VIVE3D: Viewpoint-Independent Video Editing using 3D-Aware GANs

Anna Frühstück[2*], Nikolaos Sarafianos[1], Yuanlu Xu[1], Peter Wonka[2], Tony Tung[1]
[1] Meta Reality Labs Research, Sausalito [2] KAUST

afruehstueck.github.io/vive3D

Figure 1. We propose **VIVE3D**, a novel method that creates a powerful personalized 3D-aware generator using a low number of selected images of a target person. Given a new video of that person, we can faithfully modify several facial attributes as well as the camera viewpoint of the head crop. Finally, we seamlessly composite the edited face with the source frame in a temporally and spatially consistent manner, while retaining a plausible composition with the static components of the frame outside of the generator's region. The dotted squares in the center frame denote the reference regions for the three different camera poses in the column below.

Abstract

*We introduce VIVE3D, a novel approach that extends the capabilities of image-based 3D GANs to video editing and is able to represent the input video in an identity-preserving and temporally consistent way. We propose two new building blocks. First, we introduce a novel GAN inversion technique specifically tailored to 3D GANs by jointly embedding multiple frames and optimizing for the camera parameters. Second, besides traditional semantic face edits (*e.g. *for age and expression), we are the first to demonstrate edits that show novel views of the head enabled by the inherent properties of 3D GANs and our optical flow-guided compositing technique to combine the head with the background video. Our experiments demonstrate that VIVE3D generates high-fidelity face edits at consistent quality from a range of camera viewpoints which are composited with the original video in a temporally and spatially consistent manner.*

1. Introduction

Semantic image editing has been an active research topic for the past few years. Previous work [21] uses Generative Adversarial Networks (GANs) to produce high-fidelity results in the image space. The most popular backbone is StyleGAN [26–29] as it generates high-resolution domain-specific images while providing a disentangled latent space that can be utilized for editing operations. To edit real photographs, there are typically two steps: The first step maps the input image to the latent space of a pre-trained generator. This is usually accomplished either through encoder-based embedding or through optimization, such that generator can accurately reconstruct the image from the latent code [53]. The second step is semantic image manipulation, where one latent input representation is mapped to another to obtain a certain attribute edit, (*e.g.* changing age, facial expression, glasses, or hairstyle). While existing approaches produce impressive results on single images, extending them to videos is far from straightforward. Among the challenges that arise are: (1) people tend to move their

*This work was conducted during an internship at Meta RL Research.

heads freely in videos (instead of assuming frontal image inputs), (2) the inversion of multiple frames should be coordinated, (3) the inverted face and edits need to be temporally consistent and (4) the compositing of the edited face with the original frame must maintain boundary consistency.

A recent set of approaches has focused on 3D-aware GANs where a 2D face generator is combined with a neural renderer. Given a latent code, a 2D image and the underlying 3D geometry are generated, thus allowing for some camera movement while rendering the head of the person.

In this paper, we tackle the problem of viewpoint-independent face editing in videos. The edited face is rendered from novel views in a temporally-consistent manner. Specifically, we use a 3D-aware GAN in the temporal domain and apply facial image editing techniques per frame that are temporally smooth regardless of the rendered view. Compared with other GAN-based video editing approaches [5, 48], our method is the first to perform viewpoint-independent video editing while showing the full upper body of the person in the video with high fidelity.

VIVE3D takes a video of a person captured from a monocular camera as input. The captured person can move freely across time, talk, and make facial expressions while their body can be visible. Unlike all prior work that learns a generator and performs edits on the exact same video, we disentangle these steps. Hence the output of our approach can be a different video of the same person or the same video. In both cases, the face has undergone one or more attribute edits and is rendered from a novel view. To accomplish this challenging task, we introduce several novel components, each addressing one challenge of the problem at hand. Specifically, we first propose a simple yet effective technique to create a personalized generator by inverting multiple frames at the same time. The simultaneous inversion of N frames exposes the generator to a variety of facial poses and expressions, which results in a larger capacity that we can then utilize. Our generator can generalize to new unseen videos of the same identity where the person might be wearing a different shirt, a result not demonstrated in the literature so far. In addition, we propose to optimize the camera pose of the 3D-aware GAN during inversion to obtain an accurate estimate which angle the face was captured from. Finally, we introduce an optical flow-based compositing method to properly place the novel view of the edited face back into the original frame while ensuring that the end result is temporally and spatially consistent. Our experimental work provides a wide range of qualitative and quantitative results to demonstrate that VIVE3D accomplishes semantic video editing with changing camera poses in a faithful way. In summary, our contributions are:

- A new 3D GAN inversion technique that jointly embeds multiple images while optimizing for their camera poses.
- A complete attribute editing framework and an optical flow-based compositing technique to replace the edited face in the original video.
- VIVE3D is the first 3D GAN-based video editing method and the first that can change the camera pose of the face.

2. Related Work

GAN Inversion. GANs are a powerful tool for semantic editing. Most editing techniques are tailored to StyleGAN, the state-of-the-art of 2D GANs [26–29]. Several editing techniques [10, 18, 19, 25, 34] build upon StyleGAN as it uses an intermediate disentangled latent space, usually referred to as $\mathbf{w}$-space. Before editing, a latent space representation of the input image has to be recovered using a process typically referred to as Inversion or Projection [1, 2, 12, 50]. Refer to [52] for a survey of inversion techniques. In contrast to optimization-based inversion techniques, learning-based approaches attempt to obtain faster latent space correspondences by training encoders [4, 36, 47]. In order to retain the generalization ability of the $\mathbf{w}$-space while providing a high-quality inversion, Pivotal Tuning [37] has successfully shown that trained generators can overfit to target images while still maintaining a navigable latent space. Recent works study 3D GAN inversion [30, 31], attempting to infer a 3D representation for a reference image.

