Fast Text-to-3D-Aware Face Generation and Manipulation via Direct Cross-modal Mapping and Geometric Regularization

Jinlu Zhang [*1] **Yiyi Zhou** [*1] **Qiancheng Zheng** [1] **Xiaoxiong Du** [1]
Gen Luo [1] **Jun Peng** [2] **Xiaoshuai Sun** [1] **Rongrong Ji** [1]

Abstract

Text-to-3D-aware face (T3D Face) generation and manipulation is an emerging research hot spot in machine learning, which still suffers from low efficiency and poor quality. In this paper, we propose an ***E**nd-to-End **E**fficient and **E**ffective* network for fast and accurate T3D face generation and manipulation, termed E^3-FaceNet. Different from existing complex generation paradigms, E^3-FaceNet resorts to a direct mapping from text instructions to 3D-aware visual space. We introduce a novel *Style Code Enhancer* to enhance cross-modal semantic alignment, alongside an innovative *Geometric Regularization* objective to maintain consistency across multi-view generations. Extensive experiments on three benchmark datasets demonstrate that E^3-FaceNet can not only achieve picture-like 3D face generation and manipulation, but also improve inference speed by orders of magnitudes. For instance, compared with Latent3D, E^3-FaceNet speeds up the five-view generations by almost 470 times, while still exceeding in generation quality. Our code is released at `https://github.com/Aria-Zhangjl/E3-FaceNet`.

1. Introduction

Text-to-3D-aware face (T3D Face) generation is an emerging research hot spot in machine learning (Toshpulatov et al., 2021; Zhang et al., 2023b; Wu et al., 2023; Xia & Xue, 2023). Compared with the well-developed 2D face generation (Xia et al., 2021; Patashnik et al., 2021; Peng et al., 2022a;b; Du et al., 2023), T3D Face not only requires to depict the intricate details of different facial components but also to model the 3D shapes and positions under different poses (Toshpulatov et al., 2021; Xia & Xue, 2023). Additionally, the semantic consistency between cross-modal (Zhou et al., 2019; Luo et al., 2022; Zhou et al., 2021; Wu et al., 2023) and cross-view (Or-El et al., 2022; Chan et al., 2022; Xia & Xue, 2023) generations is also a main obstacle.

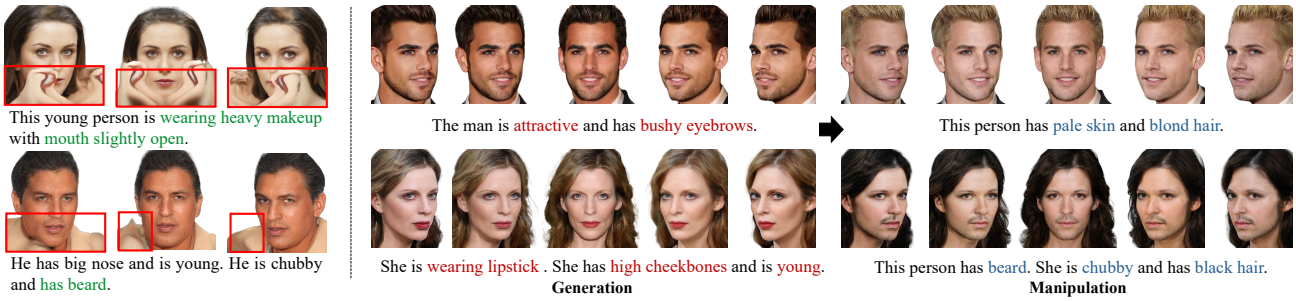
To overcome these challenges, most of existing methods often adopt a multi-stage pipeline for high-quality T3D Face generation (Zhang et al., 2023b; Wu et al., 2023; Rowan et al., 2023; Li & Kitani, 2023). For instance, Wu et al. (2023) propose two independent networks to generate face polygon mesh and texture map separately, and then synthesize the textured mesh by the renderer (Li et al., 2018). Li & Kitani (2023) first generate the latent code via a diffusion model (Rombach et al., 2022) and then feed it to a pre-trained 3D generator for T3D Face generations. Moreover, most methods also require test-time tuning in 2D image space (Zhang et al., 2023b; Canfes et al., 2023; Zhang et al., 2023a) or the manual editing of facial attributes (Wu et al., 2023). Despite effectiveness, the training and inference of existing methods are still time-consuming, greatly impeding their real-world applications.

Unlike previous works, we are keen to fast and accurate T3D face generation and manipulation via direct cross-modal mapping. In particular, we aim to use the text description to impact the whole process of 3D face generation, including the learning of the implicit 3D scene representations, *i.e.*, NeRF (Mildenhall et al., 2021), and the subsequent 2D upsampling. This intuition has been well validated in the recent progress of T2D face generation (Peng et al., 2022a; Sun et al., 2022; Peng et al., 2022b), where the text feature can be directly used to modulate the sampled latent code for adversarial image generation. Adopting this direct cross-modal mapping strategy can help T3D Face methods avoid the complex processing steps in existing multi-stage pipelines, thereby substantially improving inference speed.

However, direct text-to-3D modeling is still intractable, which suffers from two main issues. First, most T3D face datasets contain only single-view face images aligned with

[*]Equal contribution [1]Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, 361005, P.R. China. [2]Peng Cheng Laboratory, Shenzhen, 518000, P.R. China. Correspondence to: Xiaoshuai Sun <xssun@xmu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).



(a) Direct text-to-3D mapping **(b) E^3-FaceNet with Style Code Enhancer and Geometric Regularization**

Figure 1. **(a)** The 3D-aware portraits generated by direct cross-modal mapping. Without any 3D regularization and semantic enhancement, the model tends to generate faces with artifacts (labeled in **red** box) and fails to generate the accurate face attribute (shown in **green** color). **(b)** Multi-view results of T3D face generation and manipulation by our E^3-FaceNet. With the proposed Style Code Enhancer and Geometric Regularization, E^3-FaceNet can well align the text instructions and generate high-quality 3D-aware face images without obvious defects (**red** for generation and **blue** for manipulation). Notably, the speeds of five-view image generation and manipulation of E^3-FaceNet are only 0.64s and 0.89s, respectively. The backgrounds of these images are removed for better comparison.

text descriptions, lacking accurately matched T3D face examples from multi-view collections (Yu et al., 2023). Consequently, a cross-modal mapping T3D face model is often hard to be fully supervised during training, leading to the generation of obvious artifacts, as shown in Fig.1(a). In addition, the visual semantic space of T3D face generation is much larger than that of T2D face, so it is more challenging to ensure semantic consistency among cross-modal and cross-view generations. For instance, the model is prone to overlooking the attributes in the text prompt, as shown in Fig.1(a). In this case, direct mapping for T3D face still remains an open problem.

To remedy these issues, we propose a novel *End-to-end Efficient and Effective* T3D Face generation and manipulation network in this paper, termed E^3-FaceNet. Concretely, E^3-FaceNet is built based on a pre-trained unconditional 3D generation network called *StyleNeRF* (Gu et al., 2021). In E^3-FaceNet, the text representation is directly applied to modulate the sampled noise for end-to-end text-guided 3D generation. To improve semantic alignment, we also propose a novel *Style Code Enhancer* to inject text information into the process of adversarial image rendering, which also enables E^3-FaceNet with the ability of text-driven face editing. Meanwhile, to generate high-quality 3D geometric details, we also propose a novel learning objective for E^3-FaceNet, which leverages both basic and high-order geometric information to regularize the generated faces in 3D space, thereby helping the model synthesize more vibrant and natural-looking 3D faces. Equipped with these two innovative designs, E^3-FaceNet achieves efficient and precise controls over 3D face synthesis, as shown in Fig.1(b).

To validate our E^3-FaceNet, we conduct extensive experiments on three widely-used benchmarks, namely *Multi-Modal CelebA-HQ* (Xia et al., 2021), *CelebAText-HQ* (Sun et al., 2021) and *FFHQ-Text* (Zhou, 2021), and compare E^3-FaceNet with a set of state-of-the-art (SOTA) methods of 2D and 3D face generations, including (Xia et al., 2021; Peng et al., 2022a;b; Wu et al., 2023; Aneja et al., 2023). The experimental results show that our method not only performs better than existing T3D face methods in generation quality, 3D consistency and semantic alignment, but also improves the inference speed by orders of magnitude *e.g.*, 0.64*s* *v.s.* 302*s* of Latent3D (Canfes et al., 2023). Compared with T2D face methods, E^3-FaceNet also excels in single-view image quality, *e.g.*, 12.46 FID *v.s.* 53.38 FID of StyleCLIP (Patashnik et al., 2021) on MMCelebA-HQ. Moreover, our method also supports accurate 3D face manipulation with very limited time cost, *e.g*, only 0.17*s* per view. These results well validate the effectiveness and efficiency of E^3-FaceNet towards fast and accurate T3D Face generation and manipulation.

Conclusively, the contribution of this paper is three-fold:

- We propose a novel cross-modal mapping method for fast and accurate T3D-Face generation and manipulation, termed E^3-FaceNet.
- We propose a novel *geometric regularization* to avoid the artifacts caused by direct cross-modal mapping and an innovative module called *Style Code Enhancer* to ensure the semantic consistency of T3D generation.
- On a bunch of 2D and 3D benchmarks, our E^3-FaceNet outperforms a set of compared methods in terms of image quality and semantic consistency, and improves the speed of T3D Face generation by orders of magnitude.

2. Related Work

2.1. 3D-aware Image Generation

The task of 3D-aware image generation (Xia & Xue, 2023) is to generate high-quality renderings that are consistent

across multiple views. Recent studies have been conducted in representing 3D scenes with neural implicit functions(Mescheder et al., 2019; Park et al., 2019; Mildenhall et al., 2021), among which *neural implicit representation* (NIR) (Mescheder et al., 2019; Mildenhall et al., 2021; Park et al., 2019) emerges as an effective alternative for learning-based 3D reconstruction with multi-view 2D supervision. This representation is then applied to 3D image synthesis under single-view image supervision (Schwarz et al., 2020; Chan et al., 2021; Gu et al., 2021; Or-El et al., 2022; Chan et al., 2022). In this paper, we extend the representative implicit method, *i.e.,* StyleNeRF (Gu et al., 2021), for text-conditioned 3D face generation. By incorporating text semantics into the latent code, our proposed method enables the end-to-end mapping from text words to 3D faces.

2.2. Text-to-3D Face Generation

Text-to-3D face (T3D Face) generation often aims to incorporate text information into 3D generative models for language-guided 3D face generation. Dreamface (Zhang et al., 2023b) and Describe3D (Wu et al., 2023) propose to generate text-conditioned texture maps to render 3D morphable models (3DMM) (Blanz & Vetter, 2023). Despite the effectiveness of geometric modeling, these methods require additional processes to add more facial assets like hair or wrinkle. A relevant work to this paper is TG-3DFace (Yu et al., 2023), which uses tri-plane neural representations and extends EG3D (Chan et al., 2022), a 3D-aware GAN, for end-to-end text-conditioned generation. However, its text-control mechanism is still compositional, which needs to additionally integrate a face parser and train an attribute classifier for fine-grained semantics injection. To guide the generator in learning correct priors, TG-3DFace incorporates a camera pose conditional discriminator during the training process. Different from previous methods, our proposed E^3-FaceNet seamlessly integrates text information into the whole process of 3D representation and 2D upsampling, achieving single-stage cross-modal mapping without any further interventions. Besides, we also propose a set of innovative designs to cope with the problems encountered in direct T3D Face generations, *i.e.*, 3D geometric regularization and Style Code Enhancer.

