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

Supplementary Material

1. Overview

In this supplementary material, we provide additional details to complement our main paper. We begin with a thorough description of our implementation, including model architecture, training protocols, and experimental setup (Sec. 2). We then present a comprehensive user study that offers a perceptual evaluation of our method against several baselines (Sec. 3). Subsequently, we conduct a robustness analysis to demonstrate our model's performance on diverse sketch and appearance inputs (Sec. 4). Finally, we discuss some failure cases and limitations (Sec. 6) and showcase more high-quality editing results (Sec. 7).

2. Implementation and Experimental Details

2.1. Training Details and Loss Functions

Our model is trained in a stage-wise manner, with distinct training strategies and loss functions tailored for each component: the two stages of the generation pipeline (coarse and fine) and the editing pipeline. Below, we provide a detailed description of the objective function for each stage.

Coarse Stage Training. This stage learns a mapping from a sketch and a reference appearance to a coarse but geometrically consistent UV feature map. We train the model on a synthesized multi-view dataset generated from GGHead [8]. Each sample includes two images of the same identity from different viewpoints. One view provides the appearance reference and its corresponding sketch, while the other serves as the ground-truth target, I_{target}.

To enable end-to-end training, we introduce a temporary MLP decoder D_g that translates the predicted coarse UV map into a renderable set of 3D Gaussian primitives. The optimization objective is cross-view consistency: the rendered result I_{render}, produced from the target viewpoint, should match I_{target}. We employ a combination of an $\mathcal{L}_1$ loss, a VGG-based [6] perceptual loss $\mathcal{L}_{\text{perceptual}}$, and a color consistency loss $\mathcal{L}_\delta$. The total loss $\mathcal{L}_{\text{coarse}}$ is:

$$\begin{aligned}\mathcal{L}_{\text{coarse}} = \lambda_{c1}\mathcal{L}_1(I_{\text{target}}, I_{\text{render}}) + \lambda_{c2}\mathcal{L}_{\text{perceptual}}(I_{\text{target}}, I_{\text{render}}) \\ + \lambda_{c3}\mathcal{L}_\delta(I_{\text{target}}, I_{\text{render}}) \quad (1)\end{aligned}$$

where the weights are set to $\lambda_{c1} = 0.1$, $\lambda_{c2} = 0.0066$, and $\lambda_{c3} = 0.007$.

Fine Stage Training. In this stage, we switch to a single-view dataset (FFHQ [7]) to enhance generalization and photorealism. The training has two complementary supervision paths:

1. We reuse the temporary decoder D_g to produce a preliminary rendering, denoted as $I_{\text{U-Net}}$. This output is supervised against the reference image I_{ref} **using an LPIPS [11] loss** to ensure the U-Net predicts meaningful modulation parameters.
2. The final, high-quality output from the full GGHead pipeline, I_{final}, is supervised to enhance fine details and photorealism.

The overall loss $\mathcal{L}_{\text{fine}}$ combines supervision on the final output using LPIPS, $\mathcal{L}_1$, $\mathcal{L}_{\text{perceptual}}$, and an adversarial loss $\mathcal{L}_{\text{GAN}}$:

$$\begin{aligned}\mathcal{L}_{\text{fine}} = \lambda_{f1}\mathcal{L}_{\text{lpips}}(I_{\text{ref}}, I_{\text{U-Net}}) + \lambda_{f2}\mathcal{L}_1(I_{\text{ref}}, I_{\text{final}}) \\ + \lambda_{f3}\mathcal{L}_{\text{perceptual}}(I_{\text{ref}}, I_{\text{final}}) \\ + \lambda_{f4}\mathcal{L}_{\text{GAN}}(I_{\text{final}}) \quad (2)\end{aligned}$$

where the weights are $\lambda_{f1} = 0.1$, $\lambda_{f2} = 1.0$, $\lambda_{f3} = 0.0066$, and $\lambda_{f4} = 0.4$.

Editing Stage Training. This stage teaches the model to seamlessly merge content using our UV Mask Fusion strategy. We train on synthetic heads from GGHead [8] with random masks simulating user edits. Our model then generates a fused result, which is rendered from the original viewpoint as I_{fused}. The optimization objective is twofold: first, to ensure that I_{fused} remains consistent with the original rendered image I_{orig} in the unedited regions; second, to ensure that the complete I_{fused} image conforms to the data distribution of real faces. The total loss $\mathcal{L}_{\text{edit}}$ achieves this by supervising the fused output against the original, unedited rendering:

$$\begin{aligned}\mathcal{L}_{\text{edit}} = \lambda_{e1}\mathcal{L}_\delta(I_{\text{orig}}, I_{\text{fused}}) + \lambda_{e2}\mathcal{L}_1(I_{\text{orig}}, I_{\text{fused}}) \\ + \lambda_{e3}\mathcal{L}_{\text{perceptual}}(I_{\text{orig}}, I_{\text{fused}}) \\ + \lambda_{e4}\mathcal{L}_{\text{GAN}}(I_{\text{fused}}) \quad (3)\end{aligned}$$

where the weights are set to $\lambda_{e1} = 1.0$, $\lambda_{e2} = 0.0066$, $\lambda_{e3} = 0.1$, and $\lambda_{e4} = 0.4$.

