In-N-Out: Face Video Inversion and Editing with Volumetric Decomposition

Yiran Xu [1] Zhixin Shu[2] Cameron Smith[2] Jia-Bin Huang[1] Seoung Wug Oh[2]

[1]University of Maryland, College Park, [2]Adobe Research

https://in-n-out-3d.github.io/

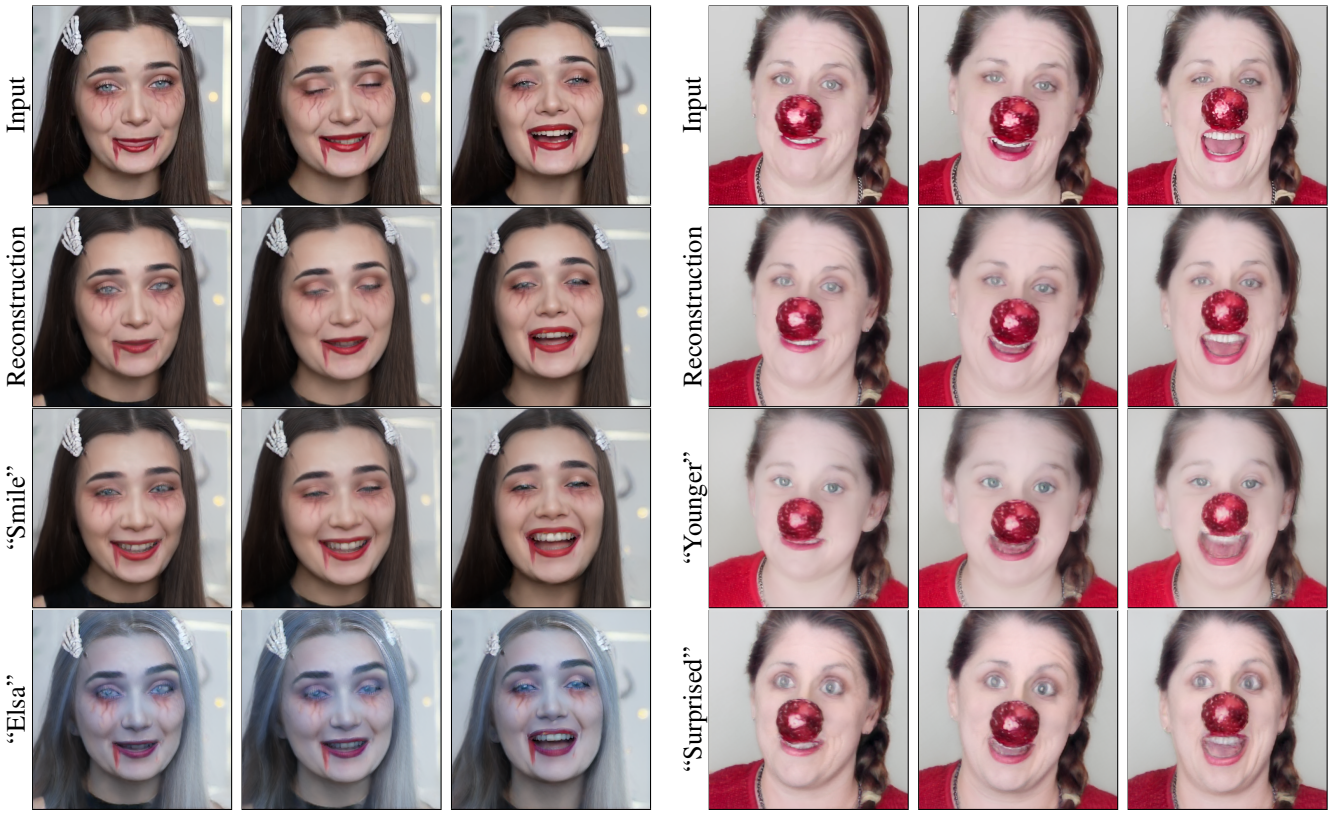


Figure 1. **Semantic editing for *out-of-distribution* data.** We present a method for reconstructing and editing an out-of-distribution video using a pre-trained 3D-aware generative model (EG3D [9]). Our method explicitly model and reconstruct the disoccluders in 3D, allowing faithful reconstruction of the input video while preserving the semantic editing capability. Here we showcase the reconstruction and editing results "Smile", "Younger" [45], "Elsa", "Surprised" [40].

Abstract

3D-aware GANs offer new capabilities for creative content editing, such as view synthesis, while preserving the editing capability of their 2D counterparts. Using GAN inversion, these methods can reconstruct an image or a video by optimizing/predicting a latent code and achieve semantic editing by manipulating the latent code. However, a model pre-trained on a face dataset (e.g., *FFHQ) often has difficulty handling faces with out-of-distribution (OOD) objects,* e.g., *heavy make-up or occlusions. We address this issue by explicitly modeling OOD objects in face videos. Our core idea is to represent the face in a video using two neural radiance fields, one for in-distribution and the other for out-of-distribution data, and compose them together for reconstruction. Such explicit decomposition alleviates the inherent trade-off between reconstruction fidelity and editability. We evaluate our method's reconstruction accuracy and editability on challenging real videos and showcase favorable results against other baselines.*

1. Introduction

GAN-based image/video editing. *GAN inversion* [3, 42, 52, 57, 64] projects a real image onto the latent space of a pre-trained GAN to obtain a latent code (so that the input image can be reconstructed by feeding the latent code into a pre-trained GAN). By changing the latent code, one can achieve many creative semantic editing effects [18, 24, 40, 45] for images. For the video domain, recent methods also achieve faithful and temporally consistent editing [53, 58, 60]. However, these *2D methods* often lack explicit viewpoint control of the generated contents.

3D-aware GANs. With the rapid development of 3D reconstruction and representations [6, 11, 35, 36], high-quality 3D-aware GANs [9, 21, 39] have emerged as a powerful tool for 3D content generation. Equipped with a 3D representation (*e.g.*, neural radiance field [9, 21] or SDF [39]), these methods first render a low-resolution feature map and RGB image, followed by an upsampler to yield a high-resolution image. 3D-aware GANs offer explicit camera pose control and 3D geometry consistency while inheriting the generation capacity and editability of 2D GANs [26–29]. This enables applications such as novel view synthesis and image/video editing [31, 48]. To accurately reconstruct a face's 3D geometry and texture, it is necessary to utilize more frames or viewpoints from a face video.