2.3. Text-Guided 3D Face Manipulation

Despite the advancement (Jetchev, 2021; Michel et al., 2022; Aneja et al., 2023; Canfes et al., 2023), text-guided 3D face manipulation still have obvious drawbacks in inference speed and generalization. Build upon TB-GAN (Gecer et al., 2020), Latent3D (Canfes et al., 2023) optimizes an intermediate layer using a CLIP-based loss (Radford et al., 2021) to generate UV-texture maps. However, the editing quality for textures and expressions are limited due to reliance on 3D scan data. Another method, *i.e.,* ClipFace (Aneja et al., 2023), uses two mappers to predict texture and expression latent codes but requires training new mappers for each text instruction, greatly limiting its efficiency and generalizations. In contrast, our E^3-FaceNet can handle different editing instructions via predicting the style code offset without instance level optimization, achieving efficient and effective 3D face manipulation.

3. Method

3.1. Preliminary

We first recap the preliminaries of our E^3-FaceNet, including *Neural Radiance Field* (NeRF) (Mildenhall et al., 2021) and the 3D generation network *StyleNeRF* (Gu et al., 2021).

NeRF represents a continuous scene as function $\mathcal{F}(\mathbf{x}, \mathbf{v}) = (c, \sigma)$, which takes a 3D point $\mathbf{x} \in \mathbf{R}^3$ and a viewing direction $\mathbf{v} \in \mathbf{R}^2$ as input and a RGB color $c \in [0, 1]^3$ together with a scalar volume density $\sigma \in [0, \infty]$ as output. To better model high-frequency details, Mildenhall et al. (2021) map each dimension of $\mathbf{x}$ and $\mathbf{v}$ to high-dimension feature space with Fourier features ζ. For given near and far bounds t_n and t_f, the expected rendered color $\hat{\mathbf{c}}_\theta$ of ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{v}$ is obtained by

$$\hat{\mathbf{c}}_\theta(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{v})dt, \tag{1}$$

where

$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right). \tag{2}$$

In practice, NeRF formulates $\mathcal{F}$ as MLPs and the parameters θ is optimized over a set of input images and their camera poses by minimizing

$$L = \sum_{\mathbf{r}\in\mathcal{R}_i} \|\hat{\mathbf{c}}_\theta(\mathbf{r}) - \mathbf{c}_{\mathrm{GT}}(\mathbf{r})\|^2 \tag{3}$$

where $\mathcal{R}_i$ denotes a set of input rays and $\mathbf{c}_{\mathrm{GT}}$ is the corresponding GT color.

To enable the control of style attributes, StyleNeRF formalizes 3D representations by conditioning NeRF with a style code $w = f(z), z \in \mathcal{N}(0, I)$ as

$$\phi^n_{\boldsymbol{w}}(x) = g^n_{\boldsymbol{w}} \circ \ldots \circ g^1_{\boldsymbol{w}} \circ \zeta(\mathbf{x}), \tag{4}$$

where $\phi^n_{\boldsymbol{w}}(\mathbf{x})$ is the nth layer feature of point $\mathbf{x}$, f is the mapping network that projects the noise vector z to the style space $\mathcal{W}$, and $g^i_w(.)$ is the ith MLP modulated by w.

Then the final predicted RBG value c and density σ can be obtained by

$$\sigma_{\boldsymbol{w}}(\mathbf{x}) = h_\sigma \circ \phi^{n_\sigma}_{\boldsymbol{w}}(\mathbf{x}), c_{\boldsymbol{w}}(\mathbf{x}, \boldsymbol{v}) = h_c \circ [\phi^{n_c}_{\boldsymbol{w}}(\mathbf{x}), \zeta(\boldsymbol{v})], \tag{5}$$

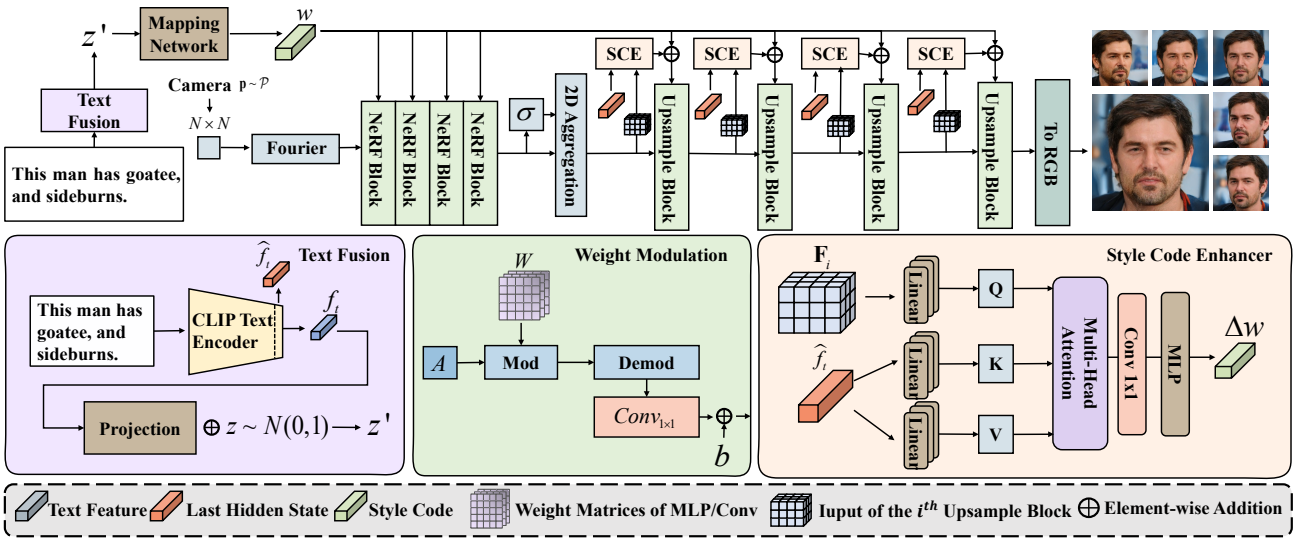


Figure 2. The overall structure of E^3-FaceNet. E^3-FaceNet uses the extracted text features to module the sampled noise, thereby achieving text-guided 3D-aware generation. To improve fine-grained semantic alignment, we introduce a *Style Code Enhancer* (SCE) to predict style code offsets in each upsample block, which also enables E^3-FaceNet to perform fast face editing through the interpolation of offsets.

where h_σ and h_c are linear projections. In order to improve efficiency, StyleNeRF first renders a low-resolution feature map by approximating Eq.1 as:

$$\hat{\mathbf{c}}_\theta(\mathbf{r}) = h_c \circ \left[\phi_{\boldsymbol{w}}^{n_c,n_\sigma}\left(\mathcal{A}(r)\right), \zeta\left(\mathbf{v}\right)\right], \tag{6}$$

where $\phi_{\boldsymbol{w}}^{n,n_\sigma}\left(\mathcal{A}(\boldsymbol{r})\right) = g_{\boldsymbol{w}}^{n} \circ \ldots \circ g_{\boldsymbol{w}}^{n_\sigma+1} \circ \mathcal{A}(\boldsymbol{r})$ and $\mathcal{A}(r) = \int_{t_n}^{t_f} T(t) \cdot \sigma_w(r(t)) \cdot \phi_w^{n_\sigma}(r(t)) dt$. These aggregated features will be upsampled to the desired resolution in 2D space.

By incorporating NeRF into the style-based architecture, StyleNeRF supports style mixing and simple editing by exploring the style space, indicating that this latent space is semantically editable. Inspired by StyleNeRF and previous adversarial T2D face methods (Sun et al., 2022; Peng et al., 2022b), we resort to mapping text information onto this style space for precise language-guided 3D generation.

3.2. The Overall Framework

The overall structure of E^3-FaceNet is depicted in Fig.2. In principle, E^3-FaceNet aims to build a deterministic function to directly map the text description T and the noise vector z to the 3D-aware face image I of the camera pose p:

$$G : (z, T, p) \rightarrow I, \tag{7}$$

where G denotes the generation network, $z \in Z$ is sampled from *Gaussian* latent space, and $p \in P$ is the camera pose.

To realize Eq.7, we first extract the feature of the description f_t by the CLIP text encoder and map it to the latent space Z. Then we modulate the random noise $z \in \mathcal{R}^d$ by $z' = z \oplus f_t'$, where f_t' is the projected text feature and $\oplus$ denote to element-wise addition.

Subsequently, the corresponding style code w is obtained by a mapping network $w = f(z') \in \mathcal{W}$. In this simple way, we can effectively combine text information with the style code, and impact both 3D and 2D representation learnings.

However, this injection is still of limited influence, and achieving well alignment between the synthesis and text semantics remains a challenge (Yu et al., 2023; Wu et al., 2023). Meanwhile, an inherent problem of training a 3D generative model using 2d unposed supervision is the absence of multi-view information (Gu et al., 2021; Xia & Xue, 2023). As shown in Fig.1(a), the model tends to generate 3D faces with obvious artifacts and low semantic alignment. To overcome these challenges, we further propose a novel *Style Code Enhancer* and a *geometric regularization*.

3.3. Style Code Enhancer

As discussed above, it is often difficult to control the entire generation with a single style code w in 2D image generation (Abdal et al., 2020; Saha et al., 2021). And this problem will become more prominent in T3D face generation, since the 3D visual semantic space is much larger than that of the 2D one (Xia & Xue, 2023). Previous studies (Karras et al., 2019; Patashnik et al., 2021; Xia et al., 2022) have shown the varying levels of detail captured by different layers in a 2D image generator. Based on this, we divide the T3D face generation process into two parts: NeRF blocks generate coarse textual semantics like facial contour and shape, and 2D upsample blocks render intricate facial attributes. We introduce a *Style Code Enhancer* (SCE) to enhance fine-grained text semantics incorporation and improve the impact of style codes in the rendering process.

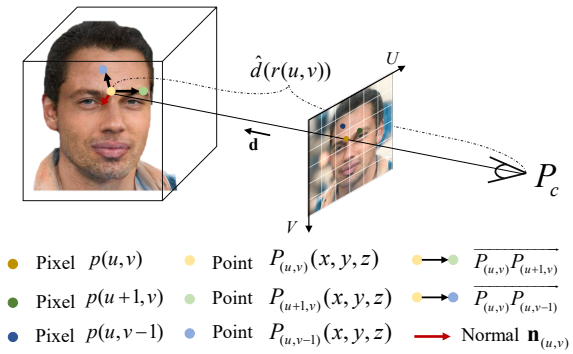


Figure 3. Illustration of the corresponding 3D location $P_{(u,v)}$ and normal vector $n_{(u,v)}$ for a given pixel $p_{(u,v)}$. To generate smooth and natural-looking faces with the absence of multi-view information, we propose to regularize these two terms in 3D space.

