Temporally consistent semantic video editing

Yiran Xu[1], Badour AlBahar[2], and Jia-Bin Huang[1]

[1] University of Maryland, College Park
[2] Virginia Tech

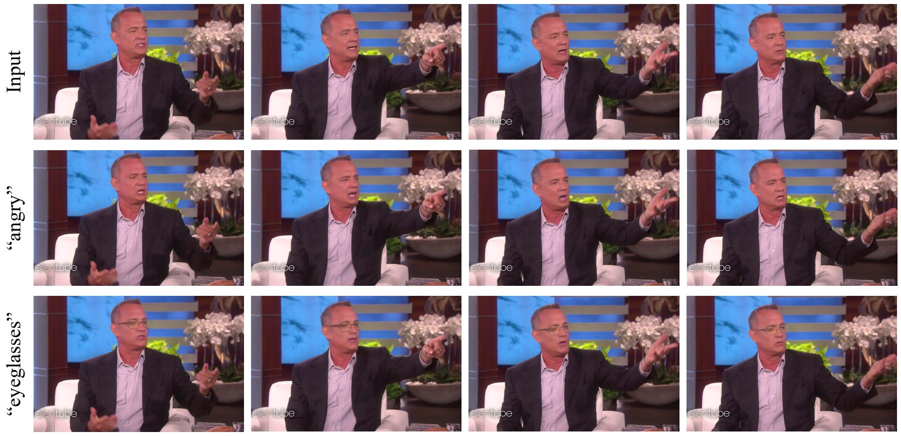


Fig. 1. Temporally consistent video semantic editing. We present a method for editing the semantic attributes of a video using a pre-trained StyleGAN model. Here we showcase free-form text based editing from SytleCLIP [44] to make the person appear “angry” (2nd row) or wear “eyeglasses” (3rd row).

Abstract. Generative adversarial networks (GANs) have demonstrated impressive image generation quality and semantic editing capability of real images, e.g., changing object classes, modifying attributes, or transferring styles. However, applying these GAN-based editing to a video independently for each frame inevitably results in temporal flickering artifacts. We present a simple yet effective method to facilitate temporally coherent video editing. Our core idea is to minimize the temporal photometric inconsistency by optimizing both the latent code and the pre-trained generator. We evaluate the quality of our editing on different domains and GAN inversion techniques and show favorable results against the baselines.

Keywords: Video editing, GAN editing, video consistency

1 Introduction

Generative adversarial models (GANs) [16] have shown remarkable ability to generate photorealistic images in various domains such as faces and common objects [10,28,29]. GANs take a latent code (usually sampled from a Gaussian distribution) as input and produce an image as the output. *GAN inversion* techniques allow us to project a *real image* onto the latent space of a pretrained GAN and retrieve its corresponding latent code. The pretrained GAN generator can then reconstruct that image using the estimated latent code. Modifying this estimated latent code opens up exciting new opportunities to perform a wide range of high-level editing tasks that are traditionally challenging, e.g., changing semantic object classes, modifying high-level attributes of the object/scene, or even applying 3D geometric transformations. We refer to the modification of the latent code with a semantic change in the image as *semantic editing*.

Semantic editing in images. A recent line of research work [66,1,2,21,65,6,49] has shown promising results in reconstructing an input image by either optimizing the latent code (or latent variables) or directly predicting the latent code via an image encoder. These GAN inversion techniques enable interesting semantic photo editing applications. For image-level editing applications, several approaches [22,51,52] find specific semantic directions in the latent space, e.g., changing poses, colors, or age, while others [15] aim to change the global style, e.g., photo → sketch. We denote them as *In-domain* and *Out-of-domain* editing, respectively. With these *image* GAN inversion-based semantic editing approaches, how can we extend them to *videos*?

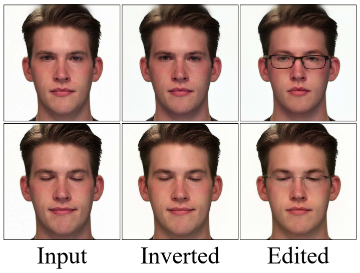


Fig. 2. Issues with per-frame editing. While current methods achieve faithful inversion and photorealistic editing, the results are inconsistent across frames (*eyeglasses*) and may fail to preserve details of the input video (*lips*).

Per-frame editing. One straightforward way is to apply existing GAN inversion techniques [21,65,6,49] for each frame in a video *independently*. Figure 2 shows an example of applying a StyleCLIP mapper [44] on two frames. The input and the independently reconstructed frames look plausible when viewed individually, but two edited frames exhibit inconsistency (e.g., the frame of the eyeglasses). Recently, Yao et al. [62] learns to predict per-frame semantic editing directions for editing face videos. However, the edited videos suffer from apparent temporal flickering and fail to preserve facial identity.

Our work. In this paper, we present a method for *temporally consistent* video semantic editing. We start from the existing GAN inversion approaches [6,49] to obtain the latent code for each frame. We first modify the latent code to achieve the initial per-frame editing results. However, such a direct editing approach results in temporal inconsistencies in the modified video's appearance or style. To deal with this challenge, we propose to compute bi-directional optical flow estimated from a frame pair sampled from the video. We can then adjust the latent code and the generator to minimize the photometric loss (along with valid flow vectors). We present a two-phase optimization strategy. In the first phase, we update only the latent codes via an MLP (with generator parameters frozen) to adjust the consistency of the detailed appearance. In the second phase, we finetune the generator with a local regularization to maintain the editability of the latent space. Our two-phase optimization approach helps achieve significantly improved temporal consistency while preserving the edited contents.

Concurrent work. Two concurrent work [57,7] also apply StyleGAN for video editing. These methods either use per-frame pivot tuning [49] for maintaining the similarity between the edited and input frame [57] or apply latent vector smoothing [7] with StyleGAN3 [27]. Our method differs in 1) the use of explicit temporal consistency optimization and 2) the applicability of performing both in-domain and out-of-domain editing.

Our contributions:

- We tackle a task on GAN-based semantic editing in videos. We propose a simple yet effective flow-based approach to mitigate the temporal inconsistency of a directly (frame by frame) edited video.
- We present a two-phase optimization approach for updating the latent code *and* generator to preserve the video details.
- Our method is agnostic and can be applied to different GAN inversion and editing approaches.

2 Related Work

Generative adversarial networks. The quality and resolution of generated images have been achieved rapidly in recent years [28,29,25,27,10]. These GAN models can map a random latent code (a noise vector) to a photorealistic image. Many recent efforts have been devoted to improving the generator architectures [24,28,29,27], training strategies [10], loss function designs [41,18], and regularization [42]. Our work builds upon existing pretrained StyleGAN models as they demonstrate disentangled latent space for editing. Instead of *generating synthetic images*, our goal is to *edit real videos*.