2.2. Evaluation Details

2.2.1. Datasets

Our framework is trained and evaluated on a combination of synthetic and real-world data, tailored to the specific requirements of each stage and task as described in our training strategy (Sec. 2.1).

Training Data. Our training process utilizes a total of 90,000 samples. For the multi-view **Coarse Stage**, we use 40,000 samples generated by rendering latent codes from

the pre-trained GGHead model from multiple viewpoints. For the single-view **Fine Stage**, we leverage 50,000 real-world portraits from the FFHQ dataset to enhance photorealism and generalization.

Evaluation Datasets. For evaluation, we curated two independent test sets. The **Generation Set** consists of 100 hand-drawn sketches sourced from artists, each paired with a reference appearance randomly sampled from the held-out FFHQ test set. This collection encompasses a wide spectrum of artistic styles, ranging from abstract contours to detailed line work, and covers diverse identities across genders and ethnicities, along with various facial expressions. The second set, the **Editing Set**, is similarly built upon 100 real-world portraits from FFHQ (disjoint from the training set). For each portrait, we collected authentic user edits on its automatically extracted base sketch. These edits span a comprehensive range of modifications, including local feature adjustments (e.g., eyes, mouth), hairstyle changes, accessory additions, and global shape deformations. Consequently, each sample provides the original image, the user's edited sketch, and the corresponding edit mask. To ensure a fair comparison, all edits are initiated from the original image's viewpoint, with multi-view renderings demonstrated in the paper.

2.2.2. Baseline Comparison Protocol

To provide a comprehensive and fair comparison, we established a detailed protocol for evaluating each baseline method on our two primary tasks: sketch-based generation and sketch-based editing. For all comparisons, we rendered the output from the original input viewpoint and two additional novel viewpoints to assess 3D consistency.

Generation Comparison. For the generation task, we compared our method against S3D [10], Nano-LAM [3, 5], and SketchFaceNeRF [4].

- **S3D [10]:** We observed that S3D exhibits poor generalization to free-form, artistic sketches, with direct application resulting in non-plausible and geometrically distorted outputs. To accommodate this limitation and create a fair testbed, we devised a two-stage process: first, we used DeepFaceDrawing [2] to generate a photorealistic face from the input hand-drawn sketch. Second, we applied S3D's own sketch extraction method to this generated face and used the resulting "clean" sketch as the final input to S3D. It is important to note that S3D's framework does not support an explicit appearance reference image for controlling color and style.
- **Nano-LAM:** For this two-stage pipeline, we first used the Nano-Banana model [3] to generate a 2D portrait, and then employed LAM [5] to lift it to a 3D head. We provided Nano-Banana with both the sketch and the reference appearance image, using the following prompt: `"Please generate a realistic and natural human face, with a geometry that is highly consistent with the first sketch image and a color style that matches the second RGB face image."`
- **SketchFaceNeRF [4]:** This method requires both a hand-drawn sketch and an RGB reference image as input. However, a significant inconvenience is its requirement for the camera pose of the input sketch. To provide this, we first generated a proxy face using DeepFaceDrawing [2] and then estimated its camera pose, which was then fed into SketchFaceNeRF. In contrast, our method requires only a sketch and an appearance image, with no need for camera pose pre-estimation.

Editing Comparison. For the editing task, which is performed on real face images, we compared our method against Nano-LAM [3, 5], MagicQuill [9], and SketchFaceNeRF [4].

- **Nano-LAM:** We followed a similar two-stage process. We provided the **Nano-Banana model [3]** with the real RGB face image, the original sketch, and the edited sketch. The prompt used was: `"Image 1 is the original real face, Image 2 is the sketch consistent with the original real face, and Image 3 is the sketch edited based on Image 2. Please apply the edits from Image 3 relative to Image 2 onto the real face in Image 1, and output the resulting realistic and natural human face."` The resulting 2D edited image was then lifted to a 3D head using **LAM [5]**.
- **MagicQuill [9]:** As a ControlNet-based method, we provided MagicQuill with the original face, the edited sketch, and the edit region mask as its core control inputs. We used a simple positive prompt: `"a realistic and natural human face"`.
- **SketchFaceNeRF and Ours:** To ensure a fair comparison between the NeRF-based SketchFaceNeRF and our 3DGS-based method, we first performed an inversion step for each real test image to obtain its 3D representation (for EG3D [1] and our head model, respectively). The sketch-based editing was then performed on these high-fidelity inverted 3D representations.

