TUM School of Computation, Information and Technology
Technical University of Munich

Master's Thesis in Robotics, Cognition, Intelligence

Emotion-Driven Editing of Gaussian Avatars

Emotionsgesteuerte Bearbeitung von Gaussian Avataren

Supervisor Prof. Dr. Matthias Nießner

Advisor Shivangi Aneja, M.Sc. Informatics

Author Abhinav Utkarsh, B.Tech. Information Technology

Date February 5, 2025 in Munich, Germany

Disclaimer

I confirm that this Master's Thesis is my own work and I have documented all sources and material used.

Munich, Germany, February 5, 2025 [redacted]
(Abhinav Utkarsh, B.Tech. Information Technology)

Acknowledgements

This thesis would not have been possible without the invaluable guidance of my exceptional advisor, Shivangi Aneja. From the earliest stages, when optimization challenges appeared insurmountable, she provided innovative ideas and insightful strategies that shaped the direction of my research. Her profound expertise in 3D avatar generation, optimization techniques, computer graphics, and deep learning proved to be an indispensable asset. Furthermore, her flexibility—whether in adapting the project scope or introducing cutting-edge state-of-the-art (SOTA) research as it emerged—was both pivotal and inspiring. Her ideas were not only remarkable but also engaging to implement, leading to outcomes that were both impressive and deeply fulfilling throughout the process.

I am deeply grateful to my supportive parents, who have been my unwavering pillars of strength throughout my studies in Germany. From boarding the flight to Germany until today, their steadfast encouragement has been a constant source of comfort and motivation. During challenging times, their warm words of wisdom and guidance gave me the reassurance to persevere and find the light at the end of the tunnel. Their unconditional emotional and financial support has been invaluable, allowing me to focus wholeheartedly on my studies and navigate this journey abroad. For this, and so much more, I will always remain profoundly thankful.

My gratitude also extends to the Chair of Visual Computing and AI, led by Professor Dr. Matthias Nießner at the Technical University of Munich, for providing the infrastructure, technical expertise, and resources that were pivotal to the successful completion of this thesis work. Their support played a crucial role in overcoming the challenges inherent in this endeavor.

Finally, I would like to express my heartfelt thanks to everyone who contributed, directly or indirectly, to my journey—your support has been invaluable and deeply appreciated.

Abstract

High-fidelity 3-D head avatars are becoming indispensable for emerging spatial computing devices, such as Apple's Vision Pro, where virtual "spatial persona" calls and other immersive experiences demand natural, expressive facial renderings. Yet, the challenge of creating avatars that convincingly display a broad range of emotions—complete with texture cues like wrinkles, skin-tone variations, and shadows—remains largely unsolved. Traditional 3D pipelines based on purely geometric blend-shapes or one-shot 2-D editing often omit these subtleties, leading to expressions that look hollow or appear inconsistent when viewed from varying spatial positions or angles.

To tackle this gap, we propose *Emotion-Driven Editing of Gaussian Avatars (EMO-GA)*, a pipeline designed to create expressive and emotionally rich 3D avatars. The process begins by transforming neutral or mildly expressive frames into "pseudo" emotion exemplars via text-prompted diffusion (e.g., UltraEdit), injecting details like deeper cheek contours for happiness or shading shifts for anger. Next, photometric FLAME-based tracking is applied to multi-view data, producing accurate head geometry to which the Gaussian splats are rigged. A baseline Gaussian avatar is then constructed, augmented with a color MLP and patch-based photometric constraints to capture both global shape and fine texture cues. To finalize the pipeline, we optimize the shared FLAME expression offsets and per-Gaussian texture features against the noisy "pseudo" edits generated by diffusion models. This optimization involves a delicate balance: while geometry updates capture the overall structure of the expression, texture adjustments are critical for fine details like shading, lip color, and subtle skin-tone shifts. To avoid over-fitting to diffusion inconsistencies and to maintain spatiotemporal consistency, we use shared expression offsets across frames, coupled with iterative cycles of geometry and texture refinement. By incorporating robust regularization strategies, such as view weighting and patch-based photometric losses, the pipeline ensures that both geometry and texture converge coherently to reflect the desired emotional state without introducing any significant flicker or causing nearly zero identity loss.

Our results show that geometry alone fails to capture the richness of facial affect; textural refinement is indispensable for natural emotional expressions. EMO-GA integrates these elements into a single, view-consistent representation, eliminating the need for massive labeled 3D emotion corpora. By achieving nuanced expression edits and preserving multi-view fidelity, this pipeline paves the way for more authentic avatar interactions in spatial FaceTime calls, virtual collaboration platforms, and other scenarios where subtle emotional detail truly matters.