GAN inversion. GAN inversion [66,61] allows us to reconstruct real images by projecting them onto a pretrained GAN's latent space. These techniques facilitate interesting photo editing applications. They can be split into encoder-based [40,55,43,58,48,6,56,11,48,56], optimization-based [45,1,2,21,54,17,13,14], and hybrid methods [65,8,49]. Our method is *agnostic* to different GAN inversion approaches for initializing the latent code, e.g., our experiments explore using PTI [49] for in-domain editing and Restyle encoder [6] for out-of-domain editing.

Semantic image editing in latent space. Semantic image manipulation and editing allow us to change the content and style of an image. It can be grouped into In-Domain editing and Out-of-Domain editing. Targeting at finding semantic directions in the latent space of a pretrained generator, in-domain editing [51,22,52,60,63,37,44,3,5,59,4,50] manipulates the attributes of the object, but keeps the same style. Out-of-domain [33,23,15], however, aims to change the style of the image. These techniques usually perform well on a single image but fail to maintain temporal consistency if applied to a video.

Semantic video editing. Recent and concurrent work [62,57,7] explore *video editing* with a pre-trained StyleGAN. The methods in [62,57] apply per-frame editing and show coherent editing without using any temporal information. However, these methods support only in-domain editing. For *localized editing* (e.g., adding eyeglasses), we find that the method in [62] produces inconsistency and fails to preserve identity. The work [7] applies temporal smoothing on the *inverted latent vectors* in StyleGAN3 [27]. Our approach, in contrast, directly minimizes the temporal photometric inconsistency at the *synthesized frames*.

Video editing and temporal consistency. Temporal consistency is one critical criterion in video editing. Existing methods achieve temporal consistency often by enforcing the output videos to satisfy the constraints imposed by 2D optical flow [12,20]. Alternatively, several methods first estimate an unwarped 2D texture map (either explicitly [46] or implicitly [30]) and then perform editing. The editing can then be propagated to the original video via the estimated UV mapping. Several *blind* methods enhance the temporal consistency as a *post-processing* step [9,34,36]. However, they typically have difficulty in handling videos with significant appearance changes. Our work shares similar ideas with these methods to enforce temporal consistency, using the optical flow fields estimated from the initial edited video. Instead of directly optimizing the *pixel values*, our core idea is to leverage the pretrained generator, update the latent code and generator to achieve temporal consistent *and* photorealistic results.

3 Method

3.1 Overview

GAN Inversion. Given an input video $V_{input} = \{I_1, \cdots, I_T\}$ of T frames, our goal is to semantically edit all the video frames while preserving the temporal coherence of the edited video. To edit the input video V_{input}, we first align its frames by using a facial alignment method [19]. Then we use existing GAN inversion techniques (e.g., [49,6]) to invert the frames back to the latent code such that the inverted frame $I_t^{inv} = G(W_t^{inv}; \theta^{inv})$ is similar to the input frame: $I_t^{inv} \approx I_t$. With the inverted frames, we can edit the inverted video $V_{inv} = \{I_1^{inv}, I_2^{inv}, \cdots, I_T^{inv}\}$ by *independently* editing its frames I_t^{inv}. We denote this frame-by-frame editing approach as "direct editing".

In-domain and out-of-domain GAN-based editing. Commonly used *image-based* editing techniques via a GAN include (1) in-domain and (2) out-of-domain editing. We refer to an *in-domain editing* [51,22,52,60] as the editing that only manipulates the latent code, given a *fixed* pretrained generator. That is, the generator parameters θ^{inv} remain frozen ($\theta^{inv} = \theta^{edit}$), and only the latent code W_t^{edit} is updated. The in-domain editing usually changes semantic attributes

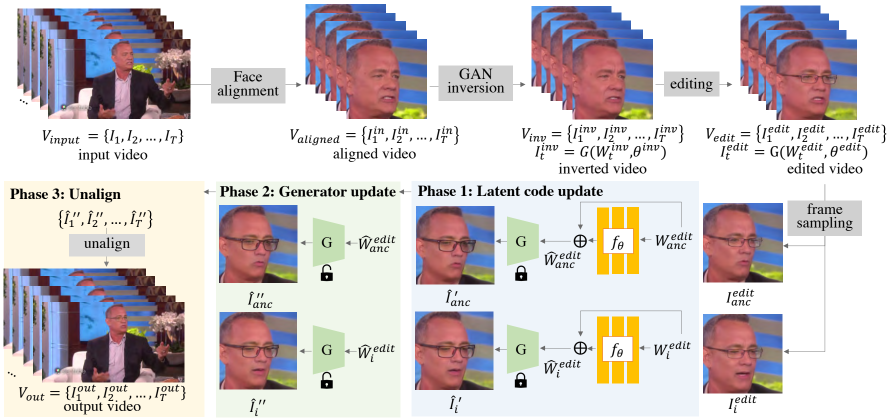
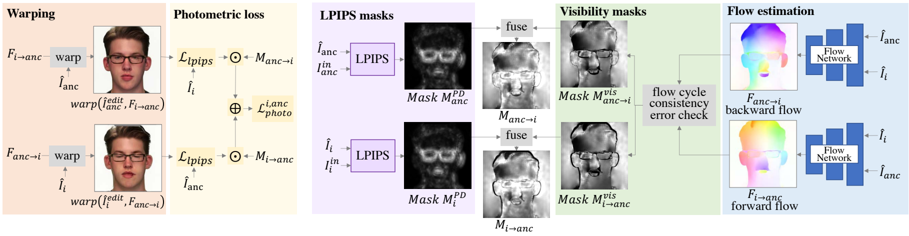

Face alignment
GAN inversion
editing
$V_{input} = \{I_1, I_2, \ldots, I_T\}$
input video
$V_{aligned} = \{I_1^{in}, I_2^{in}, \ldots, I_T^{in}\}$
aligned video
$V_{inv} = \{I_1^{inv}, I_2^{inv}, \ldots, I_T^{inv}\}$
$I_t^{inv} = G(W_t^{inv}, \theta^{inv})$
inverted video
$V_{edit} = \{I_1^{edit}, I_2^{edit}, \ldots, I_T^{edit}\}$
$I_t^{edit} = G(W_t^{edit}, \theta^{edit})$
edited video
Phase 3: Unalign
$\{\hat{I}_1'', \hat{I}_2'', \ldots, \hat{I}_T''\}$
unalign
$V_{out} = \{I_1^{out}, I_2^{out}, \ldots, I_T^{out}\}$
output video
Phase 2: Generator update
G
$\hat{W}_{anc}^{edit}$
$\hat{I}_{anc}''$
$\hat{W}_i^{edit}$
$\hat{I}_i''$
Phase 1: Latent code update
G
f_θ
W_{anc}^{edit}
$\hat{W}_{anc}^{edit}$
$\hat{I}_{anc}'$
W_i^{edit}
$\hat{W}_i^{edit}$
$\hat{I}_i'$
frame sampling
I_{anc}^{edit}
I_i^{edit}