2.2.3. Extended Comparisons with Nano-LAM

In the main paper, we compared our method with Nano-LAM, a two-stage baseline utilizing Nano-Banana (Gemini 2.5 Flash Image) [3] for 2D generation and LAM [5] for 3D lifting. While prompt engineering can successfully guide Nano-Banana to generate realistic 2D portraits, this pipeline suffers from two fundamental limitations. First, the generation process of Nano-Banana takes over 10 seconds, which severely hinders real-time interactivity. Second, relying on LAM for single-view 3D lifting inevitably introduces geometric distortions, 3D inconsistencies, and artifacts under

large viewpoint rotations (see Figure 1). In contrast, our feed-forward architecture achieves real-time performance (∼0.3s) while strictly preserving high-fidelity geometry and multi-view consistency.

Figure 1. **Comparisons with Nano-LAM.** While Nano-LAM achieves realistic 2D results via prompt engineering, its 3D lifting process leads to severe 3D inconsistencies and artifacts under large viewpoint rotations. In contrast, our method maintains strictly consistent and high-fidelity geometry across all views.

3. User Study

Given the extensive qualitative and quantitative comparisons, we conducted a perception study to fully testify our method from the perspective of human viewers. Specifically, we evaluate our method on two tasks: facial generation from hand-drawn 2D sketches, and facial editing by modifying the corresponding 2D sketches.

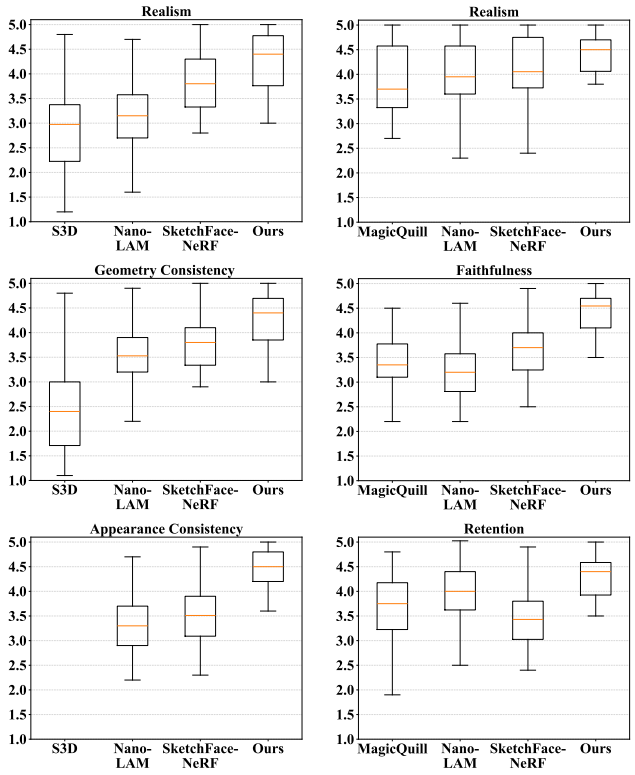
Experimental Design. For the **facial generation task**, we compare our method against three state-of-the-art baselines: S3D, SketchFaceNeRF, and Nano-LAM. We prepared 10 cases to cover as much diversity (such as drawing style and personal attributes) as possible. Each case consists of an input 2D hand-drawn sketch, a reference appearance facial image, and the facial images generated by the compared methods. To demonstrate 3D consistency, we displayed the results rendered from three different viewpoints. Users were invited to rate the generated facial images on a 1-5 Likert scale (higher is better) based on three criteria: *Realism*, *Geometry Consistency* with the input sketches, and *Appearance Consistency* with the reference appearance images.

For the **facial editing task**, we compare our method against MagicQuill, SketchFaceNeRF, and Nano-LAM. Similarly, we prepared 10 cases covering varying editing regions and styles. Each case consists of an original facial image, a modified sketch where edited regions are emphasized, and the edited facial images. For fairness, we only display the results from the viewpoints of the original facial images since MagicQuill is a 2D-only method. Users were asked to rate the results based on: *Realism*, *Retention* of the unchanged regions, and *Faithfulness* to the edited sketches.

In total, 38 people (28 males and 10 females, aged 18-40) participated in this study. We collected answers for all 20 cases from each participant, resulting in a comprehensive dataset for statistical analysis.

Statistical Analysis. Fig. 2 plots the statistics of the evaluation results. We found significant effects for all criteria through one-way ANOVA tests followed by paired t-tests.

