RIGID: Recurrent GAN Inversion and Editing of Real Face Videos

Yangyang Xu[1], Shengfeng He[2], Kwan-Yee K. Wong[1], and Ping Luo[1,3*]
[1]Deapartment of Computer Science, The University of Hong Kong
[2]School of Computing and Information Systems, Singapore Management University
[3]Shanghai AI Laboratory

(a) Original (b) IGCI / 120,000s (c) Latent-T. / 49.2s (d) STIT / 1,600s (e) TCSVE / 3,400s (f) RIGID / 54.5s

Figure 1: Comparisons of video inversion and editing with existing methods. Number in each column denotes the average editing time over 100 frames. Our RIGID achieves temporal coherent inversion and editing performances with much less time cost. s. Please refer to the arXiv version to watch this figure as a video clip.

Abstract

*GAN inversion is indispensable for applying the powerful editability of GAN to real images. However, existing methods invert video frames individually often leading to undesired inconsistent results over time. In this paper, we propose a unified recurrent framework, named **R**ecurrent v**I**deo **G**AN **I**nversion and e**D**iting (RIGID), to explicitly and simultaneously enforce temporally coherent GAN inversion and facial editing of real videos. Our approach models the temporal relations between current and previous frames from three aspects. To enable a faithful real video reconstruction, we first maximize the inversion fidelity and consistency by learning a temporal compensated latent code. Second, we observe incoherent noises lie in the high-frequency domain that can be disentangled from the latent space. Third, to remove the inconsistency after attribute manipulation, we propose an in-between frame composition constraint such that the arbitrary frame must be a direct composite of its neighboring frames. Our unified framework learns the inherent coherence between input frames in an end-to-end manner, and therefore it is agnostic to a specific attribute and can be applied to arbitrary editing of the same video without re-training. Extensive experiments demonstrate that RIGID outperforms state-of-the-art methods qualitatively and quantitatively in both inversion and editing tasks. The deliverables can be found in https://cnnlstm.github.io/RIGID.*

1. Introduction

Generative adversarial networks (GANs) have demonstrated powerful generative ability in synthesizing high-quality faces from a latent code [9, 11, 12, 47]. It is evidenced that the latent space of a well-trained GAN is semantically organized, and shifting the latent code along with a specific direction results in the manipulation of a corresponding attribute [23, 24, 5, 19, 45, 32]. Hence, many works migrate this power to real face processing by inverting a real face image to a latent code [49, 16, 39, 48, 40, 38]. Although

*Corresponding author: pingluo@hku.hk.

this two-combo strategy becomes a standard for editing high-resolution images, applying it to real videos has less been explored. A naive inversion and editing for each frame can undoubtedly produce incoherence in the resulted video.

Different from processing images, maintaining temporal coherence is the core issue for video editing [18, 28, 26]. Specifically, both the GAN inversion and attribute manipulation may introduce discontinuity across frames. IGCI [39] proposes the first attempt to invert consecutive images simultaneously. They leverage the continuity of the inputs to optimize both the reconstruction fidelity and editability of the outputs, but they fail to consider the temporal correlation between results (see flickering in Fig. 1b). Recent work STIT [30] implicitly recovers the original temporal correlations by the faithful inversion of each frame. It fine-tunes an individual generator for every input video such that the generator can capture all the reconstruction details, and TCSVE [37] extends this idea by proposing a temporal consistency loss that applies on the edited videos. Although they works well for most cases, they are video- and attribute-specific (needs to retrain the model for a new video or a new target attribute), and thus suffers from the expensive training cost and poor generalization ability.

In this paper, we aim to design a unified approach that learns the temporal correlations between successive frames for both inversion and editing, and it can be generalized to other target attributes without re-training. To this end, we propose a **R**ecurrent v**I**deo **G**AN **I**nversion and e**D**iting (RIGID) framework, which evolves and enables the image-based StyleGAN [11, 12] generator to output temporally coherent frames. The coherence is realized in both inversion and editing tasks. Given the current and previous frames, we formulate the inversion as the combination of an image-based inverted code and a temporal compensated code, while the latter amends the code with inter-frame similarity for an accurate and consistent inversion. On the other hand, we observe that the main sources of temporal incoherence, like "flickering", belong to high-frequency artifacts. This motivates us to disentangle the main video content from high-frequency artifacts in the latent space, and thus the "incoherence" can be shared with all the other frames. To build the temporal correlations after attribute manipulation, we propose a self-supervised "*in-between frame composition constraint*" that applies to consecutive edited frames. It enforces any intermediate frame that can be composed by the warping results of their neighbors, which guarantees the smoothness of generated videos. RIGID is trained on the video episodes with several tailored losses. During the inference, it inverts video frames sequentially and therefore can handle videos with arbitrary lengths and support live stream editing. More importantly, once our model is trained, it is attribute-agnostic that can be reused for arbitrary attribute manipulations without re-training. As shown in Fig. 1f, RIGID achieves temporal coherent inversion and editing with far less inference time (compared to those scene- and attribute-specific methods like STIT). Extensive experiments demonstrate the superiority over state-of-the-art methods in terms of quantitative and qualitative evaluations.

In summary, our contributions are three-fold:

- We propose a recurrent video GAN inversion framework that unifies video inversion and editing. It learns the temporal correlation of generated videos explicitly.
- We model temporal coherence from both inversion and editing ends. For inversion, we discover the temporal compensated code and disentangle high-frequency artifacts in the latent space. For editing, we present a novel "*in-between frame composition constraint*" to confine a continuous video transformation.
- We achieve attribute-agnostic editing that can vary editing attributes on the fly, avoiding expensive re-training of the model. Extensive experiments demonstrate the effectiveness of our method over state-of-the-arts.