Fig. 3. Video editing with flow-based temporal consistency. Given an input video of T frames V_{input}, we first spatially align the video frames using an off-the-shelf face landmark detector. We then use existing GAN inversion techniques [49,6] to obtain the inverted frames $\{I_1^{inv}, I_2^{inv}, \cdots, I_T^{inv}\}$ and their corresponding latent code in the $\mathcal{W}^+$-space of StyleGAN $\{W_1^{inv}, W_2^{inv}, \cdots, W_T^{inv}\}$. We independently perform semantic editing on these inverted frames to obtain $\{I_1^{edit}, I_2^{edit}, \cdots, I_T^{edit}\}$ and their corresponding latent code $\{W_1^{edit}, W_2^{edit}, \cdots, W_T^{edit}\}$. To achieve temporal consistency, we choose an anchor frame I_{anc}^{edit} as the reference frame, and each time sample another frame I_i^{edit} from the edited video. To generate a temporally consistent edited video, we first refine the latent codes of the directly edited video W_{anc}^{edit} and $\{W_i^{edit}\}_{i \neq anc}$ to $\hat{W}_{anc}^{edit}$ and $\{\hat{W}_i^{edit}\}_{i \neq anc}$ by optimizing an MLP f_θ (phase 1). These refined latent codes result in the temporally consistent frames $\hat{I}_{anc}'$ and $\hat{I}_i'$. To further improve the temporal consistency, we keep the refined latent codes $\hat{W}_{anc}^{edit}$ and $\hat{W}_i^{edit}$ and only update the generator parameters (phase 2). This will generate $\hat{I}_{anc}''$ and $\hat{I}_i''$ with improved temporal consistency. After our two phase optimization, we finally unalign the frames to generate our final edited video V_{out} (phase 3).

such as color, age, or facial expressions. On the other hand, out-of-domain editing may involve updating the pretrained generator to produce an entirely new style (as shown in [15]). Here, the latent code remains the same $W_t^{edit} = W_t^{inv}$ and only the generator θ^{edit} changes.

Direct editing on a video. When applying both types of editing techniques to a video independently for each frame, we obtain an edited video $V_{edit} = \{I_1^{edit}, I_2^{edit}, \cdots, I_T^{edit}\}$. For each directly edited frame I_t^{edit}, there is a corresponding latent code W_t^{edit} such that $I_t^{edit} = G(W_t^{edit}; \theta^{edit})$. Due to the per-frame, independent process, the edited video V_{edit} often suffers from temporal inconsistency. Moreover, due to the poor disentanglement of this per-frame editing, not only will the edited attributes differ among frames, but other existing facial attributes also change (see the change in mouth in Fig. 5). Our



(a) $\mathcal{L}_{photo}$ computation (b) Flow and mask computation

Fig. 4. Photometric loss for temporal consistency. Given a frame pair $\hat{I}_i$ and $\hat{I}_{anc}$ (either from phase 1 or phase 2), we compute the forward and backward flows $F_{i\to anc}$ and $F_{anc\to i}$ using RAFT [53]. We then use these two flow fields to compute the visibility masks by performing a forward-backward and backward-forward flow consistency error check. For in-domain editing, we also use LPIPS to obtain a semantic mask that highlights the difference between the aligned input frames I^{in}_i and I^{in}_{anc} and our edited frames $\hat{I}_i$ and $\hat{I}_{anc}$. We then fuse both the LPIPS semantic masks and the visibility masks to get our final masks $M_{anc\to i}$ and $M_{i\to anc}$. To compute the photometric loss (Eqn. 1), we use the flows to warp the directly edited frames and utilize the fuzed masks as shown in (a).

goal is to ensure that the edited attributes remain temporally consistent while preserving the other details from the input video.

Overview of our approach. To achieve this goal, we propose a two-phase optimization approach: phase 1 updates the *latent code* via an MLP and phase 2 updates the *generator*. In both phases, we optimize the temporal photometric loss across frames. With the finetuned latent code and generator, we unalign the edited frames to produce an edited video. Figure 3 outlines our workflow. Below, we describe the details and the losses of our approach.

3.2 Flow-based temporal consistency

We present a flow-based approach to explicitly encourage temporal consistency in the edited video V_{edit}.

Frame sampling. As we cannot fit an entire video into the GPU memory, we choose to perform our optimization from a *pair of frames* at a time. We choose to use an anchor frame I^{edit}_{anc} as one of the pair, which we set as the middle frame of the video. This is inspired by recent video representation work [47], where a video is represented by a key frame and a flow network. At each iteration, we sample a latent code W^{edit}_i, corresponding to the frame I^{edit}_i and optimize the pair of frames $\{I^{edit}_{anc}, I^{edit}_i\}$. We perform our optimization in two phases (Section 3.3). In phase 1, we generate temporally consistent pairs $\{\hat{I}^{'}_{anc}, \hat{I}^{'}_i\}_{i\neq anc}$ as a result. In phase 2, we further improve the temporal consistency, recover other affected attributes brought by the per-frame editing due to the poor disentanglement, and generate the pairs $\{\hat{I}^{''}_{anc}, \hat{I}^{''}_i\}_{i\neq anc}$.

Flow estimation and warping. We use RAFT [53] to compute the forward and backward flows $F_{i\to anc}$ and $F_{anc\to i}$ of the pair $\{\hat{I}_{anc}, \hat{I}_i\}$. This pair is either the output of phase 1 $\{\hat{I}'_{anc}, \hat{I}'_i\}$ or phase 2 $\{\hat{I}''_{anc}, \hat{I}''_i\}$. We then use these two flows to warp the pair of frames $\{\hat{I}_{anc}, \hat{I}_i\}$.

Visibility masks. To highlight the *non-occluded* regions, we compute the visibility masks $M^{vis}_{anc\rightarrow i}$ and $M^{vis}_{i\rightarrow anc} \in [0, 1]$. This mask shows lower weights for occluded pixels and higher weights for the non-occluded pixels (Figure 4). To compute the visibility masks, we first compute forward-backward and backward-forward flow consistency error maps $\epsilon_{anc\rightarrow i}$ and $\epsilon_{i\rightarrow anc}$ and compute the error map by $\epsilon_{i\rightarrow anc}(p) = ||p - F_{anc\rightarrow i}(p + F_{anc\rightarrow j}(p))||_2$, where p is a pixel in the flow field. These resultant error maps are mapped to $[0, 1]$ using an exponential function such that $M^{vis}_{anc\rightarrow i} = \exp(-10\epsilon_{anc\rightarrow i})$ and $M^{vis}_{i\rightarrow anc} = \exp(-10\epsilon_{i\rightarrow anc})$.