Generation Task Results. ANOVA tests revealed highly



(a) Generation Task Results (b) Editing Task Results

Figure 2. **User Study Statistics.** Box plots of averaged user preference scores (scale 1-5, the higher, the better). (a) Comparison of facial generation with four methods: S3D [10], Nano-LAM [3, 5], SketchFaceNeRF [4], and Ours, in terms of realism, geometry consistency, and appearance consistency. (b) Comparison of facial editing with four methods: MagicQuill [9], Nano-LAM [3, 5], SketchFaceNeRF [4], and Ours, in terms of realism, retention, and faithfulness. The boxes extend from the first quartile to the third quartile, with a middle horizontal line denoting the median. The whiskers represent the minimum and maximum values excluding outliers.

significant differences among methods: Realism ($F = 17.44, p < 0.001$), Geometry Consistency ($F = 24.17, p < 0.001$), and Appearance Consistency ($F = 22.10, p < 0.001$). We conduct paired t-tests to confirm the superiority of our method. In terms of **Realism**, our method (mean: 4.31) is significantly preferred over S3D (mean: 2.98; $t = 6.35, p < 0.001$), SketchFaceNeRF (mean: 3.69; $t = 3.45, p < 0.001$), and Nano-LAM (mean: 3.04; $t = 6.87, p < 0.001$). For **Geometry Consistency**, our method achieves the highest score (mean: 4.48), significantly outperforming S3D (mean: 2.73; $t = 8.43, p < 0.001$), SketchFaceNeRF (mean: 3.48; $t = 6.09, p < 0.001$), and Nano-LAM (mean: 3.43; $t = 6.25, p < 0.001$). Regarding **Appearance Consistency**, our method (mean: 4.35) demonstrates a substantial advantage over SketchFaceNeRF (mean: 3.72; $t = 3.63, p < 0.001$) and Nano-LAM (mean: 3.57; $t = 4.54, p < 0.001$). Note that since S3D does not

support explicit appearance control, we omit the detailed statistical comparison for this metric, though its low score (mean: 2.61) reflects this limitation.

Editing Task Results. The editing task also showed significant effects across all criteria: Realism ($F = 5.43, p = 0.001$), Retention ($F = 18.10, p < 0.001$), and Faithfulness ($F = 8.80, p < 0.001$). Our method demonstrates clear superiority in all aspects. For **Realism**, our method (mean: 4.47) scores significantly higher than SketchFaceNeRF (mean: 3.92; $t = 3.01, p = 0.004$), MagicQuill (mean: 3.74; $t = 4.18, p < 0.001$), and Nano-LAM (mean: 3.80; $t = 3.75, p < 0.001$). Finally, we confirm the superiority of our method in terms of both **Retention** and **Faithfulness**. Specifically, for Retention (Unedited Region Consistency), our method (mean: 4.47) significantly surpasses SketchFaceNeRF (mean: 3.54; $t = 5.51, p < 0.001$), MagicQuill (mean: 3.41; $t = 6.65, p < 0.001$), and Nano-LAM (mean: 3.21; $t = 7.73, p < 0.001$). For Faithfulness (Edited Region Consistency), our method (mean: 4.38) also outperforms SketchFaceNeRF (mean: 3.46; $t = 5.59, p < 0.001$), MagicQuill (mean: 3.62; $t = 4.44, p < 0.001$), and Nano-LAM (mean: 3.82; $t = 3.28, p = 0.002$).

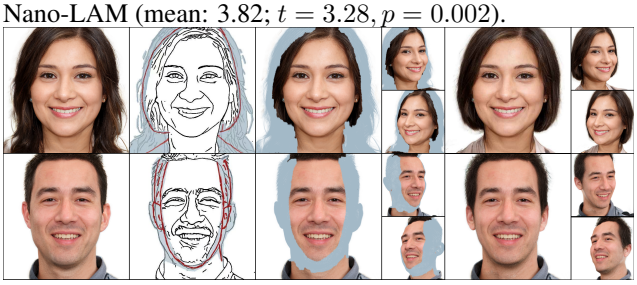


Figure 3. **Robustness to extreme contour edits and back-face leakage.** The gray regions denote the Gaussians accurately localized by our UV mask. Even when subjected to substantial and exaggerated contour edits (e.g., drastically altering the facial shape or drastically changing the hairstyle), our method precisely confines the modifications to the intended areas. The novel-view renderings further confirm that our pipeline explicitly prevents back-face leakage, leaving the occluded and unedited regions completely intact under extreme conditions.

