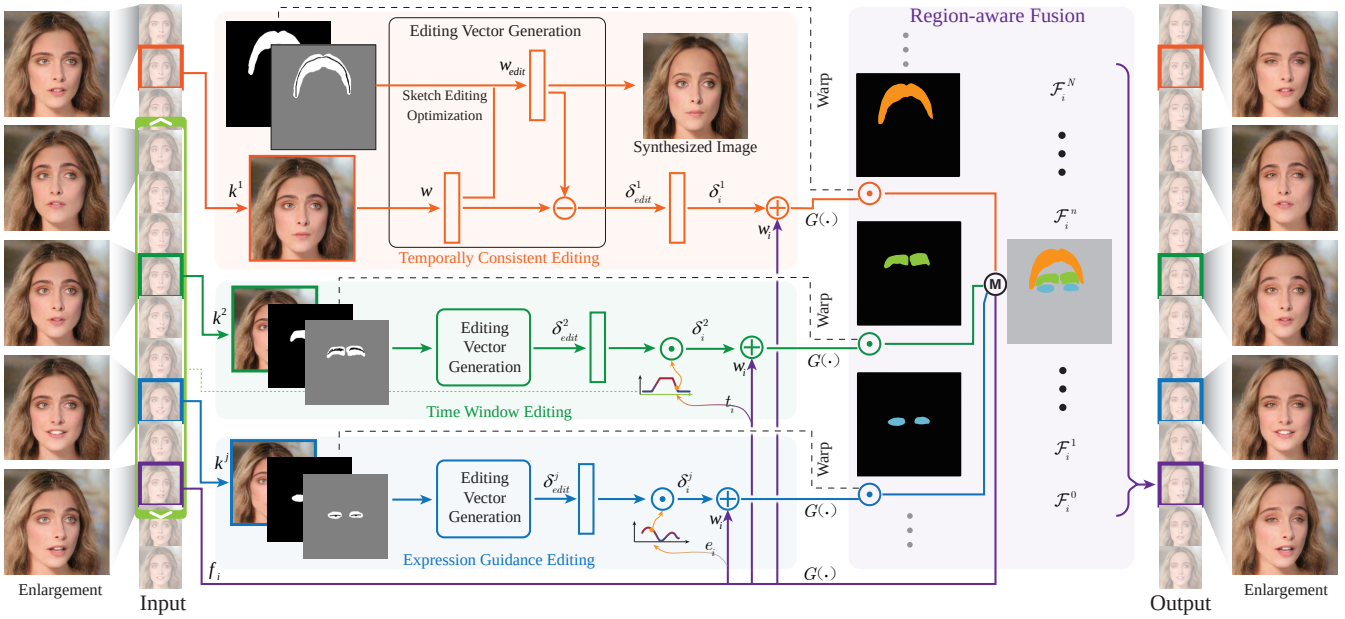


Fig. 2. An overview of our framework. Given an input video, we utilize E4E [Tov et al. 2021] to generate corresponding latent codes. Then, the Editing Vector Generation module takes the original image, user-drawn sketch and mask as input, and generates editing vectors to represent editing manipulations. The editing vectors are further propagated in different ways for each frame f_i: Temporally Consistent Editing directly copies the original editing vectors, while temporally variant editing (Time Window Editing and Expression Guidance Editing) use specific strategies to generate propagation weights for adjacent frames. Finally, multiple editing operations are fused by the Region-aware Fusion to synthesize the edited video results. Original videos courtesy of cottonbro.

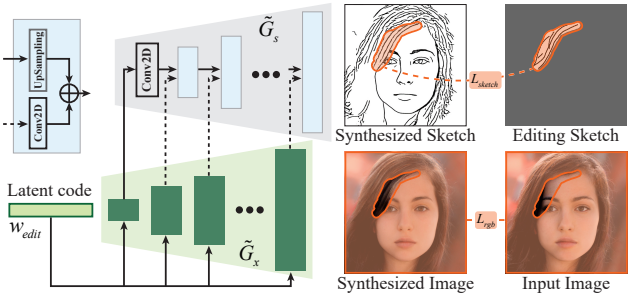


Fig. 3. We apply the original StyleGAN generator to synthesize sketches and images simultaneously. Here $\widetilde{G}_s$ generates sketches and $\widetilde{G}_x$ generates images. Given an editing sketch and its corresponding mask, we propose an optimization approach to obtaining the edited latent code w_{edit}, which is then used to synthesize the edited image.

sketch $\widetilde{G}_s(w)$, which describes the structure of generated images $\widetilde{G}_x(w)$ and is used for further editing. StyleGAN3 generator has 10 pixel border padding for each feature map $\mathcal{F}_i$. We crop the feature maps and only feed the center region for sketch synthesis since the padding border does not have direct corresponding pixels in synthetic sketches or images.

To train the sketch generation branch $\widetilde{G}_s$, we first utilize paired data to train a sketch generation network S based on the network structure and training process of Pix2PixHD [Yang et al. 2017]. The sketch generation network S takes real facial images as input and generates corresponding sketches for training $\widetilde{G}_s$. Then, we randomly sample a latent code w and feed it into $\widetilde{G}$ to generate a synthetic facial image $\widetilde{G}_x(w)$ and a predicted sketch $\widetilde{G}_s(w)$. We utilize the following losses to train $\widetilde{G}_s$:

$$L(\widetilde{G}_s) = \alpha_1 L_{VGG}(\widetilde{G}_s(w), S(\widetilde{G}_x(w))) + \alpha_2 L_{L2}(\widetilde{G}_s(w), S(\widetilde{G}_x(w))), \tag{1}$$

where L_{VGG} is the Perceptual loss used to measure the visual similarity by the pretrained VGG-19 model, and L_{L2} is a regular pixel-wise L2 loss. In our experiments, we empirically set $\alpha_1 = \alpha_2 = 1.0$.

After learning the distribution of real facial images and sketches by $\widetilde{G}$, we propose an optimization method to find the editing vector δ_{edit}, from the user's inputs of the real image x, the editing sketch s_{edit} and the mask m_{edit}, which marks the editing region. The real image x is projected into the $\mathcal{W}^+$ space to generate the original latent code w. Then, we propose to find a new latent code w_{edit}, which generates a synthetic facial image $\widetilde{G}_x(w_{edit})$ and the corresponding sketch $\widetilde{G}_s(w_{edit})$, such that the generated sketch is as close as possible to the user input in the mask region, and the generated facial image retains the original facial image outside the mask. To find w_{edit}, we minimize the following losses:

$$L_{sketch}(w_{edit}) = L_{LPIPS}(\widetilde{G}_s(w_{edit}) \odot m_{edit}, s_{edit} \odot m_{edit}), \tag{2}$$

$$L_{rgb}(w_{edit}) = L_{LPIPS}(\widetilde{G}_x(w_{edit}) \odot (1 - m_{edit}), x \odot (1 - m_{edit})), \tag{3}$$

where L_{LPIPS} is the LPIPS [Zhang et al. 2018] distance, and $\odot$ denotes pixel-wise multiplication. L_{sketch} constrains the generated image $\widetilde{G}_x(w_{edit})$ with the editing effects determined by s_{edit}, while L_{rgb} avoids undesired changes in non-editing regions. Although the LPIPS [Zhang et al. 2018] distance is specifically designed for

images, we find it is also effective for sketches, possibly due to its sensitivity for edges. The final optimization loss is:

$$L_{editing}(w_{edit}) = \beta_1 L_{sketch} + \beta_2 L_{rgb}, \tag{4}$$

where β_1 and β_2 are hyperparameters. The weights of networks are fixed and the only parameter to optimize is w_{edit}. Then, we obtain the final editing vector δ_{edit} as:

$$\delta_{edit} = w_{edit} - w. \tag{5}$$

This represents the proposed sketch editing and is later propagated to the entire video.

3.2 Temporal Editing Propagation

Given a sequence of video frames, $f_1, f_2, ..., f_N$, where N is the number of frames, we project all frames into the $\mathcal{W}^+$ space to generate a sequence of latent codes, $w_1, w_2, ..., w_N$, using the E4E [Tov et al. 2021] encoder. After generating the editing vectors δ_{edit} in Section 3.1, we design a method to propagate the editing reasonably. Specifically, the editing operations are classified into two categories: temporally consistent editing and temporally variant editing. The latter is further propagated in two different ways: expression guidance propagation and time window transformation. Since these two propagation methods are meant to generate different editing results, users define specific editing operations beforehand.

Temporally consistent editing. Some sketch editing operations have a significant influence on the entire video and show a limited relationship with facial motion or expression. These operations are mainly editing the basic shape, such as the shape of face and facial components, haircut, etc. Since the editing vectors δ_{edit} are disentangled and semantically meaningful, we directly apply them to all frames. For each frame f_i, we generate editing vectors $\delta_i = \delta_{edit}, i = 1, 2, ..., N$. They will be used to propagate the input edits to the entire video and generate final edited frames.

