Stitch it in Time: GAN-Based Facial Editing of Real Videos

Rotem Tzaban Ron Mokady Rinon Gal Amit H. Bermano Daniel Cohen-Or

The Blavatnik School of Computer Science, Tel Aviv University

Figure 1. Video editing using our proposed pipeline. Our framework can successfully apply consistent semantic manipulations to challenging talking-head videos, without requiring any temporal components or losses.

Abstract

The ability of Generative Adversarial Networks to encode rich semantics within their latent space has been widely adopted for facial image editing. However, replicating their success with videos has proven challenging. Sets of high-quality facial videos are lacking, and working with videos introduces a fundamental barrier to overcome — temporal coherency. We propose that this barrier is largely artificial. The source video is already temporally coherent, and deviations from this state arise in part due to careless treatment of individual components in the editing pipeline. We leverage the natural alignment of StyleGAN and the tendency of neural networks to learn low frequency functions, and demonstrate that they provide a strongly consistent prior. We draw on these insights and propose a framework for semantic editing of faces in videos, demonstrating significant improvements over the current state-of-the-art. Our method produces meaningful face manipulations, maintains a higher degree of temporal consistency, and can be applied to challenging, high quality, talking head videos which current methods struggle with. Our code and results are available at https://stitch-time.github.io/

1. Introduction

The advent of Generative Adversarial Networks (GANs) has brought with it a renaissance in the field of content creation and manipulation, allowing users to modify photographs in intuitive ways. In particular, the highly disentangled latent space of StyleGAN [21, 22] has been widely adapted for realistic editing of facial images. However, these semantic editing tools have been mostly restricted to images, as the editing of videos imposes an additional challenge – maintaining *temporal coherency*. Any manipulation of the video must be propagated consistently across all video frames. Prior work suggests tackling this challenge by training a GAN for video synthesis [36, 47, 51]. However, with a lack of high quality video datasets and the complications arising from an additional data dimension, video-GANs have so far been unable to match the quality of their single-image counterparts.

Instead, we propose to meet this challenge by using the latent-editing techniques commonly employed with an off-the-shelf, non-temporal StyleGAN model. We highlight a fundamental assumption about the video editing process: the initial video is already consistent. In contrast to synthesis works, we do not need to *create* temporal consistency, but only *maintain* it. Building on this intuition, we revisit the building blocks of recent StyleGAN-based editing

pipelines, identify the points where temporal inconsistencies may arise, and propose that in many cases these inconsistencies can be mitigated simply through a careful choice of tools.

We begin our investigation by identifying two types of temporal inconsistencies: local – where the transitions between adjacent frames are not smooth and display considerable jitter, and global – where inaccuracies in the GAN editing process, such as changes in identity, build up over time. We consider the recently proposed PTI [33]; A two-step approach to inversion which first finds a 'pivot' – an initial latent code that can be fed through the generator to produce an approximation of the input image. Then, the generator's weights are fine-tuned so that the specific 'pivot' code can better reproduce the target. PTI provides strong global consistency, keeping the identity aligned with the target video. However, our investigation reveals that it fares poorly on the local benchmark and produces inversions which behave inconsistently under editing operations.

At this point, we make two key observations: The generator is a highly parametric neural function, which are known to be predisposed to learning low frequency functions. As such, a small change in its inputs (the latent codes) is likely to induce only a minor variation in the generated images. Moreover, it has been shown that style-based models maintain incredible alignment under fine-tuning, particularly when transitioning to nearby domains [13, 30, 43]. As such, if the generator produces consistent editing for a set of smoothly changing latent codes - we expect any fine-tuned generator to be similarly predisposed towards temporal consistency.

With these intuitions in mind, we propose that PTI's local inconsistency arises at the first step of the process – finding the 'pivots'. More specifically, the employed optimization-based inversion is inconsistent. Highly similar frames can be encoded into different regions of the latent space, even when using the same initialization and random noise seed. On the other hand, encoder-based inversion methods utilize highly parametrized networks, and are therefore also biased to low-frequency representations. As such, an encoder is likely to provide slowly changing latents when its input only undergoes a minor change - such as when observing two adjacent video frames.

We merge the two approaches: an encoder for discovering locally consistent pivots, and generator fine-tuning to promote global consistency, and demonstrate that they already provide a strongly consistent prior. Nevertheless, they are not sufficient for editing real videos. As StyleGAN cannot operate over the entire frame, we need to stitch the edited crop back to the original video. However, inversion and editing methods typically corrupt the background extensively, making their results difficult to blend into the original frame. For this purpose, we design a novel 'stitching-tuning' operation that further tunes the generator to provide spatially-consistent transitions. By doing so, we achieve realistic blending while retaining the editing effects.