Perceptual difference mask. For in-domain editing, because the introduced editing is temporally inconsistent, we observe that the visibility masks do *not* emphasize those edited parts (e.g., eyeglasses). To highlight those edited parts, we compute the soft semantic perceptual difference masks M^{PD}_{anc} and M^{PD}_i between the pair of frames and their corresponding aligned input frames using LPIPS [64] (Figure 4). Due to the significant appearance differences, we cannot use these semantic perceptual difference masks for out-of-domain editing.

Fused masks. For in-domain editing, we fuse the visibility masks and the semantic perceptual difference masks such that $M_{anc\rightarrow i} = (M^{vis}_{anc\rightarrow i} \oplus M^{PD}_i)$ and $M_{i\rightarrow anc} = (M^{vis}_{i\rightarrow anc} \oplus M^{PD}_{anc})$. The masks will also be clamped to [0, 1]. This fusion is shown in Figure 4. On the other hand, for out-of-domain editing, $M_{anc\rightarrow i} = M^{vis}_{anc\rightarrow i}$ and $M_{i\rightarrow anc} = M^{vis}_{i\rightarrow anc}$.

Bi-directional photometric loss. We use the warped frames and the final computed masks to compute the bi-directional photometric loss to achieve a temporally consistent video. This loss measures the difference between the two frames to calculate the deviation in the non-occluded parts.

$$\begin{aligned}\mathcal{L}_{photo} = \sum_{\hat{I}_i, \hat{I}_{anc} \in P} & M_{i\rightarrow anc} \mathcal{L}_{LPIPS}(\hat{I}_{anc}, warp(\hat{I}_i, F_{anc\rightarrow i})) \\ & + M_{anc\rightarrow i} \mathcal{L}_{LPIPS}(\hat{I}_i, warp(\hat{I}_{anc}, F_{i\rightarrow anc}))\,, \end{aligned} \tag{1}$$

where $\hat{I}_t$ is either the output of phase 1 $\hat{I}'_t$ or phase 2 $\hat{I}''_t$. Intuitively, this bi-directional photometric loss ensures colors along the valid (forward-backward or backward-forward consistent) vectors across frames are as similar as possible.

3.3 Two-phase optimization strategy

We split our optimization into two phases. In the first phase, we refine the latent codes $\{W^{edit}_t\}$ by only optimizing an MLP f_θ. While in the second phase, we only update the generator weights θ^{edit}.

Motivation. We use a two-phase optimization approach for in-domain editing because we observe that only refining the latent codes (phase 1) sometimes introduces undesired changes to *other* facial attributes. We show an example in Fig. 5. When we only update the latent codes, we achieve temporal consistency of the introduced glasses; however, the mouth expression of the person changes.

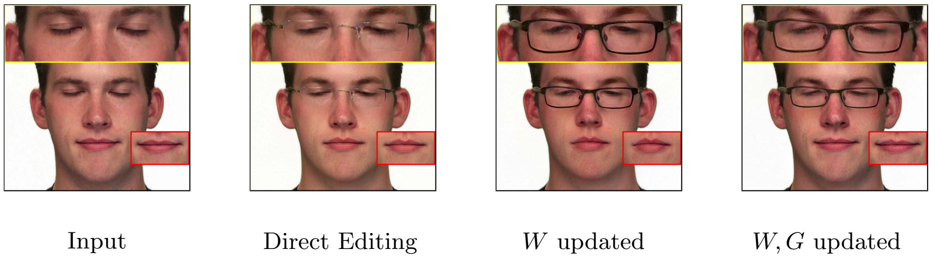


Fig. 5. Motivation for two-phase optimization. Updating latent code W brings in the eyeglasses, and tuning G with the perceptual difference mask recovers the expression in the input.

To address this in the case of in-domain editing, we update the generator weights (phase 2) using the perceptual difference mask to enforce the pixels outside the mask to be the same as the input. This will maintain the facial expression of the aligned input frame. The primary source of inconsistency for out-of-domain editing is the global inconsistency (e.g., background). Hence, updating the generator (phase 2) introduces this desired global change.

Phase 1: Latent code update. In this phase, we update the latent code W_t^{edit} using a Multi-layer Perceptron (MLP) $f_\theta = (w; \theta_f)$ implicitly. We use the same architecture as StyleCLIP mapper [44]. We use this MLP to predict a residual for the latent codes and update the parameters of the MLP instead of directly optimizing the latent codes explicitly, such that:

$$\hat{W}_t^{edit} = W_t^{edit} + \alpha f_\theta(W_t^{edit}; \theta_f)\,, \tag{2}$$

then for a pair of directly edited frames $\{I_{anc}^{edit}, I_i^{edit}\}$, we can get the updated frames $\hat{I}_i^{'} = G(\hat{W}_i^{edit})$, $\hat{I}_{anc}^{'} = G(\hat{W}_{anc}^{edit})$.

Our goal is to minimize:

$$\underset{\theta_f}{\operatorname{argmin}}\,\mathcal{L}_I = \underset{\theta_f}{\operatorname{argmin}} \sum_{t \neq anc} \mathcal{L}_{photo} + \lambda_{rf}\mathcal{L}_{rf} + \lambda_\epsilon \mathcal{L}_\epsilon\,, \tag{3}$$

where $\mathcal{L}_{photo}$ is the photometric loss, and

$$\mathcal{L}_{rf} = ||f_\theta(W_t^{edit}; \theta_f)||_1 + ||f_\theta(W_{anc}^{edit}; \theta_f)||_1 \tag{4}$$

is a regularization term to make sure we do not deviate too much from W_t^{edit}. We set $\lambda_{rf} = 0.1$ for the experiments. $\mathcal{L}_\epsilon = ||\epsilon_{anc \to i}||_1 + ||\epsilon_{i \to anc}||_1$ is the norm of error maps, and we set $\lambda_\epsilon = 10$.