Kurzfassung

High-fidelity 3D-Kopf-Avatare sind heute unverzichtbar für neuartige Spatial-Computing-Geräte wie Apple's Vision Pro, bei denen virtuelle "spatial persona"-Calls und andere immersive Anwendungen auf natürliche, ausdrucksstarke Gesichtsanimationen angewiesen sind. Dennoch stellt das Erzeugen glaubwürdiger Emotionen—einschließlich feiner Texturmerkmale wie Falten, Hauttönungsvariationen oder Schattierung—eine ungelöste Herausforderung dar. Konventionelle Methoden, die rein auf geometrischen Blend-Shapes oder einmaliger 2D-Bildbearbeitung basieren, lassen oft diese Feinheiten vermissen und führen zu Avataren, deren Mimik aus verschiedenen Blickwinkeln unvollständig oder inkonsistent wirkt.
Um diese Lücke zu schließen, schlagen wir *Emotion-Driven Editing of Gaussian Avatars (EMO-GA)* vor, eine Pipeline zur Erzeugung emotional reichhaltiger 3D-Avatare. Zunächst verwandelt unser Ansatz neutrale oder schwach expressive Frames mittels textbasierter Diffusion (z. B. UltraEdit) in "pseudo"-Emotionsexemplare und injiziert realistische Details wie ausgeprägtere Wangen für einen fröhlicheren Ausdruck oder veränderte Schatten für einen wütenden. Anschließend kommt ein photometrisches FLAME-basiertes Tracking zum Einsatz, das aus Multi-View-Daten präzise Kopfgeometrie extrahiert und die Gaussian Splats riggt. Auf dieser Grundlage wird ein baseline Gaussian Avatar aufgebaut, ergänzt durch ein Color-MLP und patch-based photometric Constraints, um sowohl globale Farbgebung als auch lokale Texturen einzufangen. In einem letzten Schritt werden die FLAME-Expressionsparameter und die per-Gaussian-Texturfeatures gegen die in der Diffusion erzeugten "pseudo" Edits optimiert. Hierbei gilt es, ein ausgewogenes Verhältnis zu wahren: Während Geometrie-Updates die grobe Struktur der Mimik einfangen, sind Texture-Änderungen entscheidend für Feinheiten wie Hautfärbung, Lippenkoloration oder subtile Schattierungen. Um ein Überanpassen an inkonsistente Diffusionsausgaben zu vermeiden und gleichzeitig spatiotemporale Kohärenz zu wahren, setzen wir auf gemeinsame Expressionsoffsets über alle Frames und zyklische Optimierungen von Geometrie und Textur. Durch zusätzliche Regularisierung—etwa durch View Weighting und Patch-basierte photometrische Verluste—wird sichergestellt, dass sowohl Geometrie als auch Textur konsistent zusammenlaufen und die gewünschte Emotion widerspiegeln, ohne Flicker oder nennenswerte Identitätsverluste zu erzeugen.
Die Ergebnisse zeigen, dass reine Geometrie allein die Vielfalt und Tiefe emotionaler Gesichtsausdrücke nicht abdecken kann; insbesondere Texture-Refinements erweisen sich als unverzichtbar für einen natürlichen Gesamteindruck. EMO-GA integriert all diese Komponenten in eine einheitliche, blickrichtungsstabile Repräsentation und kommt dabei ohne umfangreiche, gelabelte 3D-Emotionsdatensätze aus. Dadurch ermöglicht die Pipeline nuancierte Ausdrucksänderungen bei gleichzeitig hoher Multi-View-Konsistenz—ein entscheidender Fortschritt für intensivere Avatar-Interaktionen in spatial FaceTime-Calls, kollaborativen virtuellen Umgebungen und anderen Anwendungsfeldern, in denen subtile emotionale Details den entscheidenden Unterschied ausmachen.

Contents

Chapter 1

Introduction

Recent advances in virtual reality and other forms of immersive technology have placed realistic 3D head avatars at the forefront of computer graphics and vision research. These avatars are indispensable for high-fidelity remote collaboration, lifelike animations in film or gaming, and personalized digital assistants. Despite impressive progress in static as well as dynamic 3D head reconstructions [Kir+23b; Qia+24b; Qia+24a], achieving *expressive and controllable* avatars—encompassing *rich emotional variations*—remains a crucial and open challenge. Nevertheless, many existing approaches to 3D emotion manipulation rely on large datasets with detailed emotion labels [Deh+24] or on parametric blendshape models that capture only a small subset of possible expressions.

