FaceGS: Anchor-View Face Editing for 3D Gaussian Splatting

Caleb Kha-Uong*
Princeton University
caleb.uong@princeton.edu

Aaron Yang*
Princeton University
ay2707@princeton.edu

Abstract

Text-guided editing of 3D Gaussian Splatting (3DGS) scenes is especially difficult for faces: small cross-view inconsistencies quickly appear as waxy texture, blurred identity, or spotty splat artifacts. Many editing pipelines render several views, edit each rendered image independently, and consolidate the resulting 2D supervision into one shared 3D representation. This project instead studies canonical-UV editing for 3DGS faces through two complementary pipelines. First, we choose authoritative anchor views, edit those views in image space, back-project the masked edit into a canonical face UV domain, and write the edit back to face Gaussians without averaging conflicting RGB predictions. Second, we fine-tune a diffusion model to edit facial UV maps directly, then use the same writeback machinery to update Gaussian appearance. Both pipelines use a face-inclusive mask stack built from FLAME/VHAP-style regions and BiSeNet-based face parsing. For image-space edits, we compare ControlNet-InstructPix2Pix (CN-IP2P) and FLUX.1 Kontext as 2D edit backends, reporting localization, leakage, writeback coverage, edit strength, and detail-preservation diagnostics on a NeRSemble head sequence. FLUX.1 Kontext produces stronger clown-makeup anchors, while CN-IP2P is more conservative but visibly warps the edited UV canvas. For UV-space edits, a small synthetic fine-tuning set produces more coherent mustache edits than off-the-shelf image editors. Overall, canonical UVs provide a useful control surface for 3D face editing, while preserving dense edited-canvas quality during UV-to-Gaussian transfer remains the central bottleneck.

1. Introduction

3D Gaussian Splatting (3DGS) represents a scene as a collection of anisotropic Gaussians with learned position, opacity, covariance, and appearance [6]. Its real-time rendering and explicit primitives make it an attractive substrate for editing, but faces expose errors that are easy to miss in general scenes. A makeup edit should remain attached to facial surfaces across viewpoints, preserve the subject's identity and face shape, avoid changing hair or background content, and retain local detail rather than appearing as a blurred decal. Smaller structural edits, such as adding a mustache, should add new detail without degrading surrounding skin.

Existing text-guided 3D editing systems often follow a render-edit-consolidate pattern: render multiple views, edit them with a 2D diffusion model, and optimize or update the 3D representation from the edited images [5, 4, 13]. This strategy is powerful for broad scene editing, but it creates a precise failure mode for faces. If two edited views disagree about the lip contour, nose highlight, eye shadow boundary, or cheek paint, blending their RGB values can wash out the edit or introduce inconsistent texture.

To address these limitations, we propose a unified editing framework for 3DGS faces anchored in canonical UV parameterization. Rather than treating multiple rendered views as equally authoritative and risking destructive interference during consolidation, we use a canonical UV atlas as the shared editing surface. This makes edit support, view ownership, and Gaussian writeback explicit, which is crucial when small facial regions must be changed without disturbing nearby identity cues.

We explore this thesis through two complementary case studies that test the boundaries of canonical editing. First, we investigate an image-space anchor-view approach, using off-the-shelf diffusion models to perform complex cosmetic edits such as full-face clown makeup. Second, we investigate a direct UV-space approach, using a custom fine-tuned diffusion model to perform localized structural edits such as adding facial hair. Rather than directly competing these methods on a single task, we treat them as distinct probes into the canonical editing pipeline, revealing how 2D priors handle both broad albedo changes and localized structural additions. The shared goal is to avoid incompatible multi-view RGB averaging while making the remaining 2D-to-3D transfer failure modes inspectable.

Our contributions are:

- a unified 3DGS face-editing framework that leverages canonical UV parameterization to eliminate multi-

*Equal contribution.

view RGB averaging conflicts;

- an anchor-view projection strategy featuring a face-inclusive mask stack and a one-owner UV writeback policy for controlled 2D-to-3D transfer;
- a custom synthetic dataset comprising triplets of original UV textures, edited UV textures, and text prompts, designed to adapt diffusion models to non-standard canonical geometries;
- a fine-tuned UV-space editing pipeline that demonstrates the viability of applying text-guided generative edits directly to a global facial atlas;
- diagnostic evaluations measuring localization, leakage, writeback coverage, held-out edit strength, detail retention, and UV-to-Gaussian transfer quality.