Time window editing. Different from a single image, a face video often exhibits different expressions or facial movements through time. Users thus tend to edit the temporal facial movement, such as adding blink or presenting smiling at some specific time. Given an editing vector δ_{edit} at a specific frame f_t, users need to input the duration time h of the editing effects and the transition time l to the editing effects. Then, for each frame f_i, we generate smooth propagation vectors δ_i by a piecewise linear function:

$$\delta_i = \gamma \cdot \delta_{edit}, \qquad i = 1, 2, ..., M \tag{6}$$

$$\gamma = \begin{cases} 0 & i < t_1 \quad or \quad i > t_4 \\ 1 & t_2 < i < t_3 \\ \frac{i - t_1}{l} & t_1 \le i \le t_2 \\ \frac{t_4 - i}{l} & t_3 \le i \le t_4 \end{cases} \tag{7}$$

$t_1 = t - h/2 - l, t_2 = t - h/2, t_3 = t + h/2, t_4 = t + h/2 + l$, where t is the time of the edited frame f_t. Similarly, the new editing vectors will be utilized to generate synthetic facial images. In this way, we can not only generate editing effects within a specified time window, but also ensure a smooth transition where the effects gradually appear and disappear, e.g., from a neutral expression to smiling, and then from smiling to a neutral expression.

Expression guidance editing. In some situations, users only want to apply editing effects on a specific expression, while retaining the original attributes or adding different editing effects on other expressions. These editing operations include some expression-driven wrinkles (e.g., nasolabial wrinkles, dimples, etc.) and some shape editing manipulations that only affect specific expressions (e.g., making eyes smaller during smiling). In order to propagate these expression guidance editing operations, we utilize a 3D reconstruction method [Deng et al. 2019] to extract expression parameters. Specifically, given several key frames $\widetilde{k^1}, \widetilde{k^2}, \ldots, \widetilde{k^M}$, where M is the number of key frames, we extract expression parameters $\boldsymbol{e_1^k}, \boldsymbol{e_2^k}, \ldots, \boldsymbol{e_M^k}$ and generate the corresponding editing vectors $\delta^1_{edit}, \delta^2_{edit}, \ldots, \delta^M_{edit}$ (obtained in Section 3.1). Notably, some key frames could exhibit no editing operations (selected directly from the original sequence) and just serve as key reference frames to emphasize that no editing should be applied to the selected expressions. For these key frames, the editing vectors are simply set as zeros. We utilize the following strategy to propagate the expression guidance editing:

$$\delta_i = \frac{1}{C} \sum_{j=1}^{M} \exp(\cos(\boldsymbol{e_i}, \boldsymbol{e_j^k})) \cdot \delta^j_{edit}, \qquad i = 1, 2, ..., N \tag{8}$$

where $\boldsymbol{e_i}$ is an expression parameter of input frame f_i and C is the normalization term calculated by $C = \sum_{j=1}^{M} \exp(\cos(\boldsymbol{e_i}, \boldsymbol{e_j^k}))$. In our experiments, key frame editing vectors $\delta^1_{edit}, \delta^2_{edit}, ..., \delta^M_{edit}$ are generated for a specific mask region, which is assigned by users at an arbitrary key frame and warped into other key frames with the warp field generated by the method in [Siarohin et al. 2019].

3.3 Region-aware Fusion

During video editing, users can select and edit an arbitrary frame and our system generates editing vectors which represent the modifications indicated by the input sketches (Section 3.1). By utilizing the propagation method (Section 3.2), we generate editing vectors for all the frames in the video. However, the above method only supports a single editing operation in a specific mask region. In practice, users often need to select multiple frames and edit different regions within them for propagation. This would naturally generate multiple editing vectors, thus requiring a robust method to consistently fuse the editing effects in the latent space. A naïve method is to directly add all the editing vectors to the original latent code and generate edited frames from these modified latent codes. However, this easily causes undesired artifacts (as shown in the ablation study in Section 4.3). To address this issue, we design a novel approach to fuse the editing operations in different mask regions.

Given a sequence of video frames, $f_1, f_2, \ldots, f_N$, users select M key frames $k^1, k^2, \ldots, k^M$ for editing in different regions, with the corresponding M input masks $m^1, m^2, \ldots, m^M$. Utilizing the optimization and propagation methods in the above sections, for each frame f_i, we generate M editing vectors $\delta^1_i, \delta^2_i, ..., \delta^M_i$, representing different sketch editing effects. We further need to generate M masks for each predicted frame f_i, denoted as $m^1_i, m^2_i, ...m^M_i$, warped from the input masks. m^j_i marks the same region as the input mask m^j while considering the expression and head motion between frame f_i and edited key frame k^j. We use the method of [Siarohin et al. 2019] to generate a warp field, which represents head motions and propagates the input editing masks to the entire video. In order to

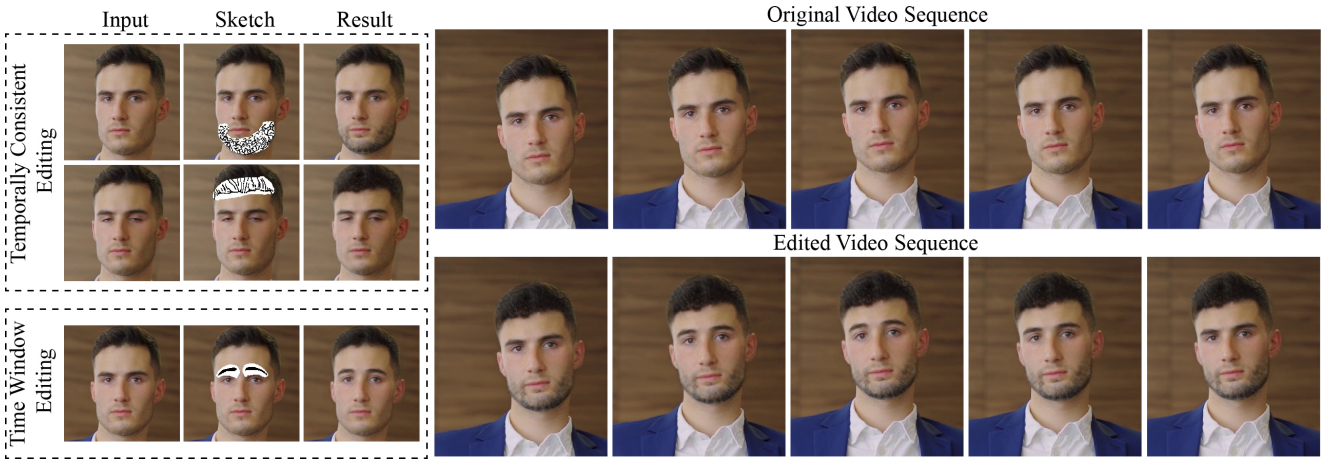


Fig. 4. The results of temporally consistent editing and time window editing. In this example, we add beard and lower his hairline. The effects are propagated to the entire sequence of video frames. Besides, an eyebrow-raising movement is added and propagated reasonably. The generated faces are merged into the original video to get complete editing results. Original videos courtesy of Mikhail Nilov.

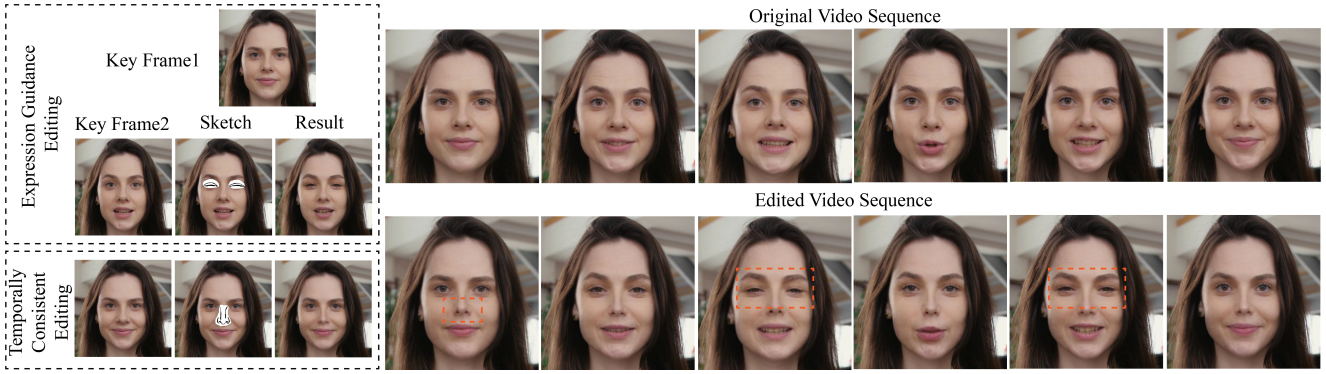


Fig. 5. The results of temporally consistent editing and expression guidance editing. In this example, we make her eyes smaller when this girl opens her mouth (Key Frame 2), while retaining the original eyes when she has a neutral expressions (Key Frame 1). Key frame 1 has no editing manipulation and thus its editing vectors are zeros. The results show that our editing effect has a close relationship with expression and the movement is modified progressively. Temporally consistent editing is also added to make her nose thinner. Original videos courtesy of SHVETS production.