Moreover, emerging cutting-edge spatial computing hardware, such as the Apple Vision Pro, demonstrates the rising need for user-friendly *"spatial persona"* avatars that can seamlessly convey a range of emotions, from subtle smiles to pronounced reactions. Such realistic avatars help avoid the *"uncanny valley"* and foster more compelling experiences in virtual encounters. However, purely geometric (blendshape) methods often lack the crucial *textural attributes*—e.g., wrinkles, shading, and skin details—that make an emotional expression genuinely *plausible* under varied viewpoints and lighting. Indeed, while the recent *Emo3D* pipeline [Deh+24] presents an extensive process to generate 3D expressions from textual prompts via a *52-dimensional blendshape vector*, it underscores how geometry alone may fall short of complete realism. Another 3D dataset, TEAD (Zhong *et al.*, 2023), further illustrates the concerted efforts in *facial expression generation (FEG)* by linking textual descriptions to blendshapes [Zho+23], yet it, too, focuses predominantly on *geometric deformations rather than textural fidelity*. In practice, collecting and training such data-hungry FEG models for every new user or scenario is prohibitively expensive and still does not guarantee the *subtle texture cues* that convey true emotional nuance.

Avatar animation has seen rapid advancements in both 2D and 3D methods aimed at enhancing expressiveness. In the 2D domain, techniques such as Audio-Driven Emotional Video Portraits [Ji+21] show how vocal cues can drive real-time facial animation in videos, while diffusion-based image-to-image editing frameworks have gained traction and can be used for emotion editing (e.g., UltraEdit with masking or ControlNet edge maps) to enable localized text-prompted, Stable Diffusion-based emotion edits in single images. Additionally, OpenAI's Sora [Ope25] offers a text-to-video model capable of animating static images based on descriptive prompts, which could potentially be used for generating emotionally expressive videos from a given image. Although these purely image-based approaches can yield compelling outcomes, they often struggle with large viewpoint changes or with preserving spatiotemporal consistency across frames.

Meanwhile, in 3D domain, pipelines like the *Emo3D* framework [Deh+24] and *EMOTE* [Dan+23] have pushed parametric blendshape models to support textual or one-hot emotion input. These methods highlight the growing demand for emotion-centric 3D editing, but since 3D emotional

data is more challenging to capture, data generation often requires multiple steps to create plausible emotional expressions. Furthermore, their reliance on curated datasets and/or purely geometric manipulations can leave out subtle texture variations (e.g., wrinkles or shading), which are crucial for conveying emotions effectively in the final 3D avatar. A more detailed overview of both 2D and 3D emotion-editing methods is presented in the Related Work (Chapter 3).

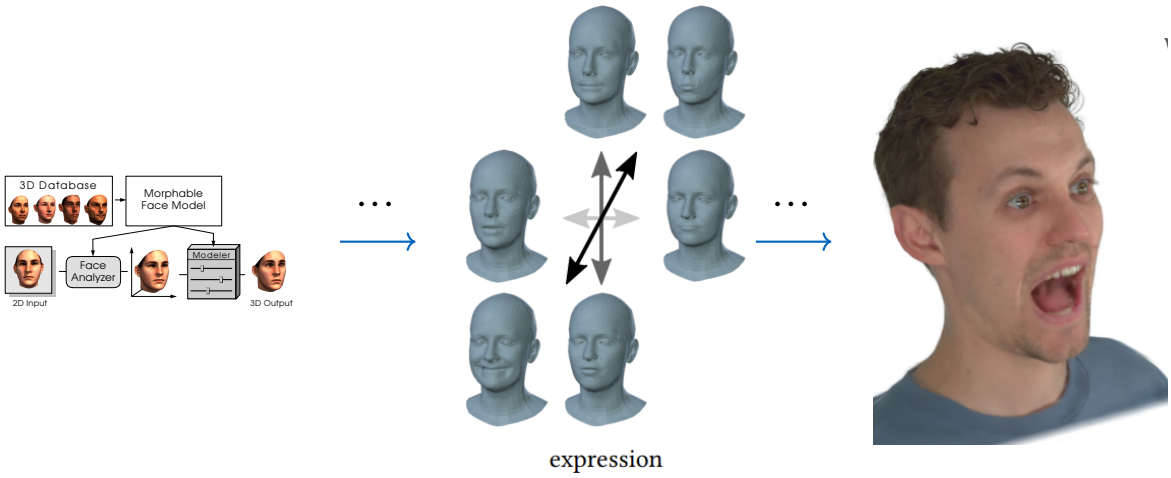
Despite this progress, current solutions face key drawbacks. First, purely geometric (blendshape) edits can lack the vital textural cues—for instance, realistic shadows, wrinkles, or eye-region shading—that convey nuanced affect. Second, data-driven 3D pipelines still often demand large annotated emotion corpora, an expensive requirement further complicated by the need to capture multi-view or volumetric data. Third, many 2D methods address only a single viewpoint and can suffer from flicker or identity drift when applied to a full video or multi-view setup. Collectively, these issues limit the realistic depiction of emotion under varied poses and lighting, making it hard to produce truly compelling, free-viewpoint emotional avatars.