2. Related Works

GAN Inversion. GAN inversion aims at inverting real images into the latent space of pre-trained generators for reconstruct and edit the real images [1, 39, 21, 33, 35, 29, 27]. Early works optimize the latent code directly for a specific image with expensive computational cost [4, 1]. Another class trains a general encoder that maps the real image to the latent code directly [21, 29, 33]. Particularly, pSp [21] proposes an encoder with pyramid architecture that inverts the real images into the $\mathcal{W}+$ latent space of StyleGAN. Based on the same architecture, e4e [29] analyzes the trade-offs between reconstruction and editability in StyleGAN's inversion. HFGI [33] introduces the distortion map for improving the fidelity reconstitution. Existing works mainly focus on the image-based GAN inversion, inverting the real videos into GANs has not been well studied.

StyleGAN-based Video Generation and Editing. State-of-the-art video GANs still cannot generate high quality results as their image counterparts [28, 6, 34, 30]. Opposite to design a video generator directly, many works use the pre-trained image generator (*e.g.*, StyleGAN) for synthesizing high-quality videos [18, 28, 43, 26]. MoCoGAN-HD [28] decomposes the motion and content in videos and generates motion trajectory in StyleGAN's latent space. StyleHEAT [43] shares the same decomposition idea, it exploits the flow field as the motion descriptor for talking face generation. Recently, StyleGAN-V [26] injects the continuous motion representations into the StyleGAN. In this paper, we aim at inverting a real video into a pre-trained StyleGAN for temporal coherence reconstruction and editing. Few works

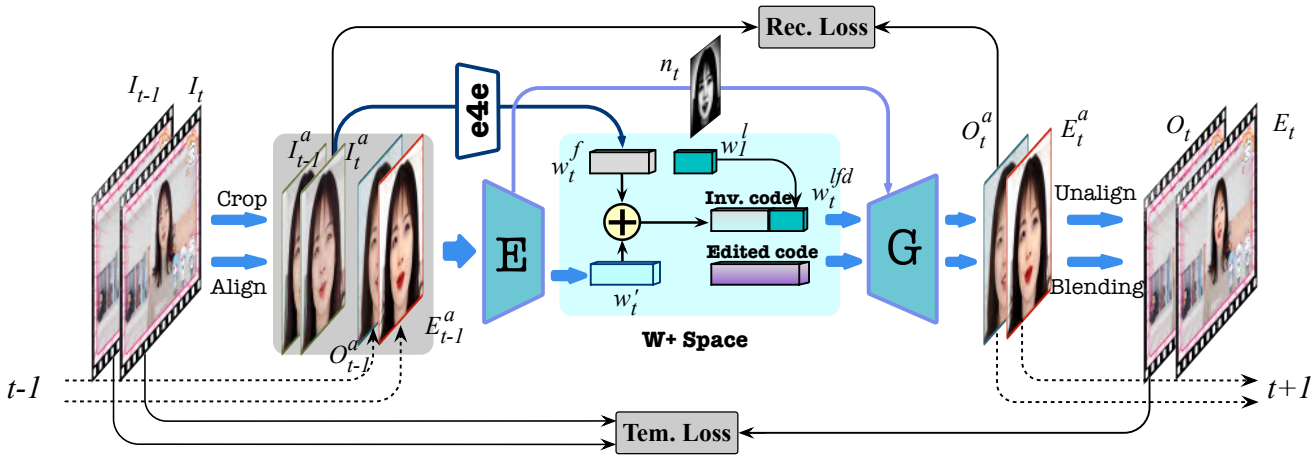

Figure 2: Overview of our RIGID. We first align the faces on two neighbor frames (I_{t-1} and I_t). Then we concatenate them with the inverted face O_{t-1} and edited face E_{t-1} in the previous step as the inputs of the recurrent encoder. The encoder learns the temporal compensated code w'_t and spatial noise map n_t, as a complement to the initial latent code w_t (acquired by the "e4e" encoder). In addition, we share the latter part of latent codes across all frames using w^l_1 for eliminating the high-frequency temporal flickering. Both the final inverted latent code w^{lfd}_t and the edited one are fed to the generator, producing the inverted and edited faces O^a_t and E^a_t. The generated faces are unaligned and blended with the original frames. Dotted lines denote recurrent inputs from the last time step or outputs to the next step. In addition, a novel *in-between frame composition constraint* is proposed that learns the temporal correlation during editing (details can be found in Fig. 3).

concentrate on this task. Latent-Transformer [42] presents a pipeline for facial video editing by inverting each frame individually, resulting in the temporal inconsistency in the edited videos. IGCI [39] introduces the consecutive frames into GAN inversion for improving the reconstruction quality and editability. STIT [30] edits the facial videos using StyleGAN2 [12] by fine-tuning the generator. TCSVE [37] also optimizes generators with a temporal consistency loss that applies on the edited videos. Above two optimization-based work need to re-optimize the generator for a new video or edit, which is time-consuming. In this paper, we propose a recurrent framework that learns the temporal correlations both in inverted and edited videos. Once it is properly trained, it supports various semantic editing methods with low computational costs.

3. Approach

3.1. Formulation

Given a real face video $\{I_t|t = 1, ...T\}$, our goal is to invert it to the latent space of pre-trained StyleGAN G to obtain a set of latent codes, and output the inverted video $\{O_t|t = 1, ...T\}$, where O_t is obtained by feeding the latent code to G. Meanwhile, we can also obtain an edited video $\{E_t|t = 1, ...T\}$ by manipulating the latent codes. The key is that both inverted and edited video should be temporal coherent.

To address that, we propose a recurrent video GAN inversion and editing framework to explicitly and simultaneously enforce temporally coherent GAN inversion and facial editing of real videos. For video inversion, as the original video of inborn temporal coherence, the best way to maintain inversion consistency is a faithful reconstruction. In addition, we propose a "*latent frequency disentanglement*" strategy for eliminating the high-frequency temporal flickering in the latent space. We also propose an *in-between frame composition constraint* that builds the temporal correlations of edited video. The overview of our RIGID can be seen in Fig. 2.

