International Journal of Computer Vision
https://doi.org/10.1007/s11263-025-02496-2

E^3DGE: Self-Supervised Geometry-Aware Encoder for Style-Based 3D GAN Inversion

Yushi Lan[1] · Xuyi Meng[2] · Shuai Yang[3] · Chen Change Loy[1] · Bo Dai[4]

Received: 5 June 2024 / Accepted: 3 June 2025

Abstract

StyleGAN has excelled in 2D face reconstruction and semantic editing, but the extension to 3D lacks a generic inversion framework, limiting its applications in 3D reconstruction. In this paper, we address the challenge of 3D GAN inversion, focusing on predicting a latent code from a single 2D image to faithfully recover 3D shapes and textures. The inherent ill-posed nature of the problem, coupled with the limited capacity of global latent codes, presents significant challenges. To overcome these challenges, we introduce an efficient self-training scheme that does not rely on real-world 2D-3D pairs but instead utilizes proxy samples generated from a 3D GAN. Additionally, our approach goes beyond the global latent code by enhancing the generation network with a local branch. This branch incorporates pixel-aligned features to accurately reconstruct texture details. Furthermore, we introduce a novel pipeline for 3D view-consistent editing. The efficacy of our method is validated on two representative 3D GANs, namely StyleSDF and EG3D. Through extensive experiments, we demonstrate that our approach consistently outperforms state-of-the-art inversion methods, delivering superior quality in both shape and texture reconstruction.

1 Introduction

This work aims to devise an effective and generic approach for encoder-based 3D Generative Adversarial Network (GAN) inversion. In particular, we focus on the reconstruction of 3D faces, requiring just a single 2D face image as the input. In the inversion process, we wish to map a given image to the latent space and obtain an editable latent code with an encoder. The latent code will be further fed to a generator to reconstruct the corresponding 3D shape with high-quality shape and texture. Besides inversion, we aim to further develop an approach to synthesize 3D view-consistent editing results, e.g., changing a neutral expression to smiling, by altering the estimated latent code.

GAN inversion (Xia et al., 2022) has been extensively studied for 2D images but remains underexplored in the 3D world. Inversion can be achieved via optimization (Abdal et al., 2019, 2020; Roich et al., 2021), which typically provides a precise image-to-latent mapping but can be time-consuming, or encoder-based techniques (Richardson et al., 2021; Wang et al., 2022; Tov et al., 2021), which explicitly learn an encoding network that maps an image into the latent space. Encoder-based techniques enjoy faster inversion, but the mapping is typically inferior to optimization. In this study, we extend the notion of encoder-based inversion from 2D images to 3D shapes.

Increasing the additional dimension makes inversion more challenging beyond the goal of reconstructing an editable shape with detail preservation. In particular,

Communicated by Bolei Zhou.

✉ Chen Change Loy
ccloy@ntu.edu.sg

Yushi Lan
yushi001@e.ntu.edu.sg

Xuyi Meng
mengxuyi@seas.upenn.edu

Shuai Yang
williamyang@pku.edu.cn

Bo Dai
doubledaibo@gmail.com

[1] S-Lab, Nanyang Technological University, Singapore, Singapore

[2] University of Pennsylvania, Pennsylvania, USA

[3] Wangxuan Institute of Computer Technology, Peking University, Beijing, China

[4] Shanghai AI Laboratory, Shanghai, China

Published online: 14 June 2025

1) Recovering 3D shapes from 2D images is an ill-posed problem, where innumerable compositions of shape and texture could generate identical rendering results. 3D supervisions are crucial to alleviate the ambiguity of shape inversion from images. Though high-quality 2D datasets are easily accessible, owing to the expensive cost of scans, there is currently a lack of large-scale labeled 3D datasets.
2) The global latent code, due to its compact and low-dimensional nature, only captures the coarse shape and texture information. Without high-frequency spatial details, we cannot generate high-fidelity outputs.
3) Compared with 2D inversion methods where the editing view mostly aligns with the *input view*, in 3D editing we expect the editing results to perform well over the *novel views* with large pose variations. Therefore, 3D GAN inversion is a non-trivial task and cannot be achieved by directly applying existing approaches.

To this end, we propose a novel **E**ncoder-based **3D G**AN inv**E**rsion framework, E^3DGE, which addresses the aforementioned three challenges. Our framework has three novel components with a delicate model design. Specifically:

Learning Inversion with Self-supervised Learning - The first component focuses on the training of the inversion encoder. To address the shape collapse of single-view 3D reconstruction without external 3D datasets, we retrofit the generator of a 3D GAN model to provide us with diverse pseudo-training samples, which can then be used to train our inversion encoder in a self-supervised manner. Specifically, we generate 3D shapes from the latent space $\mathcal{W}$ of a 3D GAN, and then render diverse 2D views from each 3D shape given different camera poses. In this way, we can generate many pseudo-2D-3D pairs together with the corresponding latent codes. Since the pseudo pairs are generated from a smooth latent space that learns to approximate a natural shape manifold, they serve as effective surrogate data to train the encoder, avoiding potential shape collapse.

Local Features for High-Fidelity Inversion - The second component learns to reconstruct accurate texture details. Our novelty here is to leverage local features to enhance the representation capacity, beyond just the global latent code generated by the inversion encoder. Specifically, in addition to inferring an editable global latent code to represent the overall shape of the face, we further devise an hour-glass model to extract local features over the residuals details that the global latent code fails to capture. The local features, with proper projection to the 3D space, serve as conditions to modulate the 2D image rendering. Through this effective learning scheme, we marry the benefits of both global and local priors and achieve high-fidelity reconstruction.

Synthesizing View-consistent Edited Output - The third component addresses the problem of novel view synthesis, a problem unique to 3D shape editing. Specifically, though we achieve high-fidelity reconstruction through the aforementioned designs, the local residual features may not fully align with the scene when being semantically edited. Moreover, the occlusion issue further degrades the fusion performance when rendering from novel views with large pose variations. To this end, we propose a 2D-3D hybrid alignment module for high-quality editing. Specifically, a 2D alignment module and a 3D projection scheme are introduced to jointly align the local features with edited images and inpaint occluded local features in novel view synthesis.

Extensive experiments show that our method achieves 3D GAN inversion with plausible shapes and high-fidelity image reconstruction without affecting editability. Owing to the self-supervised training strategy with delicate global-local design, our approach performs well on real-world 2D and 3D benchmarks without resorting to any real-world 3D dataset for training.

Compared with our previous work (Lan et al., 2023) that mainly studies 3D GAN inversion on MLP-based StyleSDF, we renovate E^3DGE as a generic 3D GAN inversion framework with the following key improvements:

1) *Representation-wise*, we explore 3D inversion over tri-plane-based GAN, characterized by EG3D (Chan et al., 2022a), the state-of-the-art GAN-based 3D generative model.
2) *Modality-wise*, we adapt our proposed method over video input, facilitating 4D monocular reconstruction.
3) *Application-wise*, we include a new practical application of our method: 3D toonification.

Since EG3D adopts a new 3D representation, it poses new challenges for 3D GAN inversion: how to design an efficient and high-fidelity encoder-based inversion pipeline that leverages a triplane-based method. An observation in our experiment is that, though with superior representation capacity, the tri-plane is more sensitive to inaccurate pose during inference. To tackle this challenge, we propose the following component for EG3D-based E^3DGE:

Pose estimation for domain adaptation - This newly proposed component addresses the problem of noisy poses of 3D GAN inversion over real images, which EG3D is especially sensitive to. Specifically, though the aforementioned designs could achieve high-quality inversion, we rely on the assumption that an accurate input pose is available at test time. However, though we directly leverage pseudo samples with ground truth pose for training, the real inputs are inclined to pair with noisy pose labels, which intensifies the domain gap between training and test. To bridge the gap, we first propose to use predicted pose during training with ground-truth poses supervising the pose estimator. Besides, we provide the option to finetune the predicted pose to boost

the performance. In addition, comprehensive experiments are conducted to evaluate the new designs of the extended EG3D-E^3DGE, including qualitative and quantitative evaluations on the introduced pose estimator, hybrid fine-tuning, and the comparison results with the corresponding baselines. New applications and inversion on more categories are also included.

To summarize, our main contributions are as follows:

- We propose the learning of an encoder-based 3D GAN inversion framework for high-quality shape and texture inversion. We show that with careful design, samples synthesized by a GAN could serve as proxy data for self-supervised training in inversion.
- We present an effective framework that uses local features to complement the global latent code for high-fidelity inversion.
- We propose an effective approach to synthesize view-consistent output with a 2D-3D hybrid alignment.
- We propose to jointly train a pose estimator to address the pose domain gap during inversion.
- We validate our method on two representative 3D GAN models, showing its generalizability to different backbones.

The rest of this paper is organized as follows. In Section 2, we review related studies in 3D-aware image synthesis, GAN-supervised training, 2D GAN inversion, and 3D reconstruction and editing with 3D GANs. Section 3 introduces the preliminary 3D representation background of the proposed method. In Section 4, the details of the proposed self-supervised 3D GAN inversion method, E^3DGE, are introduced. Section 5 further discussed applying E^3DGE over the StyleSDF backbone, which is used in our conference version submission. Section 6 validates the superiority of our method via extensive experiments and comparisons with state-of-the-art encoder-based 3D GAN inversion methods. Finally, we conclude our work in Section 7.

2 Related Work

3D-aware Image Synthesis Generative Adversarial Network (Goodfellow et al., 2014) has shown promising results in generating photorealistic images (Karras et al., 2019; Brock et al., 2019; Karras et al., 2020a) and inspired researchers to put efforts on 3D aware generation (Nguyen-Phuoc et al., 2019; Henzler et al., 2019; Pan et al., 2021a). However, these methods use explicit shape representations, i.e., voxels (Nguyen-Phuoc et al., 2019; Henzler et al., 2019) and meshes (Pan et al., 2021a) as the intermediate shape models, which lacks photorealism and is memory-inefficient. Motivated by the recent success of neural rendering (Park et al., 2019; Mescheder et al., 2019; Mildenhall et al., 2020), researchers shift to implicit function along with the volume rendering process as the incorporated 3D inductive bias. Especially, NeRF (Mildenhall et al., 2020) proposed an implicit 3D representation for novel view synthesis which defines a scene as $\{\boldsymbol{c}, \sigma\} = F_\Phi(\boldsymbol{x}, \boldsymbol{v})$, where $\boldsymbol{x}$ is the query point, $\boldsymbol{v}$ is the viewing direction from camera origin to $\boldsymbol{x}$, $\boldsymbol{c}$ is the emitted color and σ is the volume density. Researchers further extend NeRF to generation task (Chan et al., 2021; Schwarz et al., 2020) and show impressive 3D-awareness synthesis. To further increase the generation resolution, recent works (Niemeyer & Geiger, 2021; Or-El et al., 2021; Chan et al., 2022a; Gu et al., 2021; Hong et al., 2022) resort to the StyleGAN-based architecture on the 3D generation tasks with a hybrid design. By lifting the intermediate low-resolution 2D features to high resolution with a 2D super-resolution decoder, the hybrid design achieves high resolution of 1024^2. Two of the canonical works, StyleSDF (Or-El et al., 2021) and EG3D (Chan et al., 2022a), stand out as the state-of-the-art 3D GANs using SDF (Signed Distance Field)- and triplane-based 3D representations correspondingly. We select these two representative methods as the backbone for the 3D GAN inversion study.

GAN-supervised Training Previous works (Besnier et al., 2020; Pan et al., 2022; Jahanian et al., 2020, 2022; Yang et al., 2022; Zhang et al., 2021a; Ling et al., 2021) propose to use pretrained GAN to generate training dataset. Through careful design in the sampling strategy (Jahanian et al., 2022), loss functions (Pan et al., 2022) and generation process (Zhang et al., 2021a), researches show that off-the-shelf image generators could facilitate a series of downstream visual applications.

2D GAN Inversion Optimization-based 2D GAN inversion methods (Abdal et al., 2019; Fu et al., 2022) achieve photorealistic reconstruction at the cost of slow inference and lack of editability. To speed up, Encoder-based methods (Richardson et al., 2021; Wang et al., 2022; Tov et al., 2021; Chan et al., 2022b; Zhu et al., 2020) like pSp (Richardson et al., 2021) and e4e (Tov et al., 2021) have been developed and show better properties in editing through specific model design (Richardson et al., 2021; Wang et al., 2022) and training strategies (Tov et al., 2021). However, they (Richardson et al., 2021; Tov et al., 2021; Zhu et al., 2020; Abdal et al., 2019, 2020) all adopt global latent code alone for GAN inversion task, thus failing to recover high-fidelity details. Recently, HFGI (Wang et al., 2022) introduce an extra spatial consultation map to mitigate this issue, though still designed to restore 2D textures without considering 3D shape modeling. In this work, we propose a delicate design that exploits local features to recover texture details and achieves view-consistent synthesis.

