SketchFaceGS: Real-Time Sketch-Driven Face Editing and Generation with Gaussian Splatting

Bo Li[1] Jiahao Kang[1] Yubo Ma[1] Feng-Lin Liu[2] Bin Liu[1] Fang-Lue Zhang[3] Lin Gao[2*]

[1]Nanchang Hangkong University
[2]Institute of Computing Technology, Chinese Academy of Sciences
[3]Victoria University of Wellington

bolimath@gmail.com, 2307070100013@stu.nchu.edu.cn, 2304081200005@stu.nchu.edu.cn,
liufenglin21s@ict.ac.cn, nyliubin@nchu.edu.cn, z.fanglue@gmail.com, gaolin@ict.ac.cn

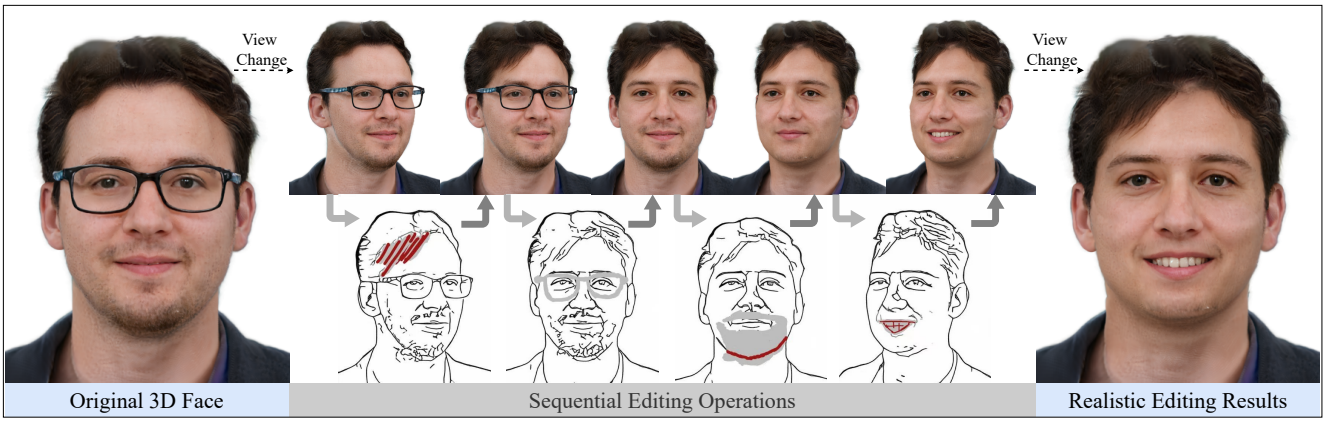

Figure 1. SketchFaceGS presents a real-time, sketch-driven editing framework tailored for 3D Gaussian Splatting (3DGS) facial heads. Starting with an original 3D face (a), sequential editing operations (b) are applied to targeted local regions. Our approach delivers photo-realistic editing results (c) while preserving the 3D consistency.

Abstract

3D Gaussian representations have emerged as a powerful paradigm for digital head modeling, achieving photorealistic quality with real-time rendering. However, intuitive and interactive creation or editing of 3D Gaussian head models remains challenging. Although 2D sketches provide an ideal interaction modality for fast, intuitive conceptual design, they are sparse, depth-ambiguous, and lack high-frequency appearance cues, making it difficult to infer dense, geometrically consistent 3D Gaussian structures from strokes—especially under real-time constraints. To address these challenges, we propose SketchFaceGS, the first sketch-driven framework for real-time generation and editing of photorealistic 3D Gaussian head models from 2D sketches. Our method uses a feed-forward, coarse-to-fine architecture. A Transformer-based UV feature-prediction module first reconstructs a coarse but geometrically consistent UV feature map from the input sketch, and then a 3D UV feature enhancement module refines it with high-frequency, photorealistic detail to produce a high-fidelity 3D head. For editing, we introduce a UV Mask Fusion technique combined with a layer-by-layer feature-fusion strategy, enabling precise, real-time, free-viewpoint modifications. Extensive experiments show that SketchFaceGS outperforms existing methods in both generation fidelity and editing flexibility, producing high-quality, editable 3D heads from sketches in a single forward pass.

* Corresponding author.

1. Introduction

Accurate 3D facial modeling remains a challenging problem in computer graphics and computer vision, underpinning applications from cinematic characters to immersive avatars. Prior mesh-based pipelines [1, 40, 44] typically rely on labor-intensive workflows to produce photorealistic faces that capture both geometry and appearance, making them impractical for non-experts and costly to achieve photoreal quality. Recent NeRF-based approaches [2, 17, 38, 45] leverage 3D GANs for controllable face synthesis; however, their practical adoption is often limited by costly rendering and 3D inconsistencies introduced by super-resolution modules. Although 3D Gaussian splatting [30] addresses real-time rendering, existing GAN-based Gaussian head generation methods [24, 31] and text-driven face-editing approaches [48, 49, 52] lack fine-grained, convenient control for detailed facial edits.

We propose an intuitive, sketch-driven approach for 3D face editing that leverages 3D Gaussian Splatting. To overcome high rendering costs and error accumulation in existing 3D sketch-based methods (e.g., SketchFaceNeRF [15]), we introduce a unified feed-forward framework that directly generates and interactively edits photorealistic 3D Gaussian head models from single-view hand-drawn sketches. Our method bridges the gap between sparse 2D sketches and dense 3D Gaussian representations via a coarse-to-fine pipeline. In the coarse stage, two parallel Transformer branches extract geometric features from the sketch and appearance features from a reference image; a fusion network then reconciles geometry–appearance conflicts across identities to produce a geometry-consistent, color-coherent coarse UV map. In the fine stage, we use a pretrained 3D-GAN as a high-frequency texture prior: a U-Net maps the coarse UV into a global latent and multi-scale spatial modulation parameters, which are applied layer-wise to condition synthesis and elevate the coarse representation into a renderable, photorealistic 3D face.

For interactive editing, we introduce UV Mask Fusion. User edits (draw/erase) are converted to 2D pixel masks and back-projected into 3D to identify the 3D Gaussians that contribute to the edited pixels. Mapping those Gaussians into the canonical UV map produces precise UV masks that separate edited and unedited regions. Instead of compositing in 3D Gaussian Splatting space, we resample each UV mask to match the resolution of the corresponding generator layer and perform layer-wise fusion in the generator's multi-scale feature maps. This suppresses seam artifacts and yields natural, seamless merges. By leveraging the model's end-to-end generative capability, our approach enables continuous, free-viewpoint, real-time editing.