3.2. Coherent Video Inversion

3.2.1 Temporal Compensated Inversion

Given a real video, we can deliver its temporal correlation to the generated video by a faithful reconstruction. Before inversion, face alignment is necessary since StyleGAN cannot handle the entire frame. After the alignment on each frame, we can obtain a set of aligned faces $\{I^a_t|t = 1, ...T\}$. We first encode them to the $\mathcal{W}+$ space using image-based inversion method (e4e [29] in this paper), and obtain the initial latent codes $\{w_t|t = 1, ...T\}, w_t \in \mathcal{W}+$. However, directly using the initial latent codes cannot reconstruct original faces accurately due to the missing of temporal context. We use a recurrent encoder that learns a temporal compensated code as a complement to the initialized one. Moreover, recovering a high-fidelity face solely in StyleGAN's $\mathcal{W}+$ latent space is too difficult due to the lack of spatial information [13, 33]. Inspired by [2], the encoder also learns a

noise map in StyleGAN's $\mathcal{N}$ space for injecting the spatial information.

Our goal is to unify inversion and editing in the same framework. As a result, both the inverted and edited faces from previous and current time steps are fed to the encoder to generate the temporal compensated code and the noise map. Here a ConvLSTM layer [25] is integrated into the encoder for modeling the spatial-temporal correlations. Specifically, at time step t, we concatenate the aligned faces I^a_{t-1}, I^a_t, the last inverted face O^a_{t-1}, and last edited face E^a_{t-1} as inputs. It outputs a temporal compensated code w'_t and spatial noise map n_t, we add w'_t with the initialized code w_t. Then both added code and noise map n_t are sent to the generator for inversion, that is:

$$\{w'_t, n_t\} = E(\texttt{cat}(I^a_{t-1}, I^a_t, O^a_{t-1}, E^a_{t-1})), \quad (1)$$

$$O^a_t = G(w_t + w'_t, n_t), \quad (2)$$

where E denotes the recurrent encoder and $\texttt{cat}(\cdot,\cdot,\cdot,\cdot)$ denotes the concatenation operator.

3.2.2 Latent Frequency Disentanglement

The fidelity inversion delivers the temporal relations from the original video to the inverted one. However, frames are inverted one by one and may lead to subtle inconsistency, *i.e.*, the unique high-frequency information in a single frame will be accumulated in a video and leads to temporal flickering. We notice that high-frequency temporal flickering mainly exists in the appearance of an image. Recent works evidenced that $\mathcal{W}+$ space is highly disentangled, and the appearance of the image is synthesized at the higher layer of StyleGAN, which is controlled by the latter part of the w latent code [12, 21]. Hence we propose a "*latent frequency disentanglement*" strategy that eliminates the temporal flickering by sharing the latter part of w latent code across all the frames.

Specifically, we first decompose a latent code into the former part and the latter part, then we reuse the latter part (corresponding to high layers of StyleGAN) of the first frame in all the following frames that unify the high frequency information, that is:

$$w^{lfd}_t = \texttt{cat}((w_t + w'_t)^f, w^l_1), \quad (3)$$

where w^{lfd}_t is the latent code after latent frequency disentanglement, $(w_t + w'_t)^f$ is the former part of latent code $w_t + w'_t$, and w^l_1 denotes the latter part of the first latent code. Now, we can get the final latent codes $\{c_t = \{w^{lfd}_t, n_t\} | t = 1, ...T\}$ for reconstructing a temporal coherent video. That is, we replace Eq. 2 with following equation:

$$O^a_t = G(w^{lfd}_t, n_t). \quad (4)$$

The face can be edited by manipulating the latent code w^{lfd}_t:

$$E^a_t = G((w^{lfd}_t + \overrightarrow{\mathbf{n}}), n_t), \quad (5)$$

where $\overrightarrow{\mathbf{n}}$ denotes the semantic direction acquired by arbitrary semantic editing techniques [23, 24, 5, 42, 41]. It is noticed that the n_t is also determined by edited face E^a_{t-1}, and it can be well cooperated with both inverted and edited code.

3.3. Coherent Video Editing

3.3.1 Post Processing

The generated faces are naturally aligned that lost the temporal coherence. We need to unaligned generated faces and blend them with the original frame I_t. Here we follow [42, 30] that blends the face region only. In particular, we first segment the face region on an original frame by a pre-trained face parsing model [44] to get the inner face mask M_{I_t}, then we blend the inverted face O^a_t with the original frame according to mask M_{I_t}, which can be presented as:

$$O_t = \texttt{B}(\texttt{UA}(O^a_t), I_t, M_{I_t}), \quad (6)$$

where $\texttt{B}$ and $\texttt{UA}$ denote the blending and unalignment respectively.

For the edited face E^a_t, its face region may be modified after editing, and the face mask of the original frame M_{I_t} cannot well fit with the edited faces. Besides, directly using the face mask of the edited face also suffers a similar problem. Hence we use the union of two masks as the blending mask, that is:

$$E_t = \texttt{B}(\texttt{UA}(E^a_t), I_t, M_{E_t} \cup M_{I_t}). \quad (7)$$

3.3.2 *In-between Frame Composition Constraint*

After post processing, we can obtain the inverted and edited videos. Compared with the inverted videos, temporal correlation in an edited video is more difficult to learn, since there is no GT edited videos for supervision. We propose a self-supervised *in-between frame composition constraint* that models the temporal correlation in an edited video.