4. Robustness Analysis

4.1. Robustness to Diverse Sketch Styles

Our framework demonstrates exceptional robustness to a wide spectrum of sketch styles beyond clean, synthetic line art. As illustrated in Fig. 5, the model effectively interprets and processes casual, messy, and highly stylized hand-drawn sketches. While striving to faithfully preserve the input's structural cues, the powerful GAN prior inherently regularizes the output to maintain photorealism, thereby successfully translating diverse artistic intents into high-quality 3D head models. Furthermore, this robustness extends to our editing pipeline. As shown in Fig. 6, variations in stroke thickness or drawing style do not negatively impact the editing outcome, ensuring consistent and precise modifications.

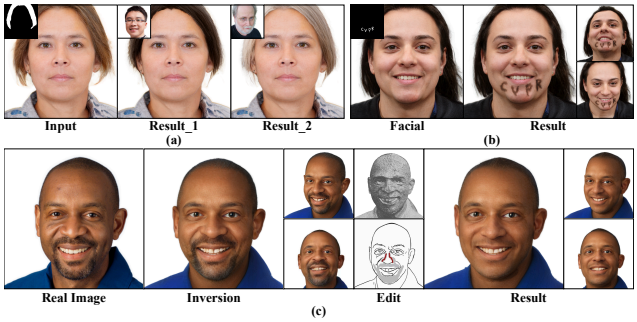


Figure 4. **Application.** Our method enables the local region appearance editing (a), interesting tattoo patterns (b), and fast 3D inversion and editing (c) of real images.

4.2. Robustness of UV Mask Fusion

Generated UV masks are essential for feature fusion. Our approach derives robust UV masks from 2D masks, effectively preventing edits from incorrectly penetrating into occluded regions such as the back of the head. Specifically, the mask generation consists of three steps. First, we extract the 2D edit region via sketch differencing and morphological dilation. Second, we cast rays through the 2D mask pixels to locate the Gaussians with the highest rendering contribution, explicitly filtering out occluded underlying primitives. Finally, leveraging the correspondence between these Gaussians and the FLAME model, we map them into the UV mask. As shown in Fig. 3, this mechanism ensures robustness against large-scale contour deformations by precisely localizing edit regions, thereby maintaining 3D-consistent editing.

4.3. Robustness to Structural Inconsistencies

A key challenge in exemplar-based generation is handling cases where the geometry defined by the sketch differs drastically from the structure of the appearance reference image. In Fig. 7, we demonstrate our model's capability to disentangle geometry from appearance. Even when the reference image presents a significantly different face shape, pose, or feature arrangement compared to the input sketch, our model strictly adheres to the geometry specified by the sketch while faithfully transferring the color palette and texture details from the reference. This confirms that our method relies on the sketch for structural guidance rather than overfitting to the geometry of the appearance image.

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 4(a), users can modify local appearance (e.g. hair color, skin tone) while preserving the

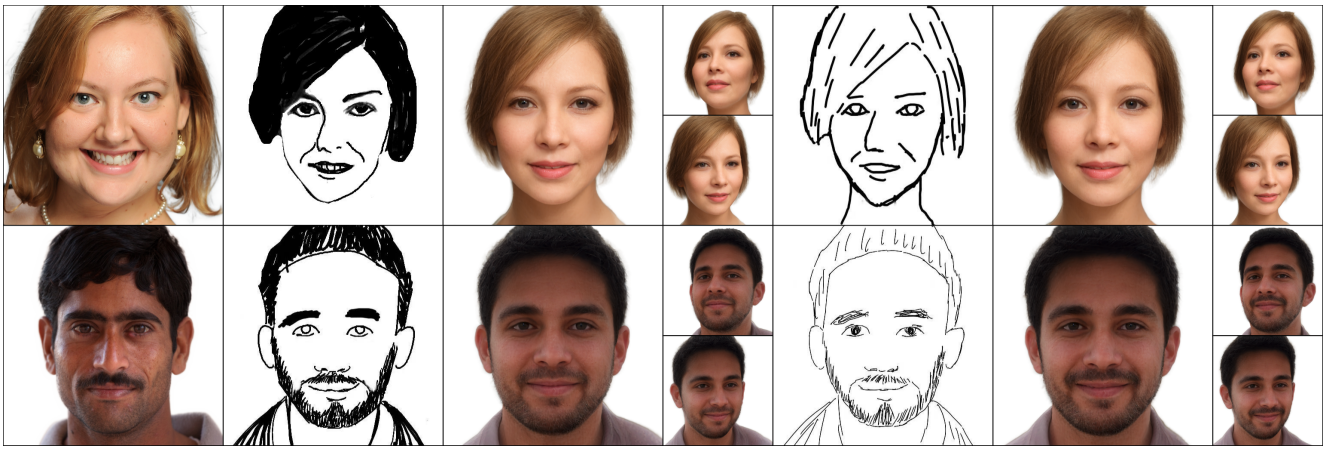


Figure 5. **Robustness to diverse sketch styles in generation.** Given a fixed appearance reference, input sketches with vastly different artistic styles but consistent underlying geometry yield perceptually similar 3D outputs that align with the intended shape.