Our main contributions are:

- We introduce the first unified, end-to-end framework to generate and interactively edit photorealistic 3D Gaussian head models from a single sketch, providing an intuitive, artist-friendly workflow.
- We propose a coarse-to-fine pipeline where parallel Transformer branches extract geometry and appearance from the sketch and reference image respectively to produce a coarse UV map. A 3D-GAN–based modulation module then enhances it with high-frequency, photorealistic details.
- We present UV Mask Fusion, a real-time editing mechanism that performs layer-wise feature fusion in the generator's multi-scale maps, avoiding spatial composition artifacts and enabling natural, stable, continuous free-viewpoint editing.

2. Related Work

2.1. 3D-aware Portrait Synthesis

3D-GANs [5, 16] are more popular in 3D-aware portrait synthesis compared with Diffusion Models [22] because of the training dependence on 2D images instead of scarce 3D data. Early 3D-GAN methods render pixels directly from 3D representations, including meshes [8, 26], voxels [14, 21], and Neural Radiance Fields [3, 46], with later works introducing additional convolution modules [37, 39, 53] to improve fidelity and resolution. Subsequent approaches [4, 17, 42] adopted StyleGAN [27] as the generator backbone. Among them, EG3D [5] proposed a triplane-based hybrid neural rendering with a super-resolution module for upsampling. However, its volume rendering remained inefficient and the super-resolution stage introduced cross-view inconsistencies. To address these issues, recent work leverages 3D Gaussian Splatting (3DGS) [30], a new explicit representation enabling native high-resolution, real-time head synthesis. GSGAN [24] introduces a hierarchical multiscale Gaussian generator that regularizes the Gaussians across levels, enabling coarse-to-fine 3D modeling. GGHead [31] binds 3D Gaussians to a template mesh and predicts their attributes as UV maps within a StyleGAN framework, achieving real-time and arbitrary resolution rendering. Our work builds on these latest advances in generative 3D Gaussian head synthesis.

2.2. Sketch-based Face Modeling

Sketching has been widely used for both facial image generation [6, 9, 32, 33] and editing [7, 25, 55, 56]. In 3D face modeling, some methods predict the coefficients of parametric models and refine surface details with displacement maps [18, 19, 54], while others use sketches to guide the deformation of a template mesh to produce diverse 3D faces [13, 19, 35]. Although effective for mesh geometry, these approaches struggle to produce photorealistic images due to the difficulty of estimating texture, materials, and lighting. To address this issue, recent re-

search propose NeRF-based generation methods. For instance, S3D [47] generates a semantic mask from a face sketch and then synthesizes a 3D face using a semantic-map-based generative network like pix2pix3d [11], but this sketch-to-mask mapping loses fine-grained texture details. SketchFaceNeRF [15] predicts a Facial NeRF directly from a sketch using a tri-plane network and enables 3D face editing via mask fusion. However, its coarse tri-plane prediction reduces generation quality, and editing requires time-consuming per-instance optimization. Additionally, sequential edits sometimes cause error accumulation, preventing its use in real-time interactive facial creation. In contrast, our model is a feed-forward framework that enables real-time, optimization-free editing, where precise UV Mask Fusion effectively prevents error accumulation.

2.3. Feed-forward 3D Reconstruction Model

Neural representations, such as NeRF [36] and 3DGS [30], have revolutionized 3D reconstruction with their high-quality novel view synthesis. To eliminate costly per-scene optimization, Large Reconstruction Models (LRMs) [23] series introduced Transformer-based feed-forward architectures, enabling fast and generalizable reconstruction of detailed 3D models from sparse inputs. This powerful paradigm has rapidly extended from general object reconstruction to the domain of human body reconstruction. For instance, Human-LRM [51] utilizes a Transformer to decode tri-plane-NeRF representations, while Human-Splat [43] employs a latent reconstruction Transformer to directly predict 3DGS human avatars from multi-view images generated by a diffusion model. In the more fine-grained task of head modeling, GAGAvatar [10] proposed a one-shot, 3D Gaussian-based method for head avatar generation, though it still requires 2D neural post-processing for ideal animation. In contrast, LAM [20] builds Gaussian features on FLAME vertices and densifies them via subdivision and interpolation to achieve fast and realistic 3D head reconstruction. Inspired by LAM's efficient feed-forward architecture, our model similarly designs a Transformer-based feature extraction network, aiming to improve the head's realism and its structural consistency with the input sketches.

3. Method

We propose a real-time 3D head generation and editing framework driven by sketches. As shown in Fig. 2, given a single portrait image and a hand-drawn sketch as inputs, our method enables intuitive and efficient manipulation of 3D head models by simply editing the sketch. Section 3.1 introduces some prerequisite knowledge related to our method. Section 3.2 provides a detailed description of the steps involved in our method during the generation phase, whereas the editing component is thoroughly explained in Section 3.3. Section 3.4 elaborates on the loss at each stage.

3.1. Preliminaries

3D Gaussian Splatting. Our work is based on 3D Gaussian Splatting [30], a point-based 3D representation that assigns 5 core attributes to each point: position $\mu \in \mathbb{R}^3$, scale $s \in \mathbb{R}^3$, rotation quaternion $q \in \mathbb{R}^4$, opacity $\alpha \in \mathbb{R}$, and color $c \in \mathbb{R}^3$. A Gaussian ellipsoid is formulated as:

$$G(x) = e^{-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)}, \quad \text{with } \Sigma = RSS^{\mathrm{T}}R^{\mathrm{T}} \tag{1}$$

where x is the world coordinates, and the scaling matrix S and rotation matrix R are derived from the scaling s and quaternion q. The color C of a pixel in image space is computed by blending N overlapping sorted points:

$$C = \sum_{i=1}^{N} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{2}$$

This representation enables high-fidelity, real-time rendering, making it an ideal choice for interactive facial content creation.

Generative Prior for 3D Gaussian Heads. To leverage the power of generative models, we adopt GGHead [31] as our core generative prior. GGHead employs a StyleGAN2 [29] backbone that maps a latent code $\mathbf{w} \in \mathcal{W}^+$ to a multi-channel UV map. This map directly parameterizes the attributes of a full 3D Gaussian head. In our framework, we repurpose GGHead's pre-trained StyleGAN backbone not for random generation, but as a powerful and controllable decoder that translates our predicted UV features into a high-fidelity 3D head model.