3D Reconstruction and Editing with 3D GANs Recent development of 3D GANs (Schwarz et al., 2020; Niemeyer &

Geiger, 2021; Chan et al., 2021; Or-El et al., 2021; Chan et al., 2022a; Gu et al., 2021) also calls for corresponding inversion frameworks. π-GAN and EG3D (Chan et al., 2022a) directly adopt 2D inversion method (Abdal et al., 2019; Roich et al., 2021), which requires expensive latent or model optimization and still introduces implausible shape artifacts. The most relevant work to ours is Lin *et al.* (Lin et al., 2022), which employs a computationally expensive optimization-based framework (Abdal et al., 2019) and combines FLAME (Feng et al., 2021; Li et al., 2017) for portrait animation. However, it fails to guarantee reasonable shape and is limited to the human face domain. In parallel, Pix2NeRF (Cai et al., 2022) introduces a feed-forward network to pre-trained π-GAN and enables single-view 3D reconstruction. However, it does not demonstrate its performance in high-quality novel view editing. Pavllo *et al.* (Pavllo et al., 2023) proposes a hybrid 3D GAN inversion framework for single-image shape reconstruction. However, the hybrid framework trades off the performance with speed, while our method achieves high-quality inversion and editing performance with fast inference speed. Some other works rely on 3D parametric models (Feng et al., 2021) or auto-decoder (Rebain et al., 2022) architecture for single-view 3D reconstruction (Feng et al., 2021) or editing (Rebain et al., 2022), which cannot leverage the strong GAN priors for high-resolution and flexible latent-based editing.

Concurrently, SPI (Yin et al., 2022) and HFGI3D (Xie et al., 2022) also proposed to conduct 3D GAN inversion given monocular inputs. However, their encoder-free design suffers from costly per-instance optimization. Live3DPortrait (Trevithick et al., 2023) also proposed an encoder-based 3D portrait reconstruction model by distilling a pre-trained EG3D (Chan et al., 2022a) generator. However, it is designed for reconstruction only and does not yield compact latent code with editing capability. Our method, however, enables this with the help of hybrid alignment, as introduced in Sec. 4.3.

GOAE (Yuan et al., 2023) proposed a two-stage 3D GAN inversion framework. However, it only demonstrates its ability on tri-plane and image inversion, while our method is verified on both SDF and tri-plane 3D representation over both image and video.

The most similar work to ours is GRAMInverter (Deng et al., 2022b), which also presents a detail-preserved 3D portrait inversion framework, leveraging the pretrained GRAM (Deng et al., 2021). However, GRAMInverter does not support latent code-based editing as in 2D GAN, which is enabled by our hybrid alignment module mentioned in Sec. 4.3. Furthermore, our method supports both SDF and NeRF-based 3D GAN framework. Besides, GRAMInverter's global-local inversion design is also included in our Sec. 4.1 and Sec. 4.2. In summary, our proposed method is a superset for GRAMInverter.

Dynamic 3D Head Avatars for Pose and Expression Editing Beyond the progress in static avatar head generation, a parallel line of work focuses on extending 3D avatar reconstruction to video for pose and expression editing. GOHA (Li et al., 2023) proposes a 3DMM-conditioned framework that lifts 2D features to 3D neural points. GPAvatar (Chu et al., 2024) and Portrait4D (Deng et al., 2024a, b), in contrast, explicitly inject FLAME (Li et al., 2017) geometry into the generative pipeline. Note that this line of work is specifically designed for 3D avatars and requires 3DMM (Blanz & Vetter, 1999) integration. Our method, however, is a generic 3D GAN inversion framework that can be applied to generic 3D datasets like ShapeNet. Besides, beyond 3D/4D reconstruction, our method also yields a compact latent code that supports semantic editing.

3 Preliminaries

Since recent 3D-aware image generative models are based on neural implicit representations, especially NeRF (Mildenhall et al., 2020), here we briefly introduce the NeRF-based 3D representation and also hybrid 3D-aware generation based on EG3D/StyleSDF for clarification.

NeRF-based 3D Representation NeRF (Mildenhall et al., 2020) proposed an implicit 3D representation for novel view synthesis. Specifically, NeRF defines a scene as $\{\boldsymbol{c}, \sigma\} = F_\Phi(\boldsymbol{x}, \boldsymbol{v})$, where $\boldsymbol{x}$ is the query point, $\boldsymbol{v}$ is the viewing direction from camera origin to $\boldsymbol{x}$, $\boldsymbol{c}$ is the emitted radiance (RGB value), σ is the volume density. To query the RGB value $C(\boldsymbol{r})$ of a point on a ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{v}$ shoot from the 3D coordinate origin $\boldsymbol{o}$, we have the volume rendering formulation,

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{v})dt, \tag{1}$$

where $T(t) = \exp(-\int_{t_n}^{t} \sigma(\boldsymbol{r}(s))ds)$ is the accumulated transmittance along the ray $\boldsymbol{r}$ from t_n to t. t_n and t_f denote the near and far bounds.

Hybrid 3D-aware Generation To achieve high-resolution novel view synthesis, hybrid 3D-aware generators (Niemeyer & Geiger, 2021; Gu et al., 2021; Chan et al., 2022a; Or-El et al., 2021) are proposed. It is typically a cascade model $G = G_\mathbf{1} \circ G_\mathbf{0}$ composed of a NeRF-based renderer $G_\mathbf{0}$ (Chan et al., 2021) and a 2D super-resolution network $G_\mathbf{1}$, as shown in Fig. 1. Both G_0 and G_1 follow the style-based architecture (Karras et al., 2019, 2020b) to accept a latent code $\mathbf{w}$ to control the style of the generated object. During generation, G_0 captures the underlying geometry with the full control of $\mathbf{w}$ and camera pose $\boldsymbol{\xi}$, and renders a low-resolution image $\mathbf{I_0}$ and an intermediate feature map $\mathbf{F}$. Then, $G_\mathbf{1}$ further upsamples $\mathbf{F}$ to obtain a high-resolution image $\mathbf{I}$ with added high-frequency details. Among them, StyleSDF (Or-

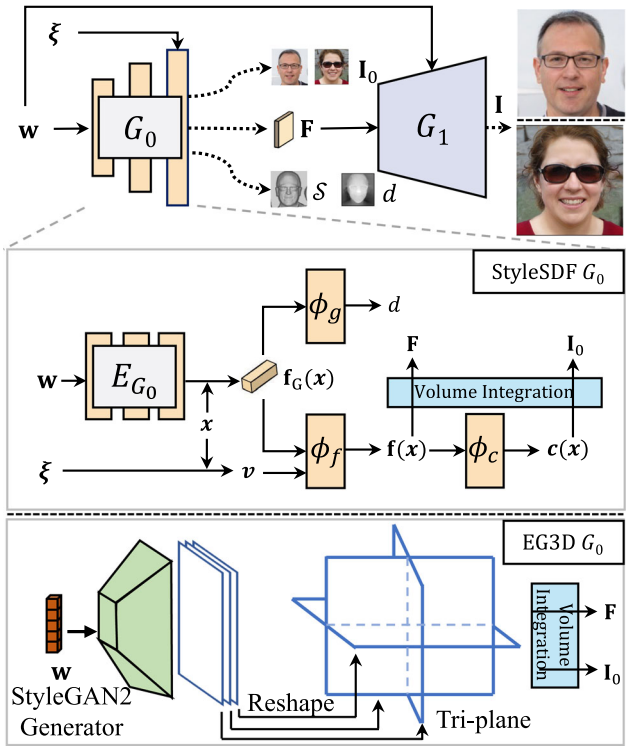

Fig. 1 StyleSDF and EG3D. Given a sampled latent code $\mathbf{w}$ and a camera pose $\boldsymbol{\xi}$, StyleSDF (Or-El et al., 2021) generates object SDF d to depict the shape with the corresponding image $\mathbf{I}$, while EG3D (Chan et al., 2022a) adopts density σ as the shape descriptor. Both methods adopt a hybrid synthesis pipeline, where a low-resolution image $\mathbf{I}_0$ is first synthesized and further up-sampled to high-res image $\mathbf{I}$

El et al., 2021) adopts NeRF-based MLP as G_0 for 3D-aware high-quality surface synthesis. In comparison, EG3D (Chan et al., 2022a) introduces axis-aligned plane (Peng et al., 2020) as G_0 and achieves state of the art performance on several benchmarks with faster rendering efficiency and sharper geometry details compared against previous work (Chan et al., 2021) and StyleSDF. EG3D also enjoys the flexible style control for semantic editing as in StyleGAN (Karras et al., 2019). Therefore, in this work, we mainly explore EG3D as the base model for the GAN inversion study (Section 4). Our method is not limited to EG3D and could be easily extended to other style-based 3D GAN variations (Or-El et al., 2021; Gu et al., 2021). Please refer to Sec. 5 and the appendix for the technical details of the StyleSDF-based E^3DGE study.

4 E^3DGE with EG3D Backbone

An effective 3D GAN inversion should be capable of **1)** reconstructing plausible 3D shape given single-view input, **2)** maintaining high-fidelity texture, and **3)** allowing view-consistent semantic edits. To achieve these goals, we propose the E^3DGE framework with three novel components: In Sec. 4.1, we leverage 3D GAN to generate pseudo 2D-3D paired samples for 3D supervisions, and train an inversion encoder E_0 to estimate the latent of plausible 3D shapes from a 2D image; In Sec. 4.2, we train a local encoder E_1 to extract pixel-aligned features to enrich texture details for high-fidelity inversion; In Sec. 4.3 introduces a hybrid alignment module for view-consistent semantic editing; Finally, in Sec. 4.4 we propose to jointly estimate the input camera pose and fine-tune the estimated code to alleviate the domain gap for better performance on the real-world inversion. We also include a notation table in the Tab. 1 of appendix.

4.1 Self-supervised Inversion Learning

In this section, we propose to mitigate the lack of large-scale high-quality 2D-3D paired datasets by retrofitting pre-trained 3D GANs to provide pseudo samples for training our inversion encoder. We demonstrate the model trained from pseudo samples can rival and even outperform the methods learned from real data on the 3D GAN inversion task. We detail the process as follows.

Global Encoder for 3D GAN Inversion With the style-based G, we build our encoder E_0 based on pSp (Richardson et al., 2021) for inversion. Given a target image $\mathbf{I}$, E_0 predicts its latent code $\hat{\mathbf{w}} = E_0(\mathbf{I})$. Given the corresponding camera pose $\boldsymbol{\xi}$, the reconstructed image is obtained by $\tilde{\mathbf{I}} = G(\hat{\mathbf{w}}, \boldsymbol{\xi})$ to approximate $\mathbf{I}$. In addition, we would like its 3D shape predicted by G_0 to be plausible enough.

Distill 3D GANs as 3D Supervisions Different compositions of shape and texture could lead to identical 2D-rendered images. 3D supervision is needed to alleviate such shape-texture ambiguity. In the lack of large-scale high-quality 2D-3D paired samples, we formulate GAN Inversion as a *self-training* task, where samples synthesized from itself are leveraged to boost the reconstruction fidelity in both 2D and 3D domains. As shown in Fig. 1 and Fig. 2, we synthesize paired 3D shape information $\mathcal{S}$ and 2D image $\mathbf{I}$ from latent code $\mathbf{w}$ and camera pose $\boldsymbol{\xi}$ using G to train E_0. To extract the 3D shape information $\mathcal{S}$ of each synthetic shape, we first sample a point set $\mathcal{P} = \{\mathcal{P}_\mathcal{O}, \mathcal{P}_\mathcal{F}\}$ where $\mathcal{P}_\mathcal{O}$ and $\mathcal{P}_\mathcal{F}$ contain points sampled from the surface and around the surface, respectively. Then, we calculate the geometry descriptor $\boldsymbol{\sigma}_i$ and $\mathbf{D}_i$ for each 3D point $\boldsymbol{x}_i \in \mathcal{P}$, and $\mathcal{S}$ is defined as the set of geometry descriptors of all 3D point in $\mathcal{P}$:

$$\begin{aligned} \mathcal{S} = \{\{\boldsymbol{\sigma}_i, \mathbf{D}_i\}_{i=1}^{|\mathcal{P}|} \mid \\ \boldsymbol{x}_i \in \mathcal{P}, \boldsymbol{\sigma}_i = G_0(\mathbf{w}, \boldsymbol{x}_i), \mathbf{D}_i = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))dt\}, \end{aligned} \tag{2}$$

where $\boldsymbol{\sigma}_i$ is the density of point $\boldsymbol{x}_i$, $\mathbf{D}_i$ is the depth of point $\boldsymbol{x}_i$ under the given view direction $\boldsymbol{\xi}$ and σ is the density of the point along the ray. Note our method is not limited

to the triplane-based shape representation and can be easily extended to SDF-based methods, as detailed in Sec. 5. Moreover, given different camera poses, we can generate a diverse 2D-3D dataset to help alleviate the shape-texture ambiguity, i.e., for each shape $\mathcal{S}$, various images $\mathbf{I} = G(\mathbf{w}, \boldsymbol{\xi})$ can be rendered by randomly sampling $\boldsymbol{\xi}$ from a predefined pose distribution $p_{\boldsymbol{\xi}}$. Finally, we define $\mathcal{X} = \{\mathcal{S}, \boldsymbol{\xi}, \mathbf{I}\}$ as a training sample for $E_{\mathbf{0}}$.