In this thesis—titled ***Emotion-Driven Editing of Gaussian Avatars (EMO-GA)***—We propose a self-supervised pipeline that combines 2D diffusion-based editing (to provide rich emotional cues) with a 3D Gaussian avatar representation (for view-consistent geometry and texture). Specifically, our method uses diffusion-model outputs (e.g., from UltraEdit) as "pseudo-ground-truth" to guide both geometry (blendshape offsets) and texture (via per-Gaussian color features), thereby addressing the shortcomings of purely geometric or purely image-based approaches. While we optimize geometry to capture structural aspects of expression, as we will see, optimizing only geometry seldom depicts emotions strongly; hence, we also refine texture to ensure more expressive and natural emotional representations. By leveraging multi-view input for baseline reconstruction, the framework maintains high fidelity and avoids large-scale 3D emotion labels or costly volumetric capture of expression data.
Key contributions of this work are:

- **Bridging 2D and 3D:** We show how diffusion-driven 2D edits can be back-propagated into a 3D avatar, allowing free-viewpoint emotional expressions without requiring labeled 3D emotion corpora.
- **Self-Supervised Photometric Guidance:** Through masked image-to-image edits (UltraEdit) and a multi-view photometric loss, the pipeline captures subtle expression changes and texture details (e.g., denser happier cheeks, changes in skin tone when portraying intense emotions (such as anger)
- **Shared Geometry and Texture Optimization:** A shared expression offset in FLAME [Li+17] avoids frame-by-frame jitter, while the per-Gaussian color MLP refines textural fidelity under varied viewpoints.
- **Lightweight, Flexible Representation:** By using Gaussian splatting rather than full neural volumes, the resulting avatar offers real-time rendering and simpler rigging for practical adoption in spatial computing hardware.

Ultimately, by focusing on *both geometry and texture*—via the expressive power of Gaussian splatting—this work addresses a significant limitation in *blendshape-only pipelines* and offers a more comprehensive strategy for *emotion-driven* edits. This *Emotion-Driven Editing of Gaussian Avatars* approach not only advances the realism of facial expressions but also lowers barriers to adoption, since no extensive 3D emotion dataset curation is required.

1.1 3D Avatars: From 3D Morphable Models to NeRF and 3D Gaussian Splatting



1999 - 3DMM (Blanz and Vetter) 2017 - FLAME 2024 - Gaussian Avatars (SOTA)

Figure 1.1: Evolution of 3D Face Models: From 3D Morphable Models (3DMM) [BV99], which pioneered parametric face modeling, to FLAME [Li+17], which introduced articulated pose and expression control, and now to Gaussian Avatars [Qia+24a], which integrates rigging with photorealistic rendering for real-time, high-fidelity facial animation.

1.1.1 Introduction and Overview

The concept of a *3D Morphable Model (3DMM)* was first popularized by Blanz and Vetter [BV99]. In essence, 3DMMs represent a statistical framework for modeling 3D facial shapes and their variations. By analyzing a collection of scanned faces, one can extract a set of *principal components* that capture both *inter-person* identity differences (e.g., facial proportions, jaw structure) and *intra-person* expression differences (e.g., raised eyebrows, mouth openings). This parametric model then serves as a powerful tool for tasks such as facial expression transfer, identity editing, and animation.

Figure 1.1 illustrates the timeline of 3D face modeling advancements, starting with the introduction of 3DMMs in 1999. These models laid the foundation for FLAME (2017), which enhanced 3DMMs with articulated pose and expression control. FLAME's incorporation of joint-based deformations made it suitable for realistic head and facial movements. The evolution continues with Gaussian Avatars (2024), which combine parametric modeling with photorealistic rendering and real-time animation capabilities, marking the state-of-the-art in 3D face modeling.

Early 3DMMs primarily relied on *Principal Component Analysis (PCA)* to decompose shape variations:

$$\mathbf{S}(\boldsymbol{\alpha}) = \overline{\mathbf{S}} + \sum_{i=1}^{M} \alpha_i \mathbf{S}_i, \tag{1.1}$$

where $\overline{\mathbf{S}}$ is the average (mean) face mesh, $\mathbf{S}_i$ are the principal component directions, M is the number of components, and α_i are the scalar coefficients. By tuning these coefficients, one can synthesize novel faces that lie within the "space" of observed variations.

1.1.2 Progressions: FLAME and Beyond

Although the initial *Basel Face Model (BFM)* [Pay+09b] introduced more diverse identities, it still used PCA for expression modeling and lacked an articulated jaw or neck. In contrast, *FLAME* (Faces Learned with an Articulated Model and Expressions) [Li+17] incorporates:

- *Articulated Joints:* FLAME includes articulated joints for the neck, jaw, and eyeball rotation, allowing for realistic head and facial movements.
- *Expression Blendshapes:* A set of additive deformations that capture non-rigid motions, such as smiles, frowns, or other subtle changes in facial expressions.
- *Linear Blend Skinning (LBS):* A deformation technique commonly used in character animation, providing a low-dimensional parameter vector for pose and expression.