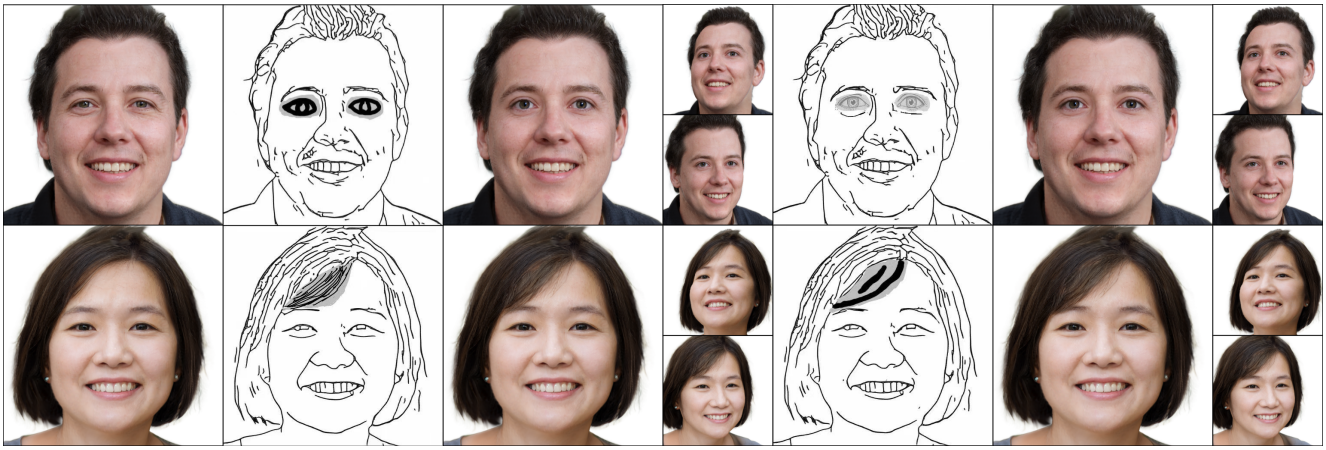


Figure 6. **Robustness to sketch styles in editing.** We demonstrate editing results using sketches with varying stroke styles and thicknesses. The editing performance remains stable and unaffected by the visual characteristics of the stroke lines.

geometry. Our method also supports complex edits such as adding tattoos in Figure 4 (b) thanks to the effective appearance control and precise UV feature fusion.

Fast Inversion and Editable Reconstruction. Given a real facial image, we extract its sketch and use our 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 one NVIDIA RTX 3090, yielding a high-fidelity, renderable 3D head that closely matches the input photo. As shown in Figure 4 (c), users can further apply effective local editing for personalized avatar creation.

6. Failure Cases and Limitations

Despite its robustness, our model has limitations. It may struggle with sketches that are extremely abstract or deviate significantly from human facial topology (e.g., anime-style characters), as the GGHead prior is trained on realistic faces. Furthermore, as shown in Fig. 8, reference images with heavy occlusions (e.g., a hand covering part of the face) or extreme, unnatural lighting can sometimes lead to minor texture artifacts. Future work could explore incorporating more diverse generative priors or style-aware feature extractors to address these limitations.

7. Additional Facial Editing Results

To further demonstrate the capabilities of our framework, we provide more qualitative results for the editing task in Fig. 9, Fig. 10, and Fig. 11. We categorize these examples into three groups: (1) precise editing of core facial components, (2) modifications to hair, facial hair, and eyebrows, and (3) editing of face shape and glasses.

References

[1] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, pages 16123–16133, 2022. 2

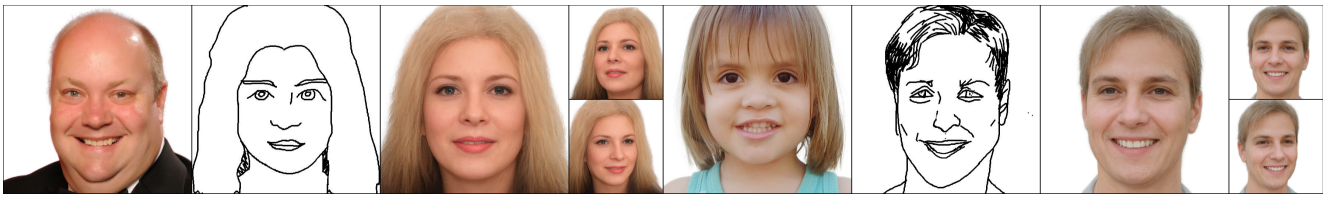


Figure 7. **Generation under significant structural conflict.** We pair input sketches with reference images that have drastically different geometric characteristics. Our model correctly synthesizes the geometry defined by the sketch (Result) while adopting the color style from the reference (Appearance), effectively resolving this conflict.