The reason we use an MLP to update the latent code *implicitly* is that we observe that *explicitly* optimizing the latent codes results in an unstable optimization when using a large learning rate. However, the running time becomes too long when using a small learning rate. To address this, we introduce an MLP to predict the residual and update the latent codes *implicitly*. This leads

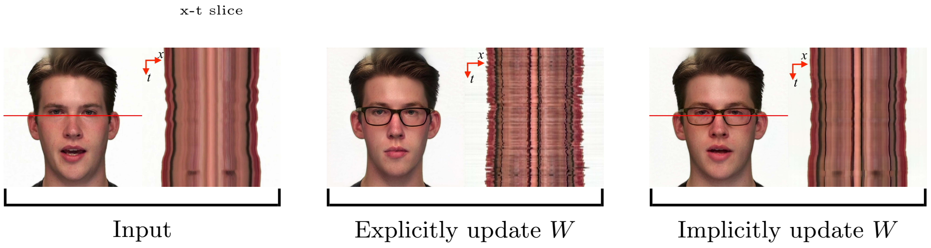


Fig. 6. x-t slices between updating latent codes explicitly and implicitly with an MLP. We visualize the optimized frames and an x-t slice at $y = 500$. Explicitly updating latent code W gives us an unstable x-t scanline, while updating W implicitly with an MLP gives a smooth scanline.

to a more stable optimization. We show an example of x-t scanline in Fig. 6 to demonstrate the effectiveness of introducing the MLP.

Phase 2: Generator update. For in-domain editing, in this phase, we use the updated latent codes $\{\hat{W}_t^{edit}\}_{t=1}^{T}$ from phase 1, and our goal is to finetune the generator only to minimize:

$$\hat{\theta}^{edit} = \underset{\hat{\theta}^{edit}}{\operatorname{argmin}}\, \mathcal{L}_{II} = \underset{\hat{\theta}^{edit}}{\operatorname{argmin}} \sum_{t\neq anc} \mathcal{L}_{photo} + \lambda_{\epsilon}\mathcal{L}_{\epsilon} + \lambda_r\mathcal{L}_r + \lambda_M\mathcal{L}_M \,, \tag{5}$$

$$\mathcal{L}_M = (1 - M_i^{PD})\mathcal{L}_{LPIPS}(\hat{I}_i^{''}, I_i^{in}) + (1 - M_{anc}^{PD})\mathcal{L}_{LPIPS}(\hat{I}_{anc}^{''}, I_{anc}^{in}) \,. \tag{6}$$

M_i^{PD} is the perceptual difference mask computed between $\hat{I}_i^{''} = G(\hat{W}_t^{edit}; \hat{\theta}^{edit})$ and aligned input I_i^{in}, and $\mathcal{L}_{LPIPS}(\cdot,\cdot)$ is the LPIPS distance loss [64]. We initialize $\hat{\theta}^{edit}$ as θ^{edit}. The LPIPS term also plays a role to maintain the sharpness of the edited frames. This is because the consistency can be achieved by pushing all the frames to become blurry.

Here, $\mathcal{L}_r$ is the regularization loss for the generator and λ_r is the strength of regularization. We introduce this loss to help prevent the generator G from losing its latent space editability as we do not wish to *ruin* its pretrained latent space. Therefore, similar to [49], we use this *local regularization* to preserve the editing ability of our generator. More specifically, we first obtain a latent code W_r by linearly interpolating between the current latent code $\hat{W}_t^{edit}$ and a randomly sampled code W_z with an interpolation parameter α_{interp}: $W_r = \hat{W}_t^{edit} + \alpha_{interp}\frac{W_z - \hat{W}_t^{edit}}{||W_z - \hat{W}_t^{edit}||_2}$. This gives us a new latent code in a local region around $\hat{W}_t^{edit}$. To ensure that we do not lose the editing capability of the original generator, we add a penalty on the distance between the generated image from the new generator and the old one such that:

$$\mathcal{L}_r = \mathcal{L}_{LPIPS}(x_r, \hat{x}_r) + \lambda_{\ell_2}^{r}\mathcal{L}_{\ell_2}(x_r, \hat{x}_r) \,, \tag{7}$$

where $x_r = G(W_r; \theta^{edit}), \hat{x}_r = G(W_r; \hat{\theta}^{edit})$, $\lambda_{\ell_2}^{r}$ is the weight for ℓ_2 loss. This regularization can alleviate the side effects from updating G within a local area.

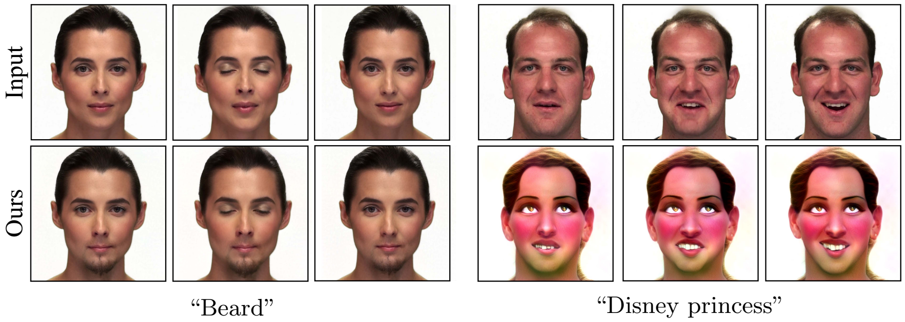


Fig. 7. Visual results on RAVDESS dataset [39]. We show both in-domain ("beard") and out-of-domain ("Disney princess") editing results. Our results maintain consistent changes with time preserving the temporal coherence.

This is desirable since for a video, the latent codes for the same identity tend to gather locally.

For out-of-domain editing, unlike in-domain editing, we cannot rely on the perceptual difference mask, so the optimization goal reduces to:

$$\hat{\theta}^{edit} = \underset{\hat{\theta}^{edit}}{\operatorname{argmin}} \mathcal{L}_{II} = \underset{\hat{\theta}^{edit}}{\operatorname{argmin}} \sum_{t \neq anc} \mathcal{L}_{photo} + \lambda_r \mathcal{L}_r + \lambda_\epsilon \mathcal{L}_\epsilon \,. \tag{8}$$

To compensate for the regularization effect of the perceptual difference mask, we freeze the last eight layers of the synthesis network in G to avoid blurry results. As all the computations, including the GAN generator, flow estimation network, spatial warping, and photometric losses, are *differentiable*, we can backpropagate the errors all the way back. After phase 1 and 2, we will have $\{\hat{W}_t^{edit}\}_{t=1}^{T}$ and $G(\cdot; \hat{\theta}^{edit})$ as a result.

3.4 Phase 3: Unalign

After our two-phase optimization, we perform *stitch tuning* approach [57] as post-processing to put the aligned frames back to the original video to generate our final edited video. Note that this is only feasible for the in-domain editing because the out-of-domain editing has a global appearance compared to the input video.