Shape, Pose, and Expression Blendshapes: FLAME models a template face $\mathbf{T} \in \mathbb{R}^{3N}$ (with N vertices) and refines it through three types of blendshapes, each responsible for different modes of deformation. Specifically, let

$$\mathbf{M}(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) = W\Big(\mathbf{T} + \underbrace{B_S(\boldsymbol{\beta})}_{\text{shape}} + \underbrace{B_P(\boldsymbol{\theta})}_{\text{pose}} + \underbrace{B_E(\boldsymbol{\psi})}_{\text{expression}}, \mathbf{J}, \mathbf{W}\Big). \tag{1.2}$$

Here, $W(\cdot)$ denotes the standard linear blend skinning (LBS) function, $\mathbf{J}$ is a set of learned joint locations, and $\mathbf{W}$ are per-vertex blend weights. The individual blendshape terms are:

1. **Shape Blendshapes** $B_S(\boldsymbol{\beta})$:

$$B_S(\boldsymbol{\beta}; \mathbf{S}) = \sum_{n=1}^{|\boldsymbol{\beta}|} \beta_n \mathbf{s}_n, \tag{1.3}$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{|\boldsymbol{\beta}|}]^T$ are shape coefficients, and $\{\mathbf{s}_n\}$ are orthonormal principal components learned (e.g., via PCA) from a set of neutral faces.

2. **Pose Blendshapes** $B_P(\boldsymbol{\theta})$:

$$B_P(\boldsymbol{\theta}; \mathbf{P}) = \sum_{n=1}^{K} \Big(R_n(\boldsymbol{\theta}) - R_n(\boldsymbol{\theta}^*)\Big)\mathbf{p}_n. \tag{1.4}$$

Here, $R_n(\boldsymbol{\theta})$ is a rotation matrix derived from pose parameters $\boldsymbol{\theta}$ (e.g., neck or jaw angles), $R_n(\boldsymbol{\theta}^*)$ is a reference (neutral-pose) rotation, and $\{\mathbf{p}_n\}$ represent the learned pose-correction basis. While this term is linear in the space of rotation *matrices*, it is *nonlinear* with respect to the underlying pose parameters.

3. **Expression Blendshapes** $B_E(\boldsymbol{\psi})$:

$$B_E(\boldsymbol{\psi}; \mathbf{E}) = \sum_{n=1}^{|\boldsymbol{\psi}|} \psi_n \mathbf{e}_n, \tag{1.5}$$

where $\boldsymbol{\psi} = [\psi_1, \ldots, \psi_{|\boldsymbol{\psi}|}]^T$ are expression coefficients, and $\{\mathbf{e}_n\}$ form an orthonormal basis for non-rigid facial motions like smiling, frowning, or lip-puckering.

Articulated Model and Skinning. Once the final mesh $\mathbf{T} + B_S + B_P + B_E$ is computed, FLAME applies a standard linear blend skinning (LBS) function $W(\cdot, \mathbf{J}, \mathbf{W})$ to place the mesh in the global coordinate frame, accounting for the rigid transformations defined by the joint locations $\mathbf{J}$ and blend weights $\mathbf{W}$. This process enables plausible deformations around the neck, jaw, and eyes as these joints rotate.

1.1.3 Applications and Limitations

3DMMs (including FLAME) are used extensively in industry for face rigging, AR/VR avatars, and performance capture. However, they typically store only *geometry* in a linear or piecewise-linear fashion, omitting high-frequency skin details like wrinkles under dynamic lighting or muscle bulges near the cheeks. Consequently, while 3DMMs excel at providing explicit and interpretable control (mouth open/close, eyebrow raise, etc.), they often struggle to achieve *photorealistic* texture fidelity.

This trade-off between *controllability* (parametric, mesh-based) and *realism* (rich texture, complex lighting) has motivated newer methods such as Neural Radiance Fields (NeRF) and 3D Gaussian Splatting, which capture fine-grained appearance details beyond a PCA-based mesh space.

Figure 1.2: Demonstration of a 3D Morphable Model using the FLAME framework [Li+17]. Each column illustrates variations in shape, pose, and expression parameters. From left to right: (1) deviations in *shape* (identity), (2) *pose* changes around the neck and jaw joints, and (3) *expression* blendshapes for mouth articulation.

1.1.4 Bridging Realism and Control: NeRF to Gaussian Splatting

From Discrete Voxels to Continuous Functions: Introduced by Mildenhall *et al.* [Mil+21], *Neural Radiance Fields (NeRF)* represent a paradigm shift in 3D scene reconstruction. Instead of discretizing space into voxels or mesh vertices, NeRF parameterizes a *continuous* 5D function $F_\theta(\mathbf{x}, \mathbf{d})$, where $\mathbf{x} = (x, y, z)$ is a 3D location in space and $\mathbf{d}$ is a 2D viewing direction (e.g., θ and ϕ in spherical coordinates). The conceptual pipeline of NeRF is illustrated in Figure 1.3, detailing how spatial and directional inputs are processed to produce view-dependent colors and densities.