Core challenges. We aim to develop a *3D-aware video editing* system that can be applied to in-the-wild Internet videos. Given a video, our main challenge is dealing with *out-of-distribution (OOD)* objects, *e.g.*, heavy make-ups or occlusions, because a) GAN is pre-trained only on natural faces, absent of complicated texture or large occlusion, and b) the editability performance drops significantly when we force a pre-trained GAN to model OOD objects. This is commonly known as the *reconstruction-editability trade-off* [52]. Existing GAN inversion methods assume that a *single* corresponding latent code for the input frame can be found in the latent space [47, 57], *i.e.*, they reconstruct the in-distribution natural face and the OOD objects *together*. Since an OOD object cannot be represented by a single latent code, existing methods either cannot reconstruct them faithfully [48] or can reconstruct them (through fine-tuning the generator) but sacrifice the editability [43] (shown in Figure 2).

Our work. In this work, we propose a new perspective to resolve the issue by drawing inspiration from recent composite volume rendering works that compose multiple radiance fields during rendering [19, 33, 55, 59]. Our core idea is to *decompose* the 3D representation of the video with the OOD object into an *in-distribution* part and an *out-of-distribution* part, and compose them together to reconstruct the video in a composite volumetric rendering manner. We use EG3D [9] as our 3D-aware GAN backbone and leverage its tri-plane representation to model this composed rendering pipeline. For the in-distribution component (*i.e.* natural face), we project pixels onto EG3D's $\mathcal{W}^+$ space. For the out-of-distribution part, we use an additional tri-plane to represent it. Later, we combine these two radiance fields together in a composite volumetric rendering manner to reconstruct the input. During the editing stage, we edit the in-distribution part, *i.e.* the latent code w, *without* touching the OOD part. Previous StyleGAN-based editing approaches [40, 45] can be directly applied to the in-distribution part. The advantages of our work are three-fold: a) by composing two parts together, we achieve a higher-fidelity reconstruction, b) by editing only the in-distribution part, we maintain the editability, and c) by leveraging 3D-aware GANs, we can render the input face video from novel viewpoints.

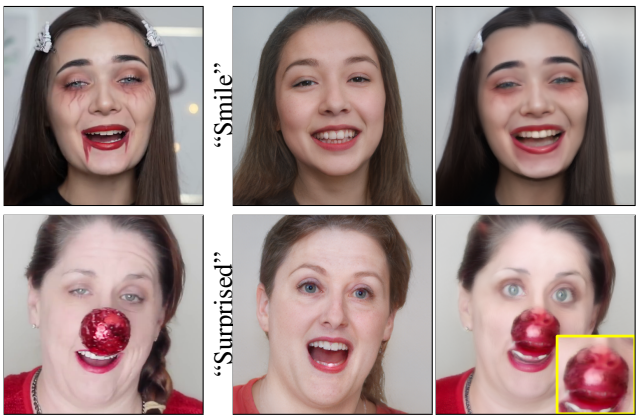


Figure 2. **Motivation.** GAN inversion techniques cannot deal with frames with OOD elements. IDE-3D [48] can produce faithful editing, but fails to preserve the identity of the input face. PTI [43] provides higher reconstruction fieldity, but the edibility suffers.

We evaluate our proposed method on online face videos and show improved reconstruction and editing quality than the previous methods. We demonstrate interesting 3D-aware editing results, such as semantic editing, novel view synthesis, and OOD object removal.

Our contributions. In sum, our contributions are:

- We propose a novel 3D-aware video editing that can manipulate monocular portrait videos with the presence of out-of-distribution objects (e.g., accessories and heavy make-up). See results in Figure 1.
- We incorporate composite volume rendering into 3D-aware GAN inversion.
- Our method reconstructs 3D shapes of faces with OOD objects faithfully and demonstrates novel 3D-aware video editing applications.

2. Related Work

3D-aware GANs. StyleGANs [26–29] have achieved high-quality photorealistic 2D image generation and have been successfully applied to various image editing applications [18, 24, 40, 45]. Significant progresses have also been made to lift 2D image generation to 3D space, using various 3D representations, for both higher quality generation and to enable 3D-aware applications such as view synthesis [9, 10, 16, 20, 21, 37, 39, 44, 46, 48]. These methods usually take two stage pipeline that first renders a raw image (usually also with feature maps) in low resolution and then upsamples the rendered image to high resolution. In our work, we leverage EG3D [9], a state-of-the-art 3D-aware GAN, as our generator architecture. EG3D builds upon a StyleGAN2 [29] backbone architecture and nicely inherits the qualities of a well-behaved latent space that naturally allows effective GAN inversion and editing applications.

GAN inversion and editing. GAN inversion has been widely studied for 2D GANs. These techniques can largely be categorized as (a) encoder-based methods [4, 8, 32, 38, 42, 42, 51, 52, 52, 54] in which a neural network encoder is trained to project an input image to the latent space of the generator; (b) optimization-based methods [1, 2, 13, 14, 22, 25, 41, 50] where the latent code is recovered via optimizing loss functions between the generator output and a target image; and (c) hybrid methods [5, 7, 43, 65] which combine both approaches. Some recent works [31, 48, 62] have also investigated 3D-aware GAN inversion. Unlike previous methods focusing on single-image inversion, we consider multi-shot (or video) inversion. With video input, our approach can better reconstruct 3D shapes. More importantly, we propose a new mechanism to allow high-quality 3D-aware GAN inversion of *out-of-distribution faces* even with significant occlusion. As we demonstrate in the experiments, previous approaches have difficulty handling these challenging cases. With our GAN inversion, we can modify the latent code to perform high-quality semantic image editing [18, 24, 40, 45] or video editing [53, 58, 60].