fuse different editing operations, we replace specific regions of the original frame's feature maps with the new edited features by using the following formulation:

$$\mathcal{F}_i^j = \widetilde{\mathcal{F}}_i^j \odot down(m_i^j) + \mathcal{F}_i^{j-1} \odot (1 - down(m_i^j)), \tag{9}$$

where $\widetilde{\mathcal{F}}_i^j = G(w_i + \delta_i^j)$, with the initial feature maps being $\mathcal{F}_i^0 = G(w_i)$ and G being the StyleGAN3 generator. We down-sample the mask m_i^j to make sure it has the same resolution as feature maps $\mathcal{F}_i^j$ and $\mathcal{F}_i^{j-1}$. The feature map $\mathcal{F}_i^j$ is updated step by step for $j = 1, 2, ..., M$, corresponding to M editing operations. As described in Section 3.1, StyleGAN3 generates 14 intermediate feature maps, from a low resolution to a high resolution in a progressive manner. We manipulate the middle 5 feature maps from resolution 32×32 to 128×128, which are further modified by the original latent code w_i in higher resolution to generate the final fusion frame with multiple editing operations. We apply the fusion manipulation for all frames $f_i, i = 1, 2, .., N$ and synthesize the aligned edited video.

The generated faces are merged into the original video to synthesize final video with the edited results in the following way. We first utilize a face parsing model [Yu et al. 2021] to generate face masks for different facial regions, both for the input and edited frames, and calculate the union of them. Then, the masks are dilated with border pixels blurred to serve as pixel color composition weights for merging. The images are further projected back to the original video according to the alignment parameters to synthesize the final edited video results. The intermediate results and more implementation details can be found in the supplementary document.

4 EXPERIMENTS

In this section, we discuss the results of our approach and extensive experiments that show the superiority of our method to existing or alternative solutions. We first present the results of diverse video editing operations, including the editing of base shape, time window editing, and expression guided manipulation, corresponding to our

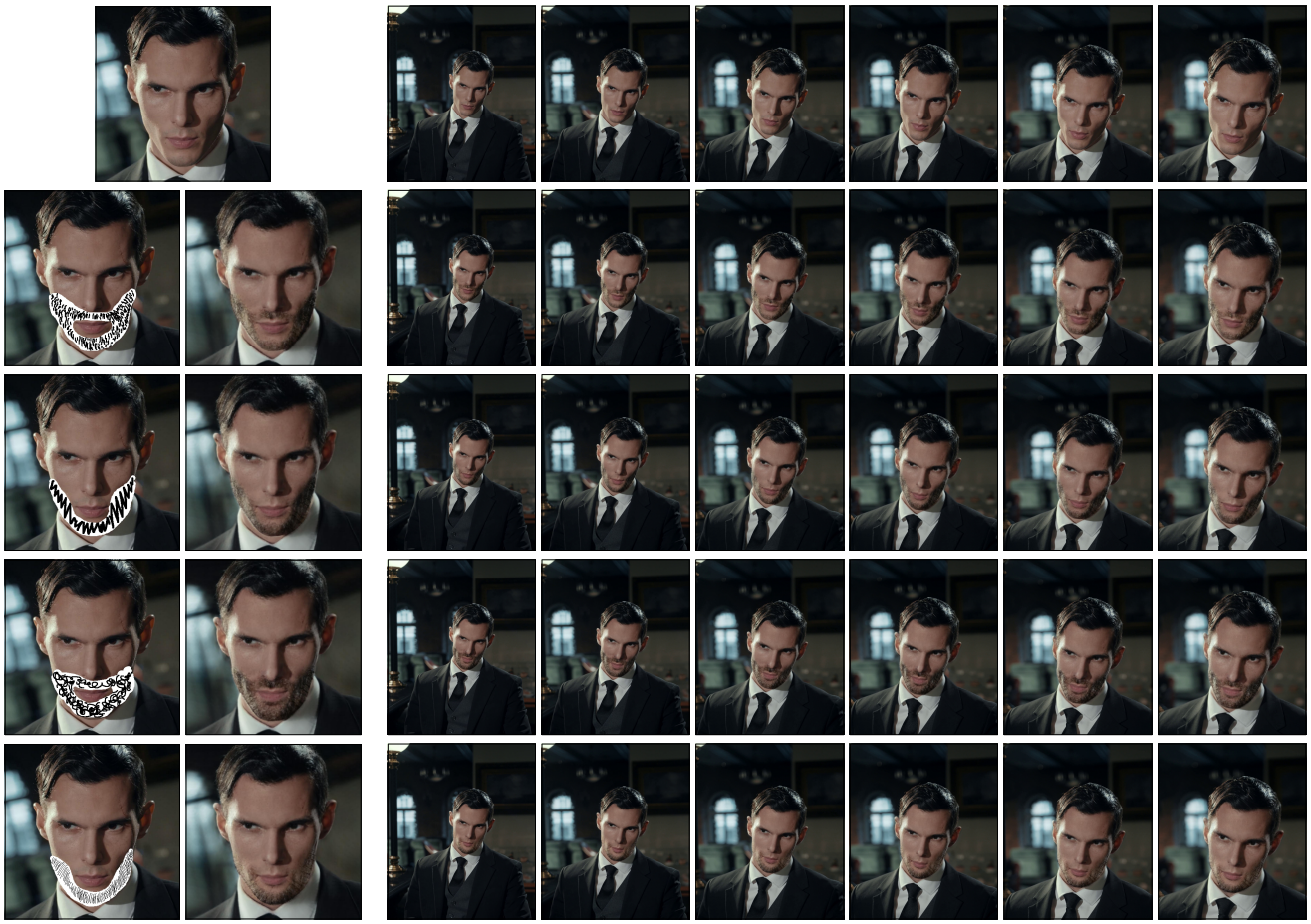


Fig. 6. The results of adding beard with different drawing styles. The 1st column shows the hand-drawn sketches and masks on a selected key frame and the 2nd column shows the corresponding editing results. The key frame before editing is given in the 1st row. For the remaining columns: the 1st row shows the original frames and the rest are propagation results by setting the editing manipulation as temporally consistent editing. Our method generates high-quality results under diverse drawing styles. Original videos courtesy of cottonbro.

Fig. 7. Our method can propagate editing manipulations across frames involving pose changes. The 1st row shows the original frames and the 2nd row presents the edited frames. The key frame highlighted with an orange border is edited into the frame shown underneath with a green border. Original videos courtesy of Tima Miroshnichenko.

different propagation strategies. Then, we evaluate our approach from three aspects, namely comparison with the state-of-the-art approaches, ablation study, and perception study, to testify the better performance of our method than alternative approaches.

	5°	15°	25°	35°	45°
Mean	-0.0250	0.0631	0.1139	0.1484	0.1714
Max	0.1551	0.1190	0.1502	0.1747	0.1962

Table 1. Statistical analysis of the relationship between editing vectors and latent codes of different poses. For each rotation angle (1st row), the latent codes of the front faces are subtracted from the projected latent codes to generate pose vector differences. Then, the Pearson correlation coefficients are calculated between editing vectors and vector differences. We use 10 videos and show the mean (2nd row) and maximum values (3rd row). All values are less than 0.2, which supports that editing manipulations and pose changes have low correlation.