The network outputs:

$$(\mathbf{c}, \sigma) = F_\theta\big(\mathbf{x}, \mathbf{d}\big), \tag{1.6}$$

where $\mathbf{c} = (r, g, b)$ denotes view-dependent emitted color, and σ is volume density (loosely speaking, how "opaque" that point in space is).

Volume Rendering Equation: To synthesize an image from a novel viewpoint, one casts rays through the scene and integrates color contributions along each ray via a differentiable volume rendering approach:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} \Big(\alpha_i \prod_{j=1}^{i-1} \big(1 - \alpha_j\big) \Big) \mathbf{c}_i, \quad \text{where} \quad \alpha_i = 1 - e^{-\sigma_i \Delta t_i}. \tag{1.7}$$

Here, Δt_i is the distance between sampled points along the ray, and σ_i is the density at the i-th sample. This rendering equation is *fully differentiable*, allowing the network parameters θ to be optimized to minimize a photometric loss with respect to multi-view images.

Advantages and Extensions: Because NeRF learns *both* geometry (encoded in σ) and appearance (**c**) directly from images, it can capture ultra-fine texture details and non-Lambertian reflectance in a more expressive way than traditional meshes or point clouds. Moreover, improvements such as Dynamic NeRF [Gaf+21], NeRSemble [Kir+23b], and other variants address moving scenes (e.g., talking faces), enabling compelling free-viewpoint replays of human performance.

Challenges in Expression Editing: While NeRF-based reconstructions often look extremely realistic, they pose significant difficulty for *expression manipulation*:

- *Black-box Representation:* The learned MLP weights do not easily translate to a rig or blendshape system for direct control of the mouth or eyes.
- *Overfitting to Observed Poses:* If the subject's dataset lacks certain expressions, the model has limited ability to extrapolate new ones.
- *Rendering Speed:* Classical NeRFs can be slow to render, though recent acceleration techniques (e.g., InstantNGP) have improved speed significantly.

Hence, although NeRF is a milestone for reconstructing *static or dynamic* scenes in high fidelity, it remains *less flexible* when one desires direct, user-driven edits to geometry or texture. This motivates hybrid approaches that preserve NeRF-level realism while reintroducing more explicit control structures—leading naturally to 3D Gaussian Splatting, which can be rigged more readily for real-time expression changes.

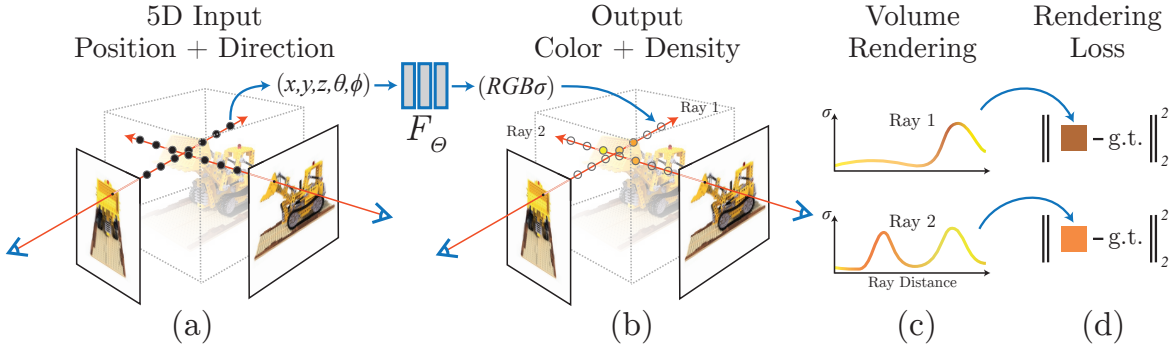


Figure 1.3: Conceptual pipeline of Neural Radiance Fields (NeRF) [Mil+21]. (a) A 5D input comprising spatial coordinates (x, y, z) and viewing direction (θ, ϕ) is passed into a neural network F_θ, which outputs a view-dependent color (R, G, B) and volume density σ. (b) Multiple points along a ray are sampled and processed by the network to obtain their respective colors and densities. (c) These outputs are integrated using the volume rendering equation to compute the pixel color for the corresponding ray. (d) The rendered image is optimized by minimizing the photometric loss between the predicted pixel colors and the ground truth image.