Concretely, SCE at the ith upsample block first fuses the input image features $\mathbf{F}_i$ with the text one $\hat{f}_t$ via a cross-modal attention (Rombach et al., 2022):

$$\begin{aligned}\mathbf{F'}_i &= Attention(Q^i, K^i, V^i) \\ &= softmax(\frac{Q^i {K^i}^T}{\sqrt{d}} \cdot V^i),\end{aligned} \tag{8}$$

with

$$Q^i = W_Q^i \cdot \mathbf{F}_i, K^i = W_K^i \cdot \hat{f}_t, V^i = W_V^i \cdot \hat{f}_t. \tag{9}$$

Here, $\hat{f}_t$ refers to the last hidden state of the text encoder.

Then $\mathbf{F'}_i$ is used to predict the offset Δw_i of the style code, denoted as $w_i' = w + \Delta w_i$, as well as to modulate the current upsample block by

$$w_i' = w + \Delta w_i = w + MLP(Conv_{1\times1}(\mathbf{F'}_i)). \tag{10}$$

To ensure the preservation of global semantics, a regularization on Δw is added, formulate as

$$L_\Delta = \sum_i \|\Delta w_i\|_1. \tag{11}$$

Equipped with Style Code Enhancer, E^3-FaceNet can better embed fine-grained text information into the rendering process, as it refines and augments missing semantic information based on both the given text and the intermediate visual features while maintaining the image quality.

T3D face manipulation. Notably, the proposed SCE also enables E^3-FaceNet to perform 3D manipulation. Specifically, we first obtain the predicted style code offsets, w_{ori} and w_{edit}, corresponding to the original text prompt and manipulation instruction using Eq.8-Eq.10. Then a simple linear interpolation of two style code offsets is applied to refine the final style code offset Δw by

$$\Delta w = (1-\lambda)\Delta w_{ori} + \lambda \Delta w_{edit} \tag{12}$$

where $\lambda \in [0, 1]$ is a scale to control the editing degree. The updated style code is then used to influence the rendering process of the generated faces to edit the corresponding attribute while preserving the overall facial identity.

3.4. Geometric Regularization in 3D Space

Due to the lack of 3D supervision from the unposed 2D training dataset (Gu et al., 2021), a direct text-to-3D image mapping is prone to generating incorrect 3D geometries and shapes of human faces. To address this issue, we also equip E^3-FaceNet with a novel regularization on both basic geometric property and high-order geometric features.

Specifically, we focus on the points' *3D location* and *normal vector* corresponds to each pixel of the rendered feature map, as shown in Fig.3. These two geometric regularizations are designed to enforce smoothness and coherence among neighboring points, obtaining a more accurate representation of the 3D face's shape and facial attributes.

Formally, the estimated depth $\hat{d}$ of a ray $\mathbf{r}$ can be defined by

$$\hat{d}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma_w(\mathbf{r}(t))tdt. \tag{13}$$

Therefore, the 3D location $P_{(u,v)}(x,y,z)$ of pixel $p(u,v)$ in the world coordinate can be easily obtained via

$$P_{(u,v)}(x,y,z) = P_c + \hat{d}(\mathbf{r}_{(\mathbf{u},\mathbf{v})}) \cdot \mathbf{d}, \tag{14}$$

where P_c is the camera coordinate and $\mathbf{d}$ is the normalized viewing direction. $\mathbf{d}$ can be easily obtained by the sampled camera pose p and the pre-defined camera intrinsic matrix. Then the 3D location constraint is defined by

$$L_{loc} = \sum_{(u,v)} \sum_{(i,j)\in\{-1,1\}^2} \|P_{(u+i,v+j)} - P_{(u,v)})\|_1. \tag{15}$$

In terms of the normal vector of P, previous works formulate it as $n = -\frac{\nabla_{\boldsymbol{x}}\sigma}{\|\nabla_{\boldsymbol{x}}\sigma\|}$ (Boss et al., 2021; Srinivasan et al., 2021). Since we can calculate the gradient $\nabla_{\boldsymbol{x}}\sigma$ directly, we turn to approximate n by virtual normal (Yin et al., 2019) to reduce GPU memory and computation.

Following (Klasing et al., 2009) and (Yin et al., 2019), we assume that local 3D points locate in the same plane of which the normal vector is the surface normal. Therefore, for each 3D point $P_{(u,v)}$, we select it two neighboring points, *e.g.*, $P_{(u+1,v)}$ *and* $P_{(u,v-1)}$, to establish a plane and compute the normal vector by

$$n_{(u,v)} = \frac{\overrightarrow{P_{(u,v)}P_{(u+1,v)}} \times \overrightarrow{P_{(u,v)}P_{(u,v-1)}}}{\left\|\overrightarrow{P_{(u,v)}P_{(u+1,v)}} \times \overrightarrow{P_{(u,v)}P_{(u,v-1)}}\right\|}. \tag{16}$$

In this case, we perform regularization on the normal vectors, defined by

$$L_{normal} = \sum_{(u,v)} \sum_{(i,j)\in\{-1,1\}^2} \|n_{(u+i,v+j)} - n_{(u,v)}\|_1. \tag{17}$$

Conclusively, the final smooth regularization loss can be written by

$$L_{reg} = L_{loc} + L_{normal}. \tag{18}$$

By integrating these regularizations, E^3-FaceNet significantly enhances face synthesis, achieving a more accurate 3D shape and improved fidelity.

3.5. Objective Function

In terms of optimization, E^3-FaceNet adopts a non-saturating GAN objective (Goodfellow et al., 2014) with *R1* regularization (Mescheder et al., 2018) for stable training. The objective for the discriminator is

$$\begin{aligned} L_D = &-\frac{1}{2}[\mathbb{E}_{I\sim P_{dala}} logD(I) + \mathbb{E}_{\hat{I}\sim p_G} log(1 - D(\hat{I})) \\ &+ \mathbb{E}_{I\sim P_{dala}} logD(I,T) + \mathbb{E}_{\hat{I}\sim p_G} log(1 - D(\hat{I},T))] \\ &+ \gamma \mathbb{E}_{I\sim P_{dala}}[(\|\nabla D(I)\|^2 + \|\nabla D(I,T)\|^2)]. \end{aligned} \tag{19}$$

In terms of the generator, its objectives are 5-fold. First, E^3-FaceNet adopts the conditional loss (Mirza & Osindero, 2014) in adversarial training for text-guided generation, which is formulated as

$$L_{adv} = -\frac{1}{2}[\mathbb{E}_{\hat{I}\sim p_G} logD(\hat{I}) + \mathbb{E}_{\hat{I}\sim p_G} logD(\hat{I},T)]. \tag{20}$$

Meanwhile, NeRF-path regularization loss $L_{NeRF-path}$ (Gu et al., 2021) is used to enforce 3D consistency. To achieve fast convergence, we also adopt a contrastive loss L_{clip} that is defined by

$$L_{clip} = -\sum_{i=1}^{n} \log \frac{\exp(E_T(T_i) \cdot E_I(\hat{I}_i))}{\sum_{j=1}^{n} \exp(E_T \cdot (T_i) \cdot E_I(\hat{I}_j))}, \tag{21}$$

where T_i and $\hat{I}_i$ are the input text prompts and corresponding generated image. In this case, the overall objectives are:

$$L_G = L_{adv} + L_{reg} + L_{\Delta} + L_{clip} + \beta L_{NeRF-path}, \tag{22}$$

where β is a hyper-parameter, and L_{Δ} is the style regularization term defined in Eq.11.

4. Experiment

4.1. Experimental Settings

Datasets The benchmarks used in this paper include MMCelebA (Xia et al., 2021) , *CelebAText-HQ* (Sun et al., 2021) and *FFHQ-Text* (Zhou, 2021). MMCelebA consists of 30,000 face images while CelebAText-HQ has 15,015 face images, each of which corresponds to 10 different text descriptions. FFHQ-Text is a small-scale face image dataset with diverse facial attributes. This dataset contains 760 female FFHQ faces with 9 descriptions for each image. We only use the training split of MMCelebA to train our E^3-FaceNet and evaluate it on the test split along with other 2D methods trained on the same dataset. In terms of 3D methods, as none of the available models can be trained on MMCelebA, we conducted zero-shot validation on the CelebAText-HQ dataset and FFHQ-Text for a fair comparison, which can also assess the generalization.

Table 1. Comparison with existing T3D Face methods. Our E^3-FaceNet not only has better performance but also a faster speed.

Method	CelebAText		FFHQ-Text		Inference [1]
	MVIC(↑)	SA (↑)	MVIC(↑)	SA(↑)	Time
Latent3D	0.7835	**0.2688**	0.7845	0.2758	301.92s
Describe3D	0.7973	0.2449	0.8027	0.2402	22.93s
E^3-FaceNet	**0.8521**	0.2623	**0.8467**	**0.2778**	**0.64s**

Metrics Following the settings of (Yu et al., 2023; Wu et al., 2023; Aneja et al., 2023), we evaluate the generation quality of our method and the compared methods from three aspects: (1) *Image-Quality*, *i.e.*, the reality and fidelity of the rendered 2D face images, using *Fréchet Inception Distance* (**FID**) (Heusel et al., 2017) and *Kernel Inception Distance* (**KID**) (Bińkowski et al., 2018) metrics, (2) *Semantic-Alignment* (**SA**), *i.e.*, using the text-image similarity of CLIP to measure whether the generated face align with the given description, and (3) *Multi-View Identity Consistency* (**MVIC**), *i.e.*, calculating the mean *Arcface cosine similarity* (Schroff et al., 2015) scores between the face images synthesized from the same 3D face but rendered from different camera poses. For better evaluations, we also conduct a user study from the perspectives of *Identity Preservation* (**IP**), *Semantic-Alignment* (**SA**) and *Editing-Quality* (**EQ**). For more details refer to our appendix.

Implementation Details In this paper, we use StyleNeRF (Gu et al., 2021) as our base model. Similarly, we adopt NeRF++ (Zhang et al., 2020), a variant of NeRF, as the backbone. We follow the settings of (Gu et al., 2021) and use NeRF++ to produce a feature map at 32x32 resolution, which is then progressively upsampled to 512x512 resolution. We trained our E^3-FaceNet with a learning rate of 2.5e-4 for both the generator and discriminator. To expedite convergence, we initialized the model with pre-trained weights of StyleNeRF, which were trained on the FFHQ dataset at a resolution of 512. We set $\gamma = 0.5$ and $\beta = 0.2$. The entire process for our model takes approximately about 5 days on 4 V100 GPUs.

4.2. Experimental Results

Comparison with T3D Face methods. We first quantitatively compare our E^3-FaceNet with two strong T3D Face methods, namely Latent3D (Canfes et al., 2023) and De-

[1]We test the speed of five-view generation with one V100 GPU.