Since our method first applies sketch editing on several key frames and then propagates the manipulations to a sequence of video frames, we compare our approach with the state-of-the-art methods from two branches: 1) sketch facial editing methods for single images, include SC-FEGAN [Jo and Park 2019], Deep-PS [Yang et al. 2020], DeepFaceEditing [Chen et al. 2021] and pSp [Richardson

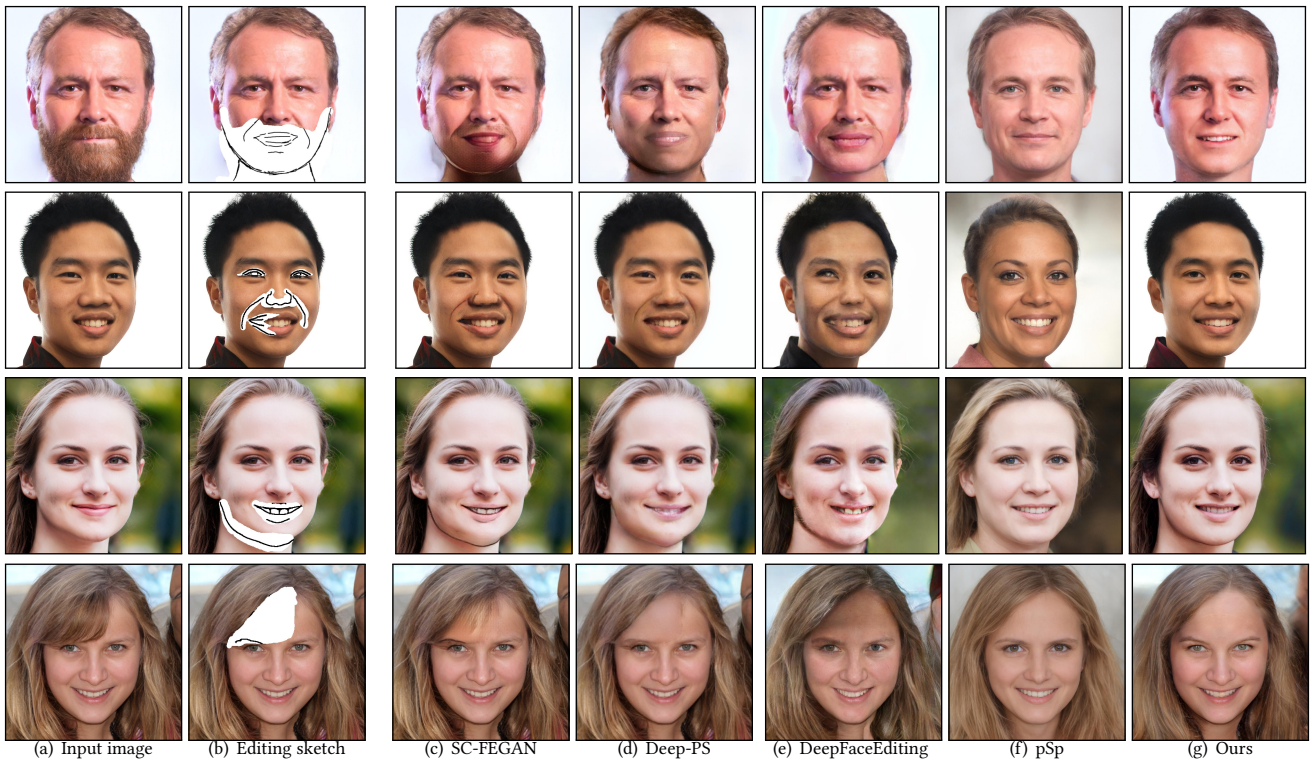


Fig. 8. Comparisons between our approach and existing sketch-based facial image editing methods. In each row, (a) is the original input image, (b) is the user-specified sketch for editing in specific mask region(s). (c) (f) are the results generated by the existing methods and (g) shows our results. Compared with (c) (e), our method generates more realistic results in local edited regions and fuses editing operations better for entire images. Compared with pSp [Richardson et al. 2021], our method retains the identity of input images well, while the identity of pSp's results is often changed dramatically.

et al. 2021], each of which takes a user-input mask and/or a sketch as input and utilizes different approaches to generate edited results for single frames; 2) sketch editing propagation methods include Few-Shot Vid2Vid [Wang et al. 2019] and First-Order [Siarohin et al. 2019]. They both propagate sketch editing manipulations or editing effects to a whole video. Then, we conduct ablation studies to show the necessity of our carefully designed components, especially the latent code optimization module and the region-aware fusion strategy. Finally, a perception study is conducted to further prove the better performance of our method compared with alternatives.

All the above experiments were carried out on a PC with an Intel i7-7700CPU, 64GB RAM, and two Nvidia RTX 2080Ti GPUs. And DeepFaceVideoEditing is implemented in Pytorch [Paszke et al. 2019] and Jittor [Hu et al. 2020]. To fine-tune the StyleGAN generator, we utilize the default hyperparameters in PTI [Roich et al. 2021], where the ADAM [Kingma and Ba 2014] optimizer with 0.0003 learning rate is used. Unless otherwise stated, we perform the fine-tuning optimization for 200 steps. When optimizing the latent editing vectors, we use the ADAM optimizer with 0.0005 learning rate, while tuning other hyperparameters to generate visually the best results. The sketch generator is trained with the FFHQ dataset [Karras et al. 2019], with the paired sketch-image data synthesized as in [Chen et al. 2020]. The test videos were collected from the pexels website[1], which contain high-resolution face videos that are free to use.

4.1 Results

Figure 4 shows an example of fusing multiple editing operations using the fusion strategy described in Section 3.3 to generate edited frames. The generated facial images are merged into the original frames with a pretrained face parsing network [Yu et al. 2021]. The temporally consistent editing controls the facial base shape (beard and hairline in Figure 4), which has a significant influence on the entire videos. The time window editing adds specific facial movements and generates progressive modification results, e.g., the eyebrow raising in Figure 4. Another type of temporally variant editing has a close relationship with expression. As shown in Figure 5, we make her eyes smaller when she opens her mouth, while retaining the original big eyes during a neutral facial expression. The proposed editing is propagated reasonably to synthesize realistic results.

Our method is robust for diverse face videos and hand-drawn sketch styles. Since the StyleGAN generator can randomly generate faces with great diversity, our method inherits this advantage and can handle diverse face videos. We have shown examples involving the diversity of gender, age, ethical groups, and human face scales.

[1]https://www.pexels.com/

For example, Figs. 4, 10, and 12 show examples of diverse ethical groups. An example with facial scale change is included in Figure 6. Please refer to the supplementary document for a child example (Figure 8) and an old man example (Figure 9). Our method is also robust for different drawing sketch styles. As shown in Figure 6, diverse styles in sketches by different users without professional training in drawing are well captured by our method. These users were given a short training on how to use our system and then instructed to add beard to a selected key frame in this example. In our approach, the editing operations are represented as editing vectors, and then added on the latent codes of the input frames. Since the sketch optimization and editing propagation lie in the latent space of StyleGAN, high-quality images/videos are generated for various sketch styles. Our system is relatively insensitive to edits and makes different drawing styles lead to broadly similar results. Nevertheless the generated facial details are still somewhat influenced by different drawing styles, as shown in Figure 6 where the thicker strokes would produce the thicker breads. This phenomenon is evident by looking at the third row and fifth row in Figure 6.

In Figure 7, we show that our method can propagate editing manipulations even for a video involving large pose changes. To explain the propagation robustness, we conducted a statistical analysis of the relationship between editing vectors and projected latent codes. Specifically, we randomly selected 10 videos with large pose changes and generated 10 editing vectors for propagation. For the convenience of data analysis in specific angles, we first preprocessed these videos to have the consistent rotation direction, from front to right profile, by flipping some of the videos that have reverse pose changes. The frames of different poses were projected by E4E [Tov et al. 2021] into the latent space, and then subtracted from the latent code of the front face to generate vector differences. Finally, the Pearson correlation coefficients were calculated between the editing vectors and the vector differences. As shown in Table 1, the maximum values of coefficients are less than 0.2, supporting that the editing vectors and pose vector differences have a low correlation. So the editing manipulations and pose changes have a weak interaction and are generally disentangled, explaining the robustness of our method for pose changes.

4.2 Comparisons

Sketch facial editing. In our framework, after users utilize sketches and masks to edit selected key frames, the edits are propagated to all the other frames. Since the editing sketch is first applied onto a single key frame, we compare our method with sketch-based facial editing methods for images, as shown in Figure 8. Since sketches are user-drawn instead of edge maps extracted from images, we use the pretrained models for all sketch-base editing methods in our experiments. SC-FEGAN [Jo and Park 2019] was trained on edge maps extracted from real images, so it is not robust enough for hand-drawn sketches and generates fuzzy and disharmonious results in local edited regions. Utilizing a sketch refinement network to process input sketches, Deep-PS [Yang et al. 2020] is more robust and synthesizes better results in local regions. However, it is based on an image completion framework and thus often generates artifacts on the boundaries of mask regions. Besides, although it utilizes the sketch refinement module, it still generates undesirable artifacts for some challenging cases, such as opening mouth in the 3rd row in Figure 8. DeepFaceEditing [Chen et al. 2021], a more recent portrait editing method with complete sketches as inputs, synthesizes edited results without boundary artifacts. However, it is still not robust enough for hand-drawn sketches and generates artifacts in edited components. pSp [Richardson et al. 2021] also takes complete sketches as conditional inputs and synthesizes high-quality facial images. However, the generated results are hard to retain the identity because pSp only utilizes sketches to infer coarse structures and thus generates inaccurate reconstruction results. We optimize the latent code with the constraints of user-drawn sketches and the original input images, enabling our model to generate more realistic results in local edited regions while retaining the identity of the original faces. Besides, since the optimized latent codes are on the distribution of real facial images, our method is more robust for editing based on hand-drawn sketches.