Originally developed for novel view synthesis in static scene reconstruction, Neural Radiance Fields (NeRF) [Mil+21] emerged in 2020 as a powerful approach to learning continuous volumetric representations from multi-view images. While initially used for reconstructing complex environments, researchers quickly adapted NeRF for human head modeling due to its ability to synthesize highly realistic appearances with fine-grained details. Gafni et al. [Gaf+21] extended NeRF to handle dynamic faces using a single monocular video, and more recent efforts like NeRSemble [Kir+23b] incorporate multi-view data to improve reconstruction quality under significant head motion and expression changes.

While NeRF-based approaches offer superior texture fidelity compared to 3DMMs, they have inherent limitations: they encode geometry and appearance in a black-box neural network, making direct edits (e.g., modifying expressions or refining facial details) more challenging. Consequently, research has shifted towards finding a middle ground between explicit control, as in 3DMMs, and high-fidelity photorealism, as in NeRF.

Recently, Gaussian avatars have emerged as the leading approach due to their ability to combine realistic texture representation with computational efficiency. Unlike NeRF, which requires expensive volumetric rendering, 3D Gaussian splatting provides a lightweight yet high-quality representation that allows for real-time rendering and easy manipulation of facial expressions. This balance of photorealism and controllability positions Gaussian-based avatars as the current state-of-the-art in 3D head modeling. The progression from 3D Morphable Models to FLAME and ultimately to Gaussian-based avatars is summarized in Figure 1.1, illustrating the shift from parametric models to photorealistic, efficient representations.

1.1.5 3D Control with Realism : Gaussian Splatting and Avatars

From Volumes to Gaussians: *3D Gaussian Splatting* [Ker+23; Mar+25] has emerged as an alternative to volumetric NeRF. Instead of encoding a scene as a dense MLP or voxel grid, Gaussian Splatting represents the object with a *cloud of anisotropic 3D Gaussians*, each of which has a position μ_k, a covariance matrix Σ_k, and a color feature $\mathbf{v}_k$. Rendering is performed by projecting these Gaussians to the image plane, accumulating color contributions via a *point-based* or *elliptical-splat* rasterization:

$$\mathbf{C}(\mathbf{p}) = \sum_{k=1}^{K} w_k \mathbf{v}_k, \quad w_k = \exp\Big(-\tfrac{1}{2}(\mathbf{p}-\mu_k)^T \Sigma_k^{-1}(\mathbf{p}-\mu_k)\Big). \tag{1.8}$$

Here, $\mathbf{p}$ is the projection of a 3D point onto the 2D image plane, and w_k measures how much that Gaussian contributes to pixel $\mathbf{p}$. Because Gaussians can be adaptively placed in high-detail regions (e.g., eyes, lips) and sparser in low-detail areas (e.g., forehead), one can achieve high-quality results with fewer primitives compared to naive per-voxel sampling.

Advantages Over NeRF: A key benefit is *real-time rendering*: once the Gaussians are learned, splatting them onto the screen is essentially a *point-based rasterization*, which can be hardware-accelerated. Additionally, each Gaussian is an *explicit primitive* with well-defined location, orientation, size, and color. This stands in contrast to a NeRF's implicit MLP parameters. As a result, it is *easier to manipulate or "rig"* these primitives to follow pose or expression changes. Figure 1.4 provides a visual demonstration of these concepts. On the left (Figure 1.4(a)), a Gaussian Avatar is shown at its full scale, rigged to a FLAME mesh. On the right (Figure 1.4(b)), the Gaussians are scaled down to emphasize the individual splats used for rendering.

Gaussian Avatars: Recent methods like *Gaussian Avatars* [Qia+24a] extend the idea of 3D Gaussian Splatting to human heads or full bodies, binding each Gaussian to a skeleton or parametric rig (e.g., FLAME). Suppose each Gaussian is associated with the nearest face triangle or joint in a parametric model. When the head moves or expressions change, the Gaussians transform according to simple *linear blend skinning* or a similar deformation scheme:

$$\mu_k' = \sum_{j=1}^{J} w_{k,j} \mathbf{LBS}_j(\mu_k), \tag{1.9}$$

where $\mathbf{LBS}_j$ is the linear blend skinning transformation for joint j, and $w_{k,j}$ is the weight that Gaussian k has for that joint. This yields an *animatable* Gaussian cloud, enabling real-time

expressions and head motion with a fraction of the computational overhead required by dynamic NeRF methods.

Limitations and Next Steps: While Gaussian Avatars excel at efficiency and can exhibit high-quality textures, their editing capabilities have mostly been limited to re-enactment or retargeting of *observed* expressions, as opposed to generating novel or *unseen* emotions. Directly painting new textures onto Gaussians or adjusting their covariance for wrinkles is conceptually possible, but requires additional optimization or guidance signals. This thesis builds on the *Gaussian Avatar* framework by introducing a 2D diffusion-based editing component, allowing us to incorporate new expressions or emotional states *not* present in the training data.