3D GAN-Supervised Training As shown in Fig. 2 (**a**), given a training sample $\mathcal{X}$, the forward process is represented as:

$$\begin{aligned} \hat{\mathbf{w}} &= E_{\mathbf{0}}(\mathbf{I}) \\ \{\tilde{\mathbf{I}}, \hat{\mathcal{S}}\} &= G(\hat{\mathbf{w}}, \boldsymbol{\xi}, \mathcal{P}) \end{aligned} \tag{3}$$

where $\hat{\mathbf{w}}$ is the estimated latent code and $\hat{\mathcal{S}} = \{\{\hat{\boldsymbol{\sigma}}_i, \hat{\mathbf{D}}_i\}_{i=1}^{|\mathcal{P}|} \mid \boldsymbol{x}_i \in \mathcal{P}\}$ is the estimated 3D shape information conditioned on $\tilde{\mathbf{w}}$ and $\mathcal{P}$.

To achieve 3D supervision, for all points we would like the estimated $\hat{\mathcal{S}}$ to approximate the ground truth $\mathcal{S}$. Specifically, we supervise both the predicted density $\hat{\boldsymbol{\sigma}}$ as well as the depth $\hat{\mathbf{D}}$ (Deng et al., 2022a) for surface points $\boldsymbol{x}_i \in \mathcal{P}_{\mathcal{O}}$ and supervise the predicted density for the remaining points $\boldsymbol{x}_i \in \mathcal{P}_{\mathcal{F}}$, leading to the geometry loss:

$$\mathcal{L}_{geo}^{\mathcal{O}} = \mathbb{E}_{\mathcal{X}}\left[\frac{1}{|\mathcal{P}_{\mathcal{O}}|}\sum_{i=1}^{|\mathcal{P}_{\mathcal{O}}|} \lambda_{g1}|\hat{\boldsymbol{\sigma}}_i - \boldsymbol{\sigma}_i| + \lambda_{g2}\|\hat{\mathbf{D}}_i - \mathbf{D}_i\|_1\right] \tag{4}$$

$$\mathcal{L}_{geo}^{\mathcal{F}} = \mathbb{E}_{\mathcal{X}}\left[\frac{1}{|\mathcal{P}_{\mathcal{F}}|}\sum_{i=1}^{|\mathcal{P}_{\mathcal{F}}|} \lambda_{g3}|\hat{\boldsymbol{\sigma}}_i - \boldsymbol{\sigma}_i|\right] \tag{5}$$

$$\mathcal{L}_{geo} = \mathcal{L}_{geo}^{\mathcal{O}} + \mathcal{L}_{geo}^{\mathcal{F}}, \tag{6}$$

where λs are loss weights and supervision of depth $\hat{\mathbf{D}}_i$ is only imposed for points over the surface. We also impose code reconstruction loss $\mathcal{L}_{code} = \|\hat{\mathbf{w}} - \mathbf{w}\|_2$ to regularize the learning and 2D supervisions $\mathcal{L}_{rec}$ to minimize the reconstruction error between $\tilde{\mathbf{I}}$ and $\mathbf{I}$ as in pSp (Richardson et al., 2021). The overall loss is $\mathcal{L} = \mathcal{L}_{\text{geo}} + \mathcal{L}_{\text{code}} + \mathcal{L}_{\text{rec}}$, which is detailed in Sec. 6.

4.2 Local Features for High-Fidelity Inversion

To facilitate the discussion in the following sections, we first take a look at the details of EG3D. The unique design of EG3D lies in its 3D renderer $G_{\mathbf{0}}$: after a StyleGAN2 generator, the last layer of the synthesized feature maps are reshaped into three orthogonal planes, e.g., tri-plane, where the queried features over three planes are volume rendered into feature map $\mathbf{F}$ and view-consistent image $\mathbf{I}_0$. Specifically, $E_{G_{\mathbf{0}}}$ extracts a global feature $\mathbf{f}_{\text{G}}(\boldsymbol{x}) = E_{G_{\mathbf{0}}}(\boldsymbol{x}, \mathbf{w})$. Based on $\mathbf{f}_{\text{G}}$, ϕ_g and ϕ_f compute density $\sigma(\boldsymbol{x}) = \phi_g(\mathbf{f}_{\text{G}}(\boldsymbol{x}))$ and the last-layer feature $\mathbf{f}(\boldsymbol{x}, \boldsymbol{v}) = \phi_f(\mathbf{f}_{\text{G}}(\boldsymbol{x}), \boldsymbol{v})$ of $G_{\mathbf{0}}$, respectively. $\mathbf{f}$ could be directly transformed to color $\boldsymbol{c}(\boldsymbol{x}, \boldsymbol{v}) = \phi_c(\mathbf{f}(\boldsymbol{x}, \boldsymbol{v}))$ or being volume integrated to $\mathbf{F}$ and sent to $G_{\mathbf{1}}$ for high resolution synthesis. For simplicity, we omit $\boldsymbol{v}$ in the following. The volume rendering process is the same as StyleSDF, which is depicted in the middle of Fig. 1. The main difference of EG3D-based representation is that $\mathbf{f}_{\text{G}}$ is decoded from tri-plane, while StyleSDF decodes the global feature from a stack of MLP layers.

Local Feature for Detailed Textures The global latent code $\hat{\mathbf{w}}$ is a compact representation of the predicted scene. How-

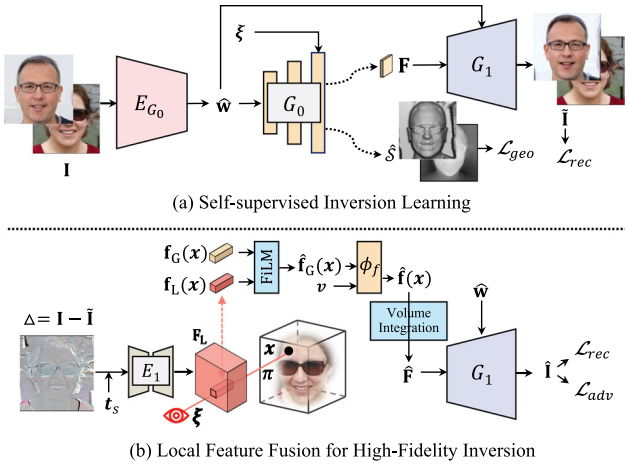

Fig. 2 E^3DGE **for 3D GAN inversion**. (**a**) We augment the training of the encoder $E_{\mathbf{0}}$ with 3D supervision $\mathcal{L}_{geo}$ for plausible 3D shape prediction. (**b**) We augment the representation capacity of the global latent code $\hat{\mathbf{w}}$ with local point-dependent latent feature $\mathbf{f}_{\text{L}}$ for high-fidelity texture reconstruction. Both StyleSDF and EG3D samples are shown here

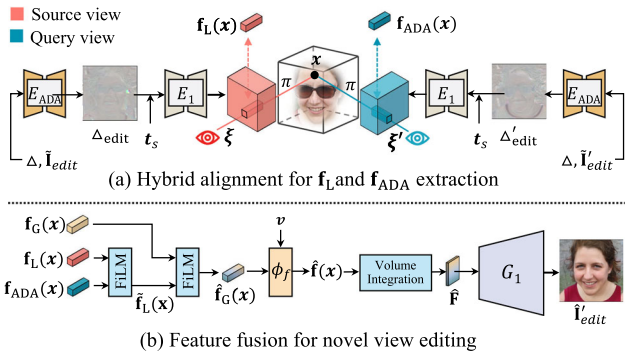

Fig. 3 **Hybrid alignment for high-quality editing.** Given code prediction $\hat{\mathbf{w}}$ from encoder $E_{\mathbf{0}}$ pre-trained in stage-I, we aim to generate high-quality view synthesis over the edited code $\hat{\mathbf{w}}_{\text{edit}}$. In (**a**), the local details Δ along with the target edited image $\mathbf{I}'_{\text{edit}}$ and depth map $\boldsymbol{t}_s(\hat{\mathbf{w}}, \boldsymbol{\xi})$ are sent to pre-trained E_{ADA} to predict aligned residual Δ'_{edit}. The original aligned residual Δ along with the 2D auxiliary residual Δ'_{edit} are processed by $E_{\mathbf{1}}$ to recover latent maps $\mathbf{F}_{\text{L}}$ and $\mathbf{F}_{\text{ADA}}$ for later fusion. In (**b**), the extracted features $\mathbf{f}_{\text{L}}(\boldsymbol{x})$ and $\mathbf{f}_{\text{ADA}}(\boldsymbol{x})$ are first fused together with a FiLM layer, and the fused result $\hat{\mathbf{f}}_{\text{L}}(\boldsymbol{x})$ further serves as conditions to modulate the global feature $\mathbf{f}_{\text{G}}(\boldsymbol{x})$. The final modulated feature $\hat{\mathbf{f}}(\boldsymbol{x})$ contains complete information, globally and locally. The volume integrated $\hat{\mathbf{F}}$ is sent to $G_{\mathbf{1}}$ for high-resolution synthesis

ever, previous works (Chan et al., 2022b; Wang et al., 2022) have validated that a low-dimensional latent code discards high-frequency spatial details and fails to reconstruct high-fidelity outputs. This phenomenon becomes more severe when lifting the 2D image to a 3D scene, which contains exponentially more information. Inspired by recent progress in few-shot 3D reconstruction (Saito et al., 2019, 2020; Yu et al., 2021; Xiu et al., 2022; Alldieck et al., 2022; Wang et al., 2021; Chibane et al., 2021), we propose to make up for the lost information by introducing pixel-aligned (local) features. As shown in Fig. 2 (b), rather than conditioning all 3D points with the same latent code $\hat{\mathbf{w}}$, we augment the representation capacity with local latent codes $\mathbf{f}_{\mathrm{L}}$ that is dependent on each point $\boldsymbol{x}$. We introduce a local hourglass (Newell et al., 2016) encoder $E_{\mathbf{1}}$ to predict a residual feature map $\mathbf{F}_{\mathrm{L}}$ based on the reconstruction residue $\Delta = \mathbf{I} - \tilde{\mathbf{I}}$,

$$\mathbf{F}_{\mathrm{L}} = E_{\mathbf{1}}(\Delta, t_s(\hat{\mathbf{w}}, \boldsymbol{\xi})), \tag{7}$$

where $t_s(\hat{\mathbf{w}}, \boldsymbol{\xi})$ is the depth map of the scene derived from the predicted density σ to serve as 3D context information. Then, the local latent code of a point $\boldsymbol{x}$ is its corresponding value in $\mathbf{F}_{\mathrm{L}}$:

$$\mathbf{f}_{\mathrm{L}}(\boldsymbol{x}) = \mathbf{F}_{\mathrm{L}}(\pi(\boldsymbol{x})) \oplus \mathbf{PE}(\boldsymbol{x}), \tag{8}$$

where π maps the 3D point $\boldsymbol{x}$ to its corresponding pixel coordinate on 2D feature map $\mathbf{F}_{\mathrm{L}}$. Since in 3D scenes, points along a ray will be projected to the same coordinate on the 2D plane, to differentiate these points, we additionally concatenate their positional encoding $\mathbf{PE}(\boldsymbol{x})$ (Mildenhall et al., 2020) in Eq. (8). In this way, the local feature $\mathbf{f}_{\mathrm{L}}$ only encodes the residual information at the projected position $\pi(\boldsymbol{x})$ but is also capable of determining where the residual information lies in the 3D scene, as well as inpainting the occluded areas along the ray.

Finally, we fuse the local latent code $\mathbf{f}_{\mathrm{L}}(\boldsymbol{x})$ with the global latent code $\mathbf{f}_{\mathrm{G}}(\boldsymbol{x}) = E_{G_{\mathbf{0}}}(\boldsymbol{x}, \hat{\mathbf{w}})$ to supplement the missing high-frequency details. Specifically, the feature fusion is based on Feature-wise Linear Modulation (FiLM) (Perez et al., 2018). As shown in Fig. 2, $\mathbf{f}_{\mathrm{L}}(\boldsymbol{x})$ is fed into two MLP layers to obtain the scale and bias modulation parameters $\mathbf{f}_{\mathrm{L}}^{\boldsymbol{\gamma}}(\boldsymbol{x})$ and $\mathbf{f}_{\mathrm{L}}^{\boldsymbol{\beta}}(\boldsymbol{x})$. Then we modulate $\mathbf{f}_{\mathrm{G}}(\boldsymbol{x})$ with FiLM

$$\begin{aligned}\hat{\mathbf{f}}_{\mathrm{G}}(\boldsymbol{x}) =& \mathrm{FiLM}(\mathbf{f}_{\mathrm{G}}(\boldsymbol{x}), \mathbf{f}_{\mathrm{L}}(\boldsymbol{x})) \\ =& \mathbf{f}_{\mathrm{L}}^{\boldsymbol{\gamma}}(\boldsymbol{x}) \cdot \mathbf{f}_{\mathrm{G}}(\boldsymbol{x}) + \mathbf{f}_{\mathrm{L}}^{\boldsymbol{\beta}}(\boldsymbol{x}).\end{aligned} \tag{9}$$

The fused $\hat{\mathbf{f}}_{\mathrm{G}}(\boldsymbol{x})$ is volume integrated to $\hat{\mathbf{F}}$ and the final high-fidelity reconstructed image is obtained as $\hat{\mathbf{I}} = G_{\mathbf{1}}(\hat{\mathbf{F}})$.