2. Related Work

3D Gaussian Splatting. 3DGS [6] models radiance fields with explicit Gaussian primitives rather than dense voxels or implicit MLP queries. We use a Nerfstudio/Splatfacto-style checkpoint [12] trained on a NeRSemble [7] head capture as the input 3D face representation.

Text-guided 3D editing. Instruct-NeRF2NeRF [5] demonstrated that 2D instruction-guided edits can supervise a 3D radiance field. GaussianEditor variants [4, 13] adapt this idea to 3DGS, using semantic tracing and localized refinement to make editing controllable. Our work follows the same broad render-edit-consolidate paradigm, but changes the consolidation step: rather than averaging many edited RGB predictions, we explicitly assign ownership in canonical face UV space.

2D editing backends. InstructPix2Pix [3] and ControlNet [17] provide instruction-guided and structure-conditioned image editing models. We test ControlNet-InstructPix2Pix (CN-IP2P) as a controlled backend and FLUX.1 Kontext [2] as a stronger image-editing backend. In our setting, these models are not the final output; they create edited anchor images that must be transferred back to the 3D face.

Face priors and parsing. FLAME [8], VHAP [10], and head-avatar methods such as GaussianAvatars [11] show the value of tying face appearance to a stable surface model. We use a FLAME/VHAP-style UV surface as the canonical coordinate system for masks and edit transfer, but keep the final representation as a 3DGS checkpoint. For image-space semantic exclusions, we use a BiSeNet-style face parser [16] through the open-source face-parsing implementation and pretrained weights from [15].

3. Method

We study two canonical editing paths. Method 1 edits one or more rendered anchor views and back-projects the trusted image-space deltas into canonical UV space. Method 2 edits the canonical UV atlas directly and skips the anchor-view editing step. Both methods ultimately use the same mask-gated UV-to-Gaussian writeback stage.

3.1. Method 1: Image Space Edits

3.1.1 Overview

Figure 1 summarizes the anchor-view stage of FaceGS. The input is a trained 3DGS face. We render a primary frontal anchor and side fill anchors from fixed cameras, edit the selected RGB anchors with CN-IP2P or FLUX.1 Kontext, and use an explicit ownership rule to decide which edited view is authoritative for each visible face region. The resulting edited anchors are then passed to the mask and UV writeback stages shown in Figures 2 and 3.

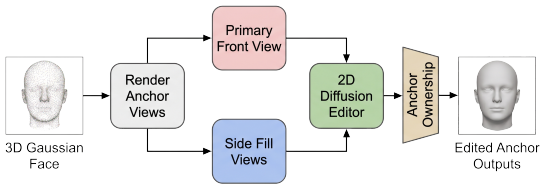

Figure 1: FaceGS anchor-view overview. A trained 3D Gaussian face is rendered from fixed anchor cameras. The 2D diffusion editor produces edited anchors, and the ownership gate records which view should be trusted before transfer into canonical UV space.

3.1.2 Anchor-view image editing

Let G be the trained Gaussian face model and let π_i denote an anchor camera. We render an image

$$I_i = R(G, \pi_i), \tag{1}$$

where R is the 3DGS renderer. A 2D editor E then produces an edited anchor image

$$\hat{I}_i = E(I_i, p), \tag{2}$$

for text prompt p. We use the same clown-makeup prompt across backends, asking for white face paint, dark eye makeup, a red nose, and red lips while preserving hair, hoodie, background, identity, and face shape. This prompt

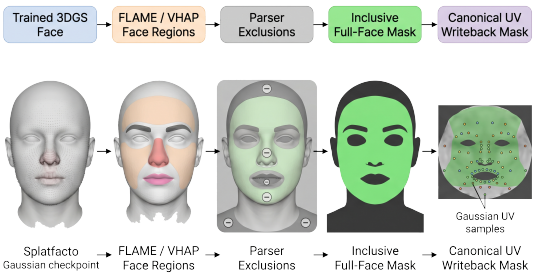

Figure 2: Face-inclusive mask construction. FLAME/VHAP-style regions define editable skin, nose, lips, and brows, while parser exclusions remove hair, background, neck, teeth, and inner-mouth regions before canonical UV writeback.