GAN-based Latent Space Editing. Once an appropriate latent space representation of an input image has been recovered, semantic edits can be applied by navigating the latent space manifold surrounding the inverted latent code. Unsupervised techniques attempt to find interesting edits without labeled data [22, 24, 41, 49]. InterfaceGAN [39, 40] is a simple and robust supervised technique that is highly recommended for practical applications, and as such we also employ it in our work. While there is a plethora of other techniques [3, 11, 44, 45, 55, 59] the development of related latent space manipulations itself is not the focus of our work. Another line of work is text-based editing which gained immense popularity during the last year [20, 35].

3D-aware GANs. Recent GAN papers attempt to discover 3D information from large collections of 2D images using Neural Radiance Fields (NeRFs) as shape representations [8, 9, 14, 23, 33, 38, 58]. While most of these papers share similar architectural ideas, EG3D [9] has emerged as a popular basis for follow-up work (*e.g.* integration of a segmentation branch [43]). We chose to build upon EG3D, but our work is also applicable to other generators with a similar latent space. For more information on 3D GAN architectures, we refer the interested reader to a recent survey [51].

Video Synthesis and Editing. One branch of work attempts to leverage 2D GANs to generate video sequences [17, 42, 46, 57]. These ideas can be extended to create 3D videos [6], which also rely on 3D NeRFs.

Figure 2. **VIVE3D Pipeline.** To create an edited video, we first need to create a personalized generator by jointly inverting selected faces and fine-tuning a pre-trained generator. We then invert the cropped face regions from a source video (which could be the same or a different video) into our personalized generator and recover the latent codes and camera poses for each target frame. We are able to perform semantic editing on the inverted stack of latent codes using previously discovered latent space directions and we can freely change the camera path around the face region. In order to composite the face with the source frame in a consistent fashion, we use optical flow to correct the position of the inset within the frame, which allows us to composite the result in a seamless and temporally consistent fashion.

GAN-based Video Editing. GAN-based video editing is the core topic of this paper. Duong *et al*. [15] employ deep reinforcement learning for automatic face aging. Latent-Transformer [55] encodes frames into the StyleGAN latent space using an encoder. They train a transformer to do attribute editing on single frames and blend the result with Poisson blending. The main competitor to our work is Stitch it in Time (StiiT) [48], which crops the faces from a video, edits them with 2D GAN techniques, and merges the edited result back to the video with some blending. However, StiiT does not learn a 3D model of the human head, overfits to a particular video, and is unable to provide edits to the viewpoint of the human head. Recently, VideoEdit-GAN (VEG) [54] attempted to improve the temporal consistency of StiiT by running a two-step optimization approach focused on localized temporal coherence. Alaluf *et al*. [5] use StyleGAN3 for video editing, to leverage its inherent alignment capabilities and reduce texture sticking artifacts. Since this is an active area of research, all these techniques are concurrent work to our method, yet we do provide comparisons to showcase the benefits of our proposed approach.

3. Method

In this section, we introduce the key components of VIVE3D to perform frame-by-frame video editing while allowing for rendering the edited face from new views. We leverage a 3D-aware generator that infers 3D geometry and camera positions while being trained solely on 2D images. We build a personalized 3D-aware generator by performing joint inversion on multiple frames and then use it to perform attribute editing, apply camera viewpoint changes, and finally composite the edited face rendered from a new view back into the original frame. An overview of our proposed approach is depicted in Fig. 2 while the personalized generator architecture is shown in Fig. 3.

3.1. Personalized 3D-Aware Generator

Face Selection and Cropping. To create a personalized 3D-aware GAN model, we start by processing a short range from the input video where N frames are selected such that they cover a range of orientations and facial expressions of the target person. We detect the facial keypoints within these frames using an off-the-shelf facial keypoint detector [7] and use them to determine the face bounding box within the frame. This is achieved by calculating a rigid transformation from the facial keypoints in the frame to the facial keypoints in a generated example image, thereby aligning the keypoints at the center of the crop in the same way as the generator's original training data. We pick a specific field of view for cropping the faces and optimizing the generator, but the field of view remains a flexible parameter that can be adapted during any later stage in the pipeline.

Simultaneous Inversion. We propose to perform multiple inversions simultaneously. EG3D has two major components in its generator. The first component uses a mapping network to map random vectors into a semantically meaningful space, called $\mathbf{w}$-space. Vectors in this space control a 3D neural field that defines a 3D representation that is rendered using volumetric rendering. The second component is a 2D upsampler that performs a $4\times$ super-resolution on the original output. We invert all selected faces simultaneously into the $\mathbf{w}$-space following a strategy similar to [37] that we discuss in detail below.