3.2. Sketch-based Gaussian Head Generation

Directly mapping a sparse sketch to a 3D head is highly ill-posed due to the inherent ambiguity between sparse line structures and dense 3D geometry. To progressively resolve this, we design a feed-forward, coarse-to-fine architecture: although the coarse stage predicts a UV feature map to establish a geometrically consistent foundation, it primarily captures low-frequency structural and chromatic information. The subsequent fine stage refines this representation via a GAN-based UV feature enhancement module that injects high-frequency, photorealistic details.

3.2.1. Coarse Stage: Transformer-based UV Prediction

Given a sketch I_{sketch} and a reference image I_{ref}, we predict a coarse but geometry-consistent UV feature map. To disentangle geometric and appearance information, we employ a parallel Transformer architecture with distinct branches for distinct inputs.

Inspired by the 3D-aware querying mechanism in LAM[20], we employ a Transformer backbone for sketch-to-geometry prediction. In this framework, a set of learnable queries, corresponding to the vertices of a canonical

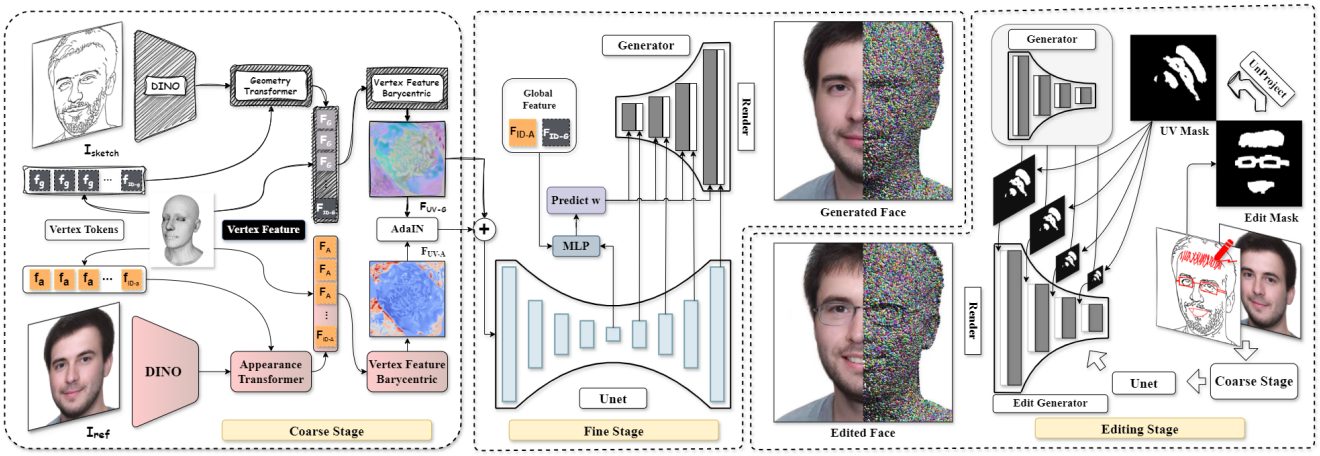

Figure 2. **Overview of the SketchFaceGS Framework.** Our method consists of two core components: (a) a sketch-based generation pipeline that translates a 2D sketch and a reference image into a photorealistic 3D Gaussian head through a coarse-to-fine process, and (b) a real-time editing pipeline that leverages a novel UV Mask Fusion method and a layer-wise feature fusion strategy to enable precise, continuous, and view-independent modifications.

head template, interact with deep features extracted from the input sketch via cross-attention to predict per-vertex geometric attributes. A similar, smaller Transformer is used to extract appearance features from the reference image. More specifically, we first extract deep features F_{sketch} and F_{ref} from the input sketch and color image using a pre-trained DINOv2[41] encoder. These image features serve as the keys and values in each Transformer's cross-attention mechanism, which operates on learnable vertex queries f_{g}, f_{a} and global identity tokens $f_{\text{ID-g}}$, $f_{\text{ID-a}}$ to produce both per-vertex features F_{G}, F_{A} and global identity vectors $F_{\text{ID-G}}$, $F_{\text{ID-A}}$:

$$F_{\text{G}}, F_{\text{ID-G}} = \mathbf{T}_{\text{G}}((f_{\text{g}}, f_{\text{ID-g}}), F_{\text{sketch}}) \tag{3}$$

$$F_{\text{A}}, F_{\text{ID-A}} = \mathbf{T}_{\text{A}}((f_{\text{a}}, f_{\text{ID-a}}), F_{\text{ref}}) \tag{4}$$

We apply barycentric interpolation to project these vertex features F_{G}, F_{A} onto the UV map, obtaining dense $F_{\text{UV-G}}$ and $F_{\text{UV-A}}$. The $F_{\text{ID-G}}$ and $F_{\text{ID-A}}$ features are utilized in the fine stage to preserve the human identity.

Notably, independent feature extraction may clash if the structures of sketch and reference differ significantly. To resolve this, we introduce a AdaIN-based alignment network G_c to align these two representations. Specifically, the network normalizes the geometric feature map $F_{\text{UV-G}}$, then applies scaling and biasing using the corresponding scalar components from appearance feature map $F_{\text{UV-A}}$, producing a new, coherent feature map:

$$F_{\text{UV-align}} = G_c(F_{\text{UV-G}}, F_{\text{UV-A}}) \tag{5}$$

Finally, the aligned appearance map $F_{\text{UV-align}}$ is concatenated with the geometric map $F_{\text{UV-G}}$ to form a coherent input for the subsequent fine stage.

3.2.2. Fine Stage: 3D UV Feature Enhancement

The coarse UV map from the previous stage establishes the foundational geometry and base color but lacks photorealistic detail. To inject high-frequency textures, we introduce a UV feature enhancement module. This module adapts the concept of generative modulation, originally popularized in 2D face restoration models such as GFP-GAN [50], and applies it to our 3D UV feature space.

Specifically, we design a U-Net that takes the coarse UV feature map as input and predicts a set of modulation parameters for our pre-trained GGHead generator. These parameters consist of a global feature vector F_{latent} and a pyramid of multi-resolution spatial features $F_{spatial}$. The feature vector F_{latent} is then aggregated with the geometric and appearance identity vectors ($F_{\text{ID-G}}, F_{\text{ID-A}}$) extracted during the coarse stage. An MLP projects this combined representation into the StyleGAN's $\mathcal{W}^+$ latent space to produce the final identity-aware latent code $\mathcal{W}$:

$$\mathcal{W} = \text{MLP}\big(\text{concat}(F_{\text{latent}}, F_{\text{ID-G}}, F_{\text{ID-A}})\big) \tag{6}$$

The predicted features guide the StyleGAN generator via two complementary mechanisms—global identity injection and localized detail modulation—to synthesize a final, detail-rich UV map, F_{output}. This map directly encodes the complete attributes for the set of 3D Gaussian primitives, which are then rendered to generate the final photorealistic image.