Note that through point projection π, the reconstruction with local prior is not limited to the original view, and naturally works for novel views. However, for views with severe occlusions or additional editing, the residual features may not fully align with the scene, leading to a failed feature fusion. We will address this issue in the next subsection with our hybrid feature alignment.

4.3 Hybrid Alignment for High-Quality Editing

Though we achieve high-fidelity reconstruction with the aforementioned designs, there is a trade-off between the *input view* reconstruction quality and *novel view* editing performance. We first analyze the reasons behind and propose a hybrid alignment module to address this issue.

Reconstruction Editing Trade-off Given an input image $\mathbf{I}$ with paired reconstruction $\tilde{\mathbf{I}}$ and residual map Δ extracted from the input view $\boldsymbol{\xi}$ with the aforementioned method, the reconstruction performance trade-offs the editing performance due to the following two reasons. First, at test time when the input image is edited $\tilde{\mathbf{I}}_{\mathrm{edit}}$ or query view $\boldsymbol{\xi}' \neq \boldsymbol{\xi}$, the residual map no longer aligns and is likely to result in wrong predictions. Second, if we supervise the models to reconstruct the input itself, the learned features are *regressive* rather than *generative* since all prediction areas are visible in the inputs. With these above-mentioned challenges, though the model could yield perfect reconstruction at training, it would result in noticeable performance degradation when rendering from novel views at test time.

Hybrid Alignment for High-Quality Editing To address the first challenge, we propose to infer aligned features with a 2D-3D hybrid alignment. Specifically, given edited latent code $\hat{\mathbf{w}}_{\mathrm{edit}}$, the initial novel-view edited image $\tilde{\mathbf{I}}'_{\mathrm{edit}} = G_{\mathbf{0}}(\hat{\mathbf{w}}_{\mathrm{edit}}, \boldsymbol{\xi}')$ is misaligned with Δ. Inspired by HFGI (Wang et al., 2022), we leverage a 2D alignment module E_{ADA} to address the misalignment. As shown in Fig. 3 (a), we first obtain $\Delta_{\mathrm{edit}} = E_{\mathrm{ADA}}(\Delta, G_{\mathbf{0}}(\hat{\mathbf{w}}_{\mathrm{edit}}, \boldsymbol{\xi}))$, transform it to residual feature map $\mathbf{F}_{\mathrm{L}}^{\mathrm{edit}}$ via Eq. (7) and retrieve the view-consistent 3D local feature $\mathbf{f}_{\mathrm{L}}$ via Eq. (8). However, to render the high-quality edited image $\hat{\mathbf{I}}'_{edit}$ from novel view $\boldsymbol{\xi}'$, $\mathbf{F}_{\mathrm{L}}^{\mathrm{edit}}$ might still suffer from occlusion due to large pose variations. To the end, we propose a hybrid alignment to further refine $\mathbf{F}_{\mathrm{L}}^{\mathrm{edit}}$ with the 2D aligned feature from E_{ADA}. Specifically, we align a 2D residue $\Delta'_{\mathrm{edit}} = E_{\mathrm{ADA}}(\Delta, \tilde{\mathbf{I}}'_{\mathrm{edit}})$ and retrieve its corresponding $\mathbf{f}_{\mathrm{ADA}}$ with $E_{\mathbf{1}}$, which fills the occlusion in a 2D manner but lacks 3D consistency. To marry the best of both, as shown in Fig 3 (b), we modulate $\mathbf{f}_{\mathrm{L}}$ with $\mathbf{f}_{\mathrm{ADA}}$,

$$\tilde{\mathbf{f}}_{\mathrm{L}}(\boldsymbol{x}) = \mathrm{FiLM}(\mathbf{f}_{\mathrm{L}}(\boldsymbol{x}), \mathbf{f}_{\mathrm{ADA}}(\boldsymbol{x})), \tag{10}$$

and further fuse $\tilde{\mathbf{f}}_{\mathrm{L}}$ with $\mathbf{f}_{\mathrm{G}}(\boldsymbol{x})$ for final prediction,

$$\hat{\mathbf{f}}(\boldsymbol{x}) = \mathrm{FiLM}(\mathbf{f}_{\mathrm{G}}(\boldsymbol{x}), \tilde{\mathbf{f}}_{\mathrm{L}}(\boldsymbol{x})), \tag{11}$$

where $\hat{\mathbf{f}}(\boldsymbol{x})$ is then integrated to $\hat{\mathbf{F}}$ for rendering the final novel-view edited image $\hat{\mathbf{I}}'_{edit} = G_{\mathbf{1}}(\hat{\mathbf{F}})$.

Novel View Training for Coherent View Synthesis To address the second challenge and enforce the model to learn generative features, during training, we sample two views $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ for each style code $\mathbf{w}$, and render the corresponding images $\mathbf{I}^{\boldsymbol{\xi}_1}$ and $\mathbf{I}^{\boldsymbol{\xi}_2}$. Then, we train the models to reconstruct plausible novel views, i.e., $G(E(\mathbf{I}^{\boldsymbol{\xi}_1}), \boldsymbol{\xi}_2) \approx \mathbf{I}^{\boldsymbol{\xi}_2}$ and $G(E(\mathbf{I}^{\boldsymbol{\xi}_2}), \boldsymbol{\xi}_1) \approx \mathbf{I}^{\boldsymbol{\xi}_1}$. By end-to-end training the E_{ADA} with this strategy, our method yields a high-quality view synthesis over edited scenes.

4.4 Pose Estimation for Domain Adaptation

Though the aforementioned strategy could effectively alleviate information loss introduced by the capacity limitation of global latent code, in reality, we observe $E_{\mathbf{1}}$ is likely to fail when the input pose $\boldsymbol{\xi}$ is noisy. This may occur since we directly adopt the ground truth pose $\boldsymbol{\xi} \in \mathcal{X}$ for training, thus the residual Δ is calculated over the reconstruction $\tilde{\mathbf{I}}$ with the perfect pose. However, the pose of real-world images estimated from COLMAP (Schönberger & Frahm, 2016; Schönberger et al., 2016) or pre-trained model (Deng et al., 2019b) tends to be noisy. In this way, the encoder $E_{\mathbf{1}}$ could not handle the residual Δ caused by slight pose misalignment.

To alleviate this issue, we propose to jointly train a pose estimator $E_{\boldsymbol{\xi}}$ over synthetic samples $\mathcal{X}$ and use the predicted pose $\tilde{\boldsymbol{\xi}}$ to calculate global reconstruction $\tilde{\mathbf{I}}_{\tilde{\boldsymbol{\xi}}} = G(\hat{\mathbf{w}}, \tilde{\boldsymbol{\xi}})$. The residual is calculated with $\Delta = \mathbf{I} - \tilde{\mathbf{I}}_{\tilde{\boldsymbol{\xi}}}$, and the remaining operations defined in Eqs. (8) and (9) stay the same. Apart from pose estimation loss $\mathcal{L}_{pose} = ||\tilde{\boldsymbol{\xi}} - \boldsymbol{\xi}||_2$, all the aforementioned loss functions in Sec. 4.1 are also imposed. We demonstrate in the experiment that this operation is crucial for high-fidelity inversion over EG3D.

Although our pose estimator could alleviate the domain gap between training and testing, we observe that it still cannot fully derive the camera pose accurately enough. Moreover, in some scenarios, the user trades off better fidelity with a reasonable time cost. Therefore, we propose to further finetune the estimated pose for a few steps in the test time, where the estimated identity code $\hat{\mathbf{w}}$ could also be finetuned together. Better quality is achieved via both test-time optimization techniques, as validated in Tab. 1 and Fig. 9.

5 E^3DGE-StyleSDF Backbone

E^3DGE also supports adopting StyleSDF (Or-El et al., 2021) as the base inversion model. Different from EG3D, as shown in the middle of Fig. 1, StyleSDF $G_{\mathbf{0}}$ can be further divided into four parts: a 8-layer MLP encoder $E_{G_{\mathbf{0}}}$, a SDF decoder ϕ_g, a feature decoder ϕ_f and a color decoder ϕ_c. To train E^3DGE on StyleSDF, as shown in the upper half in Fig 1 and Fig. 2(a), we synthesize paired 3D shape information $\mathcal{S}$ and 2D image $\mathbf{I}$ from latent code $\mathbf{w}$ and camera pose $\boldsymbol{\xi}$ using G to train $E_{\mathbf{0}}$. To extract the 3D shape information $\mathcal{S}$ of each synthetic shape, we first sample a point set $\mathcal{P} = \{\mathcal{P}_{\mathcal{O}}, \mathcal{P}_{\mathcal{F}}\}$ where $\mathcal{P}_{\mathcal{O}}$ and $\mathcal{P}_{\mathcal{F}}$ contain points sampled from the surface and around the surface, respectively. Then, we calculate the geometry descriptor d_i and $\boldsymbol{n}_i$ for each 3D point $\boldsymbol{x}_i \in \mathcal{P}$, and $\mathcal{S}$ is defined as the set of geometry descriptors of all 3D point in $\mathcal{P}$:

$$\begin{aligned} \mathcal{S} = \{\{d_i, \boldsymbol{n}_i\}_{i=1}^{|\mathcal{P}|} \mid \\ \boldsymbol{x}_i \in \mathcal{P}, d_i = G_{\mathbf{0}}(\mathbf{w}, \boldsymbol{x}_i), \boldsymbol{n}_i = \nabla_{\boldsymbol{x}_i} d_i\}, \end{aligned} \tag{12}$$

where d_i is the distance from $\boldsymbol{x}_i$ to the shape surface and $\boldsymbol{n}_i$ is the surface normal defined by the gradient of the distance w.r.t. $\boldsymbol{x}_i$. Note our method is not limited to the SDF-based shape representation and can be easily extended to volumetric-based methods (Chan et al., 2021; Pan et al., 2021b; Chan et al., 2022a).

To achieve 3D supervision, we would like the estimated $\hat{\mathcal{S}}$ to approximate the ground truth $\mathcal{S}$. Specifically, for points over the surface, their distances and normal are both considered while for points around the surface, we only supervise their distance following (Park et al., 2019; Alldieck et al., 2022), leading to geometry loss:

$$\mathcal{L}^{\mathcal{O}}_{geo} = \mathbb{E}_{\mathcal{X}}\left[\frac{1}{|\mathcal{P}_{\mathcal{O}}|}\sum_{i=1}^{|\mathcal{P}_{\mathcal{O}}|} \lambda_{g_1}|\hat{d}_i| + \lambda_{g_2}\|\hat{\boldsymbol{n}}_i - \boldsymbol{n}_i\|_1\right] \tag{13}$$

$$\mathcal{L}^{\mathcal{F}}_{geo} = \mathbb{E}_{\mathcal{X}}\left[\frac{1}{|\mathcal{P}_{\mathcal{F}}|}\sum_{i=1}^{|\mathcal{P}_{\mathcal{F}}|} \lambda_{g_3}|\hat{d}_i - d_i|\right] \tag{14}$$

$$\mathcal{L}_{geo} = \mathcal{L}^{\mathcal{O}}_{geo} + \mathcal{L}^{\mathcal{F}}_{geo}, \tag{15}$$

where λs are loss weights and $d_i = 0$ for points over the surface. We also impose code reconstruction loss $\mathcal{L}_{code} = \|\hat{\mathbf{w}} - \mathbf{w}\|_2$ to regularize the learning and 2D supervisions $\mathcal{L}_{rec}$ to minimize the reconstruction error between $\tilde{\mathbf{I}}$ and $\mathbf{I}$ as in pSp (Richardson et al., 2021). The overall loss is detailed in Sec. 6.

Among them, StyleSDF (Or-El et al., 2021) introduces the signed distance function (SDF) to serve as a proxy for the density function $\sigma(\boldsymbol{x})$ used for the volume rendering in NeRF. Specifically, StyleSDF uses $G_{\mathbf{0}}$ to predict the distance $d(\boldsymbol{x}) = G_{\mathbf{0}}(\mathbf{w}, \boldsymbol{x})$ between the query point $\boldsymbol{x}$ and the shape surface, where the density function $\sigma(\boldsymbol{x})$ can be transformed from $d(\boldsymbol{x})$ for NeRF (Mildenhall et al., 2020) to render. The incorporation of SDF leads to higher-quality geometry in terms of expressiveness view-consistency and clear definition of the surface. StyleSDF also enjoys the flexible style control for semantic editing as in StyleGAN (Karras et al.,

2019). Therefore, in this paper, we also use StyleSDF as the base model for GAN inversion study. Please refer to the conference version (Lan et al., 2023) for the technical details of our method based on StyleSDF. Note that our method is not limited to EG3D/StyleSDF and could be easily extended to other style-based 3D GAN variations (Gu et al., 2021).