We demonstrate that our proposed editing pipeline can seamlessly apply latent-based semantic modifications to faces in real videos. Although we employ only non-temporal models, we can successfully edit challenging talking head videos with considerable movement and complex backgrounds, which current methods fail to tackle. In Fig. 1, we show several frames extracted from a video edited using our method. These demonstrate our ability to alter in-the-wild scenes and maintain temporal coherence. Through a detailed ablation study, we validate each of the suggested components and demonstrate its contribution to both realism and consistency. All videos are available as part of our supplementary materials.

2. Background and Related Work

StyleGAN-based Editing StyleGAN [21, 22] employ a style-based architecture to generate high fidelity images from a semantically rich and highly structured latent space. Remarkably, StyleGAN enables realistic editing of images through simple latent code modifications. Motivated by this, many methods have discovered meaningful latent directions, using various levels of supervision. These range from full-supervision [3, 10, 14, 34, 37, 38], such as attributes labels or facial 3D priors, to completely unsupervised and zero-shot approaches [13, 17, 28, 35, 41, 42, 44].

GAN Inversion However, applying these editing methods to real images requires one to first find the corresponding latent representation of the given image, a process referred to as *GAN inversion* [8, 25, 45, 49, 54]. Multiple works have studied inversion in the context of StyleGAN. They either directly optimize the latent vector to reproduce a specific image [1, 2, 6, 15, 33, 46, 56], or train an efficient encoder over large collection of images [4, 5, 16, 19, 23, 29, 32, 40, 53]. Typically, direct optimization is more accurate, but encoders are faster at inference. Moreover, due to their nature as highly parametric neural function estimators, encoders display a smoother behavior, tending to produce more coherent results over similar inputs. We draw on these benefits in our work.

Earlier inversion methods produced code in one of two spaces: either in the native latent space of StyleGAN, denoted $\mathcal{W}$, or in the more expressive $\mathcal{W}+$, where a distinct latent code is assigned to each of the generator's layers. It has since been shown [40, 56] that $\mathcal{W}$ exhibits a higher degree of editability — latent codes in this space can be more easily manipulated while preserving a higher degree of realism. On the other hand, $\mathcal{W}$ offers poor expressiveness, resulting