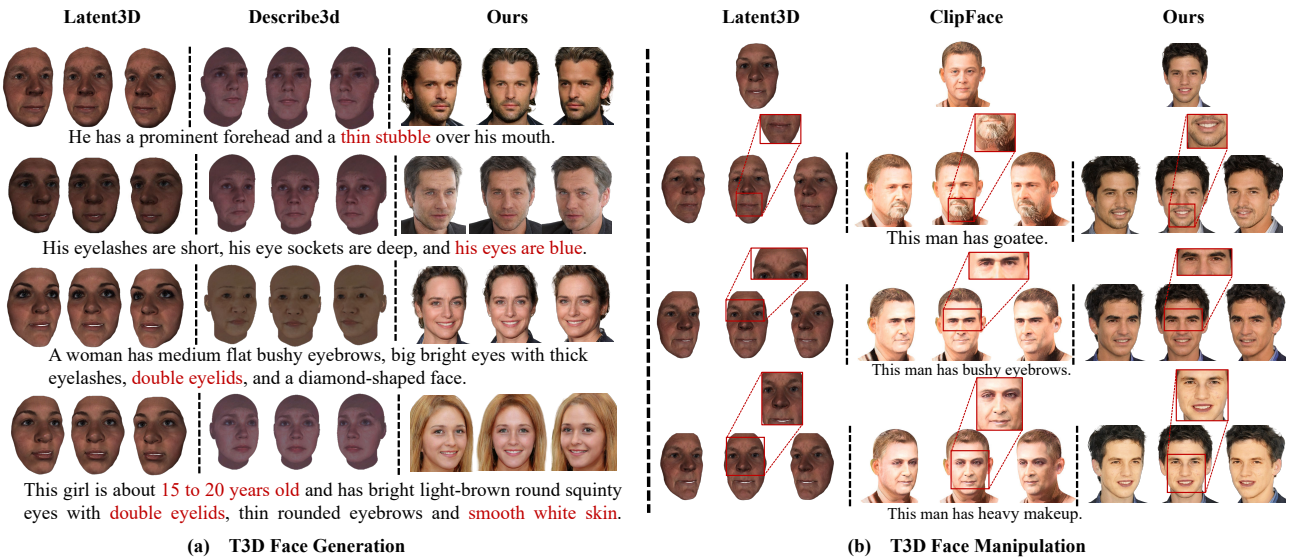


Figure 4. Comparison with T3D Face generation and manipulation methods. E^3-FaceNet can generate face images of better quality than the compared methods, while well aligning the generation and editing prompts[2]. Its inference speed is also much faster.

Table 2. Comparison with existing T3D face manipulation. The inference time is for five-view generation.

Method	IP (↑)	SA(↑)	EQ(↑)	Inference Time(↓)
Latent3D	21.92	2.8	11.60	304.44s
ClipFace	8.16	44.56	12.48	17min + 1.83s
Ours	**69.92**	**52.64**	**75.92**	**0.89s**

Table 3. Comparison with T2D face methods on MM-CelebA.

Method	FID (↓)	KID (↓)	CLIP-Score (↑)
StyleCLIP	53.38	49.89	0.2452
TediGAN	47.61	45.63	**0.2865**
OpenFaceGAN	20.72	12.58	0.2590
PixelFace	16.63	8.65	0.2583
Ours	**12.46**	**5.84**	0.2770

scribe3D (Wu et al., 2023) on CelebAText (Sun et al., 2021) and FFHQ-Text (Zhou, 2021), of which results are given in Tab.1. From this table, we can first find that E^3-FaceNet greatly outperforms the other two methods under the metric of MVIC, *e.g.*, +6.87% on CelebAText, showing great advantages in terms of multi-view identity preservation. Meanwhile, the performance of SA is also outstanding, which is slightly worse than Latent3D on CelebAText but obviously better on FFHQ. These results suggest that E^3-FaceNet can achieve excellent multi-view semantic alignment. Moreover, retaining high performance, our E^3-FaceNet also speeds up the five-view generations by orders of magnitudes. For example, compared with Latent3D and Describe3D, the inference speed of E^3-FaceNet is 471.75 $\times$ and 35.83 $\times$ faster. Notably, E^3-FaceNet achieves such performance under the setting of zero-shot generation.

Text-Driven 3D Face Manipulation. To examine the editing ability of E^3-FaceNet, we also compare it with Latent3D and ClipFace (Aneja et al., 2023) in Tab.2. Without available evaluations, we conduct a user study for this comparison. We can first observe that the identity preservation and the editing quality of E^3-FaceNet are much superior to the compared methods, *e.g.*, 69.92 *v.s.* 21.92 of Latent3D on IP and 75.92 *v.s.* 12.48 of ClipFace on EQ. Besides, E^3-FaceNet also exhibits strong cross-modal semantic consistency, *e.g.*, 52.64 *v.s.* 44.56 of ClipFace on SA. In addition, the efficiency of E^3-FaceNet is still much better than the others. For instance, ClipFace requires about 17 *mins* to train the texture mappers with another 1.8 seconds for rendering. In contrast, E^3-FaceNet only takes 0.89 seconds for the five-view editing.

Comparison with T2D Face methods. To further examine the generation ability of E^3-FaceNet, we also compare it with a set of strong T2D Face methods on MMCelebA (Xia et al., 2021; Patashnik et al., 2021; Peng et al., 2022a;b), and the result is shown in Tab.3. Theoretically, 2D face generation is relatively easier than 3D one (Toshpulatov et al., 2021). However, E^3-FaceNet also achieves outstanding performance on FID and KID for 2D image quality, indicating that E^3-FaceNet has better perceptual quality. In terms of CLIP-Score for semantic alignment, E^3-FaceNet is slightly worse than TediGAN (Xia et al., 2021), which requires an example-dependent optimization, *i.e.*, 200 steps for each text. Compared with the end-to-end 2D methods, *i.e.*, OpenFace (Peng et al., 2022b) and PixelFace (Peng et al., 2022a), E^3-FaceNet is still better in semantic consistency, well confirming the merits of our SCE. Overall, E^3-FaceNet also

[2]The backgrounds are removed for better comparison.

Table 4. Ablation study of E^3-FaceNet. L_{reg} is the proposed *geometric regularization* and SCE is Style Code Enhancer.

Setting	MMCelebA			CelebAText		FFHQ-Text	
	FID(↓)	KID(↓)	CLIP-Score(↑)	MVIC(↑)	SA(↑)	MVIC(↑)	SA(↑)
w/o L_{reg} & SCE	13.44	6.11	0.2622	0.7960	0.2499	0.7873	0.2681
Only L_{loc} in Eq.15	12.87	6.16	0.2657	0.8431	0.2547	0.8313	0.2711
Only L_{normal} in Eq.17	12.78	5.94	0.2703	0.8307	0.2570	0.8278	0.2730
L_{reg} in Eq.18	12.72	5.92	0.2652	**0.8560**	0.2540	**0.8511**	0.2704
L_{reg}+SCE	**12.46**	**5.83**	**0.2770**	0.8521	**0.2623**	0.8467	**0.2778**

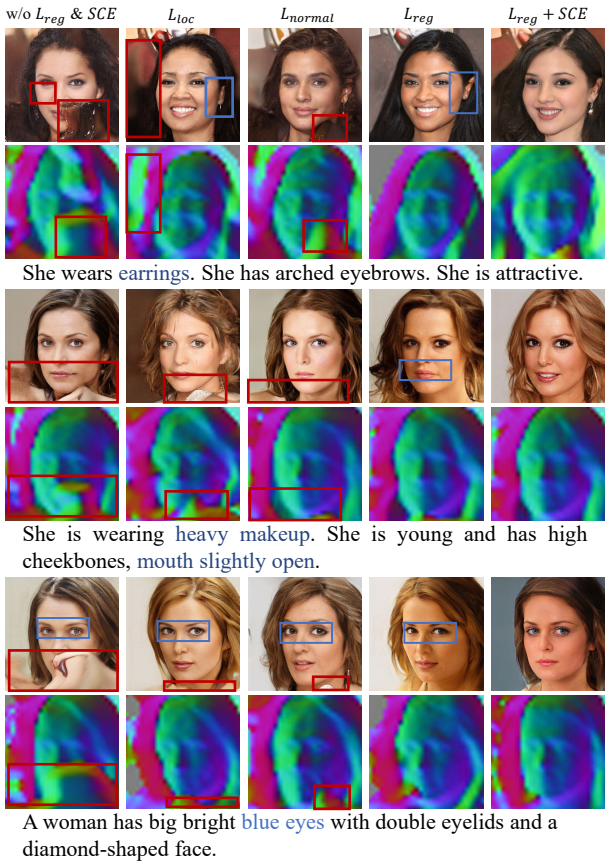


Figure 5. The visualizations of different settings of E^3-FaceNet. The bottom images are the correspondent feature map

exhibits a strong single-view generation ability, even better than existing SOTA T2D Face methods.

Ablation study. We ablate different designs of E^3-FaceNet on four benchmarks in Tab.4. Compared with the plain baseline, *i.e.*, without L_{reg} and SCE, the location loss in Eq.15 or the normal smoothness in Eq.17 can help E^3-FaceNet to generate smooth surfaces and greatly improve MVIC score. Meanwhile, their combination, *i.e.*, L_{reg}, achieves better performance. Besides, the introduction of SCE can inject finer-grained text information into the generation process, as evidenced by achieving the highest SA.

Visualizations of Text-to-3D Face Generation. We visualize the T3D face generations of E^3-FaceNet, Latent3D and Describe3D in Fig.4 (a). As can be seen, E^3-FaceNet excels in synthesizing high-quality images with better multi-view consistency, which also well aligns with the given prompts. Meanwhile, Latent3D and Describe3D are hard to reconstruct faces with finer-grained details, as exemplified by the prompt "*thin stubble*" in Fig.4 (a). Also, these two methods fall short of generating 3D faces with diverse facial attributes, such as hair or ears, lacking the ability to incorporate such elements into the synthesized faces.

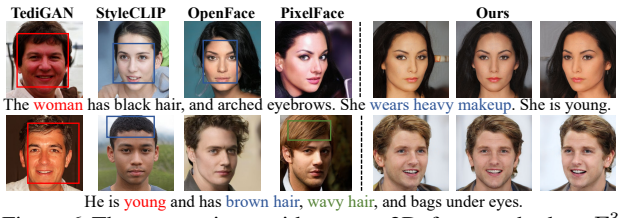


Figure 6. The comparison with text-to-2D face methods. E^3-FaceNet also excels in image quality and semantic consistency.

Visualizations of Text-Driven 3D Face Manipulation. We also present the editing results of E^3-FaceNet, Latent3D and ClipFace in Fig.4 (b). It can be observed that Latent3D struggles to respond to editing instructions, likely due to its heavy reliance on the initially generated face (Canfes et al., 2023). For instance, neither the "*goatee*" nor the "*heavy makeup*" are generated in its editions. ClipFace is slightly better and aware of the text semantics. However, without geometry optimization, ClipFace synthesizes unnatural texture maps and barely achieves multi-view consistency, *e.g.*, "*goatee* ". The identity also changes significantly. In contrast, E^3-FaceNet can well respond to the text semantics while retaining high quality and multi-view consistency.

The impacts of different designs in E^3-FaceNet. We also visualize the impacts of each design in Fig.5. Without the geometric regularization, the model tends to generate faces of artifacts, which is most severe under the base settings. In contrast, this loss term can help E^3-FaceNet synthesize 3D faces with clear contour. Additionally, with SCE, finer-grained semantics can be seamlessly injected. For example, the model successfully generates earrings, while other settings fail to capture this attribute.