6 Training

Reconstruction Loss We briefly introduce the supervision we adopt in image reconstructions in both training stages. First, we use the pixel-wise $\mathcal{L}_2$ loss,

$$\mathcal{L}_2(\mathbf{I}) = ||\mathbf{I} - \hat{\mathbf{I}}||_2. \tag{16}$$

In addition, to learn perceptual similarities, we use the LPIPS (Zhang et al., 2018) loss, which has been shown to better preserve image quality compared to the more standard perceptual loss:

$$\mathcal{L}_{\text{LPIPS}}(\mathbf{I}) = ||F(\mathbf{I}) - F(\hat{\mathbf{I}})||_2, \tag{17}$$

where $F(\cdot)$ denotes the perceptual feature extractor.

Finally, a common challenge when handling the specific task of encoding facial images is the preservation of the input identity. To tackle this, we incorporate a dedicated recognition loss measuring the cosine similarity between the output image and its source,

$$\mathcal{L}_{\text{Id}}(\mathbf{I}) = 1 - \langle R(\mathbf{I}), R(E_g(\mathbf{I})) \rangle, \tag{18}$$

where R is the pretrained ArcFace (Deng et al., 2019a) network. In summary, the total loss function is defined as

$$\mathcal{L}_{rec}(\mathbf{I}) = \lambda_1 \mathcal{L}_2(\mathbf{I}) + \lambda_2 \mathcal{L}_{\text{LPIPS}}(\mathbf{I}) + \lambda_3 \mathcal{L}_{\text{Id}}(\mathbf{I}),$$

where we set $\lambda_1 = 1, \lambda_2 = 0.8, \lambda_3 = 0.1$ as the defined loss weights. In $E_{\mathbf{0}}$ training, we supervise images $\hat{\mathbf{I}}_{\mathbf{0}}, \hat{\mathbf{I}}_{\mathbf{1}}$ of both resolutions. In $E_{\mathbf{1}}$ training, we only supervise the reconstruction of high-resolution images since the network weights to render $\hat{\mathbf{I}}_{\mathbf{0}}$ is fixed. Here, we also impose the non-saturating adversarial loss with R1 regularization (Mescheder et al., 2018) to improve the naturalness of reconstructed images, which is defined as:

$$\mathcal{L}_{adv} = -\mathbb{E}[log(D(\hat{\mathbf{I}}))], \tag{19}$$

$$\mathcal{L}_D = \mathbb{E}[log(D(\hat{\mathbf{I}}))] + \mathbb{E}[log(1 - D(\mathbf{I}))], \tag{20}$$

$$\mathcal{L}_{R1} = \lambda \|\nabla D(\hat{\mathbf{I}}; \theta_D)\|_2, \tag{21}$$

where D is initialized with the pre-trained discriminator paired with the generator and θ_D is the corresponding parameters to optimize. In summary, the overall loss is the weighted summation of the loss functions described above:

$$\mathcal{L} = \mathcal{L}_{geo} + \mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_D \mathcal{L}_D + \lambda_{R1}\mathcal{L}_{R1}, \tag{22}$$

where we set $\lambda_D = \lambda_{adv} = 0.01$ and $\lambda_{R1} = 10$ in the experiments.

7 Experiments

7.1 Implementation Details

Datasets We mainly focus on the human face domain and use both 2D and 3D datasets for extensive evaluation. To examine 2D reconstruction quality, we adopt CelebA-HQ (Karras et al., 2018; Lee et al., 2020) dataset for source view reconstruction. To further evaluate novel view synthesis performance, we synthesize 100 trajectory videos from a pretrained generator as a proxy test set. For attribute editing, we adopt InterfaceGAN (Shen et al., 2020) and Talk2Edit (Jiang et al., 2021) to search for the editing directions. To evaluate 3D shape reconstruction quality, we use NoW benchmark (Sanyal et al., 2019) that provides a rich variety of face images with ground-truth 3D scans. The 3D GANs are pre-trained on FFHQ (Karras et al., 2019). Note that our method does not rely on any external 3D data during the training process.

Network Architecture Details For $E_{\mathbf{0}}$, a modified version of the pSp encoder (Richardson et al., 2021) is deployed here for a fair comparison with existing work. For EG3D, we introduce 14 prediction heads to the pSp for the latent code prediction. For StyleSDF, since G_0 and G_1 of StyleSDF have 9 and 10 latent codes, respectively, we introduce $9+10$ extra prediction heads. We observe that early layers of $G_{\mathbf{0}}$ control the geometry of generated samples, and later $G_{\mathbf{0}}$ layers as well as decoder generator $G_{\mathbf{1}}$ control the texture and high-frequency details. Thus, we adopt the early pSp feature map of resolution 32×32 to predict the latent code of G_0 for geometry control and the pSp feature map of resolution 64×64 to predict the latent code of G_0 for texture control. We use the highest resolution feature map of pSp with resolution 128×128 to predict the latent code for G_1. We show our FiLM layer where the input features are modulated by the input conditions with predicted γ, and β. The MLP is implemented with two MLP residual blocks (Yu et al., 2021), which outputs α and β for modulation, respectively.

Training Details In this work, we directly use the officially released pre-trained GAN models from EG3D and StyleSDF. In self-supervised shape inversion learning (Sec. 4.1), due to GPU memory restriction, we sample 4 shapes per GPU each iteration for training. After $E_{\mathbf{0}}$ converges, we fix the network weights and only train the $E_{\mathbf{1}}$ for high-fidelity inversion. The hybrid alignment module E_{ADA} is trained end-to-end along

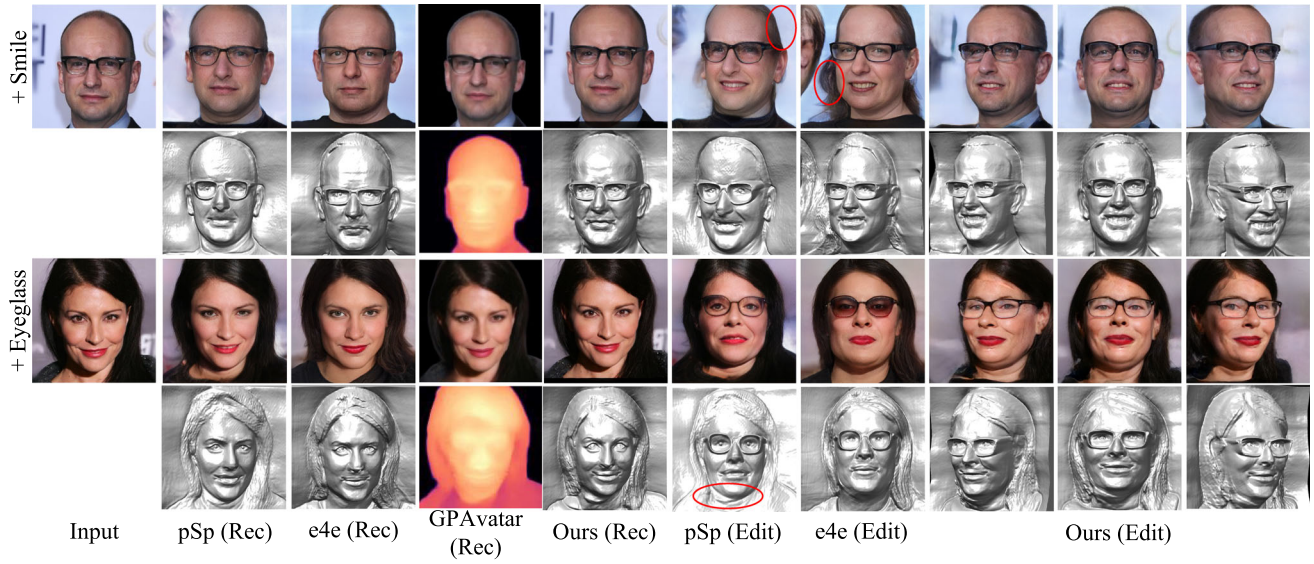

Fig. 4 Qualitative comparisons on face reconstruction (Rec) and editing (Edit) under novel views. Rec denotes "Reconstruction" and Edit denotes "Editing". Our method shows both faithful texture preservation and plausible shape reconstruction compared to the baselines

with the other modules on the FFHQ dataset and supervised with the training strategy discussed in the last paragraph of Sec. 4.3 and the loss functions defined in Eq. 22. For all the parameters, we adopt Adam optimizer with a learning rate of $5e-5$ to train the models on 4 NVIDIA Tesla V100 GPUs, with a resolution of 512^2, batch size of 24, and 48 samples along a ray for the recommended $500K$ iterations. Following (Saito et al., 2019), we filter our invisible 3D points when training from a certain view.

To train the Pose Estimator E_{ξ}, we directly append a prediction head behind the pre-trained global encoder $E_{\mathbf{0}}$ and output the spherical pose (θ, ϕ) of the input image. After the base inversion model is trained, we fix the remaining parameters and train the prediction head over synthetic images for $50,000$ iterations, which takes four days over 4 V100 GPUs. The training takes 4 days for StyleSDF-E^3DGE and 7 days for EG3D-E^3DGE. Code, dataset, and all pre-trained models are publicly available at https://github.com/NIRVANALAN/CVPR23-E3DGE.

7.2 Quantitative Evaluation

For comparison, we implement two canonical encoder-based GAN inversion approaches on StyleSDF (Or-El et al., 2021) and EG3D (Chan et al., 2022a), i.e., pSp (Richardson et al., 2021) and e4e (Tov et al., 2021), which stress reconstruction and editing quality respectively. The feed-forward 3D head reconstruction method GPAvatar (Chu et al., 2024) is also included for reference. Furthermore, we also implement optimization-based methods, including SG2 (Karras et al., 2019) and PTI (Roich et al., 2021) on EG3D and StyleSDF for extensive comparison. Besides, the concurrent optimization-based method SPI (Yin et al., 2022) is also included for comparison.

We report inversion performance for both source view reconstruction and novel view synthesis in Tabs 1-2. For source view reconstruction, the metrics are calculated on the $2,824$ images from CelebA-HQ test set (Lee et al., 2020). For novel view synthesis, the metrics are averaged from 100 videos generated from pre-trained 3D GANs, each with 250 frames covering ellipsoid camera poses trajectory. For each video, we randomly pick one image as source view input and the remaining images as ground truths with labeled poses as query views. In this way, we could extensively evaluate the view synthesis ability under occlusions and varied input viewpoints. We also compare E^3DGE against two optimization-based methods, SG2 and PTI. As demonstrated in Tab. 1, our approach substantially outperforms encoder-based baselines in terms of reconstruction quality on two settings and achieves considerably faster inference speed against optimization-based methods. For SPI, though it achieves better LPIPS score against our proposed method due to the optimization-based nature, it is over $1000\times$ slower compared to E^3DGE_{EG3D}. Besides, it is almost $8\times$ slower compared to $E^3DGE^{P+C}_{\text{EG3D}}$ and achieves worse Similarity score. Besides, SPI is specifically designed for human face inversion, while E^3DGE is a generic solution for any 3D objects modeled with 3D GAN. Notice that we do not include EG3D in Tab. 2 due to its camera pose being misaligned with StyleSDF.

Table 1 Quantitative performance on CelebA-HQ. 'T' and 'S' denote the time for texture and shape inversion, 'P' denotes the finetuning estimated pose, and 'P+C' denotes the finetuning of both pose and code, respectively. For the Time(s) column, we show the texture and geometry inversion time separately. The last two rows describe the performance of test-time optimization, where fine-tuning the pose (P) and code (C) further improves the quality. Note that compared with E^3DGE-StyleSDF, E^3DGE-EG3D does not require post-processing to output depth.

Methods	MAE ↓	SSIM ↑	LPIPS ↓	Similarity ↑	Time(s) ↓
$\text{SG2}_{\text{StyleSDF}}$	.202 ± .063	.650 ± .054	.167 ± .046	.219 ± .106	235
$\text{PTI}_{\text{StyleSDF}}$	**.062 ± .012**	**.796 ± .017**	**.027 ± .005**	**.892 ± .009**	246
$\text{pSp}_{\text{StyleSDF}}$	.150 ± .032	.696 ± .048	.270 ± .059	.498 ± .099	.29
$\text{e4e}_{\text{StyleSDF}}$	.174 ± .049	.669 ± .049	.226 ± .063	.252 ± .107	.29
E^3DGE_{StyleSDF}	.103 ± .010	**.769 ± .039**	**.136 ± .039**	.881 ± .041	**.45/.81**
pSp_{EG3D}	.163 ± .024	.689 ± .039	.264 ± .049	.455 ± .096	0.29
e4e_{EG3D}	.230 ± .021	.658 ± .019	.425 ± .029	.316 ± .068	0.29
SG2_{EG3D}	.241 ± .019	.671 ± .014	.288 ± .019	.434 ± .037	100
PTI_{EG3D}	.079 ± .005	.769 ± .012	.105 ± .011	.779 ± .027	114
SPI_{EG3D}	N/A	N/A	.086	.947	463.5
E^3DGE_{EG3D}	**.084 ± .012**	**.768 ± .026**	.163 ± .017	**.891 ± .027**	**.38**
$E^3DGE^{P}_{\text{EG3D}}$	**.076 ± .004**	**.777 ± .008**	**.153 ± .005**	**.952 ± .004**	45
$E^3DGE^{P+C}_{\text{EG3D}}$	**.064 ± .003**	**.795± .009**	**.115 ± .005**	**.974 ± .003**	60

Table 2 Quantitative performance on Novel View Synthesis.