Generally, for a triplet of consecutive frames $\{E_{t-1}, E_t, E_{t+1}\}$, the intermediate frame E_t can be composed by flow-based warping results of its neighbor frames [8, 20, 36]. Specifically, as shown in Fig. 3, let $f_{t\Rightarrow t-1}$ denotes the optical flow from E_t to E_{t-1} and $f_{t\Rightarrow t+1}$ is the flow from E_t to E_{t+1}, then frames E_{t-1} and E_{t+1} are warped using different flow. And the intermediate frame can be composed using two warped results according to a visibility map, that is:

$$\begin{aligned}\hat{E}_t =& V_{t\Leftarrow t-1} \odot \texttt{W}(E_{t-1}, f_{t\Rightarrow t-1}) + \\ & (1 - V_{t\Leftarrow t-1}) \odot \texttt{W}(E_{t+1}, f_{t\Rightarrow t+1}),\end{aligned} \quad (8)$$

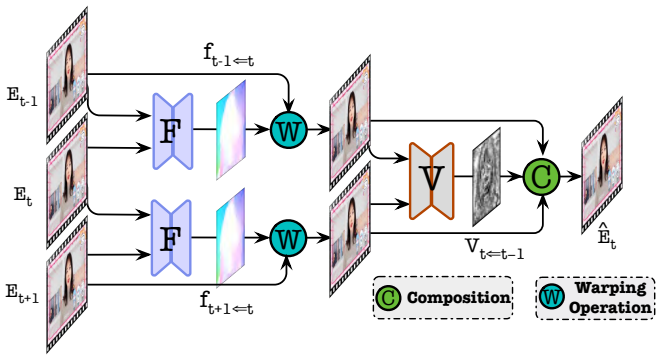

Figure 3: The in-between frame can be composed by flow-based warping results of its neighboring frames according to a visibility map $V_{t\Leftarrow t-1}$.

where $\hat{E}_t$ is the composed intermediate frame, $V_{t\Leftarrow t-1}$ is the visibility map from frame E_t to E_{t-1}, it is a one channel mask with the same resolution with aligned face frames. It denotes whether a pixel remains visible when moving from frame $t-1$ to t (0 is fully occluded and 1 is fully visible). $\mathtt{W}$ is the warping operator, and $\odot$ denotes element-wise multiplication. This equation models the temporal correlations among three consecutive frames.

For the edited videos, both $f_{t\Rightarrow t-1}$ and $f_{t\Rightarrow t+1}$ are available, the remaining challenge is to estimate the visibility mask $V_{t\Leftarrow t-1}$. Since there is no GT for edited videos, we turn to train a visible net V for estimating the mask. Particularly, we composite the in-between frame $\hat{I}_t$ on the real videos using Eq. 8. Then we align the l_1 distance between $\hat{I}_t$ and I_t for training visible net V. After training, we fix the V and adopt it to the edited frames for estimating the visibility mask and composing the in-between frame $\hat{E}_t$. We minimize the distance between $\hat{E}_t$ and E_t to form *in-between frame composition constraint*:

$$\mathcal{L}_{ibfcc} = \sum_{t=2}^{T-1} ||E_t - \hat{E}_t||_1. \tag{9}$$

This constraint enforces the edited video as smooth as real video, which guarantees the temporal coherence effectively. Note that we train the encoder on the video episodes. Particularly, we first collect the outputs $\{E_0, ..., E_T\}$ in an episode via forward propagation, then we apply this constraint to train the encoder in backward propagation (we use differentiable unalignment).

3.4. Loss Functions

Our RIGID is trained under several tailored losses. Besides the *in-between frame composition constraint* $\mathcal{L}_{ifsc}$ applied on the edited videos, it also includes reconstruction and temporal consistency losses on the inverted videos.

Reconstruction Loss. We first introduce the reconstruction loss $\mathcal{L}_{rec}$ on the inverted faces. Following [21, 33], it includes a pixel-wise $\mathcal{L}_2$ loss for minimizing the reconstruction error, and LPIPS loss for a better image quality preservation [46], which can be represented as:

$$\mathcal{L}_2 = \sum_{t=1}^{T} ||I_t^a - O_t^a||_2, \tag{10}$$

$$\mathcal{L}_{lpips} = \sum_{t=1}^{T} ||P(I_t^a) - P(O_t^a)||_2, \tag{11}$$

$$\mathcal{L}_{rec} = \mathcal{L}_2 + \alpha \mathcal{L}_{lpips}, \tag{12}$$

where T is the number of frame in each training episode, $P(\cdot)$ denotes the perceptual feature extractor, and α is the balance weight. Following [21], we set $\alpha = 0.8$.

Temporal Consistency Loss. We also introduce a warping-based temporal consistency loss $\mathcal{L}_{tc}$ for preserving the temporal consistency of inverted videos. In particular, we first calculate the optical flow between two real neighbor frames, then we warp a real frame according to the flow, meanwhile, we also warp the inverted frame by the same flow. Then we minimize the distance on two warped frames to form the temporal consistency loss, that is:

$$\hat{I}_{t-1} = \mathtt{W}(I_{t-1}, f_{t\Rightarrow t-1}), \tag{13}$$

$$\hat{O}_{t-1} = \mathtt{W}(O_{t-1}, f_{t\Rightarrow t-1}), \tag{14}$$

$$\mathcal{L}_{tc} = \sum_{t=2}^{T} ||\hat{I}_{t-1} - \hat{O}_{t-1}||_1, \tag{15}$$

where $f_{t\Rightarrow t-1}$ is the flow from frame I_{t-1} to I_t. This loss enforces the temporal correlation in the videos to be the same as the input video, and improves the temporal smoothness.

Final Loss. We get the final loss function for training the RIGID:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{tc} + \lambda_3 \mathcal{L}_{ibfcc}, \tag{16}$$

where $\{\lambda_i\}$ denote the weight factors for balancing loss terms.