GAN inversion for out-of-distribution (OOD) data. There have been attempts to invert out-of-distribution data to the GAN's latent space. Early work [1] proposes to project an image onto extended $\mathcal{W}^+$ space to achieve better reconstruction. PTI [43] fine-tunes generator with regularization for lower distortion error. StyleSpace [56] proposes to invert an image using StyleGAN's internal feature maps and tRGB blocks, which shows better reconstruction and disentanglement. Very recently, ChunkyGAN [47] proposes to compose multiple generated images, from multiple latent codes, with a set of segmentation masks to reconstruct an input image. With a similar goal in mind, we propose to leverage the radiance field of EG3D [9] and decompose the volumetric representation into an in-distribution part and an out-of-distribution part. In contrast to ChunkyGAN [47] that models an image as a collection of *2D* segments, we model the OOD and face directly in volumetric 3D representation and merge them with composite rendering.

Composite Neural Radiance Fields. Neural Radiance Fields (NeRFs) [35] have shown impressive 3D reconstruction and view synthesis results. Recently, it has been shown that 3D scenes can be decomposed into different NeRFs. When multiple radiance fields are built, one can compose them together by using a composite rendering manner [19, 33, 55, 59]. Our GAN backbone EG3D [9] use the tri-plane representation to generate 3D object from the latent code. We adopt the idea of composite volume rendering to address the out-of-distribution 3D GAN inversion problem. Specifically, we split the in-distribution and out-of-distribution parts in the tri-plane 3D representation and compose them during volume rendering.

3. Background

3.1. Neural Radiance Fields (NeRFs)

Our goal is to leverage the composability of Neural Radiance Fields [33] (NeRFs) to reconstruct the OOD object and in-distribution face, respectively. A NeRF is an implicit 3D representation with a differentiable and continuous function D (either an MLP or voxel grid features), that takes a 3D position $\mathbf{x} \in \mathbb{R}^3$ and a view direction $d \in \mathbb{R}^3$ as the input, and outputs pointwise RGB color $\mathbf{c} \in \mathbb{R}^3$ and density σ.

$$\mathbf{c}(\mathbf{x}, \mathbf{d}), \sigma(\mathbf{x}) = D(\mathbf{x}, \mathbf{d}) \,. \tag{1}$$

To compute the color of a pixel, a ray $\mathbf{r}(t_k) = \mathbf{o} + t_k\mathbf{d}$ is cast through the camera origin in the direction $\mathbf{d}$. Then, the predicted color is computed by volume rendering [34].

$$\mathbf{C}(\mathbf{r}) = \sum_{k=1}^{K} T(t_k)\alpha(\sigma(t_k)\delta_k)\mathbf{c}(t_k) \,, \tag{2}$$

where $T(t_k) = \exp(-\sum_{k'=1}^{k-1} \sigma(t_{k'})\delta_{k'})$, $\alpha = 1 - \exp(-x)$, and $\delta_k = t_{k+1} - t_k$ is the distance between two 3D points.

3.2. 3D-aware GANs

We choose EG3D [9], which consists of a tri-plane representation and a super-resolution (SR) module, as the 3D-aware GAN backbone.

Neural rendering at low resolution. Given a latent code $z \in \mathbb{R}^{512}$ and camera parameters p, EG3D first generates a corresponding tri-plane $\mathbf{T} \in \mathbb{R}^{256\times256\times32\times3}$. For each pixel, a ray $\mathbf{r}$ is cast, and points are sampled along the ray. Unlike the positional encoding [33, 49] for each point in NeRFs, EG3D projects each point onto $\mathbf{T}$ and retrieves features from three planes via bilinear interpolation. These fea-

tures are then aggregated by summation, and fed into the decoder D (*i.e.* an MLP) to predict the color and density. Volume rendering is then performed to compute the final color for each pixel. To this end, a raw RGB image with a 32-channel feature in a low resolution (*e.g.* 128×128) is generated.

Super resolution. To gain high-resolution outputs, EG3D later uses an SR module that inputs the raw image and the 32-channel feature as the input and yields a high-resolution RGB image (*e.g.* 512×512). We build our approach upon EG3D due to its rendering efficient compared to other alternatives [21, 39].

4. Method

Given an aligned, monocular face video $\mathbf{V} = [\mathbf{I}_1, \cdots, \mathbf{I}_t, \cdots, \mathbf{I}_N]$ with N frames, and binary masks $[\mathbf{M}_1, \cdots, \mathbf{M}_t, \cdots, \mathbf{M}_N]$ of the out-of-distribution (OOD) content for each frame, we aim to reconstruct the video with EG3D inversion and perform face editing. For each video, we label the first frame for $\mathbf{M}_1$, and use an off-the-shelf tracking algorithm [12] to propagate it to obtain other masks $\mathbf{M}$'s.

We show the high-level overview in Figure 3. Each time we sample one single frame $\mathbf{I}_t$ from the video $\mathbf{V}$. We aim to build *two* neural radiance fields (NeRFs) [35], one for *in-distribution* face (Section 4.1), and the other one for *out-of-distribution* (OOD) object (Section 4.2), using tri-plane representations [9]. The OOD object, for example, can be a non-face object with a rigid shape or heavy makeup with a complicated texture. Next, we combine two radiance fields (Section 4.3) to reconstruct the low-resolution frame. Finally, we finetune the super-resolution module of EG3D to get the high-resolution output (Section 4.4). After training the radiance fields, we can edit the face video (Section 4.5).

4.1. In-distribution GAN inversion

EG3D [9] G first takes a latent code $w \in \mathbb{R}^{14\times512}$ as the input to generate a tri-plane representation $\mathbf{T}^I \in \mathbb{R}^{256\times256\times32\times3}$. Then the features are sampled from $\mathbf{T}^I$ and fed into an MLP decoder D^I. The decoder predicts the colors $\mathbf{c}^I$ and densities σ^I for the 3D coordinates. Next, volume rendering [34] is used to generate a low-resolution output (128×128).

Formulation. The in-distribution inversion is similar to the normal 3D GAN inversion [31, 57] except we are inverting a face *video* (or multiple images of the same identity). For each frame, we optimize a latent code w_t and camera parameters $p_t \in \mathbb{R}^{25}$, such that we can reconstruct the input frame $\mathbf{I}_t$. We exclude pixels inside the mask $\mathbf{M}_t$ from the reconstruction loss. .