Methods	MAE ↓	SSIM ↑	LPIPS ↓	Similarity ↑
$\text{SG2}_{\text{StyleSDF}}$	.284 ± .025	.572 ± .006	.244 ± .031	.304 ± .036
$\text{PTI}_{\text{StyleSDF}}$	.186 ±.016	.652 ± .015	.215 ± .045	.795 ± .040
$\text{pSp}_{\text{StyleSDF}}$	.201 ± .010	.634 ± .005	.285 ± .029	.559 ± .043
$\text{e4e}_{\text{StyleSDF}}$	.197 ± .016	.597 ± .011	.212 ± .023	.297 ± .058
E^3DGE	**.147 ± .011**	**.694 ± .018**	**.151 ± .024**	**.901 ± .012**

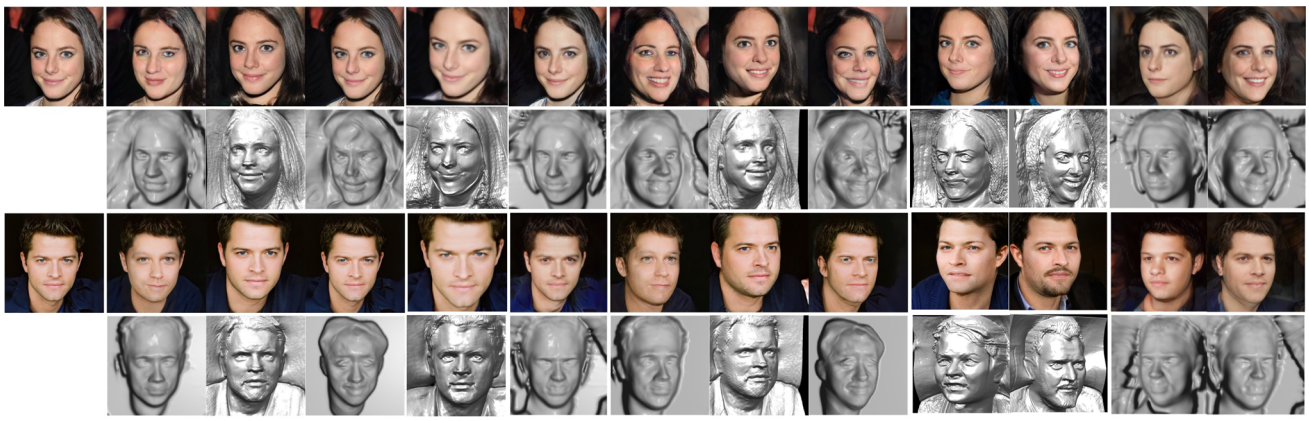

Fig. 5 Visual comparisons on optimization-based methods. 'Rec' and 'Edit' represent reconstruction and editing, respectively. For editing, we change the **'Smiling'** attribute for the first instance and **'Age'** attribute for the second instance. Besides, PTI_{EG3D} (column 7) with "Smiling" editing shows geometry-texture misalignment over the teeth, where the geometry fails to show an open mouth after editing. However, our E^3DGE_{EG3D} shows more coherent geometry and texture editing

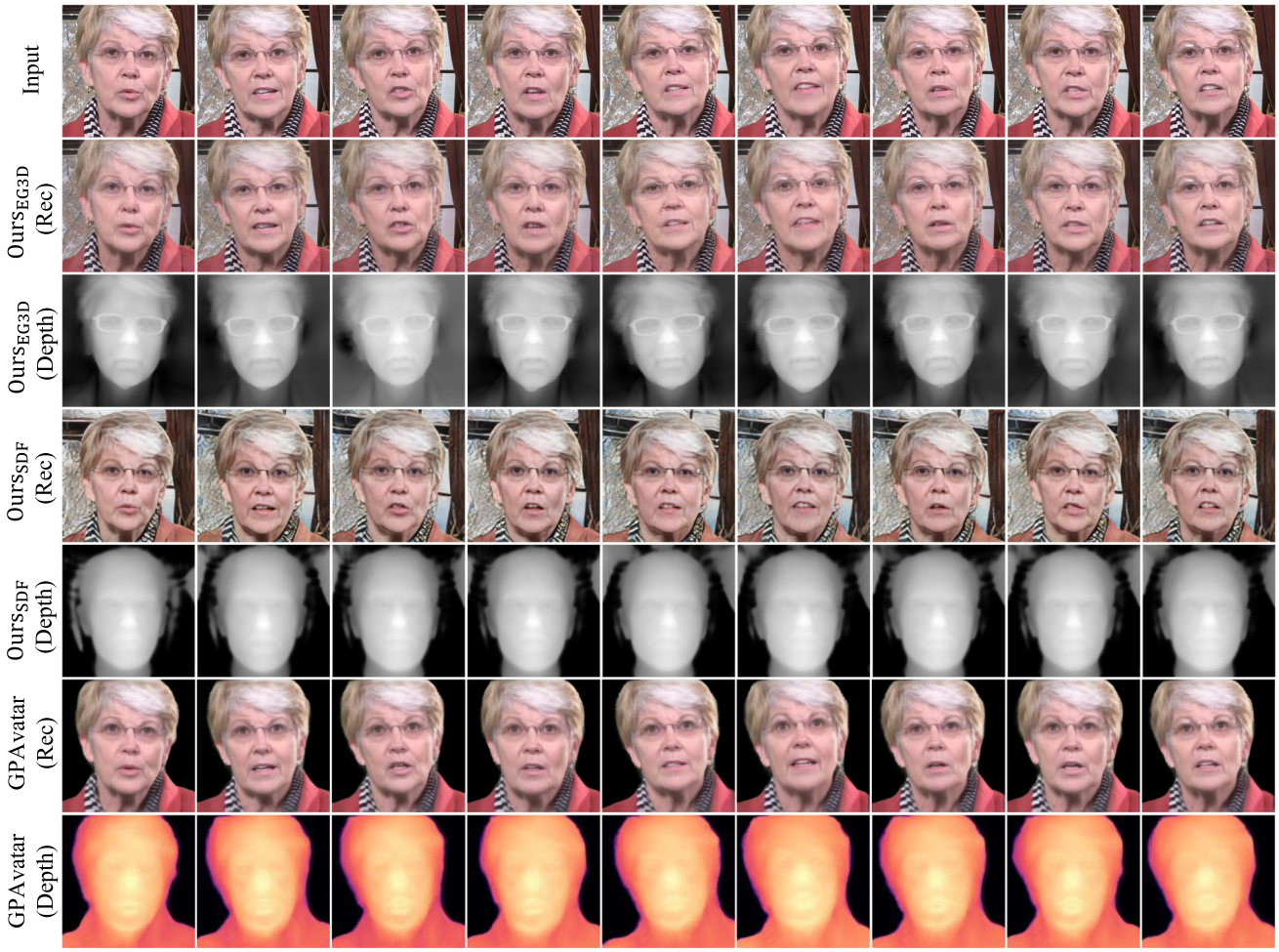

Fig. 6 **Video inversion using** E^3DGE. As shown here, the EG3D-based encoder shows better reconstruction texture quality, while StyleSDF-based E^3DGE shows smoother surface prediction. Note that we also include the depth of EG3D-E^3DGE for pose-aligned texture and shape comparison. Compared to GPAvatar, our proposed method achieves comparative visual quality with sharper geometry (depth) inversion, as reflected by the inversion of eyeglasses

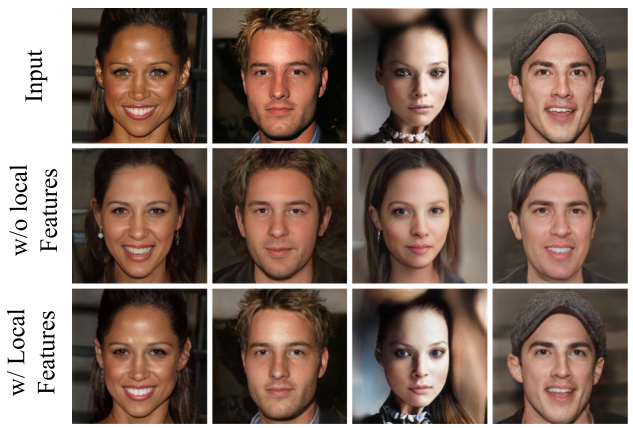

Fig. 7 **Ablation of Local Features.** Our method with pixel-aligned features shows photorealistic reconstructions

7.3 Qualitative Evaluation

Encoder Baselines We visualize both inversion and editing results against encoder baselines in Fig. 4. *Geometry-wise*, the baseline models without explicit 3D supervision tend to generate implausible intermediate shapes, a 3D plank that is only plausible from the input view. Besides, their reconstruction is not close to the "ground truth", and the reconstructed surface lacks details. Our method successfully regularizes the intermediate 3D shapes and generates plausible results with surface details and a more complete structure. For instance, our method reconstructs plausible 3D with with faithful identity preservation. Though pSp also preserves identity well, the reconstructed 3D depth is more flat. Although e4e shows better shape reconstruction, its reconstruction fails to maintain the input identity. Corresponding metrics in Tab. 4 also validate the usefulness of the direct geometry supervisions

and loss designs. *Texture-wise*, existing methods generate distorted results and suffer artifacts and identity change. In contrast, with pixel-aligned features incorporated, our method is more robust with high-fidelity results. In particular, our method captures more details and preserves the identity of different input viewpoints.

For editing, we choose the “Smile" attribute for editing. Beyond plausible shape reconstruction with high-fidelity texture inversion, and in-view synthesis over edited results, our method consistently generates high-quality edited renderings in terms of view consistency, details conservation, and identity preservation. Compared with our method, the baselines either fail to render a plausible novel view (column 5) or maintain input identity after editing (column 6), as circled. Note that though GPAvatar can achieve high-quality 3D reconstruction, it does not support semantic editing.

Optimization Baselines We also compare our method with the state-of-the-art optimization-based methods PTI and SG2 in Fig. 5. We include the performance of PTI (Roich et al., 2021) on both EG3D and StyleSDF for extensive evaluation. With more than 100× faster inference, our method achieves a comparable inversion quality. Also, the editing results produced by E^3DGE successfully preserve the local details with high-fidelity novel view editing performance. Note that the inversion result of StyleSDF tends to be inferior on 3D shape plausibility compared with EG3D, over both encoder-based inversion and optimization-based inversion. Meanwhile, in PTI-based editing, we notice the geometry-texture misalignment of PTI_{EG3D} (column 7), where the "Smiling" texture with teeth does not align with the geometry without teeth. However, our E^3DGE_{EG3D} shows coherent geometry and texture editing, which demonstrates our proposed method maintains more consistent semantics during editing. For SPI, we only include its reconstruction result since their official released code does not include the semantics editing implementation.

Video Inversion The efficiency of the encoder-based method also empowers video inversion and editing. Specifically, we experiment on HDTF (Zhang et al., 2021b) dataset and conduct inversion on each frame using E^3DGE. To allow for GAN inversion, we first preprocess the video by cropping the face from each frame using the original alignment standard adopted by StyleSDF and EG3D, respectively. To reduce oscillation between frames caused by aligning each frame individually, we smooth the affine transformation of each frame within a small sliding window using a mean filter. This operation effectively improves the visual consistency of inversed video without violating the alignment bias of pre-trained GANs.

We visualize the video inversion results in Fig. 6. As can be seen, E^3DGE shows consistent video inversion with high-quality texture and geometry. Compared with

Table 3 Ablations of Local Features and Hybrid Fusion. Our local-global model design with hybrid alignment achieves the balance of high-quality reconstruction and view synthesis.

Ablation Settings	Source View Reconstruction				Novel View Synthesis			
	MAE ↓	SSIM ↑	LPIPS ↓	ID ↑	MAE ↓	SSIM ↑	LPIPS ↓	ID ↑
Synthetic Training	.245 ± .024	.634 ± .019	.333 ± .029	.369 ± .056	.241 ± .011	.594 ± .008	.366 ± .059	.770 ± .026
+Local Features	**.074 ± .007**	**.811 ± .015**	**.075 ± .010**	**.953 ± .006**	.282 ± .103	.571 ± 0.056	.511 ± 0.031	.608 ± .123
+2D Alignment	.098 ± .005	.774 ± .038	.140 ± .040	.900 ± .032	.178 ± .007	.656 ± .009	.178 ± .012	.895 ± .018
+3D Alignment	.102 ± .009	.772 ± .015	.119 ± .016	.818 ± .029	.150 ± .011	.689 ± .022	**.140 ± .021**	.891 ± .011
Hybrid Alignment	.097 ± .008	.780 ± .016	.128 ± .017	.883 ± .017	**.147 ± .011**	**.694 ± .018**	.151 ± .024	**.901 ± .012**

Table 4 Effect of 3D Supervisions on the NoW Challenge.