In order to find a representation in $\mathbf{w}$-space, we define a "global" $\mathbf{w}_{ID}$ aiming at capturing the global identity features of the target person, and a "local" offset vector $\mathbf{o}_n$

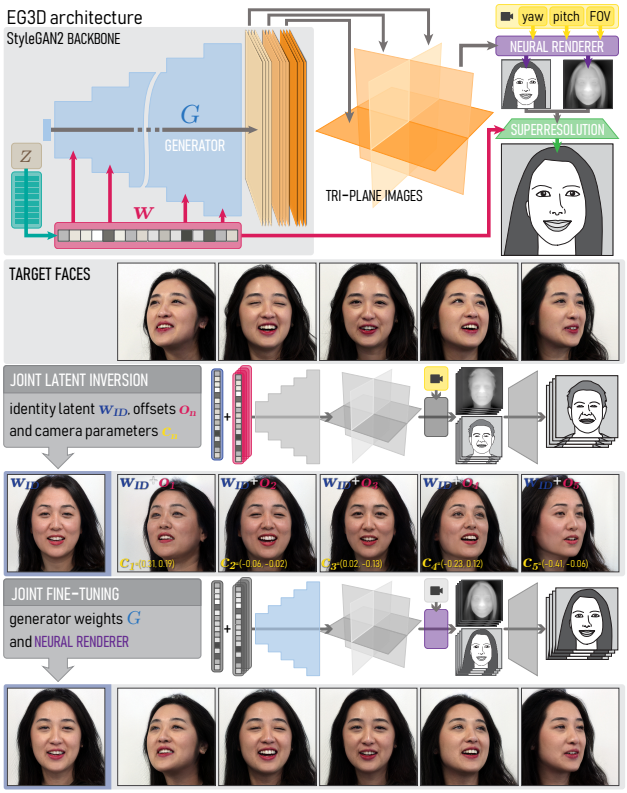

Figure 3. **Personalized Generator.** First, we run a joint inversion on N selected target faces, where we optimize a shared target person latent $\mathbf{w}_{ID}$ and an offset $\mathbf{o}_n$ for each face. This ensures the inversions share information about the target. Simultaneously, we jointly optimize for the camera pose c_n. We then fine-tune the generator to ensure it captures the fine details of the target identity. Note that the "default" latent *(left column)* implicitly captures the identity of the target person without being explicitly optimized.

for each input expression $\mathbf{F}_n$ that encodes the differences of each individual facial expression and position from the default $\mathbf{w}_{ID}$. The length of each $\mathbf{o}_n$ is regularized using an $\mathcal{L}_{L2}$ loss, aiming to keep the difference as small as possible, and capturing all similarities between the input images within the default person latent $\mathbf{w}_{ID}$. We use a combination of a perceptual loss $\mathcal{L}_{LPIPS}$ and a pixel loss $\mathcal{L}_{L1}$ for the inversion. Note that during this stage, we calculate these losses on the raw output $\mathcal{G}_{raw}(\mathbf{w}_{ID}+\mathbf{o}_n)$ of the EG3D neural renderer at 128×128 resolution because we observed that it yields sharper result quality rather than evaluating the loss at the output of the super-resolution network. We downsample our target images to the same resolution $\mathbf{D}_{128}(\mathbf{F}_n)$ to compare. To ensure that we can faithfully capture the target person's identity and expression, we use BiSeNet [56] to obtain a segmentation $\mathbf{S}_{exp}(\mathbf{F}_n)$ of the facial regions encoding the expression (eyes, mouth, eyebrows, and nose) and add an additional feature loss on this area to encourage consistent facial expressions (*e.g.* closed eyes). To obtain the inversion, in each optimization step, we sum up the losses for each face image $\mathbf{F}_n$, therefore jointly optimizing all targets simultaneously, yielding a total loss $\mathcal{L}_{inv}$.

$$\begin{aligned}\mathcal{L}_{inv} = \textstyle\sum_{n=0}^{N} & \lambda_{LPIPS}\,\mathcal{L}_{LPIPS}(\mathcal{G}_{raw}(\mathbf{w}_{ID}+\mathbf{o}_n), \mathbf{D}_{128}(\mathbf{F}_n)) + \\ & \lambda_{L1}\,\mathcal{L}_{L1}(\mathcal{G}_{raw}(\mathbf{w}_{ID}+\mathbf{o}_n), \mathbf{D}_{128}(\mathbf{F}_n)) + \\ & \lambda_{seg}\,\mathcal{L}_{LPIPS}(\mathbf{S}_{exp}(\mathcal{G}(\mathbf{w}_{ID}+\mathbf{o}_n)), \mathbf{S}_{exp}(\mathbf{F}_n)) + \\ & \lambda_{reg}\,\mathcal{L}_{L2}(\mathbf{o}_n)\end{aligned}$$

Due to the 3D awareness of the EG3D generator, the quality of the inversion into the latent space is highly sensitive to the camera parameter settings. Hence, in addition to optimizing for $\mathbf{w}_{ID}$ and $\mathbf{o}_n$, we propose to also allow the inversion to optimize for the camera parameters c_n (yaw_n and $pitch_n$) for each input expression $\mathbf{F}_n$, which reliably estimates the camera position that the face is captured from.

A key advantage of this joint optimization is that the facial characteristics of the person preserve their high fidelity even when seen from novel views. When inverting a single image of a side-facing person into the EG3D latent space, exploring other viewpoints of the inverted latent can lead to significant distortions. Often, unseen features (*e.g.* hidden ears) can be blurry or distorted, and the identity no longer resembles the input from a different viewpoint. The joint inversion, however, ensures that the different views are embedded closely enough in latent space such that even unseen views yield consistently identity-preserving outputs.