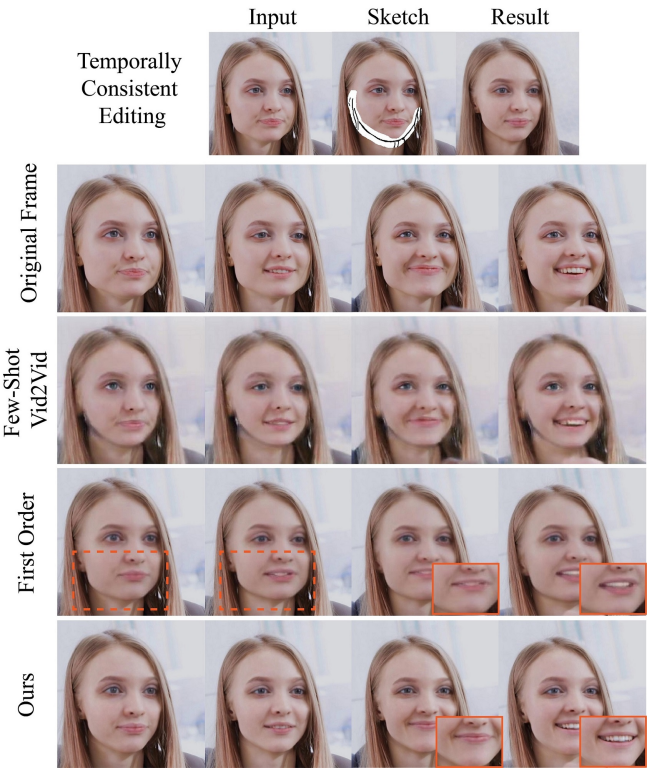


Fig. 9. The results of different propagation approaches for temporally consistent editing. Few-Shot Vid2Vid [Wang et al. 2019] converts a sequence of sketches to photo-realistic images. Since this method is trained on edge maps, it is not robust for hand-drawn sketches and often generates artifacts in edited regions. First Order [Siarohin et al. 2019] warps the single edited image to other frames driven by the original video, and thus adds undesired distortions according to the original frames and shows limited editing effects. It also generates fuzzy results in local details. Compared with these methods, our approach generates more realistic results and propagates the editing manipulations better. Original videos courtesy of Mikhail Nilov.

Sketch editing propagation. Given a sequence of video frames, users select several key frames and draw sketches and masks on

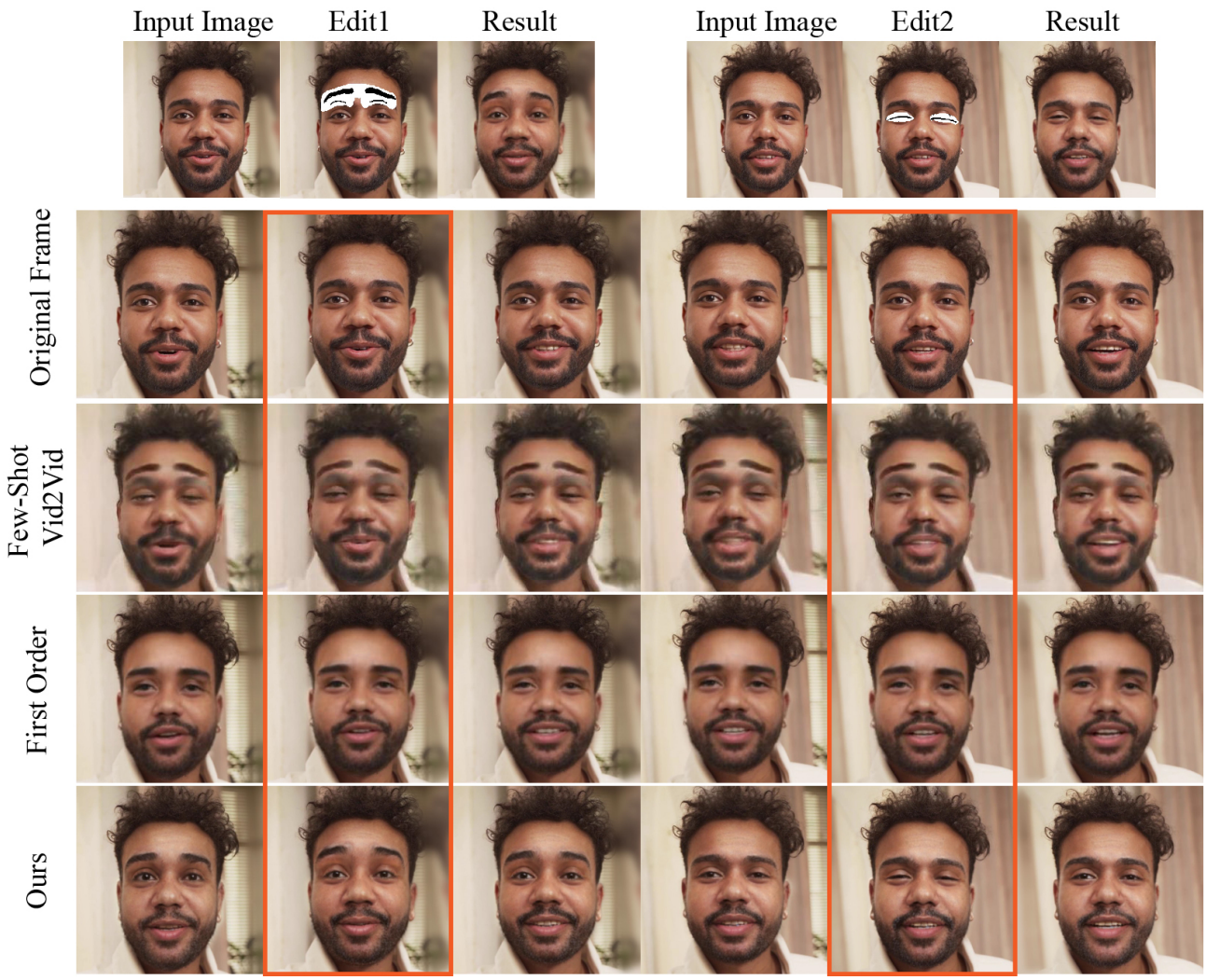


Fig. 10. The results of different propagation approaches for temporally variant editing. Users apply eyebrow raising movement and blink in this example. The editing operations are added to frames in the orange boxes and the editing effects are propagated to adjacent frames so as to achieve temporally smooth changes. Few-Shot Vid2Vid [Wang et al. 2019] propagates the editing effects to the entire video, showing no temporal variance. First Order [Siarohin et al. 2019] drives the edited frames with the movement of the original frames, so it shows no movement editing effects. Compared with previous methods, our approach generates better movement editing results and has no influence on the other frames. Original videos courtesy of ANTONI SHKRABA production.

them. The editing manipulations should be propagated reasonably. As no existing work achieves the same sketch-based facial video editing as ours, we create the following baselines for comparison. We compare our method with two possible video propagation approaches, namely, Few-Shot Vid2Vid [Wang et al. 2019] and First Order [Siarohin et al. 2019]. We first compare the temporally consistent editing effects on facial base shape, as shown in Figure 9. Few-Shot Vid2Vid [Siarohin et al. 2019] extends an image translation framework to video translation and generates a photo-realistic video from a sketch sequence. So for this method, we first extract a sequence of sketches (edge maps) from input frames using the sketch generation network in Section 3.1. To propagate the user-specified editing manipulations, we warp the user-drawn sketches and masks with a warp field generated from First Order [Siarohin et al. 2019], and then paste the hand-drawn sketches back into the original sketch sequence, only in masked regions. Finally, this new sketch sequence is utilized to synthesize video translation results. In our experiments, we train Few-Shot Vid2Vid network using its original dataset, while the inputs are changed to sketches. Since this method is trained on synthesized datasets, it is not robust for hand-drawn sketches and often generates artifacts on edited regions. The generated results are also worse than other methods. First Order [Siarohin et al. 2019] drives the edited frame generated by our method with the guidance of the original frames, and it predicts a warp field applied on the edited image and refines it with another network. Because of the warping operation on the edited

(a) Input Image (b) Input Sketch (c) Baseline1 (d) Baseline2 (e) Baseline3 (f) Ours

Fig. 11. The results of different sketch editing optimization approaches. Baseline1 optimizes the latent code only with the sketch loss and might change entire images dramatically. Baseline2 adds the RGB loss but calculates the sketch loss directly with sketch generation network S. It is hard to generate edited wrinkles and easy to change the facial identity. Baseline3 utilizes the L2 loss to replace the LPIPS loss constrained on sketches, and shows limited editing effects. Our method generates better results that maintain the facial identity well while being coherent to editing sketches.