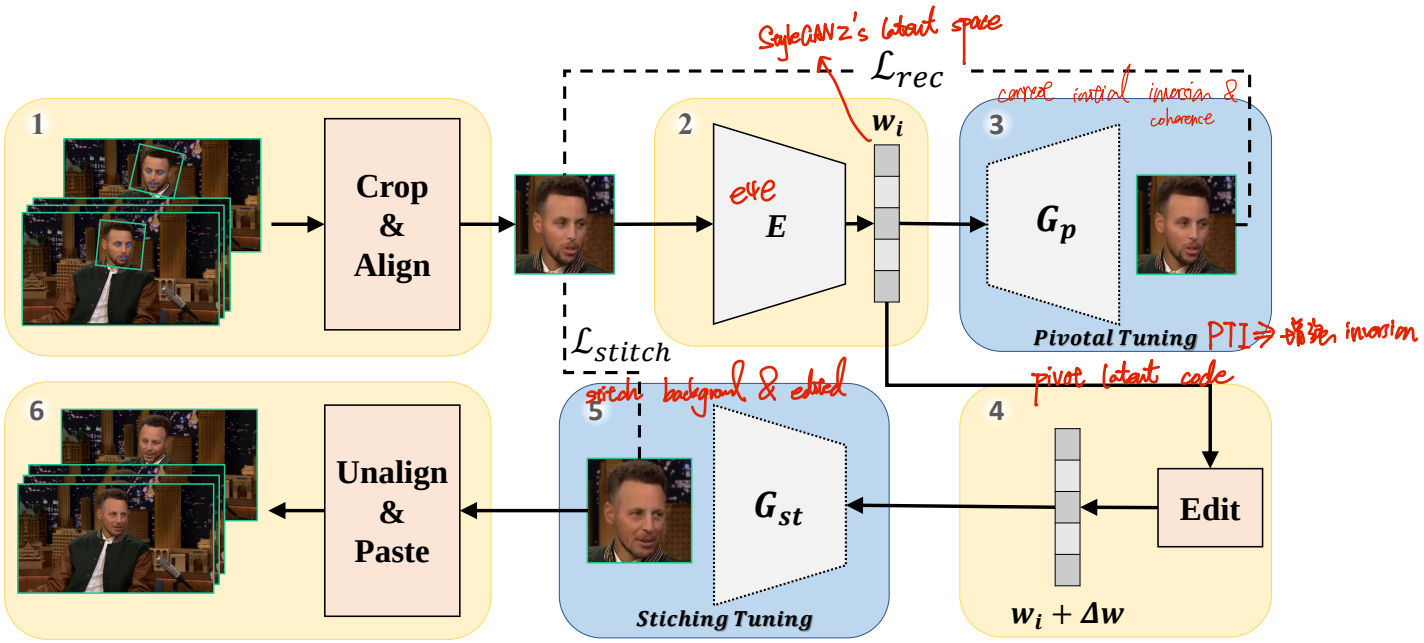

Figure 2. Our full video editing pipeline contains 6 steps. (1) Videos are split into individual frames. The face in each frame is cropped and aligned. (2) Each cropped face is inverted into the latent space of a pre-trained StyleGAN2 model, using a pre-trained e4e encoder. (3) The generator is fine-tuned using PTI across all video frames in parallel, correcting for inaccuracies in the initial inversion and restoring global coherence. (4) All frames are edited by manipulating their pivot latent codes linearly, using a fixed direction and step-size. (5) We fine-tune the generator a second time, stitching the backgrounds and the edited faces together in a spatially-smooth manner. (6) We reverse the alignment step and paste the modified face into the video.

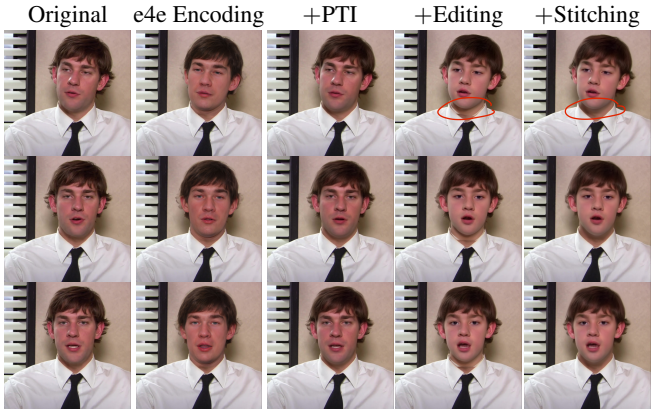

Figure 3. Visualization of our full editing pipeline. In the left column, we show three frames extracted from the source video. In the following columns, we show intermediate results of our pipeline over the same three frames. Left to right: the encoder-inversion step, the PTI fine-tuning step, the pivot editing step, and finally our stitching procedure. When not applying our stitching procedure, we use a segmentation-mask based blending procedure [48]. Note in particular the neck region, which displays considerable artifacts after the editing step which are then eliminated through our stitching-tuning approach.

in inversions that are often inconsistent with the target identity. Tov et al. [40] define this as the distortion-editability trade-off. They suggest that the two aspects can be balanced by designing an encoder that predicts codes in $\mathcal{W}+$ which reside close to $\mathcal{W}$. More recently, Roich *et al.* [33] demonstrated that one may side-step this trade-off. By fine-tuning the generator around an initial inversion code, dubbed the 'pivot', they achieve state-of-the-art reconstructions with a high level of editability. However, naïvely applying PTI to a video can result in temporal inconsistencies, as the different pivots are not necessarily coherent. A different approach to improve inversion quality was proposed by Xu et al. [46], utilizing a sequence of frames instead of a single image.

Video Generation using StyleGAN While most works have explored the use of StyleGAN in the image domain, a few recent works sought to bring its many benefits to the realm of video generation and manipulation. On the generative front, Skorokhodov et al. [36] suggest a style-based model which can produce short (e.g. 3), coherent sequences of frames. However, their method requires temporal datasets, which currently exhibit lower quality and insufficient data. Other works [12,39] train a second generator to produce temporally coherent latent codes for a pre-trained, fixed StyleGAN. They aim to disentangle spatial and temporal information. Tian et al. [39] train a temporal LSTM-based generator. Taking a different approach, Fox et al. [12] use only a single video. The produced latent sequences are

later projected to arbitrary latent codes, resulting in animating random subjects. However, while offering solid initial results, these methods do not succeed in faithfully inverting a real video or editing it.

Video Semantic Editing Many approaches suggested the editing of facial attributes in images [7, 18, 24, 26, 27]. However, applying these at the frame level typically results in temporal inconsistencies, leading to unrealistic video manipulations. To overcome this challenge, video-specific methods have been proposed. Duong et al. [11] perform facial aging in video sequences using deep reinforcement learning. Closest to our work, Yao et al. [48] propose to edit real video using StyleGAN by training a dedicated latent-code transformer to achieve more disentangled editing. These edits are applied as part of their proposed pipeline, which first includes a smoothed optical-flow based cropping-and-alignment step to reduce jitter. They then invert the frames using a $\mathcal{W}+$ encoder [32], perform the editing using their dedicated transformer, and stitch them back to the original video by employing Poisson blending using a segmentation mask. In our work, we demonstrate that by building on tools that are already highly consistent, we can achieve improved editing in more challenging settings and with fewer visual artifacts – without requiring any flow or time-based modules.