Generator Fine-tuning. We propose a variant of Pivotal Tuning [37] to jointly fine-tune the weights of the generator $\mathcal{G}_{EG3D}$ on all input faces $\mathbf{F}_n$, while keeping the detected $\mathbf{w}_{ID}$, $\mathbf{o}_n$ and camera poses c_n fixed. Here, we do not allow the weights of the upsampler of the generator to be updated as we want to preserve the generalization capabilities of the super-resolution network and prevent it from overfitting to our target images. During this fine-tuning stage, we employ perceptual and pixel losses described as follows:

$$\begin{aligned}\mathcal{L}_{tune} = \textstyle\sum_{n=0}^{N} & \lambda_{LPIPS}\,\mathcal{L}_{LPIPS}(\mathcal{G}_{ID}(\mathbf{w}_{ID}+\mathbf{o}_n), \mathbf{F}_n) + \\ & \lambda_{L1}\,\mathcal{L}_{L1}(\mathcal{G}_{ID}(\mathbf{w}_{ID}+\mathbf{o}_n), \mathbf{F}_n)\end{aligned}$$

Finally, we obtain a personalized EG3D generator $\mathcal{G}_{ID}$, fine-tuned to a set of facial expressions of the target person. We verify that the fine-tuned generator indeed provides a good generalized latent space for the target person even though it was inverted and tuned based on a low number of frames by exploring the person created by the "global" latent code, which was not explicitly fine-tuned for, as well as through a latent space walk in the fine-tuned latent space.

3.2. Frame-by-frame Video Inversion

With the personalized 3D-aware generator in hand, we are now given a video of the same person as input which can be different from the one the generator was trained on. To process our new target video, we extract the facial keypoints from each frame f to determine the location of the box to indicate the face crop within the frame. In order to stabilize the crop over time, which supports the temporal coherence

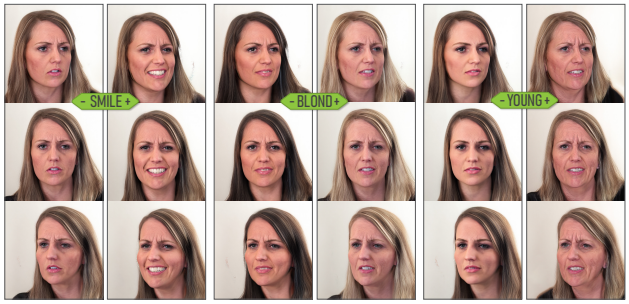

Figure 4. **InterfaceGAN edits.** We show InterfaceGAN editing directions discovered in the latent space by applying them on our personalized generator. The attribute edits are consistent in 3D.

of the inversion, we perform a Gaussian smoothing on the extracted facial keypoints along the temporal axis after extraction. However, it is important to not over-smooth, because fast motions in the video would yield distorted keypoint locations, deteriorating the inversion quality.

We then perform a frame-by-frame inversion of the extracted face regions $\mathbf{F}_f$ into the space of the fine-tuned generator $\mathcal{G}_{ID}$. Like before, we optimize for an offset $\mathbf{o}_f$ for each input frame $\mathbf{F}_f$, as well as regularizing the offset length. After inverting the first frame, each consecutive frame $\mathbf{F}_{f+1}$ is inverted starting from the previous inversion and only needs a low number (~50) of optimization steps.

$$\mathcal{L}_{vid} = \lambda_{LPIPS}\,\mathcal{L}_{LPIPS}(\mathcal{G}_{ID}(\mathbf{w}_{ID}+\mathbf{o}_f),\mathbf{F}_f) + \lambda_{L1}\,\mathcal{L}_{L1}(\mathcal{G}_{ID}(\mathbf{w}_{ID}+\mathbf{o}_f),\mathbf{F}_f) + \lambda_{reg}\,\mathcal{L}_{L2}(\mathbf{o}_f)$$

This inversion yields a stack of offsets $\mathbf{o}_f$ from the identity latent $\mathbf{w}_{ID}$ as well as camera parameters c_f (yaw_f, $pitch_f$) encoding the expression and camera position for each frame.

3.3. Attribute Editing and Novel View Synthesis

Since EG3D is built on top of StyleGAN2, we can leverage existing latent space editing techniques in order to discover semantic editing directions in the latent space of EG3D. As a proof of concept, we implemented InterfaceGAN [40] to find meaningful latent space direction vectors. We re-trained classifiers on the CelebA dataset for several facial attributes such as age, smile, gender, glasses, beard, and hair color and use these classifiers to classify the reference outputs of a set of randomized latent codes from our generator. Finally, we used an SVM to recover editing boundaries from these classified latent codes, which allows us to perform attribute editing in the EG3D latent space, as shown in Fig. 4. For a given latent space direction $\mathbf{w}_{dir}$, we apply an edit as a linear combination of the person latent $\mathbf{w}_{ID}$ with the direction, multiplied by an empirically chosen weight α_{dir}, which can be positive or negative. For our video sequence, we use the edited person latent $\mathbf{w}_{ID}{}'$ as the new identity to which we apply our video offsets $\mathbf{o}_f$.

$$\mathbf{w}_{ID}{}' = \mathbf{w}_{ID} + \alpha_{dir} \times \mathbf{w}_{dir}$$

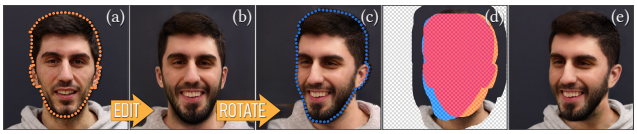

Figure 5. **Border Composition.** We calculate the composition border based on face segmentations of the target image **(a)** and the edited inset **(c)**. We unite the masks and dilate the resulting joint mask to obtain a boundary around the face regions **(d)** that should be optimized, which allows us to create the final composition **(e)**.