Following [9, 31], we obtain the camera parameters p_t by using an off-the-shelf pose detector [17]. As the input video $\mathbf{V}$ contains the face of a single identity, we use this multi-view information to recover the face details in 3D. To this end, instead of optimizing each frame's latent code *independently*, we propose to represent each frame's latent code using a canonical latent code w^{temp}, and a residual latent code w_t^{res}, such that

$$w_t = w^{temp} + a w_t^{res}\,, \tag{3}$$

where a is a factor to control the strength of the residual. In practice, we use $a = 0.7$. The benefit of this is that the canonical latent code w^{temp} is shared with all the frames, containing the common representation for the same identity. The residual latent code w_t^{res} acts as a deformation upon the template frame. During the optimization, we back-propagate gradients to w_t^{temp} and w_t^{res}.

Optimization. Our optimization objective for this stage is

$$\begin{aligned} w_t^* = \underset{w_t}{\operatorname{argmin}}\, \mathcal{L}_t^I = \underset{w_t}{\operatorname{argmin}}\, \frac{1}{||\mathbf{m}_t||_1} ||\mathbf{m}_t \odot (\mathbf{x} - \hat{\mathbf{x}})||_2^2 \\ + \mathcal{L}_{\text{mLPIPS}}(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{m}_t) + \lambda_\Delta \mathcal{L}_\Delta(w_t)\,, \end{aligned} \tag{4}$$

where w_t is the latent code at time t, $\mathbf{x}$ is the input frame $\mathbf{I}_t$, $\hat{\mathbf{x}}$ is generation output $G(w_t, p_t)$, $\mathbf{m_t} = 1 - \mathbf{M}_t$, and $\mathcal{L}_{\text{mLPIPS}}(x, \hat{x}, \mathbf{m})$ is the masked version of LPIPS loss [63]. Here, p_t is the camera parameters estimated by [17]. The masked LPIPS only considers features inside the mask $\mathbf{m}$. $\mathcal{L}_\Delta(w) = \sum_{i=1}^{13} ||\Delta_i||_2^2$ is a regularization loss adopted from [52], used to constrain the variation among style vectors in w given a latent code $w = (w_0, w_0 + \Delta_1, ..., w_0 + \Delta_{13}) \in \mathbb{R}^{14\times512}$. By minimizing the variation, we push $w \in \mathcal{W}^+$ to be closer to $\mathcal{W}$ space, which has better editability.

4.2. Out-of-distribution inversion

We use an additional tri-plane $\mathbf{T}^O$ to represent the out-of-distribution objects. However, because the object is not static across different frames, it is challenging to reconstruct it with a static radiance field. Therefore, in addition to $\mathbf{T}^O$, we use a time-varying latent code $\phi_t \in \mathbb{R}^{32}$ for each frame to represent the out-of-distribution object across the temporal domain. Both $\mathbf{T}^O$ and ϕ_t are randomly initialized from a normal distribution.

Formulation. The out-of-distribution decoder D^O takes a tuple $(\mathbf{T}^O(t_k), \phi_t) \in \mathbb{R}^{64}$ as the input, and outputs color $\mathbf{c}^O \in \mathbb{R}^3$, density $\sigma^O \in \mathbb{R}$, and blending weight $b \in [0, 1]$.

$$(\mathbf{c}^O, \sigma^O, b) = D^O(\mathbf{T}^O(t_k), \phi_t; \theta_{D^O})\,, \tag{5}$$

where $\mathbf{T}^O(t_k) \in \mathbb{R}^{32}$ is the aggregated features obtained by projecting 3D coordinate t_k onto each of the three feature planes via bilinear interpolation, then aggregated via summation [9]. The decoder D^O is an MLP with weights of θ_{D^O}. To compute the color of a pixel at time t, we use the