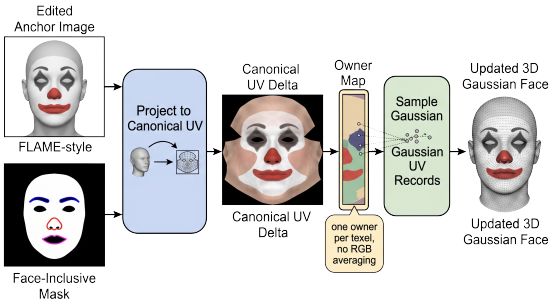

Figure 3: Canonical UV deposition and Gaussian writeback. Edited masked pixels are projected into a camera-independent UV delta. An owner map assigns each texel to one anchor, and each face Gaussian samples the atlas to update appearance without averaging inconsistent RGB predictions.

is intentionally challenging because it requires edits on several small semantic regions while leaving nearby non-face content unchanged.

We compute the image-space edit delta as

$$D_i(x, y) = \hat{I}_i(x, y) - I_i(x, y). \tag{3}$$

This keeps the edit grounded in the original rendered image. The diffusion model is responsible for producing a visually meaningful 2D edit; the rest of the pipeline is responsible for moving only the valid facial part of that edit into 3D.

3.1.3 Face-inclusive masking

A central practical issue was that ordinary face masks were too conservative for cosmetic edits. A "skin-only" mask excludes exactly the regions needed for clown makeup: lips, nose tip, and brows. We therefore build a face-inclusive edit mask from FLAME/VHAP-style canonical regions and image-space parser constraints, shown in Figure 2. The editable support is the union of face skin, nose, lips, and brows. Parser and structural exclusions remove hair, background, neck, teeth, inner mouth, and unstable non-face regions. The parser component follows the BiSeNet face-parsing family [16]; in our implementation, the practical parser weights and inference code come from the public face-parsing repository [15].

The mask is used twice. In image space, it controls which edited pixels are trusted. In canonical UV space, it controls which texels and Gaussians are allowed to receive an update.

3.1.4 Back-projecting edits to canonical UV

The edited anchor image exists only from one camera. To make it usable by the 3D face, we map edited pixels to the canonical face surface. For each visible masked pixel, the renderer/canonical bundle provides a correspondence to a face-surface UV coordinate (u, v), derived from the aligned FLAME/VHAP-style face surface. We deposit $D_i(x, y)$ into a canonical edit canvas $T(u, v)$:

$$T(u, v) \leftarrow D_i(x, y), \quad (u, v) = \mathcal{M}_i(x, y). \tag{4}$$

where $\mathcal{M}_i$ represents the mapping function from anchor image space to canonical UV coordinates. When multiple pixels or anchors map to the same texel, we resolve conflicts using the owner map described below rather than averaging RGB values. The UV canvas is camera-independent: a texel that represents the nose or lip is the same texel regardless of which camera observed it. This is what lets a 2D edit become a persistent 3D face edit.

3.1.5 One-owner writeback

If several anchors write to the same texel, RGB averaging can blur the edit because each view may contain different lighting, perspective, or diffusion hallucinations. We therefore store an owner map $O(u, v)$ and choose one source view for each texel. The front anchor owns most texels; side anchors fill only uncovered regions or regions where they provide better visibility.

Each edited Gaussian g has a canonical surface coordinate (u_g, v_g). We update its appearance by sampling the canonical edit canvas:

$$c'_g = c_g + \alpha_g T(u_g, v_g), \tag{5}$$

where c_g is the Gaussian color/appearance component and α_g is a write confidence derived from mask membership and edit coverage. Gaussians outside the edit support are left unchanged.

Figure 4: Off-the-shelf UV editing failure case. Left: original UV texture. Middle and right: InstructPix2Pix outputs for a mustache-edit prompt. The baseline can introduce facial hair, but it also distorts the UV texture, damages skin detail, and produces inconsistent structure, motivating UV-specific fine-tuning.