3.3. Sketch-based Gaussian Head Editing

Our sketch-driven geometry representation enables intuitive, interactive facial editing within the generative pipeline. Leveraging the learned features from the edited sketch, we support real-time, continuous edits to the 3D Gaussian head, including shape adjustments (e.g., facial

contours) and local feature edits (e.g., eyes and hair), producing consistent, photorealistic updates to the geometry and appearance of the generated head model. The process includes accurate localization of edits from 2D screen space to canonical UV space and coherent fusion of new and original content within the generator's feature space. The pipeline comprises two key components: UV-mask localization and layer-wise feature fusion.

UV Mask Synthesis. Our editing strategy operates within the generator's feature space to ensure seamless blending and preserve unedited regions. To achieve precise local control, we transform a user's 2D pixel-space edit into a precise mask in the parameterized UV space, where our features are defined. The user's edit (e.g., draw/erase) is first converted into a 2D pixel mask $\mathcal{M}$, which is back-projected to determine the set of 3D Gaussians contributing to the masked region. Following GaussianEditor [49], the influence weight w_i of each Gaussian is computed by accumulating its opacity–transmittance product over all masked pixels:

$$w_i = \sum_{p \in \mathcal{M}} \alpha_i(p) \cdot T_i(p), \tag{7}$$

Gaussians with nonzero weights are then projected onto the canonical UV map to generate a precise binary UV mask $\mathbf{M}_{\text{UV}}$, which is subsequently resampled to match the spatial resolution of each generator layer k, yielding a series of layer-specific masks $\mathbf{M}_{\text{UV}}^{(k)}$.

Layer-by-Layer Feature Fusion. In contrast to artifact-prone methods that composite two sets of Gaussians directly in 3D space, we propose a layer-by-layer fusion strategy that operates within the StyleGAN generator's feature space. This approach achieves a natural blend by progressively guiding the synthesis process at multiple levels of abstraction The fusion is performed at every layer of the generator. Denote the intermediate feature map of the original (unedited) head as $\mathbf{f}_k^{\text{orig}}$, and the corresponding feature map of the new (edited) head up to layer as $\mathbf{f}_k^{\text{new}}$.Instead of a simple blend, we use the features of the original head to guide the synthesis of the new one. Specifically, we create a fused feature map by selectively preserving the unedited regions from the original feature map:

$$\mathbf{f}_k^{\text{fused}} = (1 - \mathbf{M}_{\text{UV}}^{(k)}) \odot \mathbf{f}_k^{\text{orig}} + \mathbf{M}_{\text{UV}}^{(k)} \odot \mathbf{f}_k^{\text{new}}, \tag{8}$$

where $\odot$ denotes element-wise multiplication. This fused tensor then serves as the input to the next synthesis layer of the generator, which in turn produces the next feature map:

$$\mathbf{f}_{k+1}^{\text{new}} = \text{Layer}_k(\mathbf{f}_k^{\text{fused}}, \mathcal{W}_{\text{new}}). \tag{9}$$

where Layer_k is the k-th layer in StyleGAN. By iteratively applying this process, unedited regions retain stable features while edited areas are progressively updated, yielding artifact-free and view-consistent results.

3.4. Training Strategy and Loss Functions

Our model is trained in three stages, each with a tailored objective: coarse generation, fine generation, and editing. More details about the training details and loss terms are in supplemental material.

Coarse Stage. We train on synthesized multi-view pairs of the same subject. One view provides the appearance and sketch input; the other serves as the ground truth target. We introduce a temporary MLP decoder D_g (only used during training) to converts the predicted coarse UV maps into Gaussian primitives, so they can be rendered from the target viewpoint. Training encourages the rendered result to match the target image using a combination of pixel-wise alignment, perceptual similarity, and color-consistency losses (weighted 0.1, 0.0066, and 0.007).

Fine Stage. Next, we switch to single-view training to improve generalization to real images. Notably, the pre-trained backbone already offers a reliable 3D prior and prevents 2D collapse. This stage has two supervision paths: 1. Reusing the temporary decoder from the coarse stage, we guide the U-Net's modulation features using an LPIPS loss (weights 0.0066). 2. For the final rendered output from GGhead backbone, we supervise fine details and photorealism using L1, perceptual, LPIPS, and adversarial losses (weights 1, 0.0066, 0.1, and 0.4).

Editing Stage. To learn seamless fusion of edited and unedited regions, we train on synthetic heads generated by GGHead. Random masks simulate user edits, and the model blends edited content with original 3D faces. The fused output is supervised to match the unedited rendering using color-consistency, L1, perceptual, and adversarial losses (weights 1, 0.0066, 0.1, and 0.4).

4. Evaluation

We conduct extensive experiments (results in Sec. 4.1, comparisons in Sec. 4.2, ablations in Sec. 4.3) to evaluate SketchFaceGS in terms of quality, efficiency, and usability.

Implementation Details. We propose a stage-wise training strategy: the coarse stage is trained on a multi-view dataset synthesized from GGHead[31], while the fine stage is trained on the single-view FFHQ dataset[28]. Evaluation is performed on two test sets: (1) Generation set: 100 hand-drawn sketches collected from artists; (2) Editing set: 100 editing examples collected via our interactive system. For fair comparison, all baselines were re-trained or evaluated using their official codebases. For additional implementation details, please refer to the supplementary material.

4.1. Results of Our Method

Figure 3 (top) shows representative outputs synthesized from a single-view hand-drawn sketch with optional appearance references. Our pipeline reliably maps sparse stroke

into a geometrically faithful 3DGS head. Silhouettes, hairlines, and stroke-level structural cues are preserved in geometry, while the final renderings exhibit high-frequency texture and consistent illumination across novel views. Figure 3 (bottom) shows editing examples. Users can modify local regions, such as hair, glasses, and facial contours, from arbitrary viewpoints. Edits accurately follow the input sketches, and unedited regions remain unchanged and blend naturally with the modified areas.