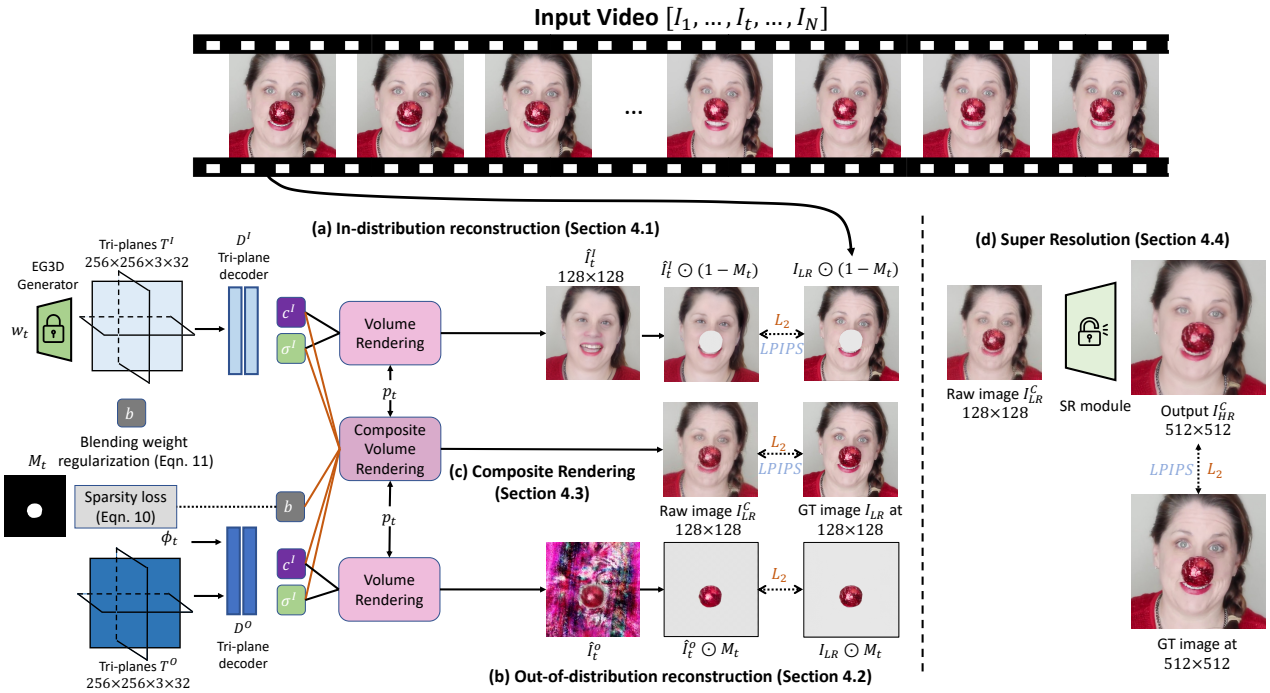


Figure 3. **Overview of our method.** Given a monocular portrait video, we use two radiance fields to represent (a) *in-distribution* face, and (b) *out-of-distribution (OOD)* item. (a) **In-distribution reconstruction** is the *GAN inversion* for the in-distribution face. We apply GAN inversion to each frame, but exclude the pixels within the OOD mask M_t. We perform GAN inversion by using pre-trained EG3D model G, where the pre-trained tri-plane generator and tri-plane decoder D^I are kept frozen. (b) For **out-of-distribution** elements, we propose to model them with a separate radiance field described by an additional tri-plane $\mathbf{T}^O$. During the training process, we optimize the tri-plane $\mathbf{T}^O$, a per-frame latent code ϕ_t, and a new decoder D^O with a masked reconstruction loss. The decoder takes as input tri-plane features $\mathbf{T}^O$ and ϕ_t and outputs color $\mathbf{c}^O$, density σ^O, and blending weight b. (c) **Composite Rendering** compose the *in-distribution* and *out-of-distribution* radiance fields together by using a composite rendering scheme (Section 4.3). (d) Finally, we finetune the **Super Resolution** module in G to achieve a better output in the high resolution. After training, we can perform various semantic edits and free-view rendering, while preserving the face identity and the OOD components.

volume rendering integral along the ray $\mathbf{r}$:

$$\mathbf{C}^O(\mathbf{r}) = \sum_{k=1}^{K} T(t_k)\alpha^O(\sigma^O(t_k)\delta_k)\mathbf{c}^O(t_k)\,. \tag{6}$$

Optimization. The optimization objective is:

$$\begin{aligned}\mathbf{T}^{O*}, \theta^*_{DO}, \phi^*_t &= \underset{\mathbf{T}^O, \theta_{DO}, \phi_t}{\operatorname{argmin}} \; \mathcal{L}^O_t \\ &= \underset{\mathbf{T}^O, \theta_{DO}, \phi_t}{\operatorname{argmin}} \sum_{ij} ||(\mathbf{C}^O_t(\mathbf{r}_{ij}) - \mathbf{C}^{GT}(\mathbf{r}_{ij})) \cdot \mathbf{M}_t(\mathbf{r}_{ij})||^2_2\,,\end{aligned} \tag{7}$$

$\mathbf{C}^{GT}(\mathbf{r}_{ij})$ is the ground-truth color at pixel (i, j).

4.3. Composite volume rendering

Now with in-distribution and out-of-distribution radiance fields, we can compose them together using the blending weight b from Eqn. 5.

Formulation. We compose two radiance fields together by

$$\begin{aligned}\mathbf{C}^C(\mathbf{r}) = \sum_{k=1}^{K} T^C(t_k)\Big(&b\alpha^O(\sigma^O(t_k)\delta_k)\mathbf{c}^O(t_k) \\ &+ (1-b)\alpha^I(\sigma^I(t_k)\delta_k)\mathbf{c}^I(t_k)\Big)\,,\end{aligned} \tag{8}$$

where $T^C(t_k) = \exp(-\sum_{k'=1}^{k-1}(\sigma^O + \sigma^I)\delta_{k'})\,.$

Optimization. The goal is

$$\begin{aligned}\mathbf{T}^{O*}, \theta^*_{DO}, \phi^*_t &= \underset{\mathbf{T}^O, \theta_{DO}, \phi_t}{\operatorname{argmin}} \; \mathcal{L}^C_t \\ &= \underset{\mathbf{T}^O, \theta_{DO}, \phi_t}{\operatorname{argmin}} \sum_{ij} ||C^C(\mathbf{r}_{ij}) - C^{GT}(\mathbf{r}_{ij})||^2_2 \\ &+ \lambda_b \mathcal{L}_b(\mathbf{r}_{ij}) + \sum_{\mathbf{M}_t(ij)\neq 1} \lambda_{spar}\mathcal{L}_{spar} + \mathcal{L}_{LPIPS}(\mathbf{I}^C_{LR}, \mathbf{I}_{LR})\,,\end{aligned} \tag{9}$$

where $\mathcal{L}_{LPIPS}$ is LPIPS loss [63], $\mathbf{I}^C_{LR}$ is the composite rendered image at low resolution (128×128), $\mathbf{I}_{LR}$ is the ground truth image also at 128×128. $\mathcal{L}_{spar}$ is a sparsity

Target | w/o finetuning SR module | w/ finetuning SR module

Figure 4. **Finetuning SR module.** Without finetuning the SR module, the output in high resolution (512×512) is blurry.

loss used to suppress the blending weights for the OOD pixels outside the mask. We use this regularization in case the OOD radiance field dominates, which makes the editing trivial, since we rely on the in-distribution part for the editing.

$$\mathcal{L}_{spar}(\mathbf{r}) = \sum_{k=1}^{K} \mathcal{L}_1(b(t_k)) \,, \tag{10}$$

weight regularizer $\mathcal{L}_b$ is adopted from [55], used to penalize the blending weight b if it is not closer to 0 or 1:

$$\mathcal{L}_b(\mathbf{r}) = \sum_{k=1}^{K} H_b(b(t_k)) \,, \tag{11}$$

where $H_b(x) = -(x\log(x) + (1-x)\log(x))$ is binary entropy. The reason behind Eqn. 11 is that objects cannot co-occupy *the same spatial location*. The entropy loss facilitates a cleaner decomposition to be either an in-distribution object (*i.e.* $b \rightarrow 0$) or an out-of-distribution object (*i.e.* $b \rightarrow 1$).

In practice, we first optimize for Eqn. 4 only. We then jointly optimize for Eqn. 7 and Eqn. 9. It takes 3.5 hours to obtain an accurate reconstruction for a video of 200 frames, on an NVIDIA RTX A4000 GPU.

4.4. Super resolution

After training in Section 4.1, 4.2, and 4.3, we can get reconstruction $\mathbf{I}^C_{LR}$ in low resolution (128×128). We observe that using the pretrained super-resolution (SR) module cannot provide a good high-resolution output as shown in Figure 4. Therefore, we finetune only the SR module in G for higher resolution at 512×512.