Figure 8. **Failure Cases.** (Left) An extreme cartoon sketch leads to an uncanny result. (Right) A reference image with a heavy shadow occlusion causes a minor artifact on the cheek of the generated face.

[2] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. Deepfacedrawing: deep generation of face images from sketches. *ACM TOG*, 39(4):72, 2020. 2

[3] Alisa Fortin, Guillaume Vernade, Kat Kampf, and Ammaar Reshi. Introducing gemini 2.5 flash image, our state-of-the-art image model, 2025. 2, 3

[4] 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 TOG*, 42(4), 2023. 2, 3

[5] 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 *Proc. ACM SIGGRAPH*, pages 1–13, 2025. 2, 3

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016. 1

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 1

[8] Tobias Kirschstein, Simon Giebenhain, Jiapeng Tang, Markos Georgopoulos, and Matthias Nießner. Gghead: Fast and generalizable 3d gaussian heads. In *Proc. ACM SIGGRAPH*, pages 1–11, 2024. 1

[9] 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 *CVPR*, pages 13072–13082, 2025. 2, 3

[10] 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. 2, 3

[11] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 1

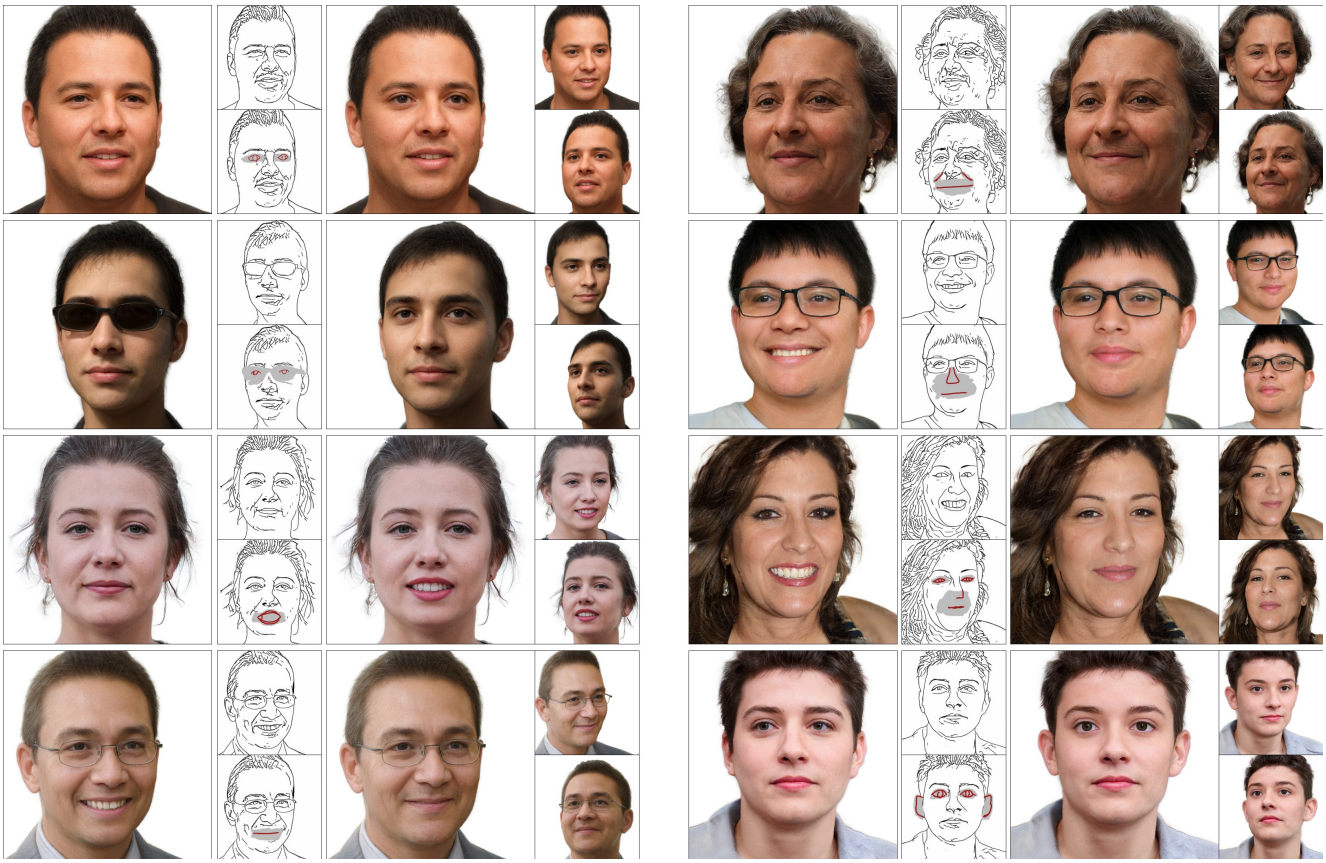


Figure 9. **Editing of core facial components.** We demonstrate the precision of our method in modifying fine-grained sensory organs. Examples include: controlling eye shape, restoring eyes after removing sunglasses, adjusting mouth opening and size, resizing the nose, and modifying ear shape.