4. Experiments

4.1. Implementation Details

We implement the proposed framework in Pytorch on a PC with an Nvidia GeForce RTX 3090. We chose the StyleGAN2 [12] pre-trained on the FFHQ dataset [11] with the resolution of 256×256 as our generator due to its strong edit ability. The corresponding w latent code has the dimension of 14×512, and we set the early 10×512 as the former part and the rest of 4×512 as the latter part in the latent frequency disentanglement. The framework is optimized by the Adam optimizer [14] with the learning rate of

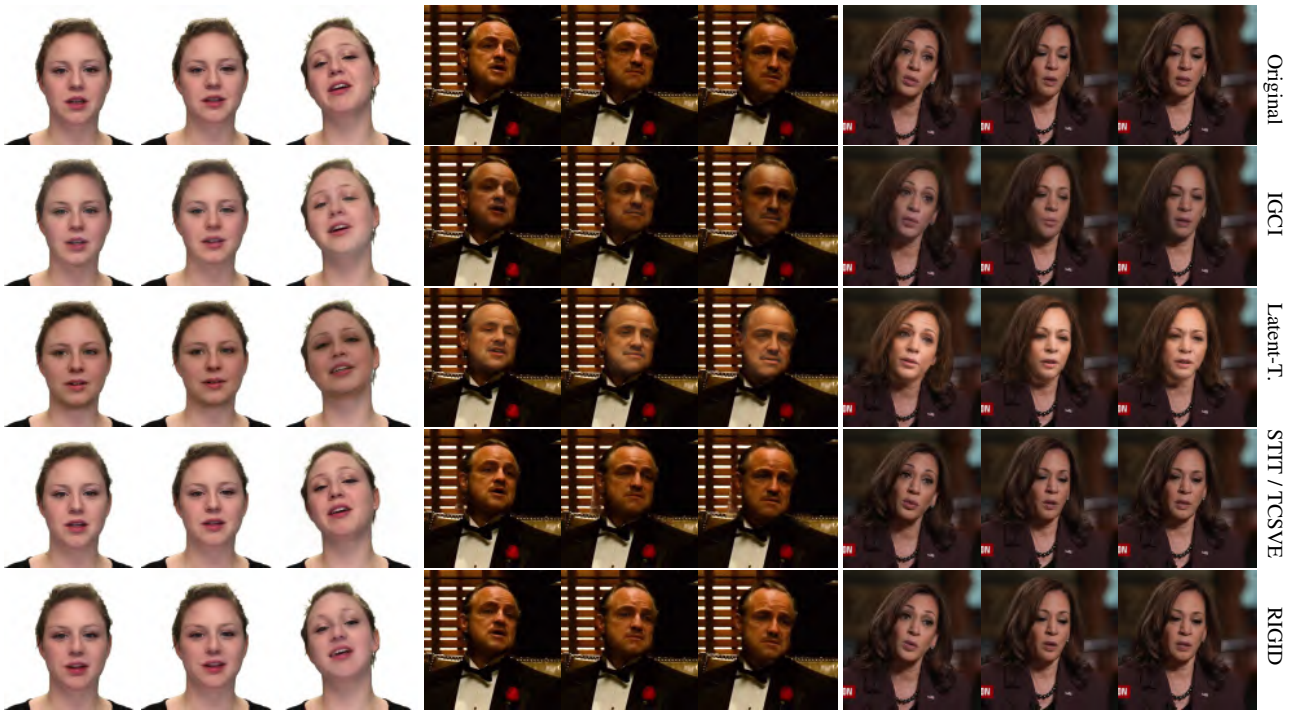

Figure 4: Qualitative comparison on the video inversion task. Our learning-based RIGID can faithfully reconstruct the video frames that is comparable to those expensive optimization-based works.

$1e^{-4}$. We empirically set the balancing weights in Eq. 16 as $\lambda_1 = 1$, $\lambda_2 = 2$, and $\lambda_3 = 5$. Limited by the memory size of GPU, we set $T = 6$ in Eq. 15, which indicates each training episode contains 6 consecutive frames. In addition, we inject the noise map n_t at the resolution of 32×32 to the generator.

The recurrent encoder consists of 7 convolutional (Conv) layers, a ConvLSTM layer, and a fully connected (FC) layer. We take "`Fused Leaky ReLU`" as activation in-between Conv layers. The ConvLSTM layer is integrated between the Conv and FC layers. The visible net has a U-Net structure [22], it takes 5 Conv layers as encoder, and 5 Transposed convolutional layers as a decoder. BatchNorm layer and leakyReLU activation are integrated into between layers. Besides, a Sigmoid function is applied on output U-Net for normalizing the visibility map values. Visible net takes the concatenation of two warped results with 6 channels as input and outputs a visibility map with 1 channel. It is trained using Adam optimizer [14] with the learning rate of $1e^-4$ with 100,000 iterations. Besides, we use pre-trained FlowNet2 [7] for predicting optical flow, and we implement warping operation using bi-linear interpolation.

4.2. Experimental Settings

Datasets. We collect datasets both under control and in the wild environment. We select 72 videos from the controlled RAVDESS dataset [17], which we called RAVDESS-72 Dataset. It contains 9,045 frames and each video contains about 120 frames. We also collect 36 videos from the Internet with the various poses, expressions, and backgrounds. We name them as In-the-Wild-36 Dataset, it contains 7,532 frames in total. We combine two datasets together, use 85 videos for training our RIGID and the rest 23 videos for testing.

Competitors and Evaluation Metrics. We compare RIGID with three works, including IGCI [39], Latent-Transformer [42], STIT [30], and TCSVE [37]. Note that STIT [30] and TCSVE [37] use the same method for inversion. We use several metrics for evaluating different methods. For the video inversion task, we use the pixel-wise Mean Square Error (MSE) and Learned Perceptual Image Patch Similarity (LPIPS) [46] that evaluates the reconstruction quality. For evaluating the temporal stability, we use the flow-based Warp Error (WE) metric. For the edited task, we follow STIT [30] that use Temporally-Local (TL-ID) and Temporally-Global (TG-ID) identity preservation metrics for evaluating the temporal coherence of edited videos. We also use Fréchet Video Distance (FVD) [31] metric on both edited and inverted videos. Inference Times (IT) over 100 frames is also reported. Please refer to supplementary for more details of competitors and evaluation metrics.