Optimization. The loss function is that

$$\mathcal{L}^{SR}(\mathbf{x}, \hat{\mathbf{x}}) = ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 + \mathcal{L}_{LPIPS}(\mathbf{x}, \hat{\mathbf{x}}) \,, \tag{12}$$

where $\mathbf{x} = \mathbf{I}_t$ and $\hat{\mathbf{x}} = SR(\mathbf{I}^C_{LR})$.

4.5. Editing

After the reconstruction, we can modify the latent code w_t to achieve semantic editing. With explicit decomposition, the out-of-distribution contents do not interfere with the semantic editing capability of in-distribution components. Here, any existing GAN-based editing approaches can be used. In our experiments, we use InterfaceGAN [45] and StyleCLIP [40].

4.6. Post-processing

We can put the aligned frames back into the original input video as an optional step. We use a face parsing algorithm [61] to get the face segmentation and paste the face back to the input frame with a Gaussian Blur to smooth the boundary. Finally, we paste the cropped, aligned frame into the full-resolution input video.

5. Experimental Results

5.1. Experimental Setup

Dataset. To evaluate how our approach works on real videos with out-of-distribution objects, we collect 20 online videos as our dataset. The OOD objects contain heavy make-up, and occlusions (*e.g.* facial masks and big glasses). For the face alignment, we use 3DDFA-v2 [23] to get the 68-point landmarks and smooth them across the frames using a sliding window for stabler cropping. After that, we convert them to EG3D's 5-point landmarks and crop the face out of the input frame. For the segmentation masks, we manually label the first frame and then use an off-the-shelf tracking algorithm [12] to get the masks for the rest of frames.

Hyperparameters. We use the Adam optimizer [30] for all the experiments. For in-distribution inversion (Secion 4.1), we optimize for 200 epochs with a learning rate of 1×10^{-3}, $\lambda_\Delta = 1 \times 10^{-3}$. For the out-of-distribution and composite rendering (Secion 4.2, 4.3), we run the optimization for 200 to 300 epochs depending on the video length with a learning rate of 5×10^{-3}, $\lambda_b = 1$, and $\lambda_{spar} = 3$. For the SR module (Section 4.4), we fine-tune the module for 100 epochs with a learning rate of 1×10^{-3}.

Metrics. We evaluate our approach from two aspects: 1) reconstruction accuracy and 2) editability to show the reconstruction-editability trade-off. For the reconstruction accuracy, we evaluate LPIPS [63], $\mathcal{L}_2$, PSNR, SSIM and ID similarity [15]. For editability, we follow [43, 47] and evaluate identity preservation after applying the editing direction. After the editing, we use ArcFace [15] to compute the similarity between the inverted and edited results.

5.2. Quantitative results

Reconstruction. We compare the reconstruction accuracy of our approach with a) hybrid method: PTI [43], b) optimization-based methods: $\mathcal{W}^+$ and $\mathcal{W}$ optimization, and c) encoder-based method: IDE-3D [48]. For PTI, we first perform a $\mathcal{W}^+$ inversion with a learning rate of 1×10^{-3} and 200 epochs. We then fine-tune the generator for 200 epochs with a learning rate of 3×10^{-5}. For $\mathcal{W}^+$ and $\mathcal{W}$

Figure 5. **Qualitative comparison of the reconstruction.** We compare our approach with $\mathcal{W}^+$ and $\mathcal{W}$ optimization, IDE-3D [48], and PTI [43]. Our method shows a better reconstruction accuracy on the OOD videos.

optimization, we use a learning rate of 1×10^{-3} and optimize for 200 epochs. For IDE-3D, we use their encoder directly for the inversion.

We report the results in Table 1. Our approach outperforms other methods by a large margin on all the evaluation metrics. This indicates that our method produces a more accurate reconstruction of the OOD videos.

Table 1. Quantitative comparison for reconstruction quality.

Metrics	LPIPS↓	$\mathcal{L}_2$↓	SSIM↑	PSNR↑	ID similarity↑
ours	**0.1981**	**0.0150**	**0.7722**	**19.6985**	**0.9831**
PTI [43]	0.3144	0.0504	0.6320	13.4483	0.9658
IDE-3D [48]	0.4999	0.1172	0.4512	9.5852	0.8251
$\mathcal{W}^+$	0.3380	0.0383	0.6557	14.7486	0.9154
$\mathcal{W}$	0.4030	0.0618	0.5787	12.4769	0.8652

Editability. We acquire editing directions from InterfaceGAN [45] ("younger", "smile") and StyleCLIP mapper [40] ("eyeglasses", "surprised", "Elsa"). Following previous work [43, 47], we measure the ID similarity between the inverted image and the edited image, as the editing should not change a person's identity. We report our results in Table 2. Our method outperforms other baselines in terms of identity preservation.

Table 2. Quantitative comparison for identity preservation after editing. Higher numbers indicate better identity preservation.

	"eyeglasses"	"surprised"	"younger"	"smile"	"Elsa"
Ours	**0.9368**	**0.9816**	**0.9457**	**0.9556**	**0.8610**
PTI	0.9049	0.9357	0.9319	0.9336	0.7945
IDE-3D	0.8767	0.9481	0.8551	0.8662	0.7871
$\mathcal{W}^+$	0.8971	0.9249	0.9290	0.9170	0.7968
$\mathcal{W}$	0.8793	0.9537	0.9068	0.9208	0.8113

5.3. Qualitative results

Inversion. We show a visual comparison regarding the reconstruction with PTI [43], IDE-3D [48], $\mathcal{W}^+$ and $\mathcal{W}$ in Figure 5. Our method provides higher-fidelity reconstruction results than other baselines, particularly for OOD regions (e.g., heavy make-up or earrings). Compared to encoder-based method IDE-3D, our method shows more consistent results with less flickering. Compared to optimization-based methods, $\mathcal{W}$, $\mathcal{W}^+$, and hybrid method, PTI, our method shows higher-fidelity reconstruction for OOD objects (Refer to our supplementary material for video comparison).

Editing. We show a qualitative comparison regarding the editing in Figure 6. Our method shows faithful editing results and less temporal flickering than other baselines. For more qualitative results, please refer to our supplementary material.