	SC-FEGAN	Deep-PS	DFE	pSp	Ours
FID↓	17.87	18.32	18.62	19.34	**17.36**
KID×10^3 ↓	**1.62**	1.86	2.72	2.50	1.71

Table 2. Quantitative comparisons between our method and existing image editing methods.

	Few-Shot Vid2Vid	First Order	Ours
FID↓	27.41	27.82	**13.25**
KID×10^3 ↓	13.31	15.26	**3.57**

Table 3. Our method outperforms the existing video propagation methods in terms of both the FID and KID metrics.

frame, the editing effects are mixed with the original frames and are less obvious and convincing compared with our method. Besides, it generates fuzzy results in the mouth region due to the limited generation ability of the refinement network. Compared with these two methods, our approach generates more realistic results that well possess propagated editing effects.

We also compare our method with the existing approaches for temporally variant editing, as shown in Figure 10. Users apply eyebrow raising and blink in this example, where the editing operations are added to the frames in orange boxes and propagated to adjacent frames utilizing the method described in Section 3.2 (time window editing). Since the editing sketches are directly pasted onto the original frames' sketches, the results of Few-shot vid2vid [Wang et al. 2019] show no temporal variance and propagate the editing manipulations in the whole video. As for First Order [Siarohin et al. 2019], the expression and movement of the generated results are driven by the original frames, so it shows no sketch movement editing effects and synthesizes results similar to the original frames. Even in the selected key frames where the editing operations are applied, First Order shows limited editing effects. In contrast, our method modifies the expression with the guidance of the editing sketches, leading to results with smooth changes and having no influence on the other non-edited frames.

Quantitative experiments. We report the Fréchet Inception Distance (FID) [Heusel et al. 2017] and Kernel Inception Distance (KID) [Binkowski et al. 2018] in Tables 2 and 3. For the quantitative comparisons on the image editing task, 15 editing examples are synthesized by each of the compared methods and local editing regions are merged into original images for a fair comparison. As shown in Table 2, our approach outperforms alternative methods except that SC-FEGAN [Jo and Park 2019] and ours are comparable under KID. More importantly, for existing image editing solutions, it is difficult to propagate the editing manipulations across video frames. As shown in Table 3, our method outperforms other video propagation methods both in FID and KID. 16 editing examples are synthesized for video comparison. Although such quantitative evaluations suggest our method is better, these metrics are not particularly suitable for evaluation since they only measure the global image quality effectively while many editing manipulations focus on local regions. Since human eyes remain the most reliable measure, we will further evaluate the performance of our method in comparison with the existing methods through a perception study in Section 4.4.

4.3 Ablation Study

We conduct ablation studies to show the necessity of the key components of our system. Since the editing vectors are generated by a carefully designed optimization strategy, we first show the results of other possible optimization baselines in Figure 11. We optimize the latent code with both the sketch loss and the RGB loss. As shown in Figure 11 (c), using only the sketch loss (Baseline1) achieves edited effects but changes whole images dramatically. We utilize an extended StyleGAN generator to model the joint distribution over real facial images and sketches and train it by leveraging a pretrained sketch generator network. So another possible approach is to directly utilize the sketch generator network to extract sketches from generated images, and then calculate the distance between it and the edited sketch to optimize the latent code (Baseline2). As show

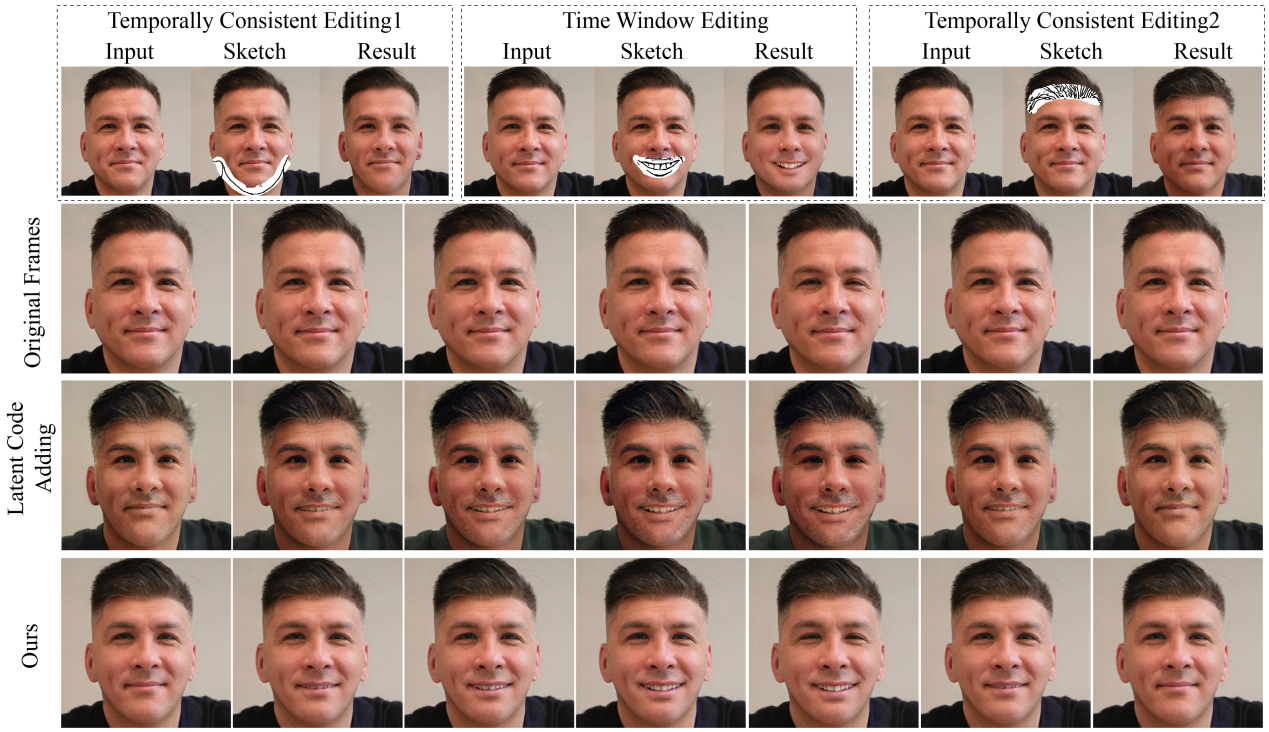


Fig. 12. The results of different fusion approaches. Users apply diverse editing operations in multiple frames for editing propagation. A naïve approach is to directly add the editing vectors generated by optimization. This approach is hard to retain image details and shows unrealistic fusion results (2nd row).Compared with this baseline, our method (last row)generates realistic fusion results while retaining the original frames' features. In this example, the temporally variant editing operation adds smiling facial movement and the two temporally consistent editing operations change the hair and face shape. Original videos courtesy of Vanessa Garcia.

in Figure 11 (d), this baseline approach changes the facial identity when applying large edits, such as adding beard, possibly due to unreasonable structure constraints. It also could not generate forehead wrinkles, which are challenging to represent. We utilize the LPIPS loss [Zhang et al. 2018] in our framework to calculate the distance between generated sketches and user-drawn sketches. Although the LPIPS loss is designed for measuring the similarity of real images, it also works well for sketches. As shown in Figure 11 (e), adding the L2 loss on sketches generates limited editing effects (Baseline3). Compared with these baselines, our method generates high-quality editing results that are consistent with the input sketches while retaining the facial identity.

Since users often apply different editing operations in multiple frames, we also present an effective editing region-aware fusion approach in Section 3.3. Taking the editing sketches in multiple frames as inputs, we can generate corresponding editing vectors by utilizing the optimization method in Section 3.1. Then, a naïve approach is to directly add them together on the original latent code and feed the fused latent code into the StyleGAN generator to synthesize results. As shown in Figure 12, temporally consistent editing1 and editing2 change the face and hair shape, respectively, and a temporally variant editing operation adds smiling movement on a generally still video. Directly adding the latent vectors shows artifacts on the jaw and changes the global facial appearance, especially when smiling is driven by a user-drawn sketch. Compared with this baseline, our method fuses multiple editing operations well and generates realistic results retaining the features of the original frames, proving the effectiveness of our region-aware fusion approach.

4.4 Perception Study

For the tasks of sketch-based facial editing and propagation, we have reported the FID and KID results in Tables 2 and 3. It can be seen from Figures 8, 9, and 10 that the results by the alternative methods exhibit artifacts in local regions. However, since such visual differences are subtle, it is difficult for existing general image quality measures to effectively capture such differences. Besides, another important aspect to evaluate is the faithfulness of generated results to user-drawn sketches. This is more difficult to measure for existing metrics. Since the most reliable criterion is human eyes, we conduct a perception study to compare the results by different methods from the perspective of human viewers.