3.2. Method 2: UV-Space Edits

3.2.1 Overview

As an alternative to the render-edit-project pipeline described in Method 1, we propose editing the canonical UV canvas directly. Rather than modifying individual anchor views in image space and resolving multi-view conflicts, we fine-tune a FLUX.1-dev-ControlNet-Canny model [9] to modify the global UV texture map. Following this direct edit, we use the same mask-guided Gaussian writeback process detailed above: each editable Gaussian samples the updated UV atlas at its canonical coordinate and applies the resulting appearance update to the 3DGS representation.

3.2.2 Synthetic Dataset Generation

Standard off-the-shelf diffusion models are trained primarily on natural perspective images. Consequently, they struggle to comprehend the flattened geometries and non-standard distortions inherent to canonical face UV maps. When tasked with editing UV textures directly, baseline models often produce structural artifacts, warped facial features, or edits that ignore the intended local geometry.

To bridge this domain gap, we fine-tune a diffusion model explicitly for UV-space editing. Supervised fine-tuning requires data triplets comprising an original UV texture, a target edited UV texture, and the corresponding text prompt. Given the absence of publicly available datasets containing such triplets, we construct a novel synthetic dataset to facilitate training.

We leverage the FFHQ-UV dataset and its accompanying UV generation pipeline [1] as our foundation. FFHQ-UV provides paired 2D facial images and corresponding UV textures. To generate a target edited UV texture for a given pair, we first perform an image-space facial-hair edit on the original 2D face using FLUX. The edited face is then processed through the FFHQ-UV pipeline to extract the altered UV texture. Pairing the original UV texture, edited UV texture, and prompt gives the supervised triplets shown in Figure 5. To keep the proof of concept tractable, we constrain data generation to facial hair additions, specifically mustaches and beards. The final curated set contains approximately 150 manually verified triplets.

Figure 5: Synthetic UV-training data generation. We edit FFHQ-UV source faces with FLUX, re-extract the edited UV texture, and pair the original UV texture, edited UV texture, and prompt as one supervised training triplet.

We fine-tune the FLUX.1-dev-ControlNet-Canny model on this synthetic set so the model sees the UV layout during training rather than being asked to generalize from natural images alone.

4. Experiments

4.1. Dataset and Setup

Our experiments utilize a NeRSemble head capture [7] and a trained Splatfacto/3DGS checkpoint comprising 244,813 Gaussians. The canonical edit bundle is constructed for the same subject, providing FLAME/VHAP-style UV correspondences and semantic region masks.

The editing setup varies depending on the chosen methodology. For image-space edits (Method 1), the primary edit source is a frontal anchor, with side anchors acting as fill views for regions poorly observed from the front. For the UV-space approach (Method 2), fine-tuning and inference are performed directly on the unedited global UV texture.

Task scopes. Because cosmetic and structural edits challenge diffusion models in different ways, we evaluate them separately. Method 1 (image space) uses broad, multi-region clown makeup, where maintaining global face structure is critical. Method 2 (UV space) uses localized facial-hair additions, where the key challenge is teaching the model the flattened topology of UV textures. We therefore treat the two methods as distinct proof-of-concept pipelines

rather than a direct comparative ablation.

Our evaluation of the image-space pipeline compares two distinct 2D editing backends:

- **CN-IP2P**: A ControlNet-InstructPix2Pix backend that provides strong structural conditioning and yields more conservative edits.
- **FLUX.1 Kontext**: A stronger image-editing backend that produced the most convincing single-view clown-makeup anchors during our trials.

4.2. Metrics

Our quantitative metrics serve as structural diagnostics rather than absolute perceptual scores and are intended to be contextualized alongside the qualitative figures. For image-space edits, we report the following quantities for both CN-IP2P and FLUX.1 Kontext under an identical evaluation setup:

- **Writeback coverage**: The fraction of the editable support (face, nose, lips, brows) that successfully receives an edit.
- **Outside-mask change**: The mean RGB change outside the edit mask; lower values indicate better preservation of identity and background.
- **Leakage ratio**: The fraction of the total RGB change that occurs outside the intended edit mask.
- **Ring drift**: The mean RGB change in a narrow band immediately surrounding the mask boundary; lower values indicate less perturbation of the protected transition region.
- **Held-out edit strength**: The mean RGB change inside the edit mask, measured on non-primary evaluation views.
- **Detail ratio**: The Laplacian standard-deviation ratio (after/before) inside the edit mask. This serves as a proxy for high-frequency detail and noise, rather than a direct score of perceptual quality.