In addition, we explore our temporal stack of latent encodings from novel views, diverging from the input views the inversion discovered. This allows us, to generate frontalized videos of the person by fixing the camera position or to define arbitrary camera trajectories around the subject.

3.4. Compositing with Source Video

After editing the inverted video, we want to recomposite the edited faces back with the source frames such that the edited video is temporally and spatially consistent. We accomplish this by running an optimization to ensure that the boundaries between the compositing of the edited face and the background of the source frame are smooth. Since cluttered video backgrounds are hard to reproduce consistently and without artifacts – especially for novel views – we define a compositing boundary region in a similar manner to [48]. To accomplish this, we need an accurate segmentation of the face and hair for both the original frame as well as the edited face. Hence, we use BiSeNet [56], a semantic segmentation technique, that accurately provides such face semantics. We use the semantic regions for both the original and edited face to form a union of their respective masks, obtaining a boundary region around the relevant face region, as illustrated in Fig. 5. We run a small number of optimization steps, optimizing for the boundary region of the edited image to appear as close as possible to the boundary region of the input frame while retaining the appearance of the edit. Finally, we use an affine transformation to reinsert the cropped face region back into the original frame and we alpha blend along the optimized boundary region to seamlessly composite the edit with the source frame.

3.4.1 Flow-based View Adjustment

During the process of re-inserting the edited face $\mathbf{F}_{edit}$ back into the source frame S, a major challenge arises when the camera pose has been modified and the face is rendered from a novel view. This is because the face is oriented within the bounding box based on the facial keypoints as described in Sec. 3.1 while upon a camera pose change, the face pivots around the keypoints. When, for instance, attempting to replace a face viewed at an angle with a frontalized face while retaining the original crop boundary of the inset, the keypoints are still roughly in the same location,

Figure 6. **View adjustment.** After cropping **(a)** and inverting **(b)** a face, we perform face editing **(c)** and change the camera viewpoint to an unseen angle **(d)**. When replacing the face in the original frame with this edit, it yields poor quality *(bottom center)* even for small angular changes, because the rotated face is in the wrong location with respect to the body. We address this by estimating the optical flow **(e)** between the face crop and the edit and use the flow direction to correct the location of the reference face based on the prospective inset **(f)**. This allows us to composite the edited face into the frame in a natural-looking fashion *(bottom right)*.

yet the mass of the head, and crucially, the neck is shifted according to the face rotation, as seen in Fig. 6, which results in the inset being disconnected from the rest of the body even when using a boundary stitching technique.

To alleviate this problem, we introduce a simple yet effective technique to reposition the reference face region within the source frame. We discover the optimal position of the updated inset with respect to the source frame by estimating the optical flow between the face segmentation in the source frame S and the face segmentation in the inset region $\mathbf{F}_{edit}$ after camera rotation. We convert the images to grayscale and use Farnebäck optical flow [16] to evaluate a dense flow field of the displacement between the edited and target faces. The optical flow is defined as a 2D displacement vector field $\mathbf{d}$ with the displacement vector at image position (x, y) given by $\mathbf{d}(x, y) = (u(x, y), v(x, y))$ where the correspondence between the two images $\mathbf{F}_f$ and $\mathbf{F}_{edit}$ is:

$$\mathbf{F}_{edit}(x + u(x, y), y + v(x, y)) = \mathbf{F}_f(x, y).$$

We then compute vector magnitudes $\|\mathbf{d}\| = \sqrt{u^2 + v^2}$ and directions $\phi = atan2(v, u)$, respectively. After eliminating all vectors with a magnitude smaller than a threshold ϵ, we create a histogram of all remaining directions. We define a dominant displacement vector $\mathbf{d}_{dom}$ from the median direction of the histogram bin with the largest count and the maximum vector length within that histogram bin. This ensures that erroneous flow directions from features that are present within one of the two images but not the other are not contributing to the final output.

The displacement vector $\mathbf{d}_{dom}$ is reprojected from inset space into frame space and is used to correct the location of the reference face crop. To ensure a smooth transition between adjacent frames, we perform temporal Gaussian smoothing of the recovered displacement vectors. We then apply our inset optimization using the updated reference areas and obtain a significantly more faithful result, allowing for large camera changes with natural-looking results.

Table 1. **Video Quality Metrics.** We compare the quality of our inversion with StiiT using reconstruction metrics on a subset of the *VoxCeleb* dataset. We also evaluate the Fréchet Inception Distance (FID) of inversion and edits with respect to the source video.