Visualizations of Text-to-2D Face Generation. In Fig.6, we also visually compare the single-view generation of E^3-

FaceNet with strong T2D Face methods (Xia et al., 2021; Patashnik et al., 2021; Peng et al., 2022a;b). As can be seen, E^3-FaceNet can achieve better semantic alignment than these 2D methods, *e.g.*, gender, age, or hairstyles. Compared with the end-to-end 2D methods, *i.e.*, OpenFace and PixelFace, E^3-FaceNet also presents better generation qualities of different views. These results well confirm the excellent generation ability of E^3-FaceNet again.

5. Conclusion

In this paper, we propose a novel and efficient E^3-FaceNet for fast and accurate T3D face generation and manipulation. E^3-FaceNet follows the principle of direct cross-modal mapping, and introduces a novel Style Code Enhancer for semantic-aligned rendering and an innovative Geometric Regularization to ensure multi-view consistency. The experiment results demonstrate that E^3-FaceNet achieves higher-fidelity generation and better semantic consistency than a set of compared T2D and T3D Face methods, and also speed up T3D face generation by orders of magnitudes.

Acknowledgements

This work was supported by National Key R&D Program of China (No.2022ZD0118201), the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U21B2037, No. U22B2051, No.U21A20472, No. 62072389, No. 623B2088, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62002305 and No. 62272401), the National Natural Science Fund for Young Scholars of China (No. 62302411), China Postdoctoral Science Foundation (No. 2023M732948), the Natural Science Foundation of Fujian Province of China (No.2021J01002, No.2022J06001), and partially sponsored by CCF-NetEase ThunderFire Innovation Research Funding (NO. CCF-Netease 202301).

Impact Statement

This paper enhances the visual quality and improve the efficiency of text-to-3D-aware face generation through *End-to-End cross modal mapping*, which will further advance the development of 3D-aware image generation. Our work may have various potential societal consequences, though none we believe need to be specifically highlighted here.

References

Abdal, R., Qin, Y., and Wonka, P. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8296–8305, 2020.

Aneja, S., Thies, J., Dai, A., and Nießner, M. Clipface: Text-guided editing of textured 3d morphable models. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

Blanz, V. and Vetter, T. A morphable model for the synthesis of 3d faces. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 157–164. 2023.

Boss, M., Braun, R., Jampani, V., Barron, J. T., Liu, C., and Lensch, H. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12684–12694, 2021.

Canfes, Z., Atasoy, M. F., Dirik, A., and Yanardag, P. Text and image guided 3d avatar generation and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4421–4431, 2023.

Chan, E. R., Monteiro, M., Kellnhofer, P., Wu, J., and Wetzstein, G. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5799–5809, 2021.

Chan, E. R., Lin, C. Z., Chan, M. A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L. J., Tremblay, J., Khamis, S., et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16123–16133, 2022.

Du, X., Peng, J., Zhou, Y., Zhang, J., Chen, S., Jiang, G., Sun, X., and Ji, R. Pixelface+: Towards controllable face generation and manipulation with text descriptions and segmentation masks. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 4666–4677, 2023.

Gecer, B., Lattas, A., Ploumpis, S., Deng, J., Papaioannou, A., Moschoglou, S., and Zafeiriou, S. Synthesizing coupled 3d face modalities by trunk-branch generative adversarial networks. In *European conference on computer vision*, pp. 415–433. Springer, 2020.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Gu, J., Liu, L., Wang, P., and Theobalt, C. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jetchev, N. Clipmatrix: Text-controlled creation of 3d textured meshes. *arXiv preprint arXiv:2109.12922*, 2021.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020.

Klasing, K., Althoff, D., Wollherr, D., and Buss, M. Comparison of surface normal estimation methods for range sensing applications. In *2009 IEEE international conference on robotics and automation*, pp. 3206–3211. Ieee, 2009.

Li, T.-M., Aittala, M., Durand, F., and Lehtinen, J. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.

Li, Y.-J. and Kitani, K. 3d-clfusion: Fast text-to-3d rendering with contrastive latent diffusion. *arXiv preprint arXiv:2303.11938*, 2023.

Lin, C.-H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.-Y., and Lin, T.-Y. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 300–309, 2023.

Luo, G., Zhou, Y., Sun, X., Wang, Y., Cao, L., Wu, Y., Huang, F., and Ji, R. Towards lightweight transformer via group-wise transformation for vision-and-language tasks. *IEEE Transactions on Image Processing*, 31:3386–3398, 2022.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for gans do actually converge? In *International conference on machine learning*, pp. 3481–3490. PMLR, 2018.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4460–4470, 2019.

Michel, O., Bar-On, R., Liu, R., Benaim, S., and Hanocka, R. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13492–13502, 2022.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Or-El, R., Luo, X., Shan, M., Shechtman, E., Park, J. J., and Kemelmacher-Shlizerman, I. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13503–13513, 2022.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.

Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., and Lischinski, D. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2085–2094, 2021.

Peng, J., Du, X., Zhou, Y., He, J., Shen, Y., Sun, X., and Ji, R. Learning dynamic prior knowledge for text-to-face pixel synthesis. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 5132–5141, 2022a.

Peng, J., Pan, H., Zhou, Y., He, J., Sun, X., Wang, Y., Wu, Y., and Ji, R. Towards open-ended text-to-face generation, combination and manipulation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 5045–5054, 2022b.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Rowan, W., Huber, P., Pears, N., and Keeling, A. Text2face: A multi-modal 3d face model. *arXiv preprint arXiv:2303.02688*, 2023.

Saha, R., Duke, B., Shkurti, F., Taylor, G. W., and Aarabi, P. Loho: Latent optimization of hairstyles via orthogonalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1984–1993, 2021.

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Schwarz, K., Liao, Y., Niemeyer, M., and Geiger, A. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33: 20154–20166, 2020.

Shi, Y., Wang, P., Ye, J., Long, M., Li, K., and Yang, X. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Srinivasan, P. P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., and Barron, J. T. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7495–7504, 2021.

Sun, J., Li, Q., Wang, W., Zhao, J., and Sun, Z. Multi-caption text-to-face synthesis: Dataset and algorithm. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2290–2298, 2021.

Sun, J., Deng, Q., Li, Q., Sun, M., Ren, M., and Sun, Z. Anyface: Free-style text-to-face synthesis and manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18687–18696, 2022.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Toshpulatov, M., Lee, W., and Lee, S. Generative adversarial networks and their application to 3d face generation: A survey. *Image and vision computing*, 108:104119, 2021.

Wang, H., Du, X., Li, J., Yeh, R. A., and Shakhnarovich, G. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12619–12629, 2023a.

Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., and Zhu, J. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023b.

Wu, M., Zhu, H., Huang, L., Zhuang, Y., Lu, Y., and Cao, X. High-fidelity 3d face generation from natural language descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4521–4530, 2023.

Xia, W. and Xue, J.-H. A survey on deep generative 3d-aware image synthesis. *ACM Computing Surveys*, 2023.

Xia, W., Yang, Y., Xue, J.-H., and Wu, B. Tedigan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2256–2265, 2021.

Xia, W., Zhang, Y., Yang, Y., Xue, J.-H., Zhou, B., and Yang, M.-H. Gan inversion: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3121–3138, 2022.

Yin, W., Liu, Y., Shen, C., and Yan, Y. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5684–5693, 2019.

Yu, C., Lu, G., Zeng, Y., Sun, J., Liang, X., Li, H., Xu, Z., Xu, S., Zhang, W., and Xu, H. Towards high-fidelity text-guided 3d face generation and manipulation using only images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15326–15337, 2023.

Zhang, H., Xu, Y., Dai, T., Tang, T. C.-K., et al. Fd-nerf: Semantics-driven face reconstruction, prompt editing and relighting with diffusion models. *arXiv preprint arXiv:2306.00783*, 2023a.

Zhang, K., Riegler, G., Snavely, N., and Koltun, V. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

Zhang, L., Qiu, Q., Lin, H., Zhang, Q., Shi, C., Yang, W., Shi, Y., Yang, S., Xu, L., and Yu, J. Dreamface: Progressive generation of animatable 3d faces under text guidance. *arXiv preprint arXiv:2304.03117*, 2023b.

Zhou, Y. Generative adversarial network for text-to-face synthesis and manipulation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2940–2944, 2021.

Zhou, Y., Ji, R., Sun, X., Su, J., Meng, D., Gao, Y., and Shen, C. Plenty is plague: Fine-grained learning for visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 44(2):697–709, 2019.

Zhou, Y., Ji, R., Luo, G., Sun, X., Su, J., Ding, X., Lin, C.-W., and Tian, Q. A real-time global inference network for one-stage referring expression comprehension. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

In this supplement, we first introduce the comparison methods and their experimental setup in Appendix A. Subsequently, we provide a comprehensive description of the evaluation metrics in Appendix B. We also present additional visualization results in Appendix C and Appendix D.

A. Introduction of Comparison Methods

In this section, we introduce the comparison methods used in the experiment and their implementation settings.

Latent3D (Canfes et al., 2023) builds upon the TB-GAN (Gecer et al., 2020), a generative model that inputs one-hot encoded facial expressions along with a random noise vector, and then outputs shape, shape-normal, and texture images. Typically, Latent3D optimizes the offset Δc for the intermediate layer $\mathbf{c}$ of TB-GAN, directing how the target attributes, as specified by the text prompt, are enhanced. It employs a combination of a CLIP-based loss, an identity loss and an $L2$ loss as

$$\arg\min_{\Delta \mathbf{c} \in \mathcal{C}} \mathcal{L}_{\text{CLIP}} + \lambda_{\text{ID}} \mathcal{L}_{\text{ID}} + \lambda_{\text{L2}} \mathcal{L}_{\text{L2}}, \tag{23}$$

where λ_{ID} and λ_{L2} are the hyper-parameters of $\mathcal{L}_{ID}$ and $\mathcal{L}_{L2}$, respectively. We use the official implement code [1] for 3D face generation and manipulation. Following their suggestions, we set $\lambda_{ID} = 0.01$, $\lambda_{L2} = 0.0$ with a learning rate of 0.01 for 100 epochs per text.

Describe3D (Wu et al., 2023) builds a dataset comprising 1,627 3D face models, each annotated with 25 labels representing facial attributes. This dataset is then used for training the entire generation pipeline, which consists of three stages: text parsing, concrete synthesis and abstract synthesis. The text parser is used to encode the input natural language into a descriptive code d, which is then divided into shape-related code d_S and texture-related code d_T. The shape-related code d_S is employed to predict 3DMM (Blanz & Vetter, 2023) parameters, while the texture-related code d_T serves as a conditional input for StyleGAN2 (Karras et al., 2020) to generate texture maps. This process is known as concrete synthesis. To boost performance, Describe3D incorporates additional optimization to finetune both the 3DMM parameters and texture parameters during the abstract synthesis stage. We use the official code [2] and the released model to generate 3D faces.