Our feed-forward approach is robust for multi-step, iterative editing. As shown in Figure 4, users can apply long sequences of diverse edits, such as altering hairstyle, facial contour, eyebrows, mouth, and eyes. Our UV Mask Fusion and layer-wise feature fusion ensure that each modification is precisely localized. This stability makes our method exceptionally well-suited for practical, creative workflows that require continuous and repeated refinements.

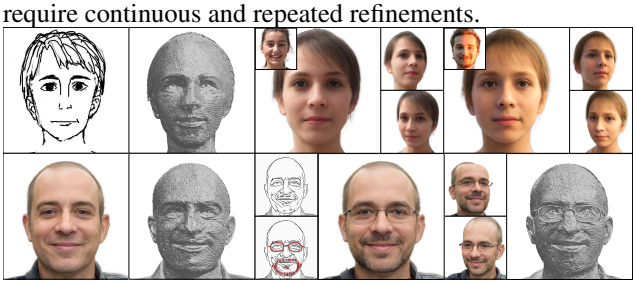

Figure 3. Given a hand-drawn sketch and a reference image, our method produces a photorealistic 3D head (Top). Our method also support detailed local editing (Bottom).

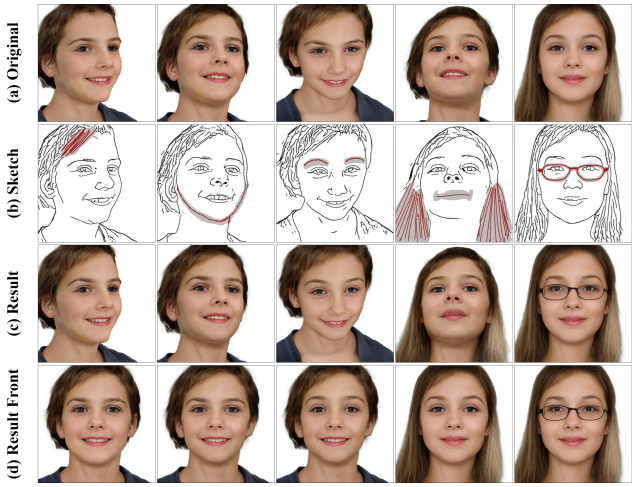

Figure 4. **Results of continuous editing over five steps.** Progressively changing the sketches in different viewpoint achieves detailed and photorealistic editing results.

Table 1. **Quantitative comparison for sketch-based 3D head generation.** Lower values indicate better performance.

Method	FID ↓	KID (×100) ↓
S3D [47]	96.03	4.50 ± 1.0
Nano-LAM [12, 20]	133.72	7.61 ± 0.9
SketchFaceNeRF [15]	94.94	4.53 ± 0.6
Ours	**92.65**	**4.00 ± 0.4**

4.2. Comparison

Sketch-to-3D Head Generation. We compare our method with leading sketch-to-3D approaches, including SketchFaceNeRF[15] and S3D[47]. We also include a two-stage baseline (Nano-LAM) that first generates a 2D image using Nano-Banana [12], and then reconstructs a 3DGS head with LAM [20]. As shown in Fig.5, S3D struggles with hand-drawn sketches and produces geometric inconsistencies. Nano-LAM produces plausible 2D portraits but tends to stylized, cartoon-like outputs that lack photorealism. SketchFaceNeRF generates reasonable 3D shapes, but its coarse tri-plane prediction often fails to recover fine-grained details from the input sketches (e.g., complex hair strands in the 2nd row), and its outputs fall short of photorealistic fidelity. By directly synthesizing Gaussian attributes through a StyleGAN-based UV manifold representation, our method recovers sharper geometry and clearer details while remaining faithful to the input sketch. The qualitative evaluation is reported in Table 1. Our method achieves the best FID and KID scores on the held-out sketch set, demonstrating improved realism and sketch faithfulness over the baselines.

Sketch-based 3D Head Editing. For editing, we compare against SketchFaceNeRF [15], MagicQuill [34], and the similar Nano-LAM 2D-to-3D editing baseline. MagicQuill, a 2D ControlNet-based method, is evaluated from the original rendering viewpoint for fairness. As shown in Figure 6, MagicQuill can perform edits but often introduces stylization and identity shifts. Nano-LAM suffers from imprecise 2D edits and poor multi-view consistency after lifting to 3D. SketchFaceNeRF produces photorealistic results but relies on slow optimization that lacks precision for fine stroke intent. Our method achieves the most accurate sketch-aligned edits while preserving unedited regions. It is also substantially faster: as reported in Table 2, we achieve the best FID/KID, an end-to-end edit latency of 0.7 s, and up to 243 FPS rendering. This enables our approach to achieve practical, interactive editing workflows. Notably, SketchFaceNeRF requires **10 s** per edit.

Table 2. **Quantitative comparison for sketch-based 3D head editing.** We report image quality (FID, KID) and interaction/rendering performance.

Method	FID ↓	KID (×100) ↓	Time (s) ↓	FPS ↑
MagicQuill	46.48	0.78 ± 0.2	∼6.0	—
Nano-LAM	74.26	3.01 ± 0.3	∼15.0	**281**
SketchFaceNeRF	62.49	2.65 ± 0.3	∼10.0	42
Ours	**44.60**	**0.69 ± 0.2**	**∼0.7**	243