Table 1: Video inversion comparisons on two datasets. ↓ denotes the lower the better and the best results are marked in **bold**, and the values of MSE are magnified 100 times.

Methods \ Metrics	RAVDESS-72 Dataset				In-the-Wild-36 Dataset				IT↓(s)
	MSE↓	LPIPS↓	WE↓	FVD↓	MSE↓(×e-2)	LPIPS↓	WE↓	FVD↓	
IGCI [39]	**0.99**	**0.05**	154.21	412.33	**2.01**	0.13	432.98	276.92	1.2×e5
Latent-T. [42]	5.68	0.12	86.356	220.46	5.31	0.22	367.32	165.35	**49.2**
STIT [30]/TCSVE [37]	**0.99**	**0.05**	**83.21**	**171.23**	2.32	**0.11**	293.83	81.03	851.5
RIGID	1.04	**0.05**	84.62	174.55	2.31	0.12	**287.32**	**74.11**	54.5

Figure 5: Qualitative comparison on video editing. RIGID uses the same post processing as IGCI and Latent-Transformer, but the edited faces can be better blended with the original background. Besides, compared with STIT and TCSVE, our RIGID supports shape editing on the face boundary (*e.g.*, "Chubby").

4.3. Evaluation on Video Inversion

Quantitative Evaluation. We first evaluate the fidelity and temporal coherence of inverted videos. Quantitative comparison can be seen in Tab. 1. We can see that both STIT, IGCI, and TCSVE have lower MSE and LPIPIS values, they optimize latent codes or generator specifically according to the video frames, which can reconstruct the target frames faithfully. IGCI presents the worse performance on WE and FVD metrics, since it optimizes the latent code for each image but fails to consider the temporal coherence of consecutive frames, making the inverted latent codes less consistent. In addition, optimization-based methods cost a lot of time during the inference, especially for IGCI, which takes about 20 minutes for a single frame. Latent-Transformer uses a learning-based encoder that accelerates the inference speed, but it processes each frame individually, both the temporal coherence and reconstruction quality cannot be guaranteed. With about 15× faster inference than STIT, RIGID achieves a comparable result on the RAVDESS-72 and In-the-Wild-36 datasets. RIGID not only inverts the frame faithfully but also preserves the original temporal relations across frames.

Table 2: Video editing comparisons on two datasets. ↓ denotes the lower the better and vice versa, the best results are marked in **bold**.

Methods \ Metrics	RAVDESS-72			In-the-Wild-36			IT↓(s)
	TL-ID↑	TG-ID↑	FVD↓	TL-ID↑	TG-ID↑	FVD↓	
IGCI [39]	0.94	0.79	834.33	0.93	0.71	499.22	1.2×e5
Latent-T. [42]	**0.99**	0.93	311.84	**0.99**	0.87	267.42	**49.2**
STIT [30]	**0.99**	**0.97**	212.04	**0.99**	0.91	211.34	1.6×e3
TCSVE [37]	**0.99**	**0.97**	201.32	**0.99**	0.92	198.46	3.4×e3
RIGID	**0.99**	**0.97**	**198.54**	**0.99**	**0.93**	**183.36**	54.5

Thanks to temporal compensated inversion, RIGID guarantees the fidelity of inverted faces.

Qualitative Evaluation. The quantitative comparison of video inversion can be seen in Fig. 4. We can see that two optimization-based works, IGCI and STIT reconstruct the frames well. Latent-Transformer utilizes the pSp encoder for inversion [21], and the results present different skin colors with original frames. Thanks to the temporal compensation inversion, our learning-based RIGID presents the competitive results with optimization-based IGCI and STIT on pixel-wise reconstruction. The video comparison can be seen in Fig. 1. Latent-Transformer cannot provide accurate reconstruction. IGCI inverts each frame faithfully, but it cannot guarantee the temporal coherence across frames, resulting in serious temporal flickering. STIT achieves high-quality inversion on a specific video at the cost of long computational times. In contrast, RIGID builds temporal relations of inverted videos by the temporal compensated inversion, yielding temporal coherent inverted videos with much less time cost.

4.4. Evaluation on Video Editing

Quantitative Evaluation. The quantitative comparison of video editing can be seen in Tab. 2. We can see that IGCI [39] has high FVD values on both two datasets. As discussed above, it cannot produce consistent latent codes, making the edited frames discontinuous. In addition, IGCI has lower values on both TL-ID and TG-ID, which evidences that identity information cannot be preserved locally and globally. Compared with IGCI, Latent-Transformer [42] uses an encoder for producing latent codes, and the edited frames are more consistent. STIT [30] optimizes the generator for each video hence requires many inference times. RIGID achieves a comparable result on the RAVDESS-72 dataset with 30× faster during the inference. As for the In-the-Wild-36 Dataset, RIGID outperforms all competitors with three metrics. This should contribute to our *in-between frame composition constraint*. It enforces the smoothness of edited frames and brings temporal coherence into edited videos.