5.4. Speed

We include a comparison of different baselines in Table 3. We compare the speed on 200 frames using a single NVIDIA RTX A4000 GPU. Our method takes more time for the optimization, but gains a significant improvement over the reconstruction-editability trade-off.

Table 3. Speed for different baselines on 200 frames.

PTI [43]	IDE-3D [48]	Ours
2.18h	95s	3.54h

5.5. Number of frames

The number of frames needed for reconstructing OOD objects depends on the number of viewpoints of the OOD object shown in a video. Our method usually takes 30-100 frames for a natural talking video. Compared to encoder-based method IDE-3D [48], our method needs more frames, but we show a higher-fidelity reconstruction and editing performance. Compared to optimization-based methods $\mathcal{W}$,

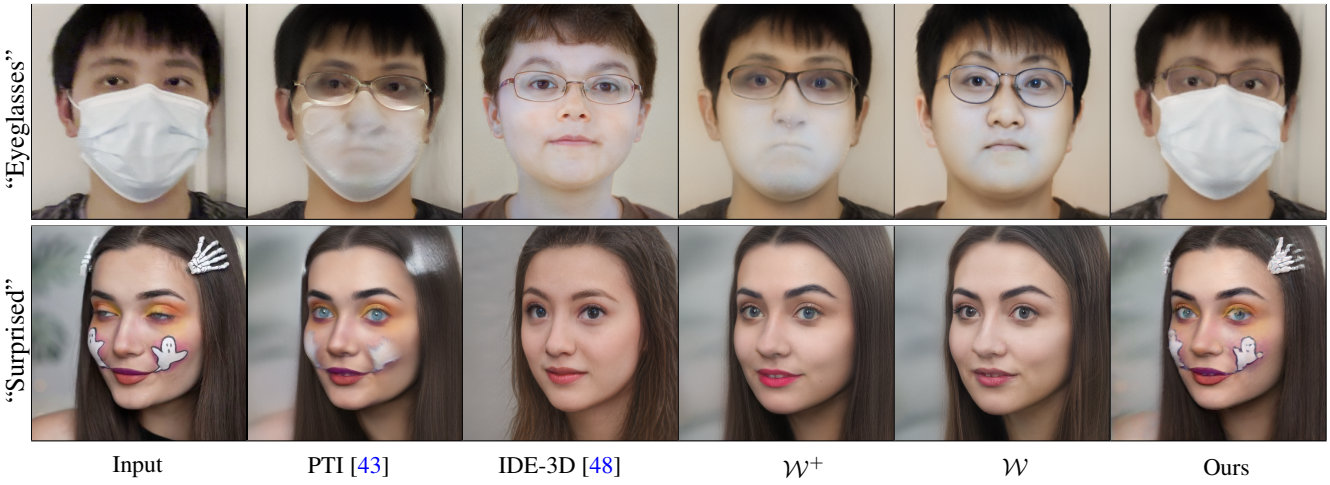


Figure 6. **Qualitative comparison of the editing.** We compare our editing results with other baselines, with different editing latent directions "Eyeglasses", and "Surprised". Our approach can preserve the original appearance details better, and shows improved editability over other baselines.

Figure 7. **Novel view synthesis.** We can synthesize novel views for a fixed frame in a video, which is challenging for 2D GANs. Each column shows different view for the same frame.

$\mathcal{W}^+$, and hybrid method PTI [43], we use the same frames as they do, but we show more accurate reconstruction and better editability in Figure 5 and Figure 6.

5.6. Other Applications

View synthesis. The use of 3D GANs supports rendering novel views after inversion. We show novel view synthesis results in Figure 7. Our method can generate 3D consistent novel views *both* for the face and OOD object.

Object removal. By setting the blending weights of the OOD objects to 0, we can remove OOD objects. We show results in Figure 8.

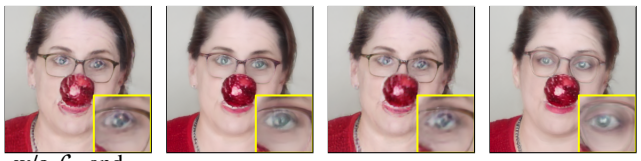
Figure 8. **OOD object removal.** By setting the blending weights inside the mask to 0, we can remove the OOD object.

5.7. Ablation Study

Table 4. Ablation study.

	Inversion		Editing
	$\mathcal{L}_2 \downarrow$	LPIPS↓	ID similarity↑
w/o $\mathcal{L}_b$ and $\mathcal{L}_{spar}$	**0.0114**	**0.1812**	0.9087
w/o $\mathcal{L}_b$	0.0153	0.2051	0.9145
w/o $\mathcal{L}_{spar}$	0.0119	0.1828	0.9088
Full method	0.0150	0.1981	**0.9361**

We introduce two loss functions, Eqn. 10 and Eqn. 11, to enhance the editability in Section 4.3. To validate the loss functions' effects, we conduct an ablation study in Table 4. Without weight regularizer $\mathcal{L}_b$ and sparsity loss $\mathcal{L}_{spar}$, the reconstruction accuracy is improved while the editability is reduced. One of the reasons is that GAN-based editing usu-



w/o $\mathcal{L}_b$ and $\mathcal{L}_{spar}$ | w/o $\mathcal{L}_b$ | w/o $\mathcal{L}_{spar}$ | Full model

Figure 9. **Ablation study on editing.** Without regularizations, the out-of-distribution component dominates ($b \rightarrow 1$) and weakens the editing. Here we show that the other results have "duplicate eyes" artifact because the editing direction "eyeglasses" is not disentangled well with other attributes, and changes the positions of the eyes, while the blending weights are the same as the reconstruction, it results in duplicated eyes.

(a) OOD dominates | (b) Double glasses | (c) Small object | (d) Extreme pose

Figure 10. **Limitations.** Our approach has some limitations. (a) Editing on where OOD blending weights dominate is challenging, (b) Adding another eyeglasses to OOD eyeglasses will result in duplicated objects, (c) Our approach has difficulty handling small objects like small earrings, and (d) extreme poses.

ally also brings unwanted changes to other attributes [45]. In Figure 9, the editing direction "eyeglasses" also moves the position of the eyes. At this time, if the blending weight b is closer to 1 for pixels outside the OOD mask, *i.e.* the OOD part has more contributions, the editing tends to keep the pixel values in the reconstruction stage. While the eyes will be moved due to the editing direction, it results in the duplicate eyes in Figure 9. In contrast, with regularization (Eqn. 10 and Eqn. 11) on the blending weights, pixels in the in-distribution part contribute more to the output, which better supports the editing since we can only edit the in-distribution part.