4 Experimental Results

4.1 Experimental setup

Implementation details. We use StyleGAN-ADA [26] as our pre-trained generator. We experiment with in-domain and out-of-domain editing techniques to validate our approach for different GAN inversion methods. Specifically, for in-domain editing, we use the PTI inversion [49] (based on e4e [56]) and StyleCLIP

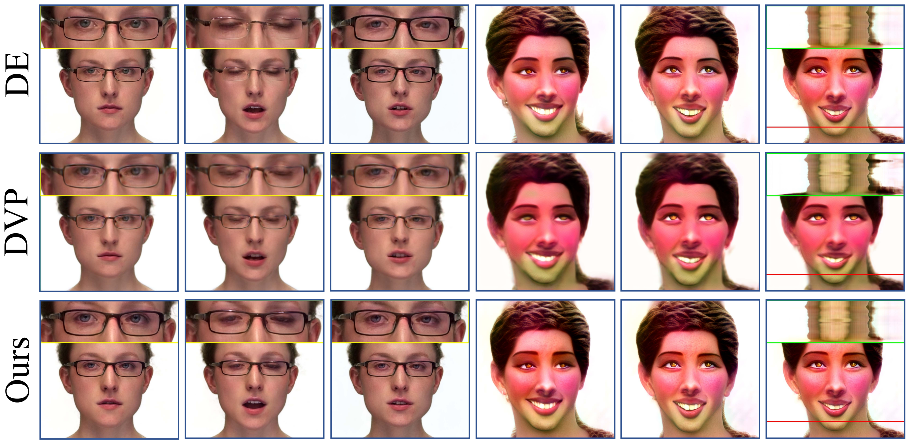


Fig. 8. Visual comparison with DVP [36]. DVP achieves temporal consistency by severely smoothing the image and hence losing its sharpness. Our method, however, can achieve a balance between consistency and sharpness. In “eyeglasses” example (left), DVP shows a different pair of eyeglasses across the time (zoom-in for better visualization), while ours remain a good consistency for the eyeglasses; in “Disney princess” (right), DVP shows a blurry result with an unstable x-t scanline, while ours is sharper and shows a stable consistency in the scanline.

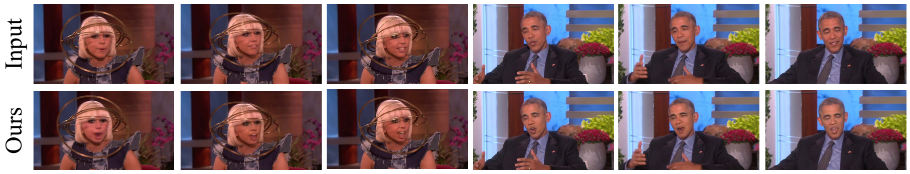


Fig. 9. Results on Internet videos. Results on the Internet videos. We change the first person to “surprised” expression, and change the second person to “angry”.

mapper [44]. For out-of-domain editing, we use the Restyle encoder [6] and the StyleGAN-NADA [15]. We will release the source code and pretrained models. In the following, we show sample results from the video frames. We encourage the readers to view the videos in the supplementary material for video results.

Datasets. We conduct our metric evaluation using 20 videos randomly sampled from RAVDESS dataset [39]. We conduct 5 types of in-domain editing for each video and 5 types of out-of-domain editing. To further demonstrate the capabilities of our method to handle *real* videos, we also apply our approach to Internet videos and show the visual results.

Metrics. We aim to evaluate the method using two main aspects: 1) temporal consistency and 2) perceptual similarity with the semantically edited frames. To

Table 1. Out-of-domain editing comparison.

	$E_{warp}\downarrow$		LPIPS↓	
Direct editing	0.0098		0.0000	
Editing categories	DVP [36]	Ours	DVP	Ours
Sketch	0.0036	0.0085	0.2404	0.1314
Pixar	0.0031	0.0025	0.1074	0.1178
Disney Princess	0.0040	0.0078	0.2062	0.1204
Elf	0.0042	0.0108	0.2289	0.1310
Zombie	0.0040	0.0085	0.2033	0.1370
Average perfomance	0.0038	0.0076	0.1972	0.1275

evaluate temporal consistency, we measure the *Warping Error* E_{warp}:

$$E_{warp}(I_t, I_{t+1}) = \frac{1}{\sum_{i=1}^{N} M_t(p_i)} \cdot \sum_{i=1}^{N} M_t(p_i)||I_t(p_i) - \hat{I}_{t+1}(p_i)||_2^2\,, \qquad (9)$$

where $\hat{I}_{t+1} = warp(I_{t+1}, F_{t\to t+1})$, N is the number of pixels, p_i is the i-th pixel, M_t is a binary non-occlusion mask, which shows non-occluded pixels, we compute it using the forward-backward consistency error the threshold in [35,38].

We also measure the LPIPS perceptual similarity score [64] (with AlexNet [32]) between the directly edited video $V^{edit} = \{I_1^{edit}, I_2^{edit}, \cdots, I_T^{edit}\}$ and the output of our phase 2 $\{\hat{I}_1^{''}, \hat{I}_2^{''}, \cdots, \hat{I}_T^{''}\}$ by measuring the averaged perceptual similarity between the corresponding frames. The purpose of these two metrics is to evaluate whether the method can achieve a balance between *temporal consistency* and *fidelity degradation*. This is an inherent trade-off. Preserving all the details of per-frame editing inevitably leads to temporal flickering artifacts. Focusing only on temporal consistency may easily lead to blurry videos. Our goal is that the final output video is visually similar to the directly (per-frame) edited video.

4.2 Out-of-domain results

Setup. We first invert the videos frame by frame using the Restyle encoder [6] (psp-based [48]). We then directly apply five different out-of-domain editing effects produced by StyleGAN-NADA [15]. We perform our two-phase optimization approach on the directly edited video using Adam optimizer [31]. For phase 1, we set the learning rate to $\alpha_I = 0.005$, and update the latent codes for 5 epochs. In Eqn. 2, we set $\alpha = 0.04$ for all the editing directions. For phase 2, we set the learning rate to $\alpha_{II} = 8 \times 10^{-4}$, and finetune G for 5 epochs. We set the regularization weight λ_r to 200.

Evaluation. Table 1 shows that our method decreases the temporal error of the directly edited video. The primary sources of inconsistency in out-of-domain editing can be seen in the flickering background and the details of the hair.

Table 2. In-domain editing comparison.