	Reconstruction Quality		**Editing Quality**		
	PSNR ↑	**SSIM** ↑	**FID** ↓		
METHOD	INVERSION		INVERSION	AGE EDIT	ANGLE EDIT
StiiT	38.0134	**0.9798**	6.8329	15.8021	19.0371
VIVE3D	**38.1259**	0.9704	**4.9852**	**9.3410**	**8.9953**

4. Experiments

We conduct a wide range of quantitative and qualitative comparisons to demonstrate the key contributions of our work along with ablation studies against baselines and simplified variants where proposed modules are removed. We showcase that VIVE3D is on par with StiiT in terms of reconstruction quality while it also greatly outperforms prior work in editing quality and identity preservation. However, a key novelty afforded by VIVE3D is the ability to render the edited face from novel viewpoints within the existing frame, a task for which comparisons are hard due to the absence of ground truth. We showcase this with qualitative results and videos provided in the supplementary material.

4.1. Quantitative Evaluation

We establish comparisons with StiiT [48], the key competitor to our work in the field of GAN video editing, and with VEG [54] for which many components (*e.g.* their stitching technique) are identical to StiiT, so we only compare facial similarity metrics. Please see the supplementary material for a detailed description of the comparison setup between our method and the competitors.

Inversion Quality. We provide a quantitative comparison of our inversion quality by measuring the reconstruction quality with respect to the input video in Table 1. We evaluate PSNR and SSIM for our method and for StiiT on a set of 16 videos from the *VoxCeleb* [32] dataset, inverting the face region and recompositing it with the source video without edits. Both methods perform well on the reconstruction of the input signal and the final reconstruction quality of our technique is on par with StiiT.

Image Quality. To evaluate the image quality produced by the respective techniques, we compute the Fréchet Inception Distance (FID), which is a commonly used quality metric for GANs. To obtain the FID for each video, we compare the set of all frames of the inverted video, as well as selected edits, with all frames of the source video. Our method is able to score very good FID scores overall (see Table 1 right), confirming the quality of our results.

Face Fidelity. We calculate the fidelity of our inversion and edits based on a facial similarity metric, ArcFace [13],

Table 2. **Face Similarity Metrics.** We evaluate the identity preservation of inversion and edits based on the cosine similarity of ArcFace features extracted from generated face crops with respect to the face crops of the source video. To evaluate coherence over time, we measure the dissimilarity between consecutive frames.

	METHOD	Similarity to Source ↑	Temporal Difference ↓
Inversion	e4e	0.6923	6.2851
	StiiT	**0.9261**	1.2361
	VIVE3D	0.9203	**1.0444**
Age Editing	StiiT	0.7891	1.4126
	VEG	0.6004	1.7159
	VIVE3D	**0.8381**	**1.2257**
Angle Editing	StiiT	0.6695	1.3102
	VEG	0.4955	1.4502
	VIVE3D	**0.8694**	**1.2761**

which extracts a 512-D feature vector capturing facial characteristics from a face region. We compute the metrics based on the respective face crops of the final inset region, scaled to 512×512px, for our method, e4e encoding, StiiT, and VEG in Table 2 and average across a set of 16 videos from the *VoxCeleb* [32] dataset. The facial similarity is evaluated both with respect to the input video (frame-by-frame) as well as temporally by calculating the dissimilarity of adjacent frames in order to survey the temporal coherence of facial characteristics. In all cases, we use the cosine similarity between the extracted ArcFace deep features. We observe that the quality of the inversion is good for both StiiT and VIVE3D, both significantly improving upon e4e encoding. VEG uses the same PTI-based inversion as StiiT and is therefore not listed. For latent space editing, our proposed VIVE3D leads the competition. VEG exhibits good temporal coherence, however, the edits contain artifacts, resulting in a deterioration in the similarity to the source video. Additionally, our technique reconstructs the facial identity faithfully even for angle edits, a task in which StiiT and VEG fail to produce plausible results due to their methods' inability to accommodate changes in the head rotation.

Resource Usage. We compare runtimes and memory requirements for the default pipeline of related methods and our method, respectively, in Table 3. We split each method, wherever applicable, into precomputation, main method, and postprocessing steps and used the hyperparameter settings according to the authors' suggestions. Note that the precomputation step in our method has to only be run once per identity and can then be applied to multiple videos. All experiments are performed on an example video consisting of 200 frames at a resolution of 1920×1080px, using a single NVIDIA A100 GPU with 40GB memory.

Table 3. **Timings and Memory Requirements.** We provide runtimes and Memory requirements for ours and competing methods.

METHOD	Total	Precompute	Main	Postprocess	GPU
StiiT	58m 53s	28m 21s	30m 19s	—	22GB
VEG	159m 54s	107m 9s	34m 41s	18m 3s	19GB
VIVE3D	35m 43s	6m 58s	14m 54s	13m 51s	21GB

Table 4. **Ablation Study.** We demonstrate the effect of removing various components of our pipeline on several quality and reconstruction metrics. We measure the face similarity using the cosine similarity of ArcFace features of the generated face crop, and the reconstruction metrics at the target video resolution.

Ablation Type	Face Similarity ↑	PSNR ↑	SSIM ↑
VIVE3D, full	**0.9101**	**35.5367**	**0.9763**
no generator fine-tuning	0.7191	33.1202	0.9616
no flow correction	0.8198	24.8845	0.9350
no regularization	0.8007	25.6537	0.9162
single input for inversion	0.7382	25.1950	0.9137

Ablation Study. We quantitatively verify the effectiveness of different architecture choices we made, as shown in Table 4. We compare several metrics with respect to the source video for our default implementation, and the ablation experiments, respectively. We run four experiments on a set of 5 videos: (1) VIVE3D without generator fine-tuning, (2) VIVE3D without flow-based adjustment, (3) VIVE3D without the joint $\mathbf{w}_{ID}$ latent and offset regularization, (4) using only a single input face for the generator inversion and fine-tuning, which in practice is almost identical to only frame-by-frame inversion in EG3D without any personalized generator. We demonstrate that all experiments deteriorate the facial fidelity as well as the reconstruction quality.