3. Method

Given a real video and a semantic latent editing direction, we aim to produce an edited video. The outcome should preserve the fidelity of the original frames while modifying them in a temporally coherent manner, achieving meaningful and realistic editing. To this purpose, we design a pipeline of six components: temporally consistent alignment, encoder-based inversion, generator tuning, editing, stitching tuning, and finally merging the results back into the original frame. In the following section, we describe in detail each core step, the tools used to implement it, and the motivation behind each choice. An overview of our full pipeline is presented in Fig. 2. In addition, a visualization of the state of the video at different stages of the editing pipeline is provided in Fig. 3.

3.1. Alignment

We employ a pre-trained StyleGAN2 model for face editing. This model, however, was trained on the FFHQ dataset, where each image was pre-processed. In particular, each image was cropped and aligned around the face. Inverting an image successfully into the latent space of the GAN thereby requires a similar crop-and-align phase. However, this pre-processing procedure consists of discrete steps (e.g. cropping) that are sensitive to the exact locations of extracted facial landmarks. This sensitivity can lead to the emergence of temporal inconsistencies, and thus we aim to reduce it. Inspired by the work of Fox et al. [12], we employ a gaussian lowpass filter over the landmarks. We find that this smoothing is sufficient to overcome any inconsistencies induced by the alignment step.

3.2. Inversion

To edit an aligned face, we must invert it into the latent space of the GAN. We do so using PTI [33], a method which first discovers a 'pivot' latent code that approximately reconstructs the input image in the GAN's more editable regions, and then fine-tunes the *generator* such that the same pivot code will produce a more accurate version of the target image. The goal of PTI was to overcome the distortion-editability trade-off [40] of inversion models, allowing for more accurate yet highly editable reconstructions. We argue that, beyond its original use, PTI can also assist in maintaining temporal coherency. The intuition here is that the original video which we wish to invert is itself temporally coherent. Therefore, if we can exactly reproduce each frame, we are guaranteed to have a coherent inversion.

In practice, however, we observe that PTI's editing performance is susceptible to inconsistencies in the pivots themselves. These manifest in two ways: First, if the pivots reside far from each other in the latent space, editing becomes less consistent. For example, the same face encoded in two different regions of the latent space might grow a different beard when the latent code is adjusted in the same direction. Second, if PTI has to 'fix' attributes (*e.g.* a lack of beard) in the inversion, then further editing of the attribute will not typically account for this fix. As a consequence, if the attribute is inconsistent between pivots (a beard appearing in only some of the inverted frames), then PTI will only need to 'fix' it in some frames, and the resulting edit will differ around these frames.

We propose that both flaws can be corrected by simply replacing PTI's optimizer-based initial inversion (*i.e.* finding the pivot) with an encoder-based version, and specifically e4e [40]. As e4e is a deep neural network with many millions of parameters, it is inherently biased towards learning lower frequency representations [31]. We expect that this property will provide a strong smoothness bias, encouraging any coherent changes between images in consecutive frames to be mapped to coherently changing latent codes. In Sec. 4.2, we analyze this property and demonstrate that the use of an encoder does indeed outperform optimization-based inversions. Note that this property relies on the smoothness of transitions between input frames, and can hence be broken by inconsistent alignment, further motivating the changes of Sec. 3.1. To reduce memory and time requirements, we employ PTI around all pivot latent codes simultaneously (as opposed to a model for each

frame).

Formally, given an N frames source video $\{x_i\}_{i=1}^N$, we denote the cropped-and-aligned frames as $\{c_i\}_{i=1}^N$. We first use the e4e encoder E to obtain their latent inversion $\{w_i\}_{i=1}^N = \{E(c_i)\}_{i=1}^N$. These latent vectors are then used as 'pivots' for PTI. Let $r_i = G(w_i;\theta)$ be the image generated from latent code w_i with a generator G parameterized by weights θ, the PTI objective is defined as:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} (\mathcal{L}_{\text{LPIPS}}(c_i, r_i) + \lambda_{L2}^{\text{P}} \mathcal{L}_{L2}(c_i, r_i)) + \lambda_R^{\text{P}} \mathcal{L}_R. \quad (1)$$

Where $\mathcal{L}_{\text{LPIPS}}$ is the LPIPS perceptual loss proposed by Zhang et al. [52], $\mathcal{L}_{L2}$ is pixel-wise MSE distance, and $\mathcal{L}_R$ is the locality regularization described by Roich et al. [33]. $\lambda_{L2}^{\text{P}}, \lambda_R^{\text{P}}$, are constants across all experiments.