Settings	Median↓	Mean↓	Std
$\text{pSp}_{\text{StyleSDF}}$	1.97	2.43	2.05
$\text{e4e}_{\text{StyleSDF}}$	2.83	3.40	2.67
$E^3DGE_{\text{StyleSDF}} + \mathcal{L}^{\mathcal{O}}_{geo}$	1.75	2.11	1.72
$E^3DGE_{\text{StyleSDF}} + \mathcal{L}^{\mathcal{F}}_{geo}$	1.71	2.09	1.70
$E^3DGE_{\text{StyleSDF}} + \mathcal{L}_{code}$	**1.66**	**2.06**	**1.69**
E^3DGE_{EG3D}	2.29	2.83	2.31

StyleSDF-based E^3DGE, the EG3D-based E^3DGE shows consistently better texture and shape reconstruction quality.

7.4 Ablation Study

Effect of 3D GAN as Supervisions We quantitatively validate the effects of 3D supervision in the NoW Challenge validation set and report the corresponding metrics in Tab. 4. Compared with 2D supervision alone, adding 3D supervision greatly improves the reconstruction quality. We also validate the benefits of all loss terms in E_0 training. Note that EG3D-based E^3DGE achieves worse performance against StyleSDF-based E^3DGE, demonstrating that EG3D achieves better visual quality at the price of a less accurate geometry surface.

Effect of Local Features As discussed, the local features preserve the image details to facilitate high-fidelity reconstruction. To validate the effectiveness of local features in texture reconstructions, we show the inversion results in Fig. 7. With the proposed local-global fusion pipeline, our model captures more details and guarantees photorealistic reconstruction. Quantitative results in Tab. 3 also validate the effectiveness of local features in high-quality inversion. The results on the video trajectories also show that without delicate design, e.g., novel view training, local features would fully collapse over novel view synthesis.

Effect of Hybrid Alignment We show the view synthesis achieved by different alignment methods in Fig. 8. To quantitatively analyze the effect of hybrid alignment, in Tab. 3 we evaluate the model performance of 3D alignment and 2D alignment individually. For both ablations, novel view training is enabled. As shown here, the 3D alignment model shows better view consistency in video prediction measured by reconstruction metrics, and the 2D alignment model shows better identity preservation. The hybrid alignment model marries the best of both and also enables semantic editing and yields better reconstruction performance on the video predictions.

Effect of Introducing Pose Estimator and Hybrid Optimization We show the inversion results achieved by different pipelines in Fig. 9 and quantitative analysis in Tab. 1. Though

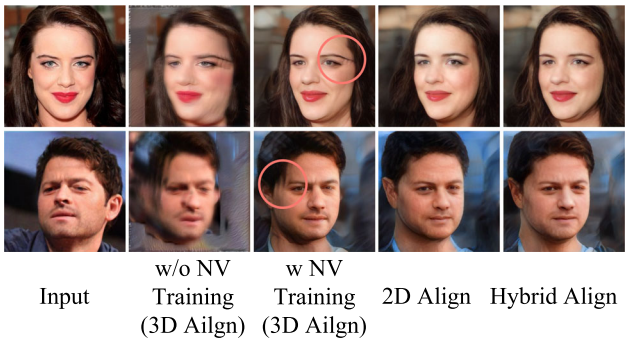

Fig. 8 Ablation of Hybrid Alignment. From left to right, we show the novel view synthesis of 3D-aligned features without novel view training, 3D alignment with novel view training, synthesis achieved using 2D-aligned features, and the final hybrid features. 3D-aligned features are view-consistent but suffer from occlusions (circled), while 2D features are visually plausible but lack some details (e.g., hair color). Our hybrid fused results share the best of both

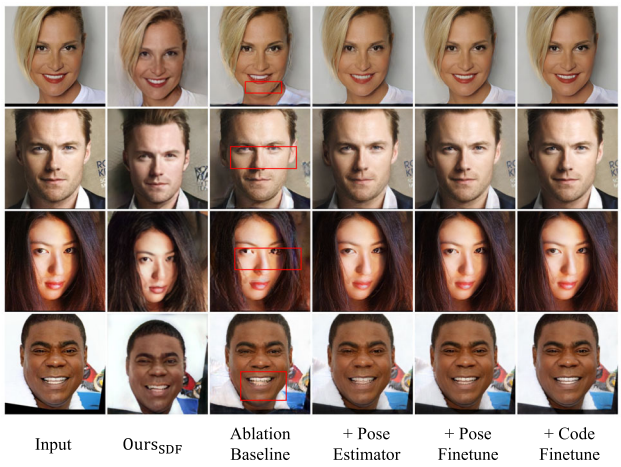

Fig. 9 Ablation of Hybrid Alignment. From left to right, we show the input, inversion with StyleSDF-E^3DGE, inversion without the jointly-trained pose estimator, inversion with jointly trained pose estimator, the inversion with finetuning camera pose and latent code correspondingly. By introducing the pose estimator and further finetuning the pose and latent code, E^3DGE achieves better inversion with fewer visual artifacts. Better zoom in

EG3D-E^3DGE preserves more texture details compared with StyleSDF-E^3DGE (column 2), adopting EG3D as the base model introduces more obvious misalignment issues due to the nature of tri-plane, and introducing a jointly trained pose estimator could alleviate this issue during inference. Furthermore, finetuning the predicted camera pose and latent code could further boost the performance.

7.5 More Results

E^3DGE on Other Categories Besides FFHQ in the paper, we show the performance of our method on AFHQ-EG3D (Fig. 10), and ShapeNet-StyleSDF (Fig. 11). As can be

Fig. 10 **The inversion and view synthesis results of AFHQ cat**

Fig. 11 E^3DGE **qualitative performance on ShapeNet Chair**

Table 5 Model parameters and computational cost evaluated with MACs.

Component	E_0 (pSp)	E_1	E_{ADA}
Parameters(M)	219.71	14.06	0.60
MACs(G)	62.95	26.07	4.03

seen, E^3DGE also achieves high-quality shape and texture inversion on both cats and chairs, demonstrating the generalizability of our method.

3D Toonification After training the E^3DGE encoder, we show that our method could be directly applied to 3D stylization. We show 3D toonify-stylized results over real-world faces using our proposed method in Fig. 12. Following (Pinkney & Adler, 2020), we finetune the pre-trained generator G for 400 iterations with 317 cartoon face images and use our pre-trained encoder E for inference. Visually inspected, the toonified results hold the cartoon style and also preserve the identity of the input image, which demonstrates the potential of applying our method over downstream tasks. Moreover, the toonification of StyleSDF-E^3DGE shows richer 3D details compared with EG3D-based E^3DGE. More results are included in the supplementary.

Computational Cost We include the computational cost of each component in the table below.

7.6 Comparisons of Inversion with MLP and Triplane-based 3D GANs

In this work, we have extended E^3DGE from MLP-based 3D GAN model (Or-El et al., 2021) to triplane-based variant (Chan et al., 2022a) and demonstrated the generalization of our design. However, as discussed before, EG3D-based inversion has its new challenges and does not outperform StyleSDF in every perspective. Here we provide our comparisons of conducting 3D GAN inversion on these two representative model architectures.

Regarding the advantages of inversion on triplane-based 3D GANs, we conclude that

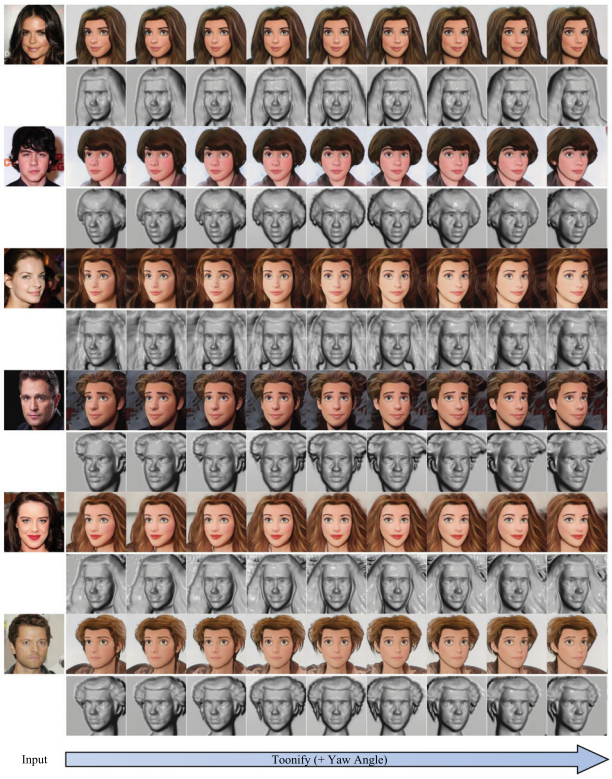

Fig. 12 **Toonification using** E^3DGE. From left to right, we show the toonification result of E^3DGE based on StyleSDF and EG3D

1) higher fidelity can be achieved on EG3D as the triplane offers more representation capacity compared to MLP-based counterparts. As shown in Fig. 5, inversion of the same input over EG3D yields better 3D and texture details compared with StyleSDF;
2) better 3D inductive bias offers more robust and flexible inversion. Specifically, noticeable shape artifacts are observed when applying 2D inversion methods like pSp, e4e, and PTI on the StyleSDF backbone, as shown in Fig. 5.

However, thanks to the triplane's strong 3D inductive bias, directly applying 2D inversion methods yields higher-quality performance with plausible shape and textures reconstructed. This also offers the hybrid inversion option, which improves E^3DGE's prediction by optimizing the inversed latent code within acceptable overhead without affecting the shape plausibility, which could not be guaranteed by StyleSDF.

However, the superior performance of triplane comes at a cost with observed limitations as the following:

1) Triplane is more sensitive to pose alignment issues, e.g., when the estimated pose is not accurate enough, the inversion of E^3DGE is likely to fail, as shown in Fig. 9. Including an extra pose estimator during training and

further finetuning the estimated pose can alleviate this issue.

2) In contrast to StyleSDF, EG3D has more complex latent space and the accurate latent code cannot be accurately acquired via a feed-forward prediction. Further optimization of the estimated poses and latent codes is required to achieve the expected performance in some challenging cases.
3) Training E^3DGE on EG3D requires more time to converge, i.e., 7 days compared with 4 days of StyleSDF, which is reasonable due to the increased model capacity and fidelity. Once trained, our model can amortize the inference time on a series of downstream applications.
4) As shown in Tab. 4, EG3D-based E^3DGE achieves worse performance against StyleSDF-based E^3DGE, which indicates that EG3D achieves better visual quality at the price of less-accurate geometry surface.

Overall, triplane-based 3D representation has shown great potential and it is worthwhile to study encoder-based inversion on it. With triplane's unique advantages and affiliated limitations, we hope our study could inspire the research and industry community to choose a suitable representation of their tasks. We also hope our work could motivate future research on improved 3D representations.

8 Conclusion and Discussions

We propose a novel 3D GAN inversion framework E3DGE for 3D GAN inversion and editing. We marry the benefits of both self-supervised global prior and pixel-aligned local prior for high-quality shape and texture reconstruction. A hybrid alignment that bridges the best of 2D and 3D features is further proposed for view-consistent editing. Benefiting from the overall system design, the proposed method has advantages in terms of both high fidelity and editability. As a pioneer attempt in this direction, we believe this work opens a new line of research direction and will inspire future works on 3D GAN inversion, few-shot 3D reconstruction and 3D-aware learning from 2D images.

Limitations and Future Work First, the fusion of global and local texture in EG3D sometimes leads to visual artifacts, as shown in Fig 9. Though we propose to alleviate this issue with an extra pose estimator and fine-tuning, how to resolve this challenge in the single stage is worth future investigation. Besides, the proposed method is affected by data bias stemming from the use of synthetic data. As the synthetic data lacks complex details and pose variations compared with real-world data, our method trained with it tends to generate a simple background and fail on extreme samples. This issue is particularly noticeable in StyleSDF-E^3DGE as shown in Fig. 6, where all the backgrounds are more blurry compared to EG3D-E^3DGE. A future direction is to leverage real data for semi-supervised training. Moreover, having two models (global and local) to model the texture introduces extra computational cost. A potential solution is to leverage the hyper network (Dinh et al., 2022) for efficient local feature incorporation to alleviate the extra computational cost of the 2D alignment module. Finally, we would explore the potential of our framework on other 3D GANs and shapes and other editing methods uniquely designed for 3D GANs. Special attention should be paid to data bias to avoid social impact on underrepresented minorities.

Acknowledgements This study is supported under the RIE2020 Industry Alignment Fund Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contributions from the industry partner(s). It is also partially supported by Singapore MOE AcRF Tier 2 (MOE-T2EP20221-0011) and the NTU URECA research program.