ClipFace (Aneja et al., 2023) also leverages the geometric expressiveness of 3D morphable models but introduces a self-supervised generative model to synthesize textures and expression parameters for the morphable model. Given a textured mesh with texture code $\mathbf{w_{init}} = \{\mathbf{w^1_{init}}, ..., \mathbf{w^{18}_{init}}\} \in \mathbf{R}^{512\times 18}$, Clipface predict the offsets formulated as:

$$\mathbf{w}^*_{\text{delta}}, \boldsymbol{\psi}^*_{\text{delta}} = \underset{\text{Wdelta}}{\arg\min} \mathcal{L}_{\text{delta}}, \tag{24}$$

where $\mathbf{w}_{\text{delta}}$ and $\boldsymbol{\psi}_{\text{delta}}$ are the optimized offsets for texture and expression, respectively. $\mathcal{L}_{\text{delta}}$ is the full training loss containing a CLIP-based loss for enhancing attributes depicted in the prompt and a regularization term for the facial expression. In practice, they employ a 4-layer MLP architecture for the mappers and we use the code from their official implement [3] in the experiments.

TediGAN (Xia et al., 2021) can perform text-guided 2D image generation by optimizing a random sampled noise $z*$ via

$$\begin{aligned} \mathbf{z}^* = \arg\min_{\mathbf{z}} ||\mathbf{x} - G(\mathbf{z})||_2^2 + \lambda_1' ||F(\mathbf{x}) - F(G(\mathbf{z}))||_2^2 \\ + \lambda_2' ||\mathbf{z} - E_v(G(\mathbf{z}))||_2^2 + \lambda_3' \mathrm{L}_{CLIP}, \end{aligned} \tag{25}$$

$$\mathrm{L}_{CLIP} = 1 - \langle E_T(T), E_I(G(z)) \rangle, \tag{26}$$

where x is the original image of z, G is the generator, F is VGG network (Simonyan & Zisserman, 2014). λ_1', λ_2' and λ_3' are the loss weights corresponding to the perceptual loss, regularization term and CLIP loss, respectively. E_v is the introduced inversion encoder, E_T and E_I are the CLIP text encoder and image encoder, respectively, $\langle \cdot, \cdot \rangle$ is the cosine similarity. Following the instructions of the official implement [4], we set $\lambda_1 = 5e-5$, $\lambda_2 = 2.0$ and $\lambda_3 = 1.0$. The learning rate is set to 1e-2 and the number of optimization steps for each instruction is 200.

[1] https://github.com/catlab-team/latent3D_code
[2] https://github.com/zhuhao-nju/describe3d
[3] https://github.com/shivangi-aneja/ClipFace
[4] https://github.com/IIGROUP/TediGAN

StyleCLIP (Patashnik et al., 2021) is another text-guided image generation and manipulation method concurrent to TediGAN, which turns to optimize the latent code w_s via

$$\arg\min_{w\in\mathcal{W}+} 1 - \langle E_T(T), E_I(G(z))\rangle. \tag{27}$$

We use the official code [5] and the learning rate is set to 0.1 with 200 optimization steps.

OpenFace (Peng et al., 2022b) is a learning-based method designed for 2D text-to-face generation task. It first constructs an effective multi-modal latent space and then directly maps a text description to a latent code. This code is then fed to a StyleGAN architecture and can perform text-guided face generation, combination and manipulation. We follow the official code [6] and use the released model weights to generate single-view text-to-2D face.

PixelFace (Peng et al., 2022a) employs pixel regression for face generation, wherein each pixel value is predicted based on the latent code and initialized pixel embeddings. The pixel embeddings are constructed using Fourier features and coordinate embeddings. To incorporate text features, PixelFace proposes a cross-modal dependency based dynamic parameter generation module to generate dynamic knowledge for pixel synthesis. We use the official code [7] and the released model for single-view generation.

B. Evaluation Metric

We utilize both 2D and 3D metrics to provide a comprehensive validation of our E^3-FaceNet. When comparing with 3D methods, we randomly sample 200 text descriptions for each dataset, *i.e., CelebAText-HQ* (Sun et al., 2021) and *FFHQ-Text* (Zhou, 2021). For each description, we generate a corresponding 3D face, then render it from five different poses. The evaluation scores for the 3D metrics are calculated inside each group, then the average score across all 200 groups is obtained. As the texts in CelebAText and FFHQ-Text are quite different from those in the Multi-Modal CelebA-HQ dataset, they can be used for cross-dataset experiments. Note that in this comparison, all the models perform zero-shot generation, thus the generalization and robustness can be effectively assessed. When comparing with 2D methods, we randomly select 30,000 descriptions from *MMCelebA* (Xia et al., 2021) and perform single-view generation for each text prompt. In this part, we compare E^3-FaceNet with a bunch of strong T2D Face generation methods trained on this dataset.

B.1. FID & KID

Fréchet Inception Distance (**FID**) (Heusel et al., 2017) and *Kernel Inception Distance* (**KID**) (Bińkowski et al., 2018) are two predominant metrics for assessing the quality of generated images. FID is defined by the Fréchet distance between the feature from the real and generated images, which can be formulated as

$$FID = ||\mu_r - \mu_g||^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}) \tag{28}$$

where Tr is the trace and (μ_r, Σ_r) and (μ_g, Σ_g) are the mean and covariance obtained from the real images and generated images feature, respectively. We use Inception-v3 (Szegedy et al., 2016) to extract the image feature and lower FID indicates better perceptual quality.

KID can be viewed as a Maximum Mean Discrepancy (MMD) (Bińkowski et al., 2018) directly on input images with the kernel

$$K(\mathcal{I}_r, \mathcal{I}_g) = k(\theta(\mathcal{I}_r), \theta(\mathcal{I}_g)) \tag{29}$$

$$k(x, y) = (\frac{1}{d}x^T y + 1)^3 \tag{30}$$

where $\mathcal{I}_r$ and $\mathcal{I}_g$ are the real and generated images, respectively, and θ is the function mapping images to Inception representations. In the main paper, we report KID$\times$1000 and a lower score indicates better generation quality.

B.2. Multi-View Identity Consistency

In line with previous studies (Wu et al., 2023; Yu et al., 2023), we employ ArcFace (Schroff et al., 2015) to extract face features from the rendered images. We calculate the average cosine similarities between any two out of five images inside

[5] https://github.com/orpatashnik/StyleCLIP
[6] https://github.com/pengjunn/OpenFace
[7] https://github.com/pengjunn/PixelFace

This man has big lips.	This woman has big lips.
This man has big nose.	This woman has big nose.
This man has black hair.	This woman has black hair.
This man has blond hair.	This woman has blond hair.
This man has brown hair.	This woman has brown hair.
This man has bushy eyebrows.	This woman has bushy eyebrows.
This man is chubby.	This woman is chubby.
This man has double chin.	This woman has double chin.
This man has goatee.	This woman has blue eyes.
This man has gray hair.	This woman has beard.
This man has mustache.	This woman has heavy makeup.
This man has narrow eyes.	This woman has pale skin.
This man has no beard.	This woman has gray hair.
This man has pale skin.	This woman has bags under her eyes.
This man has heavy makeup.	This woman wears lipsticks.
This man has grey hair and wearing eyeglasses.	This woman is attractive and has blond hair.
This man has sideburns and black hair.	This woman has big lips and a pointy nose.
This man has double chin and mustache.	This woman has bags under her eyes and bushy eyebrows.
This man has bushy eyebrows and a mustache.	This woman has heavy makeup, narrow eyes, and an oval face.
This man is wearing necktie and has beard.	This woman has wavy hair, high cheekbones, and is wearing earrings.
This man has gray hair and goatee.	This woman has black hair, bags under her eyes, and is smiling.
This man has a big nose and big lips.	This woman has pale skin and brown hair.
This man has blond hair and is young.	This woman has bangs, with mouth slightly open.
This man has grey hair and is chubby.	This woman has brown hair, rosy cheeks, and is wearing earrings.
This man has pale skin and wears lipsticks.	This woman has brown hair, pale skin, and is wearing glasses.

Figure 7. Editing instructions for text-driven 3D face manipulation comparison. We collect a total of 25 prompts for both male (left) and female (right) faces, including instructions for single-attribute editing and multi-attribute editing.

each group, and the multi-view identity consistency (MVIC) is obtained by computing the mean similarity score across 200 groups for each method on each dataset. A higher MVIC score indicates a greater level of multi-view consistency.

B.3. Semantic-Alignment

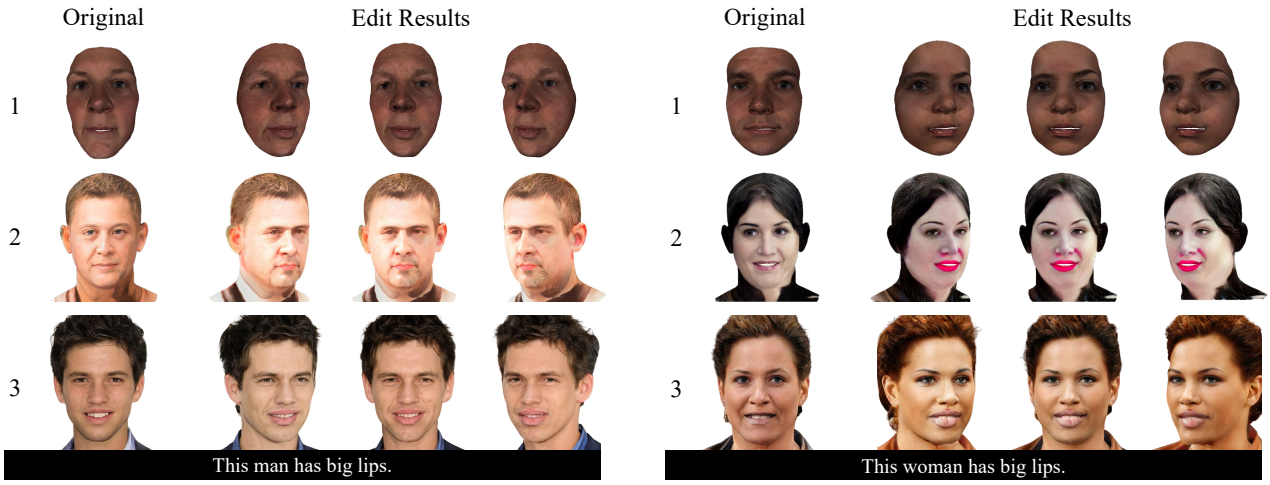
To evaluate the semantic alignment between text descriptions and the generated faces, we employ CLIP (Radford et al., 2021) to extract features from both the rendered images and text descriptions. We then measure their similarity in the CLIP space. For 3D methods, we calculate the average CLIP similarity for each group and compute the overall average CLIP similarity across all generated results. In the case of 2D methods focused on single-view generation, we calculate the CLIP score between each text description and its corresponding generated image, and then derive the mean score for this metric. A higher score indicates better semantic alignment.

B.4. User Study

There is currently no standardized metric specifically designed for evaluating face manipulation. Previous studies, such as ClipFace (Aneja et al., 2023), rely on FID which is widely used for quantitative comparison in image generation tasks. However, comparing the two methods we are evaluating, namely Latent3D (Canfes et al., 2023) and ClipFace (Aneja et al., 2023), is problematic because they are trained on different datasets. Using the aforementioned metrics under these circumstances could lead to unfair comparisons. Additionally, obtaining a reliable FID score requires a large collection of generated images. Given that both methods under evaluation rely on inference-time or example-dependent optimization, producing a sufficient quantity of edited faces is a time-intensive task. Therefore, to ensure a more accurate and effective quantitative comparison, we also carry out an additional user study, following (Xia et al., 2021; Yu et al., 2023).