3.3. Editing

Having acquired a set of temporally consistent inversions, we now turn to edit them. We demonstrate that our method works well with off-the-shelf linear editing techniques [28, 64]. We expect that, since StyleGAN itself is prone to low-frequency representations (and further motivated by path-length regularization), it will apply sufficiently consistent edits for nearby latent codes. In other words, if we edit temporally-smooth pivots using StyleGAN, we expect to generate a temporally smooth sequence. As we later demonstrate, this expectation aligns with our experimental results. Formally, given a semantic latent editing direction δw, we utilize the PTI-weights θ_p to obtain our edited frames $e_i = G(w_i + \delta w; \theta_p)$.

3.4. Stitching Tuning

As a final step, we must inject the edited face e_i back into the original video. Merely overwriting the original location of the cropped face is insufficient, as the editing process typically leads to changes in the background which are noticeable around the crop boundary. For instance, Figs. 3 and 9 demonstrate that artifacts emerge when stitching the crop naïvely. Prior works [48] elected to limit the modifications to the face area by using segmentation masks derived from the landmarks detected in the alignment step and performing Poisson blending. Such stitching, however, is still prone to inconsistencies around the border, leading to artifacts such as the appearance of ghostly outlines (see Fig. 7). We propose to tackle this limitation through a novel tuning technique inspired by PTI, referred to as *stitching tuning*. An overview of this technique is provided in Fig. 4. For each edited frame, we designate a boundary at the edge of the segmentation mask. We then briefly fine-tune our generator around the *edited pivot* with a dual objective. First, we aim to restore the boundary to its pre-inversion values, that is to blend it perfectly in the original frame. Second, we wish to retain our editing results, by requiring similarity

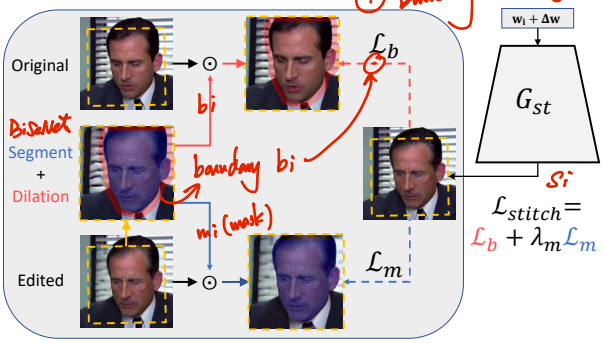

Figure 4. Outline of our stitching-tuning method. We start with generating an edited image using a modified pivot code and segment the image using an off-the-shelf segmentation network [50]. The segmentation mask is dilated, creating a boundary region. We then fine-tune the generator so that the modified pivot will provide an image that is (a) consistent with the original edit inside the face mask (blue), and (b) consistent with the original background inside the boundary mask (red). We synthesize the final image using the tuned generator and paste it inside the dilated mask region (blue + red).

to the edited frame in the area covered by the segmentation mask. This short tuning session can successfully restore the boundary and induce a smooth transition towards the center of the image, without affecting the edited face.

Formally, we first use off-the-shelf pre-trained segmentation network [50] to produce segmentation masks $\{m_i\}_{i=1}^N$ for all frames. We then perform dilation on each to obtain a set of expand masks $\{m_i^d\}_{i=1}^N$. The boundary is defined as their element-wise *xor* operator $\{b_i\}_{i=1}^N = \{m_i \oplus m_i^d\}_{i=1}^N$. Let $s_i = G(w_i + \delta w; \theta_{st})$ denote the outcome of the stitching tuning procedure, our first objective is blending the boundary to the original image:

$$\mathcal{L}_{b,i} = \mathcal{L}_{L1}(s_i \odot b_i, x_i \odot b_i), \quad (2)$$

where $\odot$ is the element-wise multiplication. The second objective is preserving the editing result:

$$\mathcal{L}_{m,i} = \mathcal{L}_{L1}(s_i \odot m_i, e_i \odot m_i). \quad (3)$$

These loss terms used to further optimize the generator weights θ_{st} for each frame:

$$\min_{\theta_{st}} \mathcal{L}_{b,i} + \lambda_m \mathcal{L}_{m,i}, \quad (4)$$

where the weights are initialized with the PTI weights θ_p, and λ_m is constant across all experiments. Finally, we perform the inverted alignment and stitch each frame s_i to the original frame x_i using the dilated masks $\{m_i^d\}_{i=1}^N$. Gaussian blur might be applied to further smooth the edges of the masks.