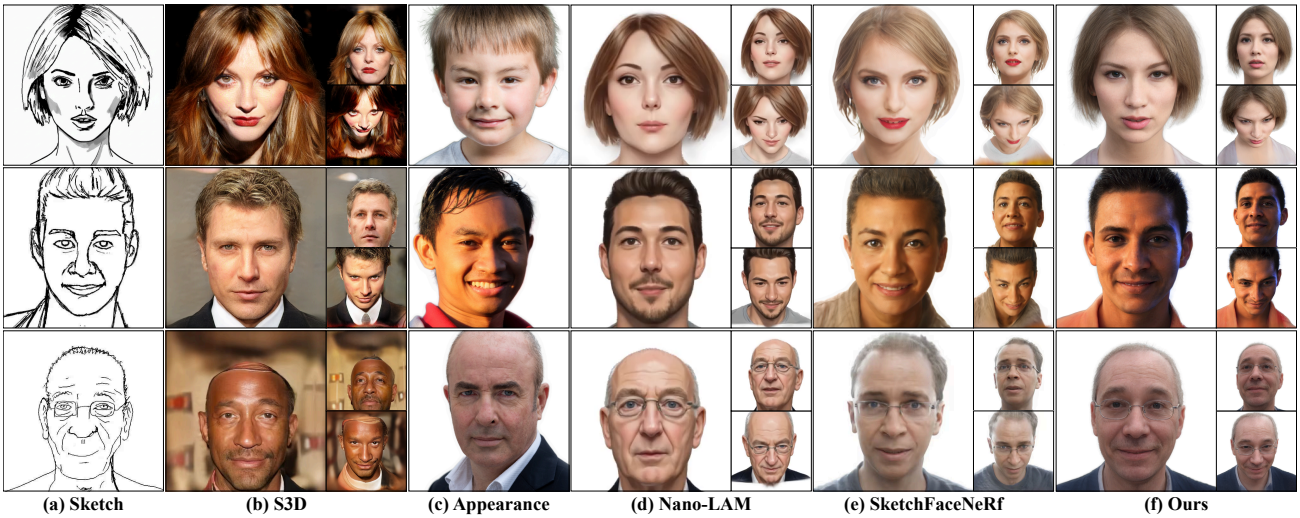

Figure 5. **Qualitative comparison for sketch-to-3D generation.** In each example, (a) is the input sketch and (c) is the reference appearance. S3D (b) shows geometric inconsistencies. Nano-LAM (d) produces cartoonish results. SketchFaceNeRF (e) lacks fine-grained detail and realism. Our method (f) generates superior results with high fidelity to both geometry and appearance.

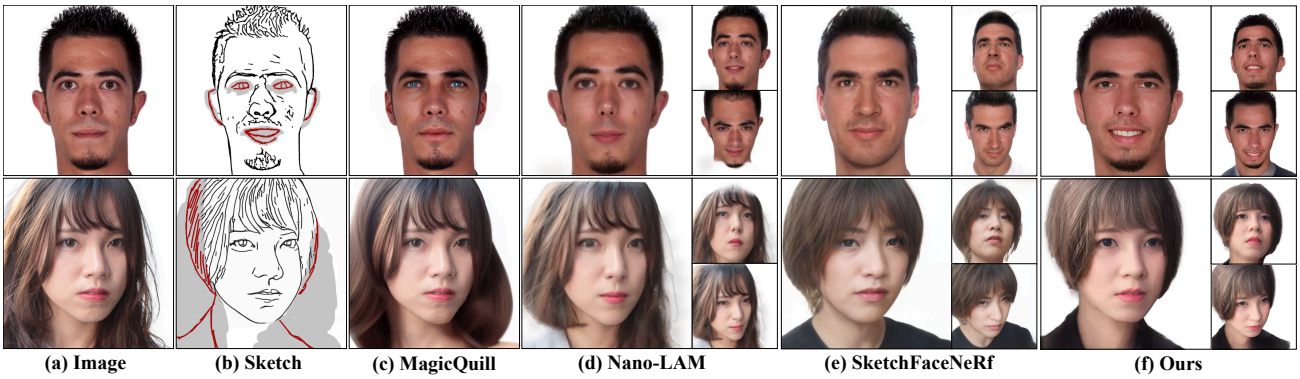

Figure 6. **Qualitative comparison for sketch-based 3D head editing.** Given an original head (a) and an edit sketch (b), MagicQuill (c) produces stylized/blurry results; Nano-LAM (d) suffers from imprecise editing with poor novel-view quality. SketchFaceNeRF (e) is optimization-based and non-interactive. Our method (f) achieves real-time, high-quality editing that faithfully follows the sketch.

4.3. Ablation study

Generation Pipeline Ablations. We examine the following settings: (a) w/o enhancement module — removing the UV Feature Enhancement stage; (b) w/o translation network — disabling the AdaIN-based alignment between geometry and appearance; (c) conv appearance encoder — replacing the appearance Transformer with a simple CNN encoder; and (d) w/o global ID feature — removing the global identity vectors. As shown in Figure 7, removing enhancement module (a) keeps coarse geometry but yields oversmoothed results without realistic details. Disabling the translation network (b) causes severe artifacts when sketch and reference identities conflict. Using a CNN encoder (c) weakens style transfer and causes color mismatches. Removing global identity vectors (d) degrades identity-specific attributes. Table 3 shows that the full model consistently outperforms all variants.

Table 3. **Quantitative ablation study for the generation pipeline.** All ablated versions show a significant drop in performance compared to our full model. Lower values are better.

Generation Ablations	FID ↓	KID (×100) ↓
Full Model (Ours)	**92.65**	**4.00** ± 0.4
w/o enhancement module	104.08	6.14 ± 1.3
w/o translation network	108.26	8.10 ± 0.7
conv appearance encoder	97.87	5.77 ± 0.5
w/o global ID feature	98.73	5.74 ± 0.6

Editing Module Ablations. We compare our layer-wise feature fusion with two alternatives: (a) direct regeneration — regenerating the entire head from the edit sketch without fusion, and (b) 3D Gaussian compositing — directly replacing affected Gaussians in 3D space. As shown in figure 8, direct regeneration (a) discards unedited context and fails to preserve identity. 3D Gaussian compositing (b) introduces conspicuous seams and mismatches at edit boundaries. In

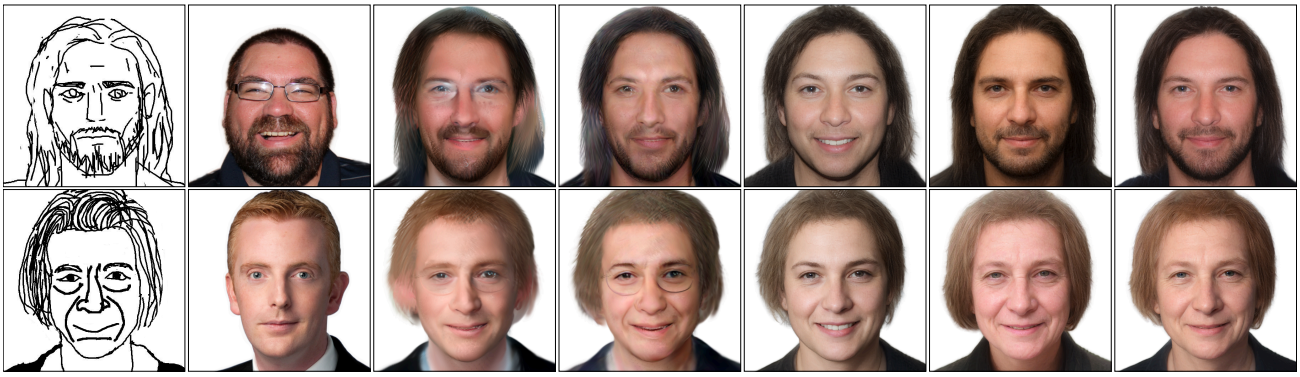

Figure 7. **Ablation study on the generation pipeline.** (a) Input sketch. (b) Input appearance. (c) Without the enhancement module, details are lost. (d) Without AdaIN alignment, identity conflicts arise. (e) Without identity vectors, identity is inconsistent. (f) With a simple CNN for appearance, style transfer fails. (g) Our full model yields the best result.