The evaluation was done through two questionnaires. The first perception study compared our method with the existing sketch-based facial image editing methods. We synthesized 15 image editing examples (the same as those used in Sec. 4.2) Then for each edited example in the questionnaire, we showed the input images, drawn sketches and masks, and edited facial results of five facial editing methods (SC-FEGAN [Jo and Park 2019], Deep-PS [Yang et al. 2020], DeepFaceEditing [Chen et al. 2021], pSp [Richardson et al. 2021]

	Realism		Faithfulness		Identity	
	mean	*t*	*mean*	*t*	*mean*	*t*
SC-FEGAN	4.10	-11.79	4.09	-11.66	3.99	-11.31
Deep-PS	2.87	-9.44	2.86	-9.57	2.79	-9.35
DFE	3.22	-12.22	3.16	-11.85	3.09	-12.28
pSp	3.27	-10.70	3.34	-11.04	3.56	-12.71
Our	1.54		1.54		1.56	

Table 4. T-test results for perceptual study on the realism, faithfulness, and identity preservation scores in the case of image editing. Under the significance $\alpha = 0.001$, our method stands out from other methods with a lower mean value, further showing that our method performs significantly better than SC-FEGAN [Jo and Park 2019], Deep-PS [Yang et al. 2020], DeepFaceEditing (abbreviated as DFE) [Chen et al. 2021] and pSp [Richardson et al. 2021].

	Realism		Faithfulness		Identity	
	mean	*t*	*mean*	*t*	*mean*	*t*
First Order	1.99	-11.14	2.13	-10.69	1.99	-9.23
Fs-Vid2Vid	2.73	-20.21	2.63	-17.85	2.75	-20.02
Our	1.28		1.25		1.25	

Table 5. T-test results for perceptual study on the realism, faithfulness, and identity preservation scores in the case of sketch editing propagation. Under the significance $\alpha = 0.001$, our method stands out from other methods with a lower mean value, further showing that our method performs significantly better than First Order [Siarohin et al. 2019] and Few-Shot Vid2Vid (abbreviated as Fs-Vid2Vid) [Wang et al. 2019] .

and ours), placed side-by-side in a random order. Participants in our perception study were required to evaluate the results in three criteria: the realism of generated edited images, the faithfulness to the input editing sketch, and the maintenance of the original identity, by sorting all the methods from best to worst. Then, the score for each method was assigned according to its sorted position (1=strongly positive, 5=strongly negative). 40 participants (23 males and 17 females; most of them have no professional training in drawing) participated in the study, leading to 40 (participants) × 15 (sketches) = 600 subjective evaluations for each method. Figure 13(a) plots the statistics of the evaluation results. We performed one-way ANOVA tests on the realism, faithfulness, and identity preservation scores and found significant effects for all three criteria: realism ($F_{(4,70)} = 64.32, p < 0.05$), faithfulness ($F_{(4,70)} = 62.74, p < 0.05$), and identity preservation ($F_{(4,70)} = 68.24, p < 0.05$). As shown in Table. 4, paired T-test is also performed to further verify that our method performs significantly better than SC-FEGAN [2019], Deep-PS [2020], DeepFaceEditing [2021] and pSp [2021] in the same three terms.

The second perception study was conducted to evaluate the results of sketch editing propagation. Similarly, we prepared 16 video editing examples (also used in Sec. 4.2) and showed the original key frames, input sketches and masks, edited video results generated by three methods: First Order [Siarohin et al. 2019], Few-Shot Vid2Vid [Wang et al. 2019], and ours. We also evaluated our method in the above three criteria. The same group of participants were invited and sorted all the methods for each criteria. The score of each method was generated according to its sorted position (1=most positive, 3=most negative). In total, this study led to 40 (participants) × 16 (sketches) = 640 subjective evaluations for each method. Figure 13(b) plots the statistics of the evaluation results. We performed one-way ANOVA tests on the realism, faithfulness and identity preservation scores and got the values: realism ($F_{(2,45)} = 327.91, p < 0.05$), faithfulness ($F_{(2,45)} = 247.66, p < 0.05$) and identity preservation ($F_{(2,45)} = 264.96, p < 0.05$). We also performed T-tests to further confirmed that our method lead to significantly better results than First Order [Siarohin et al. 2019] and Few-Shot Vid2Vid [Wang et al. 2019]. Results are shown in Table 5.

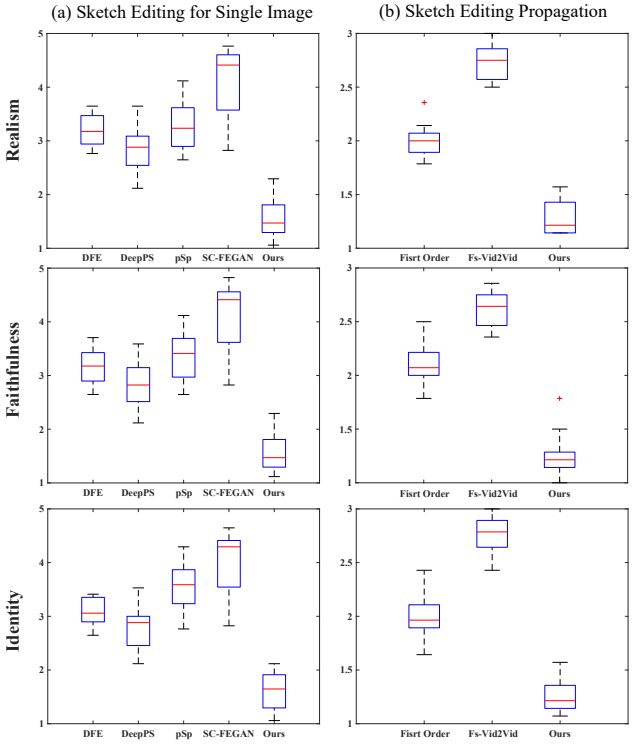
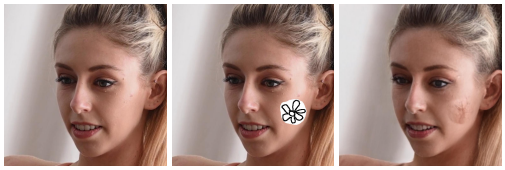


Fig. 13. Box plots of the average realism, faithfulness and identity preservation perception scores for each of the compared approaches. (a) The comparison of sketch editing for single images with five methods: DeepFaceEditing (DFE) [Chen et al. 2021], DeepPS [Yang et al. 2020], pSp [Richardson et al. 2021], SC-FEGAN [Jo and Park 2019], and ours. (b) The comparison of sketch editing propagation with three methods: First Order [Siarohin et al. 2019], Fs-Vid2Vid [Wang et al. 2019], and ours.

5 DISCUSSION AND LIMITATIONS

Although our suggested approach is effective for sketch-based video editing, there are still some limitations that should be considered. Since we utilize the pretrained StyleGAN generator to synthesize video results, the sketch editing results are limited in the StyleGAN's domain, making it hard to generate too personalized editing results. As shown in Figure 14, a flower ornament is added on the face by sketch, while our method generates an undesired scar since most of human portraits do not have the desired decoration. Furthermore, as shown in the supplementary document (Figure 2), our method cannot handle side faces well mainly because



(a) Input (b) Sketch (c) result

Fig. 14. An example of a failure case. When users want to add a flower ornament to the face via a sketched flower, our method generates a less successful result, since these examples are rare in StyleGAN's training dataset. Original videos courtesy of Marta Wave.

Fig. 15. An unsuccessful merging result. (a) is the selected frame for editing. (b) and (c) are hand-drawn sketch and fusion result, respectively. (d) shows another frame and (e) shows the corresponding propagation and fusion result. The corner of the coffee mug disappears in the fusion frame (e) because of the face merging manipulation. Original videos courtesy of Mikhail Nilov.

of the restriction of StyleGAN. Collecting a new dataset with more side faces to retrain StyleGAN might alleviate or address this issue. Besides, when designing our region-aware fusion, we assume that editing operations are likely to be applied to non-overlapping regions. When input sketches in different key frames have overlaps, our method may generate discontinuous results on the boundary of mask regions, especially for editing operations with conflicting effects in the same region (see such examples in the supplementary document, Figure 3). An appropriate mechanism needs to be devised to resolve conflicts, which we leave for future work. Moreover, when synthesized faces are realigned and merged into original videos, if some objects lie in the face boundary, the fusion strategy sometimes causes artifacts due to the restriction of facial segmentation, as shown in Figure 15. Moreover, when drawing teeth with sketches, the optimized results sometimes lead to a non-symmetric shape and position of teeth (e.g. session 4 in the video). In-depth research on the latent space of StyleGAN3 is needed to address this issue.