Qualitative Evaluation. We present the qualitative comparison on video editing in Fig. 1 and Fig. 5. We can see that ICGI presents blurry edited faces with temporal flickering, and Latent-Transformer loses temporal coherence on local details (see bangs in Fig. 1c). In addition, though they use the same post processing as our RIGID, their edited faces cannot be well blended with the original background. Without considering the temporal coherence of edited frames, their edited faces have large structure deformations from the original faces. STIT proposes a "stitching tuning" strategy for the seamless blending, it enforces the edit frames have similar transitions around the face boundary with originals. However, when the target face's boundary is close to the background, this method does not support its shape-related editing (*e.g.*, "Chubby" or "Double Chins"). As shown in the 1_{st} sample in Fig. 5, STIT fails on the "Chubby" editing. TCSVE uses the same strategy, hence it fails on this editing. More video comparisons can be found in the supplementary materials.

4.5. Ablation Study

In this section, we perform an ablation study to evaluate our RIGID on the RAVDESS-72 dataset. We develop five variants with the modification of the modules and the loss functions: 1) w/o TCC, by removing temporal compensated code w'_t. 2) w/o NM, by removing noise map n_t. 3) w/o LFD, by removing the latent frequency disentanglement in the framework. In this variant, frames in the same video have different latter codes w^l_t. 4) w/o L_{ibfcc}, by removing *in-between frame composition constraint*. 5) w/o $\mathcal{L}_{tc}$, by removing the temporal consistent loss $\mathcal{L}_{tc}$.

The quantitative comparisons with various variants can be seen in Tab. 3. Compared with RIGID, variant w/o TCC has a large WE value. Code w'_t introduces the temporal compensation to the image-based latent code, and brings the temporal coherence in inverted videos. Variant w/o NM has large MSE and LPIPS values. Since noise map n_t injects the spatial information to the generator, and improves the inversion accuracy. Both variant w/o LFD and w/o $\mathcal{L}_{tc}$ have large WE numbers. Our latent frequency disentanglement strategy unifies the high-frequency information in a video, and $\mathcal{L}_{tc}$ preserves the temporal consistency from the original to the inverted video. They learn the temporal relations effectively. We observe that variant w/o $\mathcal{L}_{ibfcc}$ has

Table 3: Quantitative comparisons with various variants on video inversion and editing. ↓ denotes the lower the better and vice visa, the best results are marked in **bold**. The values of MSE are magnified 100 times.

Variants \ Metrics	Video Inversion				Video Editing		
	MSE↓	LPIPS↓	WE↓	FVD↓	TL-ID↑	TG-ID↑	FVD↓
w/o TCC	1.31	0.07	106.52	208.74	0.98	0.94	321.98
w/o NM	2.28	0.08	96.52	201.22	0.99	0.92	334.98
w/o LFD	1.10	0.06	103.61	215.96	0.99	0.94	309.04
w/o $\mathcal{L}_{ibfcc}$	**0.93**	0.05	705.22	358.04	0.98	0.90	299.12
w/o $\mathcal{L}_{tc}$	1.08	0.05	122.08	203.54	0.99	0.97	299.34
RIGID	1.04	**0.05**	**84.62**	**174.55**	**0.99**	**0.97**	**198.54**

Figure 6: Qualitative comparison with different variants. Variant w/o LFD and w/o $\mathcal{L}_{ibfcc}$ present noticeable temporal flickering. Please refer to the arXiv version to watch this figure as a video clip.

the largest WE and FVD values on inverted videos, though it is applied to the edited frames. That is because the constraint enforces the smoothness of edited frames, and can be propagated to the latent codes that control the coherence of inverted frames. Meanwhile, variant w/o $\mathcal{L}_{ibfcc}$ has the worse performance on metrics TL-ID and TG-ID. Without the in-between constraint, the framework cannot guarantee the identity similarity across frames. By learning the temporal correlations in inverted and edited videos, the final RIGID achieves the best performance on two tasks.

Qualitative comparison with different variants can be seen in Fig. 6. We can see variant w/o $\mathcal{L}_{ibfcc}$ presents annoying flickering both on the inverted and edited videos, which evidences the effectiveness of *in-between frame composition constraint*. As talked about in the main paper, it builds the temporal relations of video frames, which guarantees the smoothness of generated videos. Variant w/o NM cannot invert the video faithfully. Variant w/o LFD may present temporal flickering. The final RIGID presents temporal coherent inversion and editing videos.

5. Conclusion and Discussions

In this paper, we propose RIGID, a novel recurrent framework that learns the temporal correlations between successive frames for both inversion and editing. RIGID first learn the temporal correlation of inverted videos by a temporal-compensated reconstruction. It takes previous and current faces as input, and produces a temporal compensated code with a spatial noise map. They, together with the image-based latent code, are fed to the StyleGAN generator to reconstruct the video frames. Meanwhile, RIGID disentangles high-frequency artifacts in the latent space for eliminating the temporal flickering. Furthermore, for learning the temporal correlation of edited video, we enforce an edited frame can be composed by the flow-based warping of its neighbor frames. RIGID achieves attribute-agnostic editing with much less time cost. Extensive experiments demonstrate the superiority of RIGID.

Limitations. As cropping and alignment of faces are needed before the inversion, those hair portions outside the cropped region cannot be edited by our method, which leads to unnatural conflicts after blending. It may be addressed by using StyleGAN3 [10] since it can generate unaligned faces. However, as indicated in [3], StyleGAN3 has a worse edit ability than StyleGAN2, which is essential in GAN inversion.

Broader Impact. RIGID can potentially be used for academic research and commercial applications. It provides a pipeline for video-based GAN inversion, and benefits the design of temporal coherent video GANs. Besides, RIGID can create more facial videos in the media industry. On the other hand, although not the purpose of this work, it can potentially be misused for the violation of portrait rights and privacy. The risk can be mitigated by face forgery detection methods, *i.e.*, we can detecte the blending boundary using [15].

Acknowledgement. This paper is partially supported by the National Key R&D Program of China No.2022ZD0161000; the General Research Fund of Hong Kong No.17200622; National Natural Science Foundation of China (No. 61972162); Guangdong Natural Science Funds for Distinguished Young Scholars (No. 2023B1515020097); and Singapore Ministry of Education Academic Research Fund Tier 1 (MSS23C002).