6. Limitations

Our method still consists of several limitations. We visualize (a)-(d) in Figure 10.

(a) Editing on OOD part. When editing on the OOD region, *e.g.* adding eyeglasses to the heavy makeup region, because the blending weights are closer to 1, the eyeglasses in the in-distribution radiance field are hard to be added.

(b) Duplicate objects. Since our OOD radiance field has no knowledge about the GAN and faces prior, when the OOD object itself is glasses, adding eyeglasses will cause duplicate objects.

(c) Small OOD objects. Our approach has difficulty reconstructing small objects. In Figure 10, we show an example of small earrings.

(d) Extreme poses. For extreme poses, our method fails at editing them.

(e) Objects with limited movement. The radiance field reconstruction suffers when the OOD object has little movement. This may introduce unwanted artifacts like "floater" in the novel views.

7. Conclusions

We have presented a novel method for face video inversion and editing. Our method mainly handles out-of-distribution objects by isolating them from the in-distribution part. Our method can achieve accurate reconstruction by building two radiance fields and then composing them together during the rendering. Then by modifying the latent code in the in-distribution part, we can obtain plausible editing results. We show that our method can achieve a better balance in the reconstruction-editability trade-off than other baselines. Finally, we also show that our method has potential for various applications, including novel view synthesis and object de-occlusion.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, 2019. 3

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

[3] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG)*, 40(3):1–21, 2021. 2

[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *ICCV*, 2021. 3

[5] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *CVPR*, 2022. 3

[6] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 2

[7] David Bau, Hendrik Strobelt, William Peebles, Bolei Zhou, Jun-Yan Zhu, Antonio Torralba, et al. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics (ToG)*, 38(4):1–11, 2020. 3

[8] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in gans. In *ICLR*, 2021. 3

[9] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022. 1, 2, 3, 4

[10] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021. 3

[11] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022. 2

[12] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. 4, 6

[13] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *CVPR*, 2020. 3

[14] Giannis Daras, Augustus Odena, Han Zhang, and Alexandros G Dimakis. Your local gan: Designing two dimensional local attention mechanisms for generative models. In *CVPR*, 2020. 3

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

[16] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *CVPR*, 2022. 3

[17] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *CVPR Workshops*, 2019. 4

[18] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 2, 3

[19] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021. 2, 3

[20] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *arXiv preprint arXiv:2209.11163*, 2022. 3

[21] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleneRF: A style-based 3d aware generator for high-resolution image synthesis. In *ICLR*, 2022. 2, 3, 4

[22] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *CVPR*, 2020. 3

[23] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. In *ECCV*, 2020. 6

[24] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33:9841–9850, 2020. 2, 3

[25] Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images to class-conditional generative networks. In *ECCV*, 2020. 3

[26] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020. 2, 3

[27] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 2, 3

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2, 3

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 2, 3

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[31] Connor Z Lin, David B Lindell, Eric R Chan, and Gordon Wetzstein. 3d gan inversion for controllable portrait image animation. *arXiv preprint arXiv:2203.13441*, 2022. 2, 3, 4

[32] Junyu Luo, Yong Xu, Chenwei Tang, and Jiancheng Lv. Learning inverse mapping by autoencoder based generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 207–216, 2017. 3

[33] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 2, 3

[34] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 3, 4

[35] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 3, 4

[36] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 2

[37] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*, 2019. 3

[38] Yotam Nitzan, A. Bermano, Yangyan Li, and D. Cohen-Or. Face identity disentanglement via latent space mapping. *ACM Transactions on Graphics (TOG)*, 39:1 – 14, 2020. 3

[39] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. In *CVPR*, 2022. 2, 3, 4

[40] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 1, 2, 3, 6, 7

[41] Ankit Raj, Yuqi Li, and Yoram Bresler. Gan-based projector for faster recovery with convergence guarantees in linear inverse problems. In *ICCV*, 2019. 3

[42] 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*, 2021. 2, 3

[43] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42(1):1–13, 2022. 2, 3, 6, 7, 8

[44] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 3

[45] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020. 1, 2, 3, 6, 7, 9

[46] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *arXiv preprint arXiv:2206.10535*, 2022. 3

[47] Adéla Šubrtová, David Futschik, Jan Čech, Michal Lukáč, Eli Shechtman, and Daniel Sỳkora. Chunkygan: Real image inversion via segments. In *European Conference on Computer Vision*, pages 189–204. Springer, 2022. 2, 3, 6, 7

[48] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *ACM Transactions on Graphics (ToG)*, 41(6):1–10, 2022. 2, 3, 6, 7, 8

[49] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 3

[50] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, Christian Theobalt, et al. Pie: Portrait image embedding for semantic control. *arXiv preprint arXiv:2009.09485*, 2020. 3

[51] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, 2020. 3

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

[53] Rotem Tzaban, Ron Mokady, Rinon Gal, Amit H Bermano, and Daniel Cohen-Or. Stitch it in time: Gan-based facial editing of real videos. *SIGGRAPH Asia 2022 Conference Papers*, 2022. 2, 3

[54] Yuri Viazovetskyi, Vladimir Ivashkin, and Evgeny Kashin. Stylegan2 distillation for feed-forward image manipulation. In *ECCV*, pages 170–186. Springer, 2020. 3

[55] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D^2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. *arXiv preprint arXiv:2205.15838*, 2022. 2, 3, 6

[56] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*, 2021. 3

[57] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 4

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

[59] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *ICCV*, 2021. 2, 3

[60] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. In *ICCV*, 2021. 2, 3

[61] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129(11):3051–3068, 2021. 6

[62] Jichao Zhang, Aliaksandr Siarohin, Yahui Liu, Hao Tang, Nicu Sebe, and Wei Wang. Training and tuning generative neural radiance fields for attribute-conditional 3d-aware face generation. *arXiv preprint arXiv:2208.12550*, 2022. 3

[63] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4, 5, 6

[64] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020. 2

[65] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*, 2020. 3