4.2. Qualitative Evaluation

Semantic Edits. First, we demonstrate that the quality of semantic edits using VIVE3D, adapting well-established latent space editing techniques for 3D GANs, is on par with the editing quality of StiiT, as shown in Fig. 7. Note that while both approaches discover latent space directions using InterfaceGAN [40], the latent spaces and discovered directions are dissimilar, yet both plausible.

Synthesizing novel views. We show that VIVE3D can accommodate changes in the camera view with natural-looking results for a wide range of views regardless of the input face orientation, as shown in Fig. 8. This is nontrivial

Figure 7. **Comparisons with StiiT on attribute editing.** We show an example of our method and StiiT editing the subject's age in the video frame *(center column)*. Both methods yield plausible but distinctly different results. Our results *(columns **(a)**, age=–1.4 and **(c)**, age=+2.3)* vs StiiT *(columns **(b)**, age=–8 and **(d)**, age=+12)*.

Figure 8. **Changing camera poses.** Our method can freely alter the camera pose and composite the result back with the source frame by fixing the divergence between the source and target pose using our optical flow correction strategy. The generated results look natural despite the static body pose.

Figure 9. **Comparisons with StiiT on viewpoint changes.** VIVE3D ***((a)** and **(c))*** produces plausible results for both positive and negative yaw changes, whereas StiiT ***((b)** and **(d))*** is unable to create natural compositions of the edit with the target frame.

as we need to ensure that the person's identity is consistent from multiple viewpoints while the body in the source frame also defines a rigid constraint to which the head alignment must be adjusted to. While StiiT produces good results for attribute edits, it cannot composite images with angular changes, despite the fact that a limited head pose change can be achieved by applying latent space manipulations. In Fig. 9, we show a comparison where StiiT is unable to generate a reasonable composition, whereas we can achieve natural-looking results for a similar head rotation.

Compositing with challenging boundaries. When parts of the head or hair are visible both inside and outside the face bounding box then compositing the edited frame back into the original input is a challenging task. In most cases, we address these scenarios by adjusting our camera parameters (*e.g.*, choosing a wider field of view for our generator) but some configurations can still be challenging, especially when different textures need to be matched. We show in Fig. 10 that our technique attempts to produce plausible inset optimizations even for such instances.

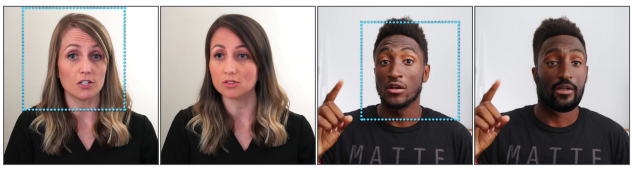

Figure 10. **Spatial Consistency.** VIVE3D composites images with challenging boundaries such as long hair *(right)*, yielding faithful hair color change results. For hard boundary cases, such as matching with a static piece of hair outside the boundary crop *(left)*, it plausibly connects the contents of the two images.

Figure 11. **Limitations.** VIVE3D inherits some limitations from the frameworks we rely on. EG3D cannot capture extreme poses well *(left)*. Large angle changes cannot be composited naturally with the static body in the source frame. For extreme edits (gender edits that change the hair structure *(right)*), it is difficult to yield temporally consistent results, both due to the entanglement of the latent space editing and the challenges of frame compositing.

Limitations. Changing the camera parameters for the head in videos with fast motion or discontinuities results in artifacts because the flow estimation becomes unstable. Furthermore, we inherit the shortcomings of EG3D (see Fig. 11): stronger entanglement of attribute edits compared to StyleGAN2, extreme camera poses are not captured in the training set, and texture sticking. Finally, since the video outside the face region is immovable, the range of possible changes is constricted to plausible compositions.

5. Conclusion

In this paper, we introduced VIVE3D, a novel framework that uses prior information encoded in 3D GANs for video editing. Our edits are identity-preserving and temporally consistent. While we enable standard semantic edits, such as age, or expressions, a distinguishing feature of our work is that we facilitate edits that alter the view of the head. This capability is not available in any 2D GAN-based prior work. The key building blocks of our work are a new embedding algorithm that jointly embeds multiple frames and optimizes for camera pose as well as flow-guided video compositing. In future work, we aim to extend our framework to include a 3D GAN for head details and another 3D GAN for the body. We also plan to investigate performance speedups by replacing various optimizations with encoders.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to embed images into the StyleGAN latent space? In *ICCV*, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN++: How to edit the embedded images? In *CVPR*, 2020. 2

[3] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *ToG*, 2021. 2

[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *CVPR*, 2021. 2

[5] 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 Workshops*, 2022. 2, 3

[6] Sherwin Bahmani, Jeong Joon Park, Despoina Paschalidou, Hao Tang, Gordon Wetzstein, Leonidas Guibas, Luc Van Gool, and Radu Timofte. 3D-aware video generation. *arXiv preprint arXiv:2206.14797*, 2022. 2