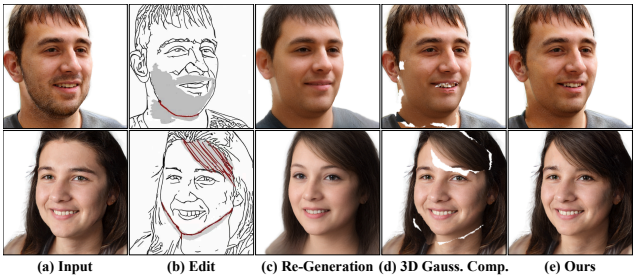

Figure 8. **Ablation study on the editing module.** Our method (e) achieves the superior results compared with naive generation (b) and direct 3DGS composition (d).

contrast, our layer-wise feature fusion produces visually coherent, artifact-free transitions by blending at multiple feature resolutions. The superiority of our approach is quantitatively validated in Table 4, where our method achieves substantially better FID and KID scores.

Table 4. **Quantitative ablation study for the editing module.** Our layer-wise feature fusion significantly outperforms alternative strategies. Lower values are better.

Editing Ablations	FID ↓	KID (×100) ↓
Full Model (Ours)	**44.60**	**0.69** ± 0.2
direct regeneration	88	3.35 ± 0.7
3D Gaussian compositing	68.4	1.89 ± 0.2

5. Applications

Local Appearance Transfer. Our pipeline decouples geometry (sketch) from appearance (reference image) and integrates local fusion to naturally support region-specific edits. As shown in Figure 9 (a), users can modify local appearance (e.g. hair color, skin tone) while preserving the geometry. Our method also supports complex edits such as adding tattoos in Figure 9 (b) thanks to the effective appearance control and precise UV feature fusion.

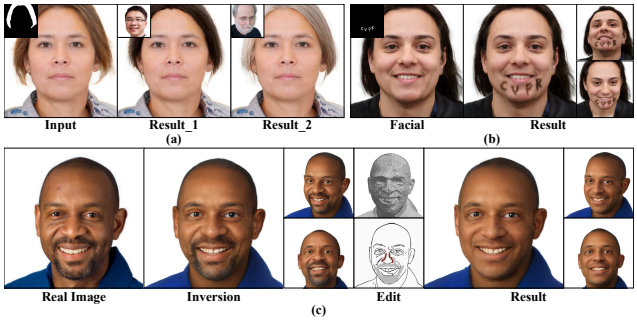

Figure 9. **Application.** Our method enables the local region appearance editing (a), interesting tattoo patterns (b), and fast 3D inversion and editing (c) of real images.

Fast Inversion and Editable Reconstruction. Given a real facial image, we extract its sketch and use our SketchFaceGS generation pipeline (with input image as appearance) to predict an initial 3DGS head capturing coarse identity features. Optional fine-tuning of the latent code $\mathcal{W}$ then achieves faithful inversion. This short optimization converges in approximately 15s on a single NVIDIA RTX 3090, yielding a high-fidelity, renderable 3D head that closely matches the input photo. As shown in Figure 9 (c), users can further apply effective local editing for personalized avatar creation.

6. CONCLUSIONS AND DISCUSSIONS

This paper presents SketchFaceGS, a real-time 3DGS-based framework for human head generation and editing to enable interactive avatar creation. For generation, we adopt a coarse-to-fine generation paradigm. In the coarse stage, a dual-transformer feedforward network extracts features from sketch and appearance images respectively, following by fusion network to predicts a coarse UV map. In the fine stage, we translate this UV map into a global latent and modulation parameters to generate high-quality 3D faces based on a pretrained 3D GAN. For editing, we intro-

duce a UV Mask Fusion strategy to ensure effective local edits while preserving unedited regions.

Limitations and future work. Despite advances in 3D face synthesis, our method still has several limitations. Trained primarily on realistic faces, it struggles to retain highly stylized features. Adding more stylized training examples or a style-aware extractor could help address this issue. Additionally, since our method is built on GGHead, it inherits limitations in handling rare accessories and extreme occlusions. Adding targeted data augmentation or accessory modules could improve results. Finally, for fast inversion, out-of-distribution inputs may require longer or multi-view optimization. Future work could explore co-training pose estimation, multi-view refinement, and interactive user-in-the-loop correction.

References

[1] Autodesk, Inc. Maya. https://www.autodesk.com/maya, 2019. 2

[2] AlexanderW Bergman, Petr Kellnhofer, TU Delft, Wang Yifan, EricR Chan, DavidB Lindell, and Gordon Wetzstein. Generative neural articulated radiance fields. 2

[3] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In Proc. CVPR, 2021. 2

[4] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini de Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 2

[5] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini de Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 2

[6] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. Deepfacedrawing: deep generation of face images from sketches. ACM Trans. Graph., 39(4), 2020. 2

[7] Shu-Yu Chen, Feng-Lin Liu, Yu-Kun Lai, Paul L. Rosin, Chunpeng Li, Hongbo Fu, and Lin Gao. Deepfaceediting: deep face generation and editing with disentangled geometry and appearance control. ACM Trans. Graph., 40(4), 2021. 2

[8] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[9] Kun Cheng, Mingrui Zhu, Nannan Wang, Guozhang Li, Xiaoyu Wang, and Xinbo Gao. Controllable face sketch-photo synthesis with flexible generative priors. In Proceedings of the 31st ACM International Conference on Multimedia, page 6959–6968, New York, NY, USA, 2023. Association for Computing Machinery. 2

[10] Xuangeng Chu and Tatsuya Harada. Generalizable and animatable gaussian head avatar. In The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024. 3

[11] Kangle Deng, Gengshan Yang, Deva Ramanan, and Jun-Yan Zhu. 3d-aware conditional image synthesis. In CVPR, 2023. 3

[12] Google Developers. Introducing gemini 2.5 flash image (aka nano-banana). https://developers.googleblog.com/en/introducing-gemini-2-5-flash-image/, 2025. Accessed: 2025-11-14. 6