Due to the lack of ground-truth 3D cosmetic edits, evaluating the UV-space method (Method 2) relies primarily on monitoring fine-tuning loss convergence, complemented by qualitative visual assessments of the edited UV canvases and their resulting 3D projections.

5. Results

5.1. Image space edits

5.1.1 Qualitative comparison

Figure 6 compares the original render with the CN-IP2P and FLUX.1 Kontext writeback results. The CN-IP2P edit is

Figure 6: Final face-edit gallery. From left to right: original 3DGS render, CN-IP2P writeback, and FLUX.1 Kontext writeback. CN-IP2P is conservative; FLUX.1 Kontext better matches the target clown makeup, but both inherit spotty artifacts from sparse or misaligned Gaussian support.

Figure 7: Edited canonical UV canvases before Gaussian writeback. CN-IP2P produces a recognizable but visibly warped clown edit, while FLUX.1 Kontext gives cleaner white paint, blue eyes, red nose, and red mouth structure. Comparing these canvases to Figure 6 shows how dense UV quality is lost during transfer to sparse Gaussian appearance parameters.

controlled and conservative, but it is qualitatively weaker for this target: the edited canvas warps the face structure, the makeup is less semantically clean, the white face paint is less coherent, and the result does not match the intended clown appearance as well as FLUX.1 Kontext. FLUX.1 Kontext produces a stronger and more readable edit, with clearer white face paint, blue eye shapes, a red nose, and a red mouth region. After writeback, those stronger 2D edits remain more visible in 3D, although the final render is still limited by the fixed Gaussian support.

5.1.2 Quantitative diagnostics

Table 1 summarizes the paper metrics. Coverage is computed from actual writeback coverage over the face-inclusive regions rather than from the full target-mask area. Because both backends use the same canonical support, their writeback coverage is identical in this run at 98.1%.

The remaining metrics reflect the expected backend tradeoff. CN-IP2P has lower outside-mask change (1.13 versus 1.37), consistent with a more controlled and conservative edit. FLUX.1 Kontext has a lower leakage ratio (12.5% versus 15.0%) and visibly stronger edited-face changes in the held-out views (65.0 versus 45.8), which

Figure 8: Side-view Gaussian render of the FLUX.1 Kontext edit. The makeup remains attached from a non-frontal view, but sparse and uneven Gaussian support makes it noisy near hair, background boundaries, and lower-density facial areas.

Method	Coverage ↑	Outside Δ ↓	Leakage ↓	Ring Δ ↓	Held-out Δ ↑	Detail ≈ 1
CN-IP2P	98.1%	1.13	15.0%	39.6	45.8	1.39
FLUX	98.1%	1.37	12.5%	55.5	65.0	1.43

Table 1: Quantitative diagnostics for the reported runs. Coverage measures writeback support over face, nose, lips, and brows. Outside Δ is mean RGB change outside the edit mask, so lower values indicate more conservative preservation of protected regions. Leakage is the fraction of total RGB change outside the edit mask. Ring Δ measures mean RGB change in a narrow band around the mask boundary; lower values indicate less boundary perturbation. Detail is a Laplacian ratio where values near 1 indicate similar high-frequency energy to the original, while larger values may reflect either desired makeup texture or unwanted high-frequency artifacts.

matches the qualitative observation that it puts more of its change into the intended facial support. The cost of this stronger edit is higher ring drift around the mask boundary (55.5 versus 39.6): FLUX perturbs the transition region more, while CN-IP2P better preserves the immediate neighborhood around the editable face support. The detail ratios for both methods are above 1, so they should not be interpreted as "more detail is always better"; in this setting they also capture high-frequency makeup texture, noise, and splat artifacts.

5.2. UV-space edits

5.2.1 Fine-tuning convergence and quality

We fine-tune the FLUX.1-dev-ControlNet-Canny model on our synthetic dataset for approximately 800 iterations. Figure 9 shows that early checkpoints often produce cartoonish or unstable facial hair, while later checkpoints produce

Figure 9: Qualitative UV-space fine-tuning progression. Rows show outputs around 40, 300, and 700 training steps. Early outputs often hallucinate stylized or unstable facial hair; later outputs better respect the face texture and place mustaches more coherently in the UV domain.