Figure 5. Multiple editing results over a single video. Our model preserves the original video details while enabling a range of semantic manipulations.

3.5. Implementation Details

For all experiments, we set $\lambda^{\mathrm{P}}_{L2} = 10$, $\lambda^{\mathrm{P}}_{R} = 0.1$ and $\lambda_m = 0.01$. When tuning a model with PTI we use a learning rate of $3e-5$ and train until each frame is observed 80 times. When tuning a model for stitching, we increase the learning rate to $3e-4$ and the number of training iterations to 100 per frame. For all other implementation details we follow PTI.

4. Experiments

We demonstrate the effectiveness of our method by applying it to a range of in-the-wild videos gathered from popular publicly available content. These include challenging scenes characterized by complex backgrounds and considerable movement. Individual frames were edited using InterFaceGAN [34] and StyleCLIP [28]. All experiments were conducted on a single NVIDIA RTX 2080. The total editing time for a single 300 frames video is roughly 1.5 hours. The full videos for each setup can be found in our supplementary materials.

4.1. Qualitative Results

In Figs. 5 and 6 we show key frames extracted from videos edited using our method. Our approach can handle challenging, highly detailed backgrounds, as well as considerable head movement and speech, all of which are beyond the scope of the current state-of-the-art. Moreover, by editing at the frame level, our method is inherently compatible with existing editing techniques, can support manipulations in multiple latent spaces, and can be easily applied for both spatially local (e.g., smile) and global (e.g., age) changes.

In Fig. 7 we compare our method to [48], the current state-of-the-art in semantic editing of faces in videos. When evaluating their method on our more challenging scenes, we observe considerable quality degradation and loss of temporal coherence. Our method, in contrast, maintains high fidelity and consistent editing without relying on any explicit temporal smoothing. Note in particular the blurry artifacts induced by [48] when blending the shirt. Moreover, the employment of optimization-based PTI results in the age changing between the different frames.

In Fig. 8 we showcase our ability to edit out-of-domain videos. Both the encoder and PTI can seamlessly adapt to animated faces. Furthermore, the alignment of fine-tuned StyleGAN models ensures we can re-use the same editing directions, as previously demonstrated by [13], [5] and [55].

4.2. Quantitative Results

We next evaluate our method quantitatively. As previously outlined, we expect that encoder-based methods will be smoother at the local level, avoiding the jitter induced by optimization techniques. On the other hand, we expect them to display considerable identity drift at the global scale - with the minor inversion inconsistencies between the frames building up over time. To validate our intuition and evaluate the temporal coherence of videos, we propose two novel metrics: temporally-local (TL-ID) and temporally-global (TG-ID) identity preservation.

In the first case, TL-ID, we aim to evaluate the video's consistency at the local level. We do so by employing an off-the-shelf identity detection network [9] to evaluate the identity similarity between pairs of adjacent video frames. To account for the effect of inconsistencies in the identity network itself, we normalize these identity preservation scores by the similarity score of each pair of frames in the original video. Finally, we average the normalized scores over the entire video, and then once again over a set of videos. Higher TL-ID scores indicate that the method produces *smooth* results, without considerable *local* identity jitter.

Our second metric, TG-ID, employs the same identity detection network and averaging scheme to evaluate the similarity between all possible pairs of video frames, not necessarily adjacent. This metric aims to capture longer-

Figure 6. Additional Video editing results using our proposed pipeline. For most modifications, our stitching framework can handle more challenging cases such as long hair.

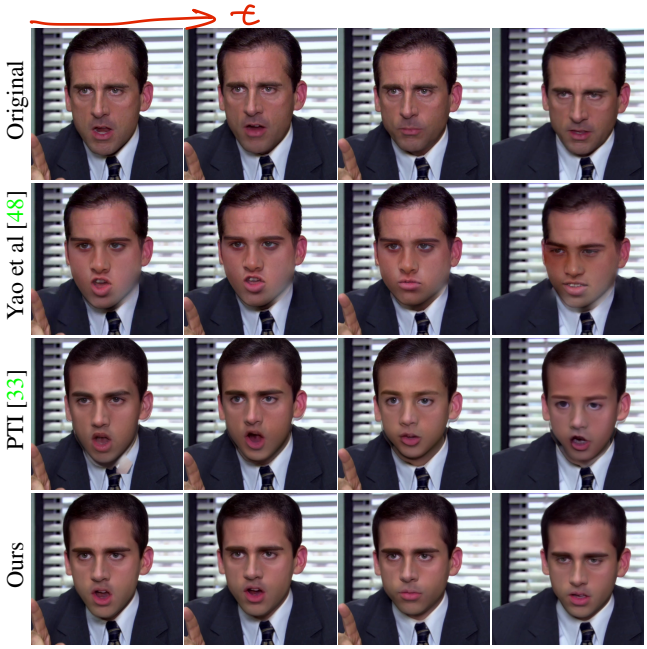

Figure 7. Visual comparison to alternative editing pipelines. Our method retains a higher degree of temporal consistency, produces realistic editing, and successfully mitigates blending-induced artifacts.