[7] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks). In *ICCV*, 2017. 3

[8] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2NeRF: Unsupervised conditional p-GAN for single image to neural radiance fields translation. In *CVPR*, 2022. 2

[9] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 2

[10] Bindita Chaudhuri, Nikolaos Sarafianos, Linda Shapiro, and Tony Tung. Semi-supervised synthesis of high-resolution editable textures for 3D humans. In *CVPR*, 2021. 2

[11] Anton Cherepkov, Andrey Voynov, and Artem Babenko. Navigating the GAN parameter space for semantic image editing. In *CVPR*, 2021. 2

[12] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 2018. 2

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

[14] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. GRAM: Generative radiance manifolds for 3D-aware image generation. In *CVPR*, 2022. 2

[15] Chi Nhan Duong, Khoa Luu, Kha Gia Quach, Nghia Nguyen, Eric Patterson, Tien D Bui, and Ngan Le. Automatic face aging in videos via deep reinforcement learning. In *CVPR*, 2019. 3

[16] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *SCIA*, 2003. 6

[17] Gereon Fox, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. StyleVideoGAN: A temporal generative model using a pretrained StyleGAN. In *BMVC*, 2021. 2

[18] Anna Frühstück, Ibraheem Alhashim, and Peter Wonka. TileGAN: Synthesis of large-scale non-homogeneous textures. In *SIGGRAPH*, 2019. 2

[19] Anna Frühstück, Krishna Kumar Singh, Eli Shechtman, Niloy J. Mitra, Peter Wonka, and Jingwan Lu. InsetGAN for full-body image generation. In *CVPR*, 2022. 2

[20] Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. StyleGAN-NADA: Clip-guided domain adaptation of image generators. *ToG*, 2022. 2

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 1

[22] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. *NeurIPS*, 2020. 2

[23] Fangzhou Hong, Zhaoxi Chen, Yushi Lan, Liang Pan, and Ziwei Liu. EVA3D: Compositional 3D human generation from 2D image collections. In *ICLR*, 2023. 2

[24] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *ICLR*, 2019. 2

[25] Cemre Efe Karakas, Alara Dirik, Eylül Yalçınkaya, and Pinar Yanardag. Fairstyle: Debiasing StyleGAN2 with style channel manipulations. In *ECCV*, 2022. 2

[26] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020. 1, 2

[27] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *NeurIPS*, 2021. 1, 2

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 2

[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. 1, 2

[30] Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 3D GAN inversion with pose optimization. In *WACV*, 2023. 2

[31] Connor Z. Lin, David B. Lindell, Eric R. Chan, and Gordon Wetzstein. 3D GAN inversion for controllable portrait image animation. In *ECCV Workshops*, 2022. 2

[32] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: A large-scale speaker identification dataset. In *INTERSPEECH*, 2017. 6, 7

[33] 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

[34] Gaurav Parmar, Yijun Li, Jingwan Lu, Richard Zhang, Jun-Yan Zhu, and Krishna Kumar Singh. Spatially-adaptive multilayer selection for c inversion and editing. In *CVPR*, 2022. 2

[35] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleClip: Text-driven manipulation of StyleGAN imagery. In *ICCV*, 2021. 2

[36] 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

[37] Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ToG*, 2022. 2, 3, 4

[38] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. VoxGRAF: Fast 3D-aware image synthesis with sparse voxel grids. In *NeurIPS*, 2022. 2

[39] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *CVPR*, 2020. 2

[40] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: Interpreting the disentangled face representation learned by GANs. *TPAMI*, 2020. 2, 5, 7

[41] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in GANs. In *CVPR*, 2021. 2

[42] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. StyleGAN-V a continuous video generator with the price, image quality and perks of StyleGAN2. In *CVPR*, 2022. 2

[43] 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. *ToG*, 2022. 2

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

[45] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. StyleRig: Rigging StyleGAN for 3D control over portrait images. In *CVPR*, 2020. 2

[46] 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

[47] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. In *SIGGRAPH*, 2021. 2

[48] Rotem Tzaban, Ron Mokady, Rinon Gal, Amit H. Bermano, and Daniel Cohen-Or. Stitch it in time: GAN-Based facial editing of real videos. In *SIGGRAPH Asia*, 2022. 2, 3, 5, 6

[49] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *ICML*, 2020. 2

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

[51] Weihao Xia and Jing-Hao Xue. A survey on 3D-aware image synthesis. *arXiv preprint arXiv:2210.14267*, 2022. 2

[52] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN inversion: A survey. *TPAMI*, 2022. 2

[53] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN inversion: A survey. *TPAMI*, 2022. 1

[54] Yiran Xu, Badour AlBahar, and Jia-Bin Huang. Temporally consistent semantic video editing. In *ECCV*, 2022. 3, 6

[55] 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

[56] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 2018. 4, 5

[57] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *ICLR*, 2022. 2

[58] Xiaoming Zhao, Fangchang Ma, David Güera, Zhile Ren, Alexander G. Schwing, and Alex Colburn. Generative multiplane images: Making a 2D GAN 3D-aware. In *ECCV*, 2022. 2

[59] Peiye Zhuang, Oluwasanmi Koyejo, and Alexander G. Schwing. Enjoy Your Editing: Controllable GANs for Image Editing via Latent Space Navigation. In *ICLR*, 2021. 2