[13] Dong Du, Xiaoguang Han, Hongbo Fu, Feiyang Wu, Yizhou Yu, Shuguang Cui, and Ligang Liu. Sanihead: Sketching animal-like 3d character heads using a view-surface collaborative mesh generative network. IEEE Transactions on Visualization and Computer Graphics, 28(6):2415–2429, 2020. 2

[14] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. Cornell University - arXiv,Cornell University - arXiv, 2016. 2

[15] Lin Gao, Feng-Lin Liu, Shu-Yu Chen, Kaiwen Jiang, Chunpeng Li, Yukun Lai, and Hongbo Fu. Sketchfacenerf: Sketch-based facial generation and editing in neural radiance fields. ACM Transactions on Graphics, 42(4), 2023. 2, 3, 6

[16] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014. 2

[17] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In International Conference on Learning Representations, 2022. 2

[18] Xiaoguang Han, Chang Gao, and Yizhou Yu. Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling. ACM Trans. Graph., 36(4), 2017. 2

[19] Xiaoguang Han, Kangcheng Hou, Dong Du, Yuda Qiu, Yizhou Yu, Kun Zhou, and Shuguang Cui. Caricatureshop: Personalized and photorealistic caricature sketching. IEEE Transactions on Visualization and Computer Graphics,IEEE Transactions on Visualization and Computer Graphics, 2018. 2

[20] Yisheng He, Xiaodong Gu, Xiaodan Ye, Chao Xu, Zhengyi Zhao, Yuan Dong, Weihao Yuan, Zilong Dong, and Liefeng Bo. Lam: Large avatar model for one-shot animatable gaussian head. In Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers, pages 1–13, 2025. 3, 6

[21] Philipp Henzler, Niloy Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 2

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020. 2

[23] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. arXiv preprint arXiv:2311.04400, 2023. 3

[24] Sangeek Hyun and Jae-Pil Heo. Gsgan: Adversarial learning for hierarchical generation of 3d gaussian splats. Advances in Neural Information Processing Systems, 37: 67987–68012, 2024. 2

[25] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In Proceedings of the IEEE/CVF international conference on computer vision, pages 1745–1753, 2019. 2

[26] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning Category-Specific Mesh Reconstruction from Image Collections, page 386–402. 2018. 2

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4401–4410, 2019. 5

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8110–8119, 2020. 3

[30] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph., 42(4):139–1, 2023. 2, 3

[31] Tobias Kirschstein, Simon Giebenhain, Jiapeng Tang, Markos Georgopoulos, and Matthias Nießner. Gghead: Fast and generalizable 3d gaussian heads. In SIGGRAPH Asia 2024 Conference Papers, New York, NY, USA, 2024. Association for Computing Machinery. 2, 3, 5

[32] Yuhang Li, Xuejin Chen, Feng Wu, and Zheng-Jun Zha. Linestofacephoto: Face photo generation from lines with conditional self-attention generative adversarial networks. In Proceedings of the 27th ACM International Conference on Multimedia, page 2323–2331, New York, NY, USA, 2019. Association for Computing Machinery. 2

[33] Yuhang Li, Xuejin Chen, Binxin Yang, Zihan Chen, Zhihua Cheng, and Zheng-Jun Zha. Deepfacepencil: Creating face images from freehand sketches. In Proceedings of the 28th ACM International Conference on Multimedia, page 991–999, New York, NY, USA, 2020. Association for Computing Machinery. 2

[34] Zichen Liu, Yue Yu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Wen Wang, Zhiheng Liu, Qifeng Chen, and Yujun Shen. Magicquill: An intelligent interactive image editing system. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 13072–13082, 2025. 6

[35] Zhongjin Luo, Jie Zhou, Heming Zhu, Dong Du, Xiaoguang Han, and Hongbo Fu. Simpmodeling: Sketching implicit field to guide mesh modeling for 3d animalmorphic head design. In The 34th annual ACM symposium on user interface software and technology, pages 854–863, 2021. 2

[36] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, page 405–421. 2020. 3

[37] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019. 2

[38] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2

[39] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2

[40] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. Ridge-valley lines on meshes via implicit surface fitting. ACM Transactions on Graphics, page 609–612, 2004. 2

[41] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193, 2023. 4

[42] Roy OrEl, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 2

[43] Panwang Pan, Zhuo Su, Chenguo Lin, Zhen Fan, Yongjie Zhang, Zeming Li, Tingting Shen, Yadong Mu, and Yebin Liu. Humansplat: Generalizable single-image human gaussian splatting with structure priors. In Advances in Neural Information Processing Systems (NeurIPS), 2024. 3

[44] Tiziano Portenier, Qinglong Hu, Attila Szabo, SiavashArjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. ACM Transactions on Graphics,ACM Transactions on Graphics, 2018. 2

[45] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. arXiv: Computer Vision and Pattern Recognition,arXiv: Computer Vision and Pattern Recognition, 2020. 2

[46] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. arXiv: Computer Vision and Pattern Recognition,arXiv: Computer Vision and Pattern Recognition, 2020. 2

[47] Hail Song, Wonsik Shin, Naeun Lee, Soomin Chung, Nojun Kwak, and Woontack Woo. S3d: Sketch-driven 3d model generation. arXiv preprint arXiv:2505.04185, 2025. 3, 6

[48] Cyrus Vachha and Ayaan Haque. Instruct-gs2gs: Editing 3d gaussian splats with instructions, 2024. 2

[49] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 20902–20911, 2024. 2, 5

[50] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 4

[51] Zhenzhen Weng, Jingyuan Liu, Hao Tan, Zhan Xu, Yang Zhou, Serena Yeung-Levy, and Jimei Yang. Template-free single-view 3d human digitalization with diffusion-guided lrm. arXiv preprint arXiv:2401.12175, 2024. 3

[52] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In European Conference on Computer Vision, pages 55–71. Springer, 2024. 2

[53] Yang Xue, Yuheng Li, Krishna Kumar Singh, and Yong Jae Lee. Giraffe hd: A high-resolution 3d-aware generative model. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 18440–18449, 2022. 2

[54] Li Yang, Jing Wu, Jing Huo, Yu-Kun Lai, and Yang Gao. Learning 3d face reconstruction from a single sketch. Graphical Models, 115:101102, 2021. 2

[55] Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. Deep plastic surgery: Robust and controllable image editing with human-drawn sketches. In European Conference on Computer Vision, pages 601–617. Springer, 2020. 2

[56] Yu Zeng, Zhe Lin, and Vishal M Patel. Sketchedit: Mask-free local image manipulation with partial sketches. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5951–5961, 2022. 2