Figure 8. Out-of-domain video editing. Our method can seamlessly adapt to other facial domains, and can handle challenging poses and expressions.

range coherence and identify slow, but consistent identity drift. For both metrics, a score of 1 would indicate that the method successfully maintains the identity consistency of the original video. Note that both metrics only compare frames *within* a given video. As such, they do not measure the similarity of the inversions to those of the original video. Rather, we focus on quantifying the temporal consistency of these inversions and the editing operations they support.

We utilize both metrics to evaluate our method against the baseline PTI [33], as well as Latent Transformer [48] which makes use of a pre-trained pSp [32] encoder for inversion. The results are shown in Tab. 1. As expected, encoder-based methods outperform the optimization-based method on local consistency, demonstrating that they do indeed provide a smooth prior. Moreover, combining an encoder with PTI results in local consistency which is just shy of 1, showing that the proposed pipeline is sufficient to inherit almost all temporal consistency from the source video.

On the global front, PTI improves identity preservation over longer time spans. While editing is still susceptible to the inconsistencies of the local pivots, the encoder-based methods lead to an identity drift that eventually surpasses the local jitter. PTI, meanwhile, constantly re-aligns the identity to that of the source video, avoiding longer term drift. Notably, PTI does demonstrate some drop in global

Model	TL-ID ↑	TG-ID ↑
Latent Transformer	0.976	0.811
PTI (optimization)	0.933	0.901
Ours	**0.996**	**0.933**

Table 1. Temporal consistency metrics. Encoder based methods display improved identity preservation at the local (adjacent frame) level, but show considerable identity drift over time. PTI, preserves a greater degree of global identity, at the cost of local jitter from inconsistent pivots. Our pipeline outperforms the alternatives and achieves a local-identity preservation score which is nearly equal to the original video (1), demonstrating our ability to maintain a high degree of consistency.

performance when compared to the local score. We hypothesize that this originates in the increased distance between pivots when optimizing around distant frames. By leveraging PTI, our method can similarly maintain a high level of global consistency, and even outperform PTI thanks to more consistent pivot codes.

4.3. Ablation Study

We further demonstrate the benefits of each component in our pipeline by conducting an ablation study. In Fig. 9 we show key frames from a video edited using our method when crucial steps of the pipeline are removed or replaced. We showcase the effects of replacing the encoder with an optimization method (w/o e4e), removing the PTI generator-tuning step, replacing the stitching-tuning step by naïvely pasting the edited image inside the segmentation mask, and finally our full pipeline.

Without an encoder, the edited frames become inconsistent when the face undergoes considerable movement or changes in expression. Without PTI, frames are less faithful to the original video, stitching performance suffers, and identity changes over longer time periods. Without stitching, artifacts appear around the hair and borders of the segmentation mask (*i.e.* the edge of the face). Our full method maintains both local and long term consistency, and seamlessly melds the edited region into the original frame.

5. Discussion

We presented a novel approach for semantic editing of facial videos. By employing smooth and consistent tools, we demonstrated that standard StyleGAN editing techniques can be readily applied to in-the-wild videos, without compromising temporal coherency.

While our method works well in many practical scenarios, it still faces some limitations. Particularly, the StyleGAN alignment process is prone to leaving portions of hair (*e.g.*, pigtails) outside the cropped region. These 'external' regions do not undergo any semantic manipulations, and may result in jarring transitions when attempting to mod-

Figure 9. Qualitative demonstration of the importance of our pipeline components. Replacing the encoder with an optimization step results in poor editing consistency. Without PTI, identity drifts over time, and stitching performance deteriorates. Replacing stitching with a mask-based blending scheme results in visual artifacts, such as sharp transitions in hair regions. Our full pipeline successfully avoids these pitfalls and generates a consistent video.

ify attributes such as hair length or color. A further limitation arises in the form of the 'texture sticking' effect investigated in StyleGAN3 [20]. The use of per-frame optimizations, rather than latent-space interpolations, significantly reduces this effect. However, in some cases it is still visible. We hope that as inversion and editing tools for StyleGAN3 emerge, they could be joined with our approach in order to obtain sticking-free results.