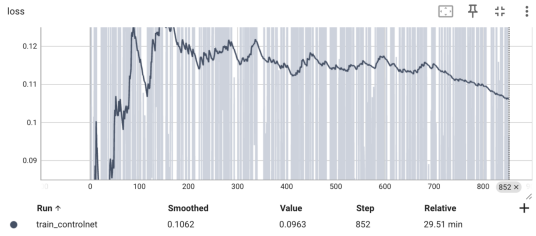

Figure 10: Fine-tuning loss over training. The smoothed loss plateaus early, so the loss curve alone does not capture the visible improvement in UV-space edit quality shown in Figure 9.

more plausible mustache structure and better preserve the surrounding UV texture.

The training loss plateaus early and decreases only modestly, even as visual quality improves substantially (Figure 10). This decoupling between scalar optimization loss and perceptual quality is common in diffusion-based editing pipelines, where small numerical changes can correspond to large improvements in texture realism and semantic placement [14]. Consequently, we evaluate the UV-space method primarily through qualitative canvas and reprojection behavior.

Compared to baseline off-the-shelf models, the fine-tuned model produces edits that are more structurally coherent within the unusual UV geometry. Figure 11 shows the available UV-space comparison: the unedited UV texture, a degraded baseline edit, and the fine-tuned mustache result. The fine-tuned output better preserves surrounding skin while adding localized facial hair with more plausible texture.

Figure 11: Direct UV-space editing comparison. Left: original UV texture. Middle: baseline off-the-shelf edit, which degrades skin texture and produces an unstable mustache region. Right: fine-tuned FLUX-ControlNet output, which localizes the mustache while better preserving nearby face texture.

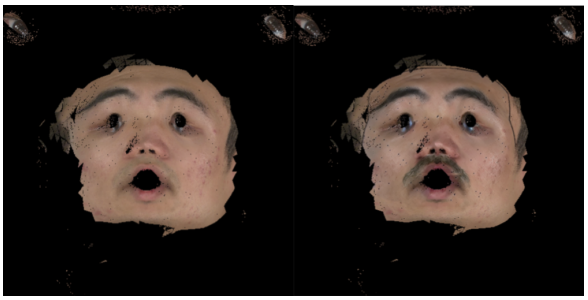

Figure 12: Generalization to a mask produced by our Gaussian projection pipeline. Left: original projected UV canvas. Right: fine-tuned UV-space edit. The generated facial hair is less clean than on FFHQ-UV training examples, but it remains localized despite the patchier splat-derived canvas.

5.2.2 Reprojection quality

However, direct UV-space editing remains bottlenecked by the same 2D-to-3D representation gap detailed in Section 5.1. Despite the sharper mustache synthesized in the UV domain (Figures 11 and 12), much of this detail is lost during the final projection back to the 3D scene. Because the underlying Gaussian primitives in the edited region may be sparse or misaligned with the new texture, the reprojection can yield a diffuse blur rather than preserving sharp individual hairs.

Nevertheless, the unedited regions of the face remain stable. Because this pipeline bypasses multi-view RGB averaging, high-frequency details in non-edited areas are largely preserved; the failure mode is concentrated in representing newly introduced structure with pre-existing skin Gaussians.

6. Discussion and Conclusion

This project demonstrates that a useful abstraction for 3D face editing is not to "average all edited views," but rather to "transfer reliable 2D evidence through a stable canonical coordinate system." By treating our two methods as complementary case studies, we identified clear tradeoffs. Method 1 shows that off-the-shelf models can provide strong semantic reasoning for cosmetic changes such as clown makeup, but anchoring to a small set of views risks baking in view-dependent lighting and editor artifacts. Method 2 shows that editing the global UV canvas directly removes multi-view RGB conflicts and can synthesize localized facial-hair texture, but the UV domain gap requires custom fine-tuning.