B.4.1. Editing Instruction

We manually collected a set of 50 editing instructions to perform face editing across various attributes, consisting of 25 instructions for male faces and 25 for female faces. In each group, we create 15 prompts for single-attribute manipulation and 10 prompts for multi-attribute manipulation. Additionally, we include some unconventional edits, such as applying "*lipstick*" to a man or adding "*beard*" to a woman, to further assess the editing capabilities of each method. The editing



(a) Examples for Male Face Manipulation

(b) Examples for Female Face Manipulation

Figure 8. Examples of user study. On the left is the original face, followed by the edited results of the given prompt under each group. To ensure anonymity, we only present these generation results with indexes for recording purposes.

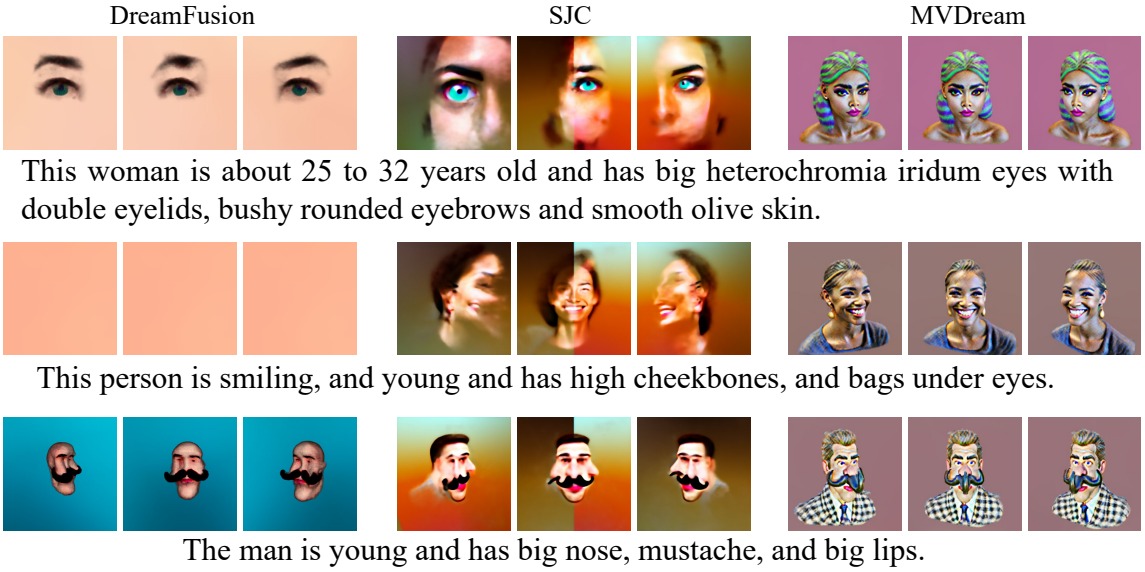


Figure 9. Results of generating 3D face of the current diffusion-distilled-based methods. Although these methods succeed in generating 3D objects, they cannot synthesize 3D human faces with high fidelity.

prompts are presented in Figure 7.

B.4.2. Detailed Experiment Settings

For Latent3D, we initially generate a 3D face and then perform optimization based on the provided prompt. For ClipFace, we follow their templates and construct the editing instruction as *"A photo of a male face with thick lips"* or *"A photo of a male face with narrow eyes"*. We train specific mappers for each instruction and randomly select a textured female face and a male face for manipulation. Regarding E^3-FaceNet, we generate the initial 3D face using two prompts, *i.e., "This is a man"* and *"This is a woman"*. Then, we edit the generated face using the proposed Style Code Enhancer, with the default edit weight set to 0.5.

B.4.3. EVALUATION CRITERIA

For each editing instruction, we render manipulated 3D faces from three different views: left, front, and right. Participants are presented with the original face and all editing results. We invite 25 individuals to assess the edited faces based on three criteria: (1) *Identity Preservation* (**IP**), selecting the edit result that best preserves the facial identity of the original face; (2) *Semantic Alignment* (**SA**), selecting the edit result that most closely aligns with the given instructions; and (3) *Editing Quality* (**EQ**), selecting the edit result with the highest perceptual quality. We calculated the percentage of times each method was chosen as the best performer by users, based on the previous mentioned criteria. As shown in Figure 8, the edit results are presented anonymously to the users.

C. Additional Comparisons

Recently, researchers have been focused on transferring pre-trained 2D image-text diffusion models to synthesize 3D objects without using any 3D data (Poole et al., 2022; Wang et al., 2023a; Shi et al., 2023; Wang et al., 2023b; Lin et al., 2023). While these methods have achieved competitive quality in 3D content creation, they fall short in generating photo-realistic human faces. Here we select three representative methods, *i.e.,* Dreamfusion (Poole et al., 2022), SJC (Wang et al., 2023a) and MVDream (Shi et al., 2023), for 3D face generation. However, as depicted in Figure 9, these methods struggle to generate high-quality 3D faces conditioned on the given input. Additionally, they all rely on test-time tuning, which significantly increases the inference time. For example, Dreamfusion takes approximately 40 minutes, and SJC takes around 20 minutes for inference on an RTX 3090 GPU. Therefore, these methods are not included in the comparison for T3D.

We also present additional generation comparisons with both 3D and 2D face generation methods mentioned in the main paper in Figure 10 and Figure 12. Our method excels in synthesizing high-quality 3D faces that align effectively with the given prompts and maintain consistency across various viewing angles. Additionally, our E^3-FaceNet method is capable of generating 3D faces with multiple facial assets, enhancing the diversity and realism of the generated results.

More results for text-driven 3D Face manipulation in shown in Figure 11, in which we show some editing examples for female faces. This figure also demonstrates the limited edit ability of Latent3D, as neither *"heavy makeup"* nor *"pale skin"* is generated in its editions. As for ClipFace, the identity changes significantly and struggles to achieve multi-view consistency. As shown from editing results of *"heavy makeup"*, ClipFace generates unnatural texture maps and the edited face appears distorted in the right view. In stark contrast, our E^3-FaceNet cannot only effectively respond to the text semantics but also maintain high image quality and multi-view consistency.

D. Additional Generation

Meanwhile, our proposed E^3-FaceNet is capable of generating diverse 3D faces conditioned on the same input sentence, as demonstrated in Figure 13. In each row, we sample different latent codes using different random seeds while inputting the same text prompts. Impressively and remarkably, these generated results showcase that E^3-FaceNet not only synthesizes high-quality and semantically aligned 3D faces but also possesses the ability for diverse 3D face generation.

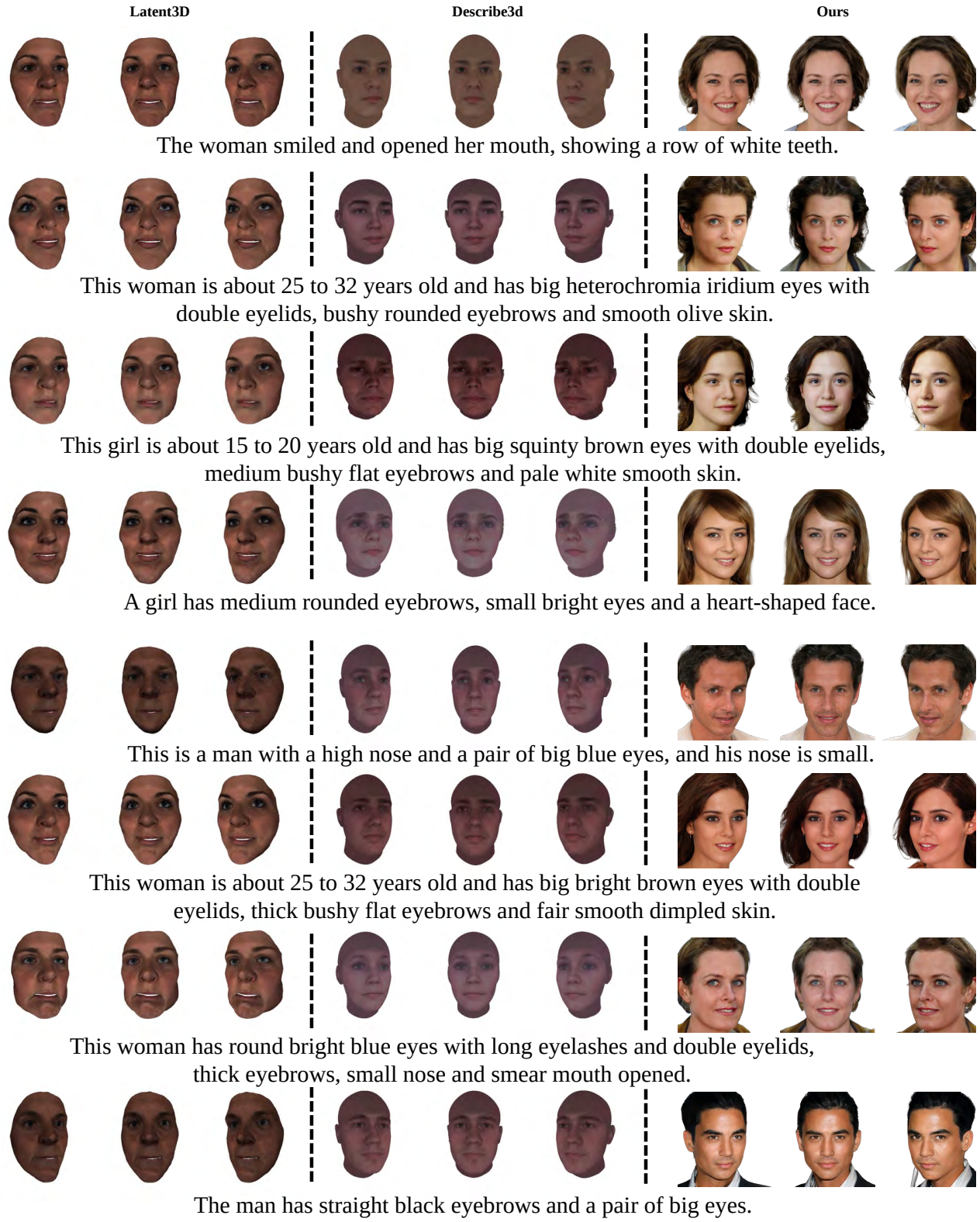


Figure 10. More comparison for 3D face generation. E^3-FaceNet excels in generating fine-grained facial attributes while maintaining multi-view consistency and preserving facial identity.