Perhaps surprisingly, our framework produces coherent videos without requiring complex machinery designed to directly enforce temporal coherence. These results indicate that spectral and inductive biases can play a crucial role in maintaining coherency, yielding significant advantages over attempts to brute-force consistency through loss terms. In addition, we highlight the challenge of stitching an edited crop to the video and propose a designated tuning scheme which can avoid the pitfalls associated with current Poisson-blending approaches. Looking forward, we hope that our approach can be improved with temporally-aware goals that are meant to supplement it, rather than serve as substitutions. For example, it may be possible to fine-tune the inversion encoder on the input video, to motivate greater consistency in the generated codes.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE international conference on computer vision*, pages 4432–4441, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 2

[3] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows, 2020. 2

[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 2

[5] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. *arXiv preprint arXiv:2111.15666*, 2021. 2, 6

[6] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. 38(4), 2019. 2

[7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. 4

[8] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018. 2

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

[10] Emily Denton, Ben Hutchinson, Margaret Mitchell, and Timnit Gebru. Detecting bias with generative counterfactual face attribute augmentation. *arXiv preprint arXiv:1906.06439*, 2019. 2

[11] Chi Nhan Duong, Khoa Luu, Kha Gia Quach, Nghia Nguyen, Eric Patterson, Tien D Bui, and Ngan Le. Automatic face aging in videos via deep reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10013–10022, 2019. 4

[12] Gereon Fox, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. Stylevideogan: A temporal generative model using a pretrained stylegan. *arXiv preprint arXiv:2107.07224*, 2021. 3, 4

[13] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators, 2021. 2, 6

[14] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties, 2019. 2

[15] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior, 2020. 2

[16] Shanyan Guan, Ying Tai, Bingbing Ni, Feida Zhu, Feiyue Huang, and Xiaokang Yang. Collaborative learning for faster stylegan embedding. *arXiv preprint arXiv:2007.01758*, 2020. 2

[17] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 2

[18] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019. 4

[19] Kyoungkook Kang, Seongtae Kim, and Sunghyun Cho. Gan inversion for out-of-range images with geometric transformations, 2021. 2

[20] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *CoRR*, abs/2106.12423, 2021. 8

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1, 2

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 2

[23] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing, 2021. 2

[24] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. *arXiv preprint arXiv:1706.00409*, 2017. 4

[25] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017. 2

[26] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3673–3682, 2019. 4

[27] Ron Mokady, Sagie Benaim, Lior Wolf, and Amit Bermano. Masked based unsupervised content transfer. In *International Conference on Learning Representations*, 2019. 4

[28] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery, 2021. 2, 5, 6

[29] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14104–14113, 2020. 2

[30] Justin N M Pinkney and Doron Adler. Resolution Dependant GAN Interpolation for Controllable Image Synthesis Between Domains. *arXiv preprint arXiv:2010.05334*, 2020. 2

[31] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 4

[32] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2, 4, 7

[33] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021. 2, 3, 4, 5, 7

[34] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 2, 5, 6

[35] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020. 2

[36] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. *arXiv preprint arXiv:2112.14683*, 2021. 1, 3

[37] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. *arXiv preprint arXiv:2004.00121*, 2020. 2

[38] Ayush Tewari, Mohamed Elgharib, Mallikarjun B R., Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Pie: Portrait image embedding for semantic control, 2020. 2

[39] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. *arXiv preprint arXiv:2104.15069*, 2021. 3

[40] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation, 2021. 2, 3, 4

[41] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. *arXiv preprint arXiv:2002.03754*, 2020. 2

[42] Binxu Wang and Carlos R Ponce. A geometric analysis of deep generative image models and its applications. In *International Conference on Learning Representations*, 2021. 2

[43] Zongze Wu, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. Stylealign: Analysis and applications of aligned stylegan models, 2021. 2

[44] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[45] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey, 2021. 2

[46] Yangyang Xu, Yong Du, Wenpeng Xiao, Xuemiao Xu, and Shengfeng He. From continuity to editability: Inverting gans with consecutive images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13910–13918, 2021. 2, 3

[47] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers, 2021. 1

[48] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13789–13798, 2021. 3, 4, 5, 6, 7

[49] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, and Minh N. Do. Semantic image inpainting with deep generative models, 2017. 2

[50] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129(11):3051–3068, 2021. 5

[51] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *Submitted to The Tenth International Conference on Learning Representations*, 2022. under review. 1

[52] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 5

[53] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. *arXiv preprint arXiv:2004.00049*, 2020. 2

[54] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pages 597–613. Springer, 2016. 2

[55] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks, 2021. 6

[56] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Improved stylegan embedding: Where are the good latents?, 2020. 2