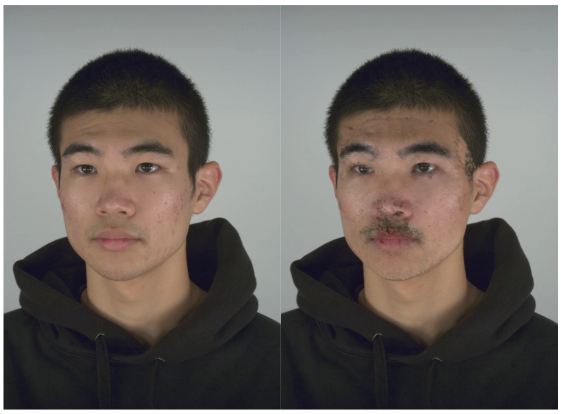

Figure 13: Reprojection of the UV-space mustache edit into the 3D Gaussian render. The edit remains localized around the mouth, but fine individual-hair detail becomes softer and noisier after transfer into the existing Gaussian appearance parameters.

The main limitation is the mismatch between a rigid canonical UV map and the flexible, point-based support of 3DGS. A FLAME/VHAP-style surface anchors facial features and makes mask ownership explicit, but it does not naturally create new geometry or non-face structures such as hair. Both methods therefore hit the same bottleneck: the 2D-to-3D representation gap. Attempting to add a volumetric mustache purely by altering the color of existing skin Gaussians stretches the physical limits of the scene. A high-fidelity 2D edit must still survive a lossy transfer into sparse Gaussian color parameters. Preserving the density and quality of the edited UV canvas during this transfer remains the central technical challenge.

Immediate future work must focus on bridging this representation gap. Rather than treating writeback as a one-shot color copy, future pipelines should dynamically densify or spawn new Gaussians in high-residual edited regions. This would ensure that sharp cosmetic boundaries or fine individual hairs are not forced onto sparse, pre-existing splats. Additionally, utilizing mesh-anchored or FLAME-rigged Gaussians could allow newly edited splats to be placed directly onto the facial surface with perfectly stable UV coordinates.

Once this reprojection bottleneck is resolved, direct UV-space editing should benefit from data scaling. Due to project constraints, our fine-tuning relied on a small synthetic dataset of roughly 150 facial hair edits. Despite this scale, the fine-tuned model showed useful structural comprehension on UV textures and improved over the off-the-shelf baseline. Scaling the data generation pipeline across more identities, face shapes, lighting conditions, and edit types such as scars, tattoos, aging effects, and full-face stylistic makeup is the most direct path toward a more general text-guided 3D face editor.

Finally, broader evaluation is limited by data availability: there are relatively few accessible, permissioned, multi-view face datasets with enough quality and calibration to train 3DGS face models. Even scarcer are datasets for edited faces. Scaling this work will require testing across more identities, capture conditions, and datasets once such data is available.

7. Acknowledgements

We thank Professor Felix Heide and Lekang Yuan for their support, guidance, and organization of an amazing semester. We also thank the creators of the NeRSemble v2 dataset for providing access to their dataset, which made our 3D face editing experiments possible.

References

[1] Haoran Bai, Di Kang, Haoxian Zhang, Jinshan Pan, and Linchao Bao. FFHQ-UV: Normalized facial UV-texture dataset for 3D face reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 362–371, 2023.

[2] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[3] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pages 18392–18402, 2023.

[4] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *CVPR*, pages 21476–21485, 2024.

[5] Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *ICCV*, pages 19740–19750, 2023.

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

[7] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Transactions on Graphics*, 42(4):161:1–161:14, 2023.

[8] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics*, 36(6):194:1–194:17, 2017.

[9] Xinsong Ma, Yiyi Zhao, Haotian Liu, Jiayi Ji, et al. FLUX.1-dev-ControlNet-Canny. `https://huggingface.co/XLabs-AI/flux-controlnet-canny`, 2024. Developed by XLabs-AI.

[10] Shenhan Qian. Vhap: Versatile head alignment with adaptive appearance priors, sep 2024.

[11] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *CVPR*, pages 20299–20309, 2024.

[12] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH*, pages 1–12, 2023.

[13] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *CVPR*, pages 20902–20911, 2024.

[14] 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. *arXiv preprint arXiv:2403.08733*, 2024.

[15] Valikhujaev Yakhyokhuja. face-parsing. `https://github.com/yakhyo/face-parsing`, 2024. GitHub repository.

[16] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision*, pages 325–341, 2018.

[17] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023.