Our method promotes a new path to editing face video by sketching and it can be improved from various perspectives. First, with our unoptimized implementation, optimizing a single frame by sketch consumes 12 seconds on average. How to support real-time sketch editing is a future work and it is nontrivial since the predicted editing vectors should be disentangled and should not influence other unedited regions. Besides, although sketches provide much freedom for human portrait editing, it is hard to edit the pose or view point only by sketches. Some other attribute-based editing approaches [Abdal et al. 2021; Shen et al. 2020; Tewari et al. 2020b] might be combined into our method for more intuitive control. Furthermore, although our method achieves a detailed control of facial shape, facial appearance has not been considered in our implementation. In the future work, color strokes could be introduced to edit the color and texture of facial video. This is a promising research direction in the future but it remains challenging, since color strokes provide limited information to find editing vectors and it is not simple to generate temporally coherent propagation results for videos.

6 CONCLUSION

This work has presented a novel sketch-based framework for intuitively editing faces in videos. The sketch-based edits are applied to multiple key frames and propagated to the whole video reasonably. We extend the original StyleGAN by adding a sketch generation branch, and then design an optimization approach to represent sketch editing operations with editing vectors. The proposed editing operations are further classified into two categories: temporally consistent editing and temporally variant editing, which are propagated to whole videos in different ways. During the video editing process, users often apply different editing operations in multiple frames, so we propose a region-aware approach to fuse different types of editing effects, by warping the input masks and replacing the corresponding region's feature maps with edited ones. Through extensive experiments, we prove that our method generates more realistic and faithful editing results compared with other possible solutions.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for the constructive comments. This work was supported by grants from the National Natural Science Foundation of China (No. 61872440, No. 62061136007 and No. 62102403), the Beijing Municipal Natural Science Foundation for Distinguished Young Scholars (No. JQ21013), the Youth Innovation Promotion Association CAS and Royal Society Newton Advanced Fellowship (No. NAF\R2\192151), the Research Grants Council of HKSAR (Project No. CityU 11212119), and the Centre for Applied Computing and Interactive Media (ACIM) of School of Creative Media, City University of Hong Kong.

REFERENCES

Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. 2021. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (TOG)* 40, 3 (2021), 1–21.

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021a. Only a matter of style: age transformation using a style-based regression model. *ACM Trans. Graph.* 40, 4 (2021), 45:1–45:12.

Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. 2021b. HyperStyle: StyleGAN Inversion with HyperNetworks for Real Image Editing. *CoRR* abs/2111.15666 (2021). arXiv:2111.15666 https://arxiv.org/abs/2111.15666

Kiran S. Bhat, Steven M. Seitz, Jessica K. Hodgins, and Pradeep K. Khosla. 2004. Flow-based video synthesis and editing. *ACM Trans. Graph.* 23, 3 (2004), 360–363.

Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying MMD GANs. In *International Conference on Learning Representations, ICLR*.

Shu-Yu Chen, Feng-Lin Liu, Yu-Kun Lai, Paul L. Rosin, Chunpeng Li, Hongbo Fu, and Lin Gao. 2021. DeepFaceEditing: deep face generation and editing with disentangled geometry and appearance control. *ACM Trans. Graph.* 40, 4 (2021), 90:1–90:15.

Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. 2020. DeepFaceDrawing: deep generation of face images from sketches. *ACM Trans. Graph.* 39, 4 (2020), 72.

Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. 2019. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. *CoRR* abs/1406.2661 (2014). arXiv:1406.2661

Shuyang Gu, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, and Lu Yuan. 2019. Mask-Guided Portrait Editing With Conditional GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546* (2020).

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NeurIPS*. 6626–6637.

Shi-Min Hu, Dun Liang, Guo-Ye Yang, Guo-Wei Yang, and Wen-Yang Zhou. 2020. Jittor: a novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences* 63, 222103 (2020), 1–21.

Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. 2016. Temporally coherent completion of dynamic video. *ACM Trans. Graph.* 35, 6 (2016), 196:1–196:11.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5967–5976.

Ondrej Jamriska, Sárka Sochorová, Ondrej Texler, Michal Lukác, Jakub Fiser, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. 2019. Stylizing video by example. *ACM Trans. Graph.* 38, 4 (2019), 107:1–107:11.

Youngjoo Jo and Jongyoul Park. 2019. SC-FEGAN: Face Editing Generative Adversarial Network With User's Sketch and Color. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems* 34 (2021).

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4401–4410.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8107–8116.

Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. 2021. Layered Neural Atlases for Consistent Video Editing. *ACM Trans. Graph.* 40, 6, Article 210 (dec 2021), 12 pages.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. 2020. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chenyang Lei and Qifeng Chen. 2019. Fully automatic video colorization with self-regularization and diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3753–3761.

Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. 2021. EditGAN: High-Precision Semantic Image Editing. *Advances in Neural Information Processing Systems* 34 (2021).

Yongyi Lu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang. 2018. Image Generation from Sketch Constraint Using Contextual GAN. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

BR Mallikarjun, Ayush Tewari, Abdallah Dib, Tim Weyrich, Bernd Bickel, Hans Peter Seidel, Hanspeter Pfister, Wojciech Matusik, Louis Chevallier, Mohamed A Elgharib, et al. 2021. PhotoApp: Photorealistic appearance editing of head portraits. *ACM Transactions on Graphics* 40, 4 (2021).

Simone Meyer, Victor Cornillère, Abdelaziz Djelouah, Christopher Schroers, and Markus H. Gross. 2018. Deep Video Color Propagation. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 128.

Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *CoRR* abs/1411.1784 (2014). arXiv:1411.1784 http://arxiv.org/abs/1411.1784

Prashant Pandey, Mrigank Raman, Sumanth Varambally, and Prathosh AP. 2021. Generalization on Unseen Domains via Inference-Time Label-Preserving Target Projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12924–12933.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NIPS*. 8024–8035.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*. 2085–2094.

Tiziano Portenier, Qiyang Hu, Attila Szabó, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. 2018. Faceshop: deep sketch-based face image editing. *ACM Trans. Graph.* 37, 4 (2018), 99:1–99:13.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 8748–8763.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2021. Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2287–2296.

Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. 2021. Pivotal Tuning for Latent-based Editing of Real Images. *CoRR* abs/2106.05744 (2021). https://arxiv.org/abs/2106.05744

Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. 2018. Artistic Style Transfer for Videos and Spherical Images. *Int. J. Comput. Vis.* 126, 11 (2018), 1199–1219.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9243–9252.

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First order motion model for image animation. *Advances in Neural Information Processing Systems* 32 (2019), 7137–7147.

Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020a. Pie: Portrait image embedding for semantic control. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–14.

Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. 2020b. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6142–6151.

Ondrej Texler, David Futschik, Michal Kucera, Ondrej Jamriska, Sárka Sochorová, Menglei Chai, Sergey Tulyakov, and Daniel Sýkora. 2020. Interactive video stylization using few-shot patch-based training. *ACM Trans. Graph.* 39, 4 (2020), 73.

Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and Sergey Tulyakov. 2021. A Good Image Generator Is What You Need for High-Resolution Video Synthesis. In *International Conference on Learning Representations, (ICLR)*.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an encoder for StyleGAN image manipulation. *ACM Trans. Graph.* 40, 4 (2021), 133:1–133:14.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. 2018. Tracking Emerges by Colorizing Videos. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII (Lecture Notes in Computer Science, Vol. 11217)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer, 402–419.

Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. 2019. Few-shot Video-to-Video Synthesis. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 5014–5025.

Yiqian Wu, Yong-Liang Yang, Qinjie Xiao, and Xiaogang Jin. 2021. Coarse-to-fine: facial structure editing of portrait images via latent space classifications. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–13.

Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. 2017. High-Resolution Image Inpainting Using Multi-scale Neural Patch Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. 2020. Deep Plastic Surgery: Robust and Controllable Image Editing with Human-Drawn Sketches. In *ECCV*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.), Vol. 12360. 601–617.

Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. 2021. A Latent Transformer for Disentangled Face Editing in Images and Videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 13789–13798.

Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. 2021. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision* 129, 11 (2021), 3051–3068.

Bo Zhang, Mingming He, Jing Liao, Pedro V. Sander, Lu Yuan, Amine Bermak, and Dong Chen. 2019. Deep Exemplar-Based Video Colorization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 8052–8061.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.

Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. 2021. Barbershop: GAN-Based Image Compositing Using Segmentation Masks. *ACM Trans. Graph.* 40, 6 (2021).

Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020. SEAN: Image Synthesis With Semantic Region-Adaptive Normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.