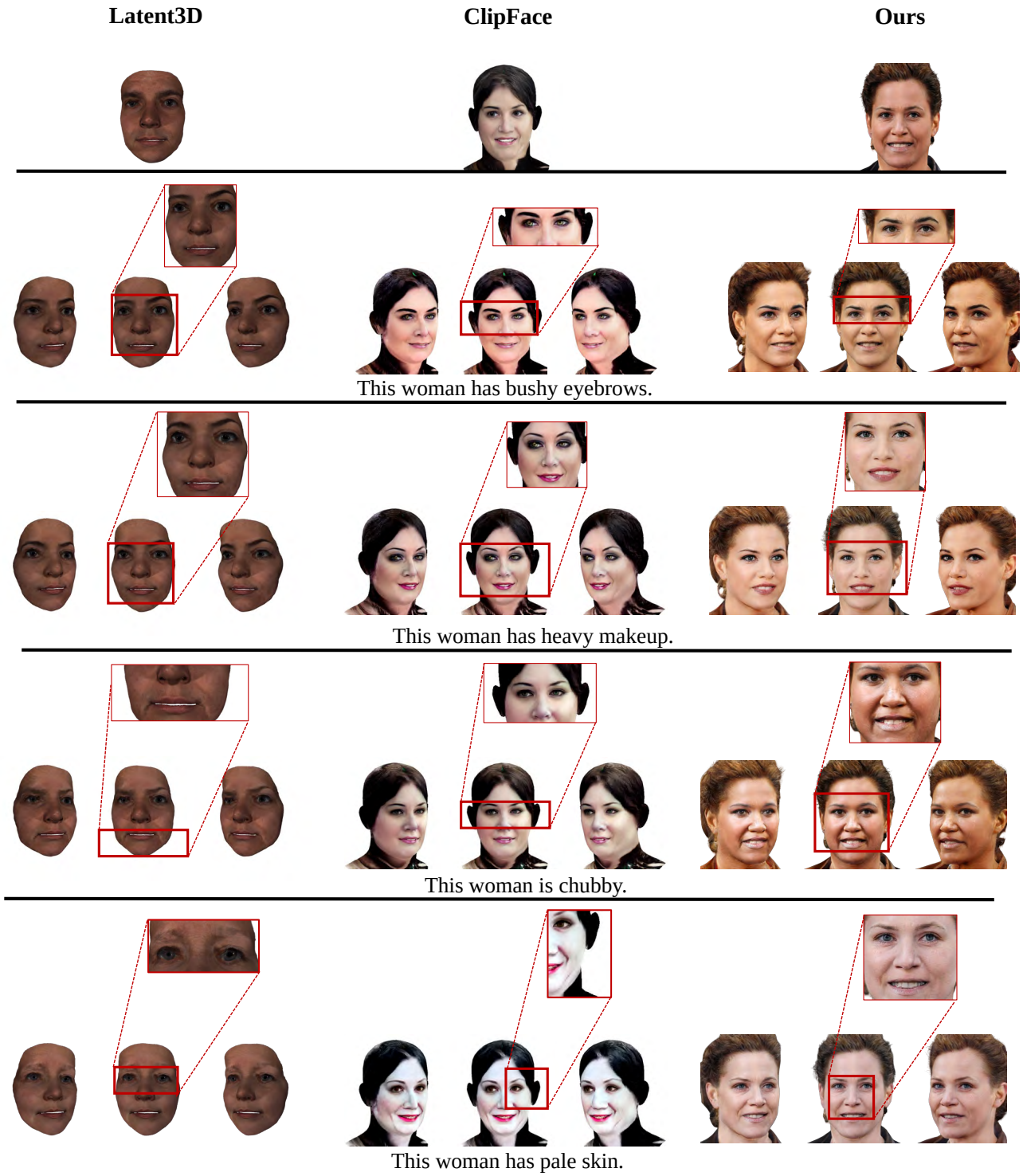


Figure 11. More comparisons for 3D face manipulation. E^3-FaceNet can faithfully edit the facial attributes described in the editing instructions while keeping other attributes unchanged. This ensures precise and targeted attribute manipulation in the generated 3D faces.

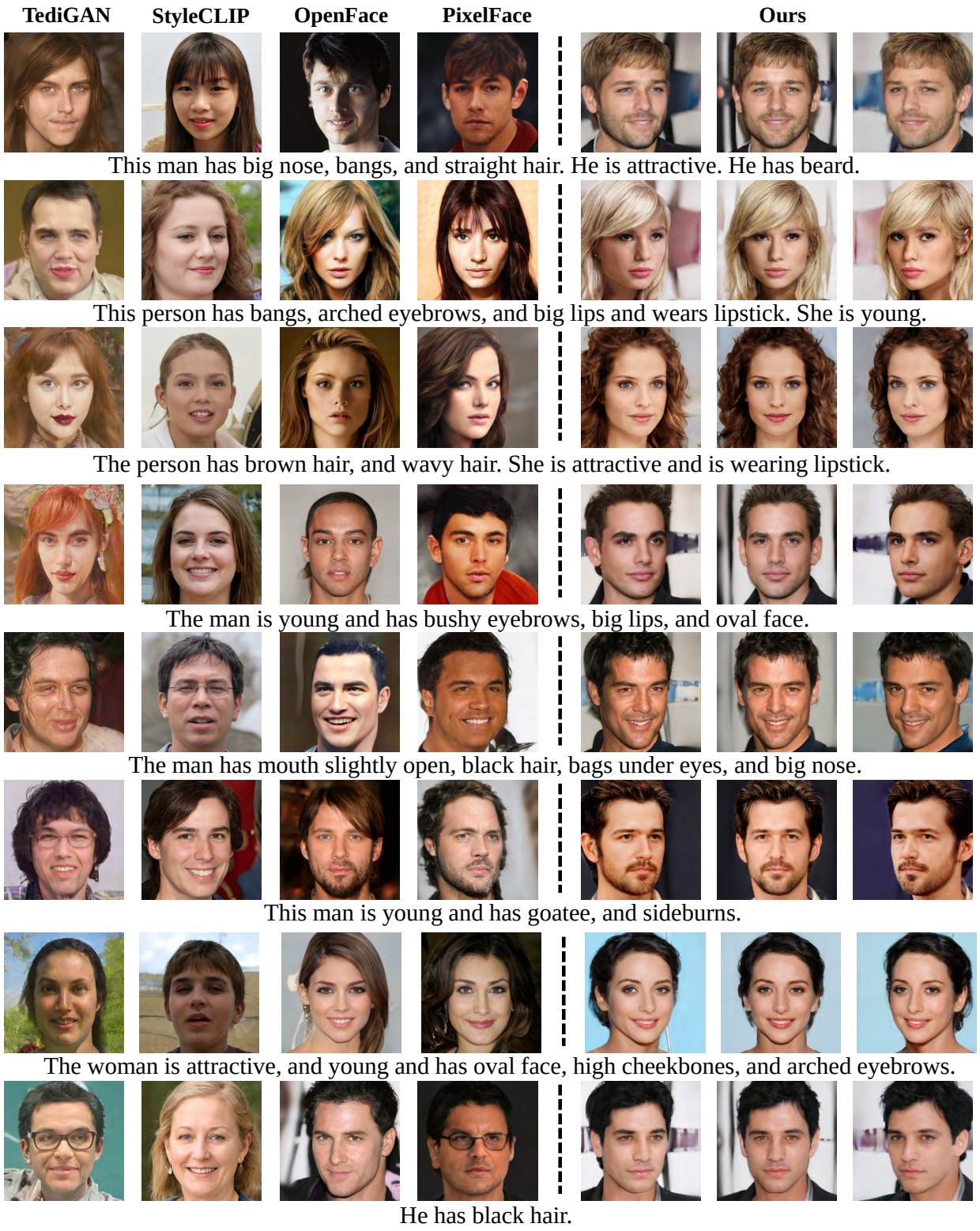


Figure 12. More comparisons with T2D Face methods. Our method showcases remarkable proficiency in maintaining high-quality visual output and ensuring semantic consistency across different views.

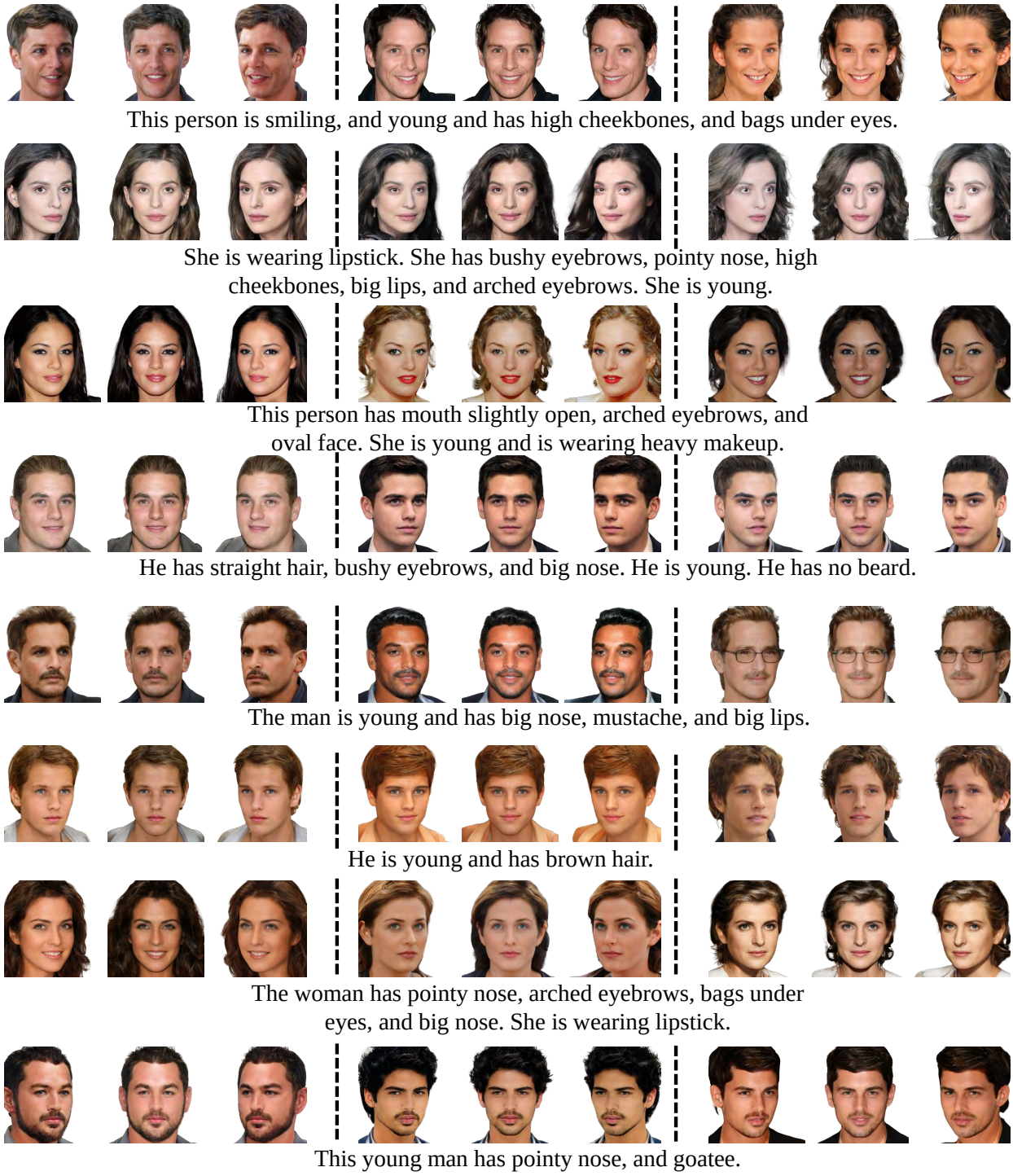


Figure 13. Diverse generation results of E^3-FaceNet. These results demonstrate that our proposed method can not only generate high-quality 3D faces that align well with the given prompts, but also achieve impressive diversity in the generated results.