		$E_{warp}\downarrow$		LPIPS↓	
Direct editing		0.0076		0.0000	
Editing categories	LT [62]	DVP [36]	Ours	DVP	Ours
angry	-	0.0033	0.0032	0.2452	0.1100
beard	0.0064	0.0038	0.0030	0.2444	0.1033
eyeglasses	0.0066	0.0039	0.0034	0.1226	0.1097
Depp	-	0.0037	0.0031	0.2452	0.2024
surprised	-	0.0035	0.0028	0.1415	0.1012
Average perfomance	0.0065	0.0036	0.0031	0.1760	0.1253

We show our visual results in Figure 7. Our method preserves the temporal consistency and maintains the sharpness of the input video.

4.3 In-domain editing results

Setup. We first invert the videos frame by frame by using the PTI method [49]. We then directly apply five different semantic editing directions discovered by StyleCLIP mapper [44]. Next, we perform our two-phase optimization approach on the directly edited video using Adam optimizer [31]. For phase 1, we set the learning rate $\alpha_I = 0.05$, and update f_θ for 10 epochs. In Eqn. 2, we set $\alpha = 0.12$ for the "eyeglasses", and $\alpha = 0.04$ for the rest of the semantic directions. For phase 2, we set the learning rate of G to $\alpha_{II} = 0.0001$, and finetune G for 5 epochs. We set the regularization weight λ_r to 200.

Evaluation. Table 2 shows that our approach improves the temporal consistency over the directly edited video baseline by a large margin. When dealing with in-domain editing, the primary source of inconsistency is the details of the newly added attributes, e.g., glasses or beard and some background flickering. We show sample visual results in Figure 7, where the introduced changes are consistent among the different frames.

4.4 Two-phase optimization strategy ablation study

We demonstrate the effect of our two-phase optimization strategy of updating the latent codes first and following that with finetuning the generator G. We compare our two-phase approach to: (1) No optimization (i.e., direct editing), (2) update latent code only (phase 1), and (3) finetune generator G only. We show the results in Table 3. When we only update generator G, we can achieve a low warping error E_{warp}. However, this is not desirable since finetuning G pushes the video to be consistent globally without modifying the local details. Therefore, the output video is different from the directly edited video (i.e., high LPIPS distance). Thus, we follow our two-phase optimization of a) updating the latent codes via an MLP f_θ (to improve local consistency), b) finetuning the generator G (to modify the global effect).

Table 3. Two-stage optimization strategy ablation study.

Optimization stage		In-domain editing		Out-of-domain editing	
Update W_t^{edit}	Update G	E_{warp} ↓	LPIPS↓	E_{warp} ↓	LPIPS↓
-	-	0.0076	0.0000	0.0098	0.0000
✓	-	0.0064	0.2108	0.0094	0.1428
-	✓	**0.0027**	0.2463	**0.0057**	0.1375
✓	✓	0.0031	**0.1253**	0.0076	**0.1275**

4.5 Comparison with Latent Transformer

We compare our method with Latent Transformer (LT) [62]. We show a quantitative comparison with two overlapped editing types, "beard" and "eyeglasses" in Table 2. LT edits video by updating the projected latent code *independently* for each frame without using temporal constraints. Our method, in contrast, uses flow-based loss to improve the temporal consistency, and our second phase uses a perceptual difference mask as a regularization to preserve the facial details other than the edited parts. As a result, our method can improve temporal consistency and preserve personal identity.

4.6 Comparison with Deep Video Prior (DVP)

We compare our method with DVP [36], a state-of-the-art blind video consistency approach, using their default setting. We show the in-domain editing comparison in Table 2 and the out-of-domain editing comparison in Table 1. For warping error E_{warp}, our method achieves improved results for in-domain editing and comparable results for out-of-domain editing. However, in terms of LPIPS distance, our visual results are more similar to the directly edited video for both in-domain and out-of-domain editing. We show visual comparison in Figure 8. DVP can achieve temporally consistent results (i.e., low E_{warp}). However, this is at the cost of losing local details in the "eyeglasses" example or excessively smoothing the results to get a blurry video as in the "Disney Princess" example.

5 Conclusions

We have presented a novel method for video-based semantic editing by leveraging image-based GAN inversion and editing. Our approach starts from direct per-frame editing, and we refine the editing results by a flow-based method to minimize the bi-directional photometric loss. Our core approach is two-phase, by adjusting the latent codes via an MLP and tuning G to achieve temporal consistency. We show that our method can achieve temporal consistency and preserve its similarity to the direct editing results. Finally, our model-agnostic method is applicable to different GAN inversion and manipulation techniques.

Potential negative impacts. Malicious use of our technique may lead to video manipulation of public figures for spreading misinformation.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: ICCV (2019) 2, 3
2. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan++: How to edit the embedded images? In: CVPR (2020) 2, 3
3. Abdal, R., Zhu, P., Mitra, N.J., Wonka, P.: Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. ACM Transactions on Graphics (TOG) **40**(3), 1–21 (2021) 4
4. Afifi, M., Brubaker, M.A., Brown, M.S.: Histogan: Controlling colors of gan-generated and real images via color histograms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021) 4
5. Alaluf, Y., Patashnik, O., Cohen-Or, D.: Only a matter of style: Age transformation using a style-based regression model. arXiv preprint arXiv:2102.02754 (2021) 4
6. Alaluf, Y., Patashnik, O., Cohen-Or, D.: Restyle: A residual-based stylegan encoder via iterative refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2021) 2, 3, 4, 5, 11, 12
7. Alaluf, Y., Patashnik, O., Wu, Z., Zamir, A., Shechtman, E., Lischinski, D., Cohen-Or, D.: Third time's the charm? image and video editing with stylegan3. arXiv preprint arXiv:2201.13433 (2022) 3, 4
8. Bau, D., Strobelt, H., Peebles, W., Zhou, B., Zhu, J.Y., Torralba, A., et al.: Semantic photo manipulation with a generative image prior. arXiv preprint arXiv:2005.07727 (2020) 3
9. Bonneel, N., Tompkin, J., Sunkavalli, K., Sun, D., Paris, S., Pfister, H.: Blind video temporal consistency. ACM TOG **34**(6), 1–9 (2015) 4
10. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis (2019) 2, 3
11. Chai, L., Wulff, J., Isola, P.: Using latent space regression to analyze and leverage compositionality in gans. In: International Conference on Learning Representations (2021) 3
12. Chen, D., Liao, J., Yuan, L., Yu, N., Hua, G.: Coherent online video style transfer. In: ICCV (2017) 4
13. Collins, E., Bala, R., Price, B., Susstrunk, S.: Editing in style: Uncovering the local semantics of gans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5771–5780 (2020) 3
14. Daras, G., Odena, A., Zhang, H., Dimakis, A.G.: Your local gan: Designing two dimensional local attention mechanisms for generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14531–14539 (2020) 3
15. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators (2021) 2, 4, 5, 11, 12
16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014) 2
17. Gu, J., Shen, Y., Zhou, B.: Image processing using multi-code gan prior. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3012–3021 (2020) 3
18. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans (2017) 3