Data Availability The datasets that support the findings of this study are all publicly online and available for research purposes. The FFHQ dataset can be downloaded from https://github.com/tkarras/progressive_growing_of_gans, the CelebA-HQ dataset can be downloaded from https://mmlab.ie.cuhk.edu.hk/projects/CelebA/CelebAMask_HQ.html, and the ShapeNet dataset can be downloaded from https://shapenet.org/. The StyleSDF pre-trained models used to generate data are available at https://stylesdf.github.io/, and EG3D models are available at https://nvlabs.github.io/eg3d/.

References

Abdal, R., Qin, Y., & Wonka, P. (2019). Image2StyleGAN: How to embed images into the stylegan latent space? In: ICCV

Abdal, R., Qin, Y., & Wonka, P. (2020). Image2StyleGAN++: How to edit the embedded images? In: CVPR

Alldieck, T., Zanfir, M., & Sminchisescu, C. (2022). Photorealistic monocular 3D reconstruction of humans wearing clothing. CVPR

Besnier, V., Jain, H., Bursuc, A., Cord, M., & P'erez, P. (2020). This Dataset Does Not Exist: Training Models from Generated Images. ICASSP

Blanz, V., & Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In: SIGGRAPH

Brock, A., Donahue, J., & Simonyan, K. (2019). Large scale GAN training for high fidelity natural image synthesis. In: ICLR, OpenReview.net, https://openreview.net/forum?id=B1xsqj09Fm

Cai, S., Obukhov, A., Dai, D., & Van Gool, L. (2022). Pix2NeRF: Unsupervised Conditional p-GAN for Single Image to Neural Radiance Fields Translation. In: CVPR

Chan, E., Monteiro, M., Kellnhofer, P., Wu, J., & Wetzstein, G. (2021). Pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In: CVPR

Chan, ER., Lin, CZ., Chan, MA., Nagano, K., Pan, B., Mello, SD., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., Karras, T., & Wetzstein G. (2022a). Efficient geometry-aware 3D generative adversarial networks. In: CVPR

Chan, KC., Xu, X., Wang, X., Gu, J., & Loy, CC. (2022b). GLEAN: Generative latent bank for large-factor image super-resolution and beyond. TPAMI

Chibane, J., Bansal, A., Lazova, V., & Pons-Moll, G. (2021). Stereo Radiance Fields (SRF): Learning View Synthesis for Sparse Views of Novel Scenes. CVPR

Chu, X., Li, Y., Zeng, A., Yang, T., Lin, L., Liu, Y., & Harada, T. (2024). GPAvatar: Generalizable and precise head avatar from image(s). In: ICLR

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019a). ArcFace: Additive angular margin loss for deep face recognition. In: CVPR

Deng, K., Liu, A., Zhu, JY., & Ramanan, D. (2022a). Depth-supervised NeRF: Fewer views and faster training for free. In: CVPR

Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., & Tong, X. (2019b). Accurate 3D face reconstruction with weakly-supervised learning: From single image to image set. In: CVPR

Deng, Y., Yang, J., Xiang, J., & Tong, X. (2021). Gram: Generative radiance manifolds for 3D-aware image generation. In: CVPR

Deng, Y., Wang, B., & Shum, HY. (2022b). Learning detailed radiance manifolds for high-fidelity and 3d-consistent portrait synthesis from monocular image. In: CVPR

Deng, Y., Wang, D., Ren, X., Chen, X., & Wang, B. (2024a). Learning one-shot 4d head avatar synthesis using synthetic data. In: CVPR

Deng, Y., Wang, D., & Wang, B. (2024b). Portrait4d-v2: Pseudo multi-view data creates better 4d head synthesizer. In: ECCV

Dinh, TM., Tran, AT., Nguyen, R., & Hua, BS. (2022). HyperInverter: Improving stylegan inversion via hypernetwork. In: CVPR

Feng, Y., Feng, H., Black, MJ., & Bolkart, T. (2021). Learning an animatable detailed 3D face model from in-the-wild images. In: SIGGRAPH., vol 40

Fu, J., Li, S., Jiang, Y., Lin, KY., Qian, C., Loy, CC., Wu, W., & Liu, Z. (2022). Stylegan-human: A data-centric odyssey of human generation. In: ECCV

Goodfellow, IJ., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S, Courville, AC., & Bengio, Y. (2014). Generative adversarial nets. In: NeurIPS

Gu, J., Liu, L., Wang, P., & Theobalt, C. (2021). StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis. In: ICLR

Henzler, P., Mitra, NJ., & Ritschel, T. (2019). Escaping plato's cave: 3D shape from adversarial rendering. In: ICCV

Hong, F., Chen, Z., Lan, Y., Pan, L., & Liu, Z. (2022). EVA3D: Compositional 3d human generation from 2d image collections. ICLR

Jahanian, A., Chai, L., & Isola, P. (2020). On the" steerability" of generative adversarial networks. ICLR

Jahanian, A., Puig, X., Tian, Y., & Isola, P. (2022). Generative models as a data source for multiview representation learning. ICLR

Jiang, Y., Huang, Z., Pan, X., Loy, CC., & Liu, Z. (2021). Talk-to-Edit: Fine-grained facial editing via dialog. In: ICCV

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In: ICLR

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In: CVPR

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020a). Analyzing and improving the image quality of StyleGAN. In: CVPR

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020b). Analyzing and improving the image quality of StyleGAN. In: CVPR

Lan, Y., Meng, X., Yang, S., Loy, CC., & Dai, B. (2023). E^3DGE: Self-supervised geometry-aware encoder for style-based 3d gan inversion. In: CVPR

Lee, CH., Liu, Z., Wu, L., & Luo, P. (2020). MaskGAN: Towards diverse and interactive facial image manipulation. In: CVPR

Li, T., Bolkart, T., Black, MJ., Li, H., & Romero, J. (2017). Learning a model of facial shape and expression from 4D scans. TOG 36(6), https://doi.org/10.1145/3130800.3130813

Li, X., De Mello, S., Liu, S., Nagano, K., Iqbal, U., & Kautz, J. (2023). Generalizable one-shot neural head avatar. NeurIPS

Lin, CZ., Lindell, DB., Chan, E., & Wetzstein, G. (2022). 3D GAN Inversion for Controllable Portrait Image Animation. arXiv abs/2203.13441

Ling, H., Kreis, K., Li, D., Kim, SW., Torralba, A., & Fidler, S. (2021). EditGAN: High-precision semantic image editing. In: NeurIPS

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., & Geiger, A. (2019). Occupancy networks: Learning 3D reconstruction in function space. In: CVPR

Mescheder, LM., Geiger, A., & Nowozin, S. (2018). Which training methods for GANs do actually converge? In: ICML

Mildenhall, B., Srinivasan, PP., Tancik, M., Barron, JT., Ramamoorthi, R., & Ng, R. (2020) NeRF: Representing scenes as neural radiance fields for view synthesis. In: ECCV

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In: ECCV

Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., & Yang, Y. (2019). HoloGAN: Unsupervised Learning of 3D Representations From Natural Images. In: ICCV

Niemeyer, M., & Geiger, A. (2021). GIRAFFE: Representing scenes as compositional generative neural feature fields. In: CVPR

Or-El, R., Luo, X., Shan, M., Shechtman, E., Park, JJ., & Kemelmacher-Shlizerman, I. (2021) StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. In: CVPR

Pan, X., Dai, B., Liu, Z., Loy, CC., & Luo, P. (2021a). Do 2D GANs know 3D shape? Unsupervised 3D Shape Reconstruction from 2D Image GANs. In: ICLR

Pan, X., Xu, X., Loy, CC., Theobalt, C., & Dai, B. (2021b). A Shading-Guided Generative Implicit Model for Shape-Accurate 3D-Aware Image Synthesis. In: NeurIPS

Pan, X., Zhan, X., Dai, B., Lin, D., Loy, C. C., & Luo, P. (2022). Exploiting Deep Generative Prior for Versatile Image Restoration and Manipulation. *TPAMI, 44*, 7474–7489.

Park, JJ., Florence, P., Straub, J., Newcombe, R, & Lovegrove, S. (2019). DeepSDF: Learning continuous signed distance functions for shape representation. In: CVPR, IEEE, https://doi.org/10.1109/CVPR.2019.00025, https://ieeexplore.ieee.org/document/8954065/

Pavllo, D., Tan, DJ., Rakotosaona, MJ., & Tombari, F. (2023). Shape, pose, and appearance from a single image via bootstrapped radiance field inversion. In: CVPR

Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., & Geiger, A. (2020). Convolutional occupancy networks. In: ECCV

Perez, E., Strub, F., de Vries, H., Dumoulin, V., & Courville, AC. (2018). FiLM: Visual reasoning with a general conditioning layer. In: AAAI

Pinkney, JN., & Adler, D. (2020). Resolution dependent gan interpolation for controllable image synthesis between domains. arXiv preprint arXiv:2010.05334

Rebain, D., Matthews, M., Yi, KM., Lagun, D., & Tagliasacchi, A. (2022). LOLNeRF: Learn from one look

Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., & Cohen-Or, D. (2021). Encoding in style: a StyleGAN encoder for image-to-image translation. In: CVPR

Roich, D., Mokady, R., Bermano, AH., & Cohen-Or, D. (2021). Pivotal tuning for latent-based editing of real images. TOG

Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., & Li, H. (2019). PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: ICCV

Saito, S., Simon, T., Saragih, J., & Joo, H. (2020). PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization. In: CVPR

Sanyal, S., Bolkart, T., Feng, H., & Black, M. (2019). Learning to regress 3D face shape and expression from an image without 3D supervision. In: CVPR

Schönberger, JL., & Frahm, JM. (2016). Structure-from-motion revisited. In: CVPR

Schönberger, JL., Zheng, E., Pollefeys, M., & Frahm, JM. (2016). Pixelwise view selection for unstructured multi-view stereo. In: ECCV

Schwarz, K., Liao, Y., Niemeyer, M., & Geiger, A. (2020). GRAF: Generative radiance fields for 3D-aware image synthesis. In: NeurIPS

Shen, Y., Yang, C., Tang, X., & Zhou, B. (2020). InterFaceGAN: Interpreting the disentangled face representation learned by GANs. TPAMI PP

Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., & Cohen-Or, D. (2021). Designing an encoder for StyleGAN image manipulation. *TOG, 40*(4), 1–14.

Trevithick, A., Chan, M., Stengel, M., Chan, ER., Liu, C., Yu, Z., Khamis, S., Chandraker, M, Ramamoorthi, R., & Nagano, K. (2023). Real-time radiance fields for single-image portrait view synthesis. In: ACM Transactions on Graphics. (SIGGRAPH)

Wang, Q., Wang, Z., Genova, K., Srinivasan, PP., Zhou, H., Barron, JT., Martin-Brualla, R, Snavely, N., & Funkhouser, TA. (2021). IBRNet: Learning Multi-View Image-Based Rendering. In: CVPR, pp 4688–4697

Wang, T., Zhang, Y., Fan, Y., Wang, J., & Chen, Q. (2022). High-Fidelity GAN inversion for image attribute editing. In: CVPR

Xia, W., Zhang, Y., Yang, Y., Xue, JH., Zhou, B., & Yang, MH. (2022). GAN Inversion: A Survey. TPAMI

Xie, J., Ouyang, H., Piao, J., Lei, C., & Chen, Q. (2022). High-fidelity 3d gan inversion by pseudo-multi-view optimization. arXiv preprint arXiv:2211.15662

Xiu, Y., Yang, J., Tzionas, D., & Black, MJ. (2022). ICON: Implicit Clothed humans Obtained from Normals. In: CVPR

Yang, S., Jiang, L., Liu, Z., & Loy, C. C. (2022). VToonify: Controllable high-resolution portrait video style transfer. *TOG, 41*(6), 1–15. https://doi.org/10.1145/3550454.3555437

Yin, F., Zhang, Y., Wang, X., Wang, T., Li, X., Gong, Y., Fan, Y., Cun, X., Shan, Y., Oztireli, C, & Yang, Y. (2022). 3d gan inversion with facial symmetry prior. arXiv preprint arXiv:2211.16927

Yu, A., Ye, V., Tancik, M., & Kanazawa, A. (2021). PixelNeRF: Neural radiance fields from one or few images. In: CVPR

Yuan, Z., Zhu, Y., Li, Y., Liu, H., & Yuan, C. (2023). Make encoder great again in 3d gan inversion through geometry and occlusion-aware encoding. ICCV

Zhang, R., Isola, P., Efros, AA., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR

Zhang, Y., Ling, H., Gao, J., Yin, K., Lafleche, JF., Barriuso, A., Torralba, A., & Fidler, S. (2021a). DatasetGAN: Efficient labeled data factory with minimal human effort. In: CVPR

Zhang, Z., Li, L., Ding, Y., & Fan, C. (2021b). Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In: CVPR, pp 3661–3670

Zhu, J., Shen, Y., Zhao, D., & Zhou, B. (2020). In-domain GAN Inversion for Real Image Editing. In: ECCV