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *CVPR*, pages 4432–4441, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, pages 8296–8305, 2020. 3

[3] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. In *ECCV WorkShop*, pages 204–220, 2023. 9

[4] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE TNNLS*, 30(7):1967–1974, 2018. 2

[5] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *NeurIPS*, 33:9841–9850, 2020. 1, 4

[6] Sangeek Hyun, Jihwan Kim, and Jae-Pil Heo. Self-supervised video gans: Learning for appearance consistency and motion coherency. In *CVPR*, pages 10826–10835, 2021. 2

[7] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, pages 2462–2470, 2017. 6

[8] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, pages 9000–9008, 2018. 4

[9] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 1

[10] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *NeurIPS*, 34, 2021. 9

[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 1, 2, 5

[12] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020. 1, 2, 3, 4, 5

[13] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In *CVPR*, pages 852–861, 2021. 3

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5, 6

[15] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *CVPR*, pages 5001–5010, 2020. 9

[16] Zhansheng Li, Yangyang Xu, Nanxuan Zhao, Yang Zhou, Yongtuo Liu, Dahua Lin, and Shengfeng He. Parsing-conditioned anime translation: A new dataset and method. *ACM TOG*, 42(3):1–14, 2023. 1

[17] Steven R Livingstone and Frank A Russo. 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. 6

[18] Elizaveta Logacheva, Roman Suvorov, Oleg Khomenko, Anton Mashikhin, and Victor Lempitsky. Deeplandscape: Adversarial modeling of landscape videos. In *ECCV*, pages 256–272, 2020. 2

[19] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, pages 2085–2094, 2021. 1

[20] Fitsum A Reda, Deqing Sun, Aysegul Dundar, Mohammad Shoeybi, Guilin Liu, Kevin J Shih, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Unsupervised video interpolation using cycle consistency. In *ICCV*, pages 892–900, 2019. 4

[21] 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 *CVPR*, pages 2287–2296, 2021. 2, 4, 5, 8

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 6

[23] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE TPAMI*, 2020. 1, 4

[24] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, pages 1532–1540, 2021. 1, 4

[25] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *NeurIPS*, 28, 2015. 4

[26] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *CVPR*, 2021. 2

[27] Haorui Song, Yong Du, Tianyi Xiang, Junyu Dong, Jing Qin, and Shengfeng He. Editing out-of-domain gan inversion via differential activations. In *ECCV*, pages 1–17, 2022. 2

[28] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *ICLR*, 2021. 2

[29] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM TOG*, 40(4):1–14, 2021. 2, 3

[30] Rotem Tzaban, Ron Mokady, Rinon Gal, Amit Bermano, and Daniel Cohen-Or. Stitch it in time: Gan-based facial editing of real videos. In *SIGGRAPH Asia*, pages 1–9, 2022. 2, 3, 4, 6, 7, 8

[31] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. In *ICLR WorkShop*, 2019. 6

[32] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *ICML*, pages 9786–9796, 2020. 1

[33] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *CVPR*, 2022. 2, 3, 5

[34] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, volume 31, 2018. 2

[35] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE TPAMI*, 45(3):3121–3138, 2023. 2

[36] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *NeurIPS*, 32, 2019. 4

[37] Yiran Xu, Badour AlBahar, and Jia-Bin Huang. Temporally consistent semantic video editing. In *ECCV*, pages 357–374, 2022. 2, 3, 6, 7, 8

[38] Yangyang Xu, Bailin Deng, Junle Wang, Yanqing Jing, Jia Pan, and Shengfeng He. High-resolution face swapping via latent semantics disentanglement. In *CVPR*, pages 7642–7651, 2022. 1

[39] Yangyang Xu, Yong Du, Wenpeng Xiao, Xuemiao Xu, and Shengfeng He. From continuity to editability: Inverting gans with consecutive images. In *ICCV*, pages 13910–13918, 2021. 1, 2, 3, 6, 7, 8

[40] Yangyang Xu, Zeyang Zhou, and Shengfeng He. Self-supervised matting-specific portrait enhancement and generation. *IEEE TIP*, 31:5332–5342, 2022. 1

[41] Huiting Yang, Liangyu Chai, Qiang Wen, Shuang Zhao, Zixun Sun, and Shengfeng He. Discovering interpretable latent space directions of gans beyond binary attributes. In *CVPR*, pages 12177–12185, 2021. 4

[42] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. In *ICCV*, pages 13789–13798, 2021. 3, 4, 6, 7, 8

[43] Fei Yin, Yong Zhang, Xiaodong Cun, Mingdeng Cao, Yanbo Fan, Xuan Wang, Qingyan Bai, Baoyuan Wu, Jue Wang, and Yujiu Yang. Styleheat: One-shot high-resolution editable talking face generation via pre-trained stylegan. In *ECCV*, pages 85–101, 2022. 2

[44] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, pages 325–341, 2018. 4

[45] Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In *ICCV*, pages 14263–14272, 2021. 1

[46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5, 6

[47] Chenxi Zheng, Bangzhen Liu, Huaidong Zhang, Xuemiao Xu, and Shengfeng He. Where is my spot? few-shot image generation via latent subspace optimization. In *CVPR*, pages 3272–3281, 2023. 1

[48] Zhixuan Zhong, Liangyu Chai, Yang Zhou, Bailin Deng, Jia Pan, and Shengfeng He. Faithful extreme rescaling via generative prior reciprocated invertible representations. In *CVPR*, pages 5708–5717, 2022. 1

[49] Yang Zhou, Yangyang Xu, Yong Du, Qiang Wen, and Shengfeng He. Pro-pulse: Learning progressive encoders of latent semantics in gans for photo upsampling. *IEEE TIP*, 31:1230–1242, 2022. 1