19. Guo, J., Zhu, X., Yang, Y., Yang, F., Lei, Z., Li, S.Z.: Towards fast, accurate and stable 3d dense face alignment. In: ECCV (2020) 4
20. Huang, J.B., Kang, S.B., Ahuja, N., Kopf, J.: Temporally coherent completion of dynamic video. ACM TOG **35**(6), 1–11 (2016) 4
21. Huh, M., Zhang, R., Zhu, J.Y., Paris, S., Hertzmann, A.: Transforming and projecting images to class-conditional generative networks. In: ECCV (2020) 2, 3
22. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: Ganspace: Discovering interpretable gan controls. In: Proc. NeurIPS (2020) 2, 4
23. Jang, W., Ju, G., Jung, Y., Yang, J., Tong, X., Lee, S.: Stylecarigan: caricature generation via stylegan feature map modulation. ACM Transactions on Graphics (TOG) **40**(4), 1–16 (2021) 4
24. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: International Conference on Learning Representations (2018) 3
25. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. arXiv preprint arXiv:2006.06676 (2020) 3
26. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. In: Proc. NeurIPS (2020) 10
27. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Proc. NeurIPS (2021) 3, 4
28. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019) 2, 3
29. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR (2020) 2, 3
30. Kasten, Y., Ofri, D., Wang, O., Dekel, T.: Layered neural atlases for consistent video editing. ACM TOG (2021) 4
31. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 12, 13
32. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012) 12
33. Kwong, S., Huang, J., Liao, J.: Unsupervised image-to-image translation via pretrained stylegan2 network. IEEE Transactions on Multimedia (2021) 4
34. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: ECCV (2018) 4
35. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: European Conference on Computer Vision (2018) 12
36. Lei, C., Xing, Y., Chen, Q.: Blind video temporal consistency via deep video prior. In: Advances in Neural Information Processing Systems (2020) 4, 11, 12, 13, 14
37. Li, B., Cai, S., Liu, W., Zhang, P., Hua, M., He, Q., Yi, Z.: Dystyle: Dynamic neural network for multi-attribute-conditioned style editing. arXiv preprint arXiv:2109.10737 (2021) 4
38. Liu, Y.L., Lai, W.S., Yang, M.H., Chuang, Y.Y., Huang, J.B.: Learning to see through obstructions. In: CVPR (2020) 12
39. Livingstone, S.R., Russo, F.A.: The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. PloS one **13**(5), e0196391 (2018) 10, 11

40. Luo, J., Xu, Y., Tang, C., Lv, J.: Learning inverse mapping by autoencoder based generative adversarial nets. In: International Conference on Neural Information Processing. pp. 207–216. Springer (2017) 3
41. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2017) 3
42. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks (2018) 3
43. Nitzan, Y., Bermano, A., Li, Y., Cohen-Or, D.: Face identity disentanglement via latent space mapping. ACM Transactions on Graphics (TOG) **39**, 1 – 14 (2020) 3
44. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2085–2094 (October 2021) 1, 2, 4, 8, 11, 13
45. Raj, A., Li, Y., Bresler, Y.: Gan-based projector for faster recovery with convergence guarantees in linear inverse problems. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5602–5611 (2019) 3
46. Rav-Acha, A., Kohli, P., Rother, C., Fitzgibbon, A.: Unwrap mosaics: A new representation for video editing. ACM TOG (2008) 4
47. Rho, D., Cho, J., Ko, J.H., Park, E.: Neural residual flow fields for efficient video representations. arXiv preprint arXiv:2201.04329 (2022) 6
48. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2021) 3, 12
49. Roich, D., Mokady, R., Bermano, A.H., Cohen-Or, D.: Pivotal tuning for latent-based editing of real images. arXiv preprint arXiv:2106.05744 (2021) 2, 3, 4, 5, 9, 10, 13
50. Saha, R., Duke, B., Shkurti, F., Taylor, G.W., Aarabi, P.: Loho: Latent optimization of hairstyles via orthogonalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1984–1993 (2021) 4
51. Shen, Y., Yang, C., Tang, X., Zhou, B.: Interfacegan: Interpreting the disentangled face representation learned by gans. TPAMI (2020) 2, 4
52. Shen, Y., Zhou, B.: Closed-form factorization of latent semantics in gans. In: CVPR (2021) 2, 4
53. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. arXiv preprint arXiv:2003.12039 (2020) 6
54. Tewari, A., Elgharib, M., Bernard, F., Seidel, H.P., Pérez, P., Zollhöfer, M., Theobalt, C., et al.: Pie: Portrait image embedding for semantic control. arXiv preprint arXiv:2009.09485 (2020) 3
55. Tewari, A., Elgharib, M., Bharaj, G., Bernard, F., Seidel, H.P., Pérez, P., Zollhofer, M., Theobalt, C.: Stylerig: Rigging stylegan for 3d control over portrait images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6142–6151 (2020) 3
56. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. ACM Transactions on Graphics (TOG) **40**(4), 1–14 (2021) 3, 10
57. Tzaban, R., Mokady, R., Gal, R., Bermano, A.H., Cohen-Or, D.: Stitch it in time: Gan-based facial editing of real videos. arXiv preprint arXiv:2201.08361 (2022) 3, 4, 10

58. Viazovetskyi, Y., Ivashkin, V., Kashin, E.: Stylegan2 distillation for feed-forward image manipulation. In: European Conference on Computer Vision. pp. 170–186. Springer (2020) 3
59. Wu, Y., Yang, Y.L., Xiao, Q., Jin, X.: Coarse-to-fine: facial structure editing of portrait images via latent space classifications. ACM Transactions on Graphics (TOG) **40**(4), 1–13 (2021) 4
60. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: Disentangled controls for stylegan image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12863–12872 (June 2021) 4
61. Xia, W., Zhang, Y., Yang, Y., Xue, J.H., Zhou, B., Yang, M.H.: Gan inversion: A survey. arXiv preprint arXiv:2101.05278 (2021) 3
62. Yao, X., Newson, A., Gousseau, Y., Hellier, P.: A latent transformer for disentangled face editing in images and videos. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 13789–13798 (2021) 2, 4, 13, 14
63. Yüksel, O.K., Simsar, E., Er, E.G., Yanardag, P.: Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. arXiv preprint arXiv:2104.00820 (2021) 4
64. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018) 7, 9, 12
65. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain gan inversion for real image editing. In: Proceedings of European Conference on Computer Vision (ECCV) (2020) 2, 3
66. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: ECCV (2016) 2, 3