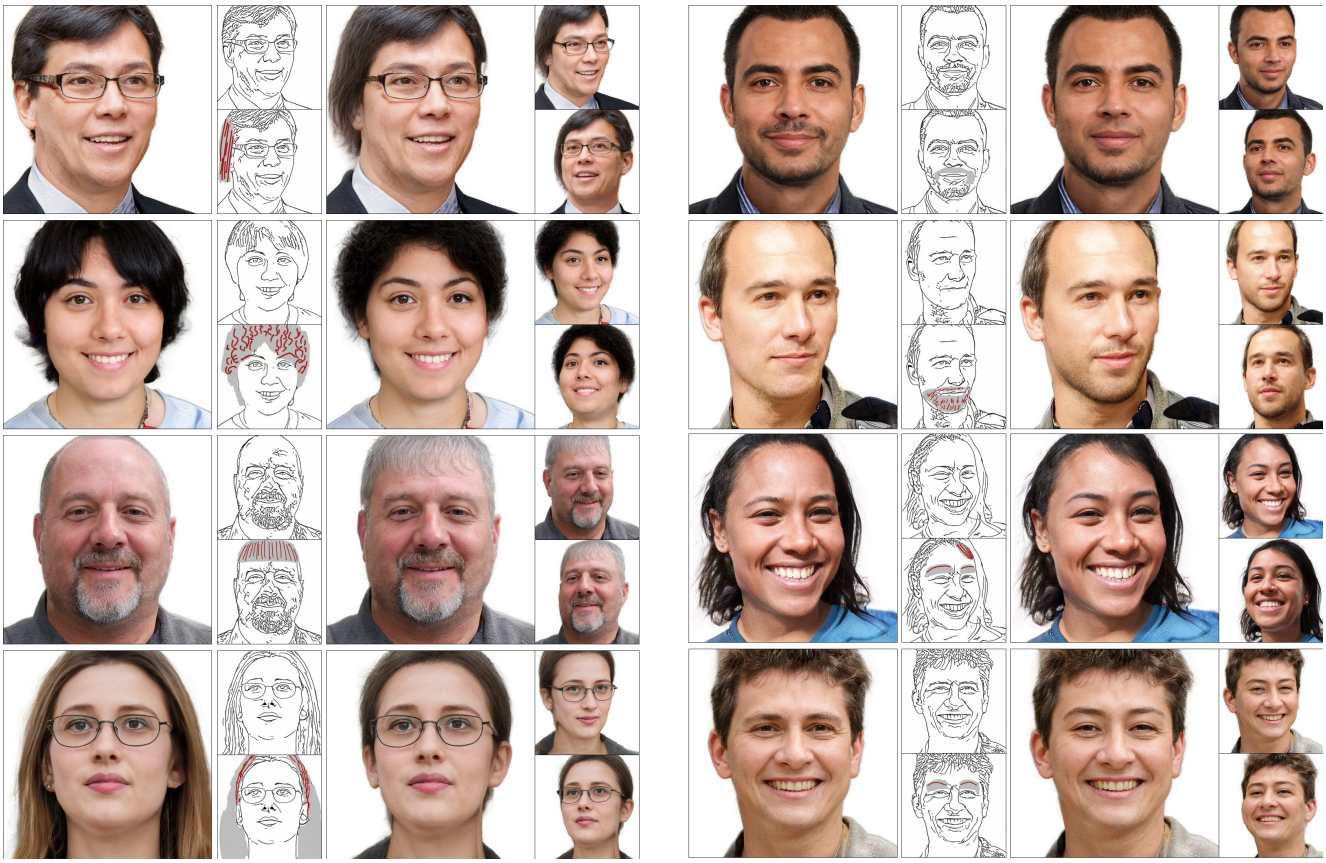


Figure 10. **Editing of hair and facial hair.** This figure showcases edits related to hair structures. Examples include precise control over various hairstyles, the addition and precise removal of beards, eyebrow adjustments, and the transformation from straight hair to voluminous curly hair.

Figure 11. **Editing of face shape and accessories.** We demonstrate the capability for global geometric deformation and object manipulation. Results show precise editing of the facial contour (face shape), as well as the addition of glasses and the modification of eyeglass frames.