(a) Gaussian avatars rigged on a FLAME mesh with Gaussians at their complete scale.

(b) Gaussian avatars scaled down to show the individual splats for the subject.

Figure 1.4: Illustration of 3D Gaussian Splatting. (a) Shows a Gaussian avatar rigged to a FLAME mesh, while (b) highlights the splatting of thousands of individual Gaussians, visualized with a zoomed crop on right top.

In summary, *3D Morphable Models* pioneered mesh-based facial parameterization for decades, while *NeRF* introduced a game-changing, implicit approach to capturing complex scene and facial detail. *3D Gaussian Splatting* now fuses the *realism* of neural rendering with *explicit* control primitives, forming the basis of next-generation *Gaussian Avatars*—a promising avenue for expressive, real-time 3D face editing and animation. The following chapters will leverage these developments to propose a novel pipeline for *emotion-driven editing* of 3D Gaussian Avatars, bridging the gap between purely geometric and purely implicit 3D solutions.

Note: While this section introduces the foundations of 3D Gaussian Splatting and Gaussian Avatars, we look into these methods in the *Related Work* section (§3.1).

1.2 Emotion Editing

1.2.1 Emotion Editing in Pixel Space (2D)

2D-based approaches to facial emotion editing are historically the most prevalent, due in large part to the abundance of labeled 2D facial datasets and the relative simplicity of operating within pixel-space rather than a full 3D geometry (as illustrated in Figure 1.5). Such methods typically involve:

1. Image-to-image translation networks [Iso+17], which learn to map an input face image to a target expression (e.g., “sad” or “angry”) by training on large corpora of labeled images.

2. Generative Adversarial Networks (GANs) [Goo+14], which can synthesize facial expressions by sampling or interpolating in a learned latent space, optionally guided by emotion labels or continuous parameters such as valence and arousal.

3. Diffusion-based methods, which iteratively denoise random latent codes to generate images matching a target attribute—e.g., "a happy face." These have achieved state-of-the-art fidelity in single-image editing, but often struggle with consistency across multiple frames. We end up utilizing this approach (§3.3.2), namely *UltraEdit*[Zha+24].

While these 2D-based strategies are not the main focus of this thesis, their strengths and limitations offer valuable insights into the broader challenges of emotion editing. Diffusion-based editing techniques, for instance, can yield remarkably high-quality images at the single-frame level but often encounter temporal and spatial consistency issues when extended to video.

A key advantage of most 2D pipelines is the ability to rapidly modify local facial regions (e.g., the mouth or eyebrows) without reconstructing the entire head in 3D. This makes them particularly attractive for social-media filters or stylized portrait effects, where minimal user intervention is preferred. For example, Audio-Driven Emotional Video Portraits [Ji+21] add explicit emotional cues into a 2D talking-head video by separating speech content from emotional style and then projecting manipulated landmarks back onto the facial region. Similarly, EmoStyle [AL24] modifies real images using continuous valence and arousal values rather than discrete categories—a reminder that many real-world emotional states reside on a continuum (e.g., "mild annoyance" vs. "intense anger").

Nevertheless, transitioning from single-image editing to true sequence-to-sequence pipelines adds considerable complexity. While some 2D works do apply edits frame-by-frame, this straightforward approach is often plagued by temporal flicker and inconsistency—especially with diffusion-based methods. As illustrated in Figure 1.5, small denoising variations can lead to visible pulsing artifacts where subtle details (wrinkles, shadows) shift across consecutive frames. Even methods with explicit temporal constraints (e.g., optical flow, landmark tracking) may struggle with extreme head poses or multi-view angles, as 2D warping or inpainting is rarely robust enough to handle large occlusions or complex lighting changes.

Finally, the lack of a physically consistent 3D model for lighting and geometry makes it challenging to embed 2D emotion editing into VR or AR contexts, where the user's viewpoint can shift arbitrarily. These limitations underscore the need for 3D approaches to achieve truly immersive and view-consistent emotion editing—the core endeavor of this thesis.

1.2.2 Emotion Editing in 3D

In contrast, 3D-based emotion editing tackles many of these shortcomings by representing the face's shape, texture, and possibly hair in a 3D format, thus enabling free viewpoint rendering, more natural integration of lighting, and the potential for geometric consistency across frames. This is crucial for applications in virtual reality systems, where the user's viewpoint may change arbitrarily, or lighting must respond realistically to head movements. Moreover, 3D representations can store high-frequency skin details and wrinkles in a way that can be rendered from any viewpoint, thereby avoiding the frequent "cut-out" look of purely 2D editing.

The challenge, however, is that 3D emotional manipulation is significantly more complex: any modifications to a 3D avatar's expression must maintain consistent geometry (i.e., face shape) while also adjusting the texture to reflect subtle changes such as fine wrinkles or shadows under the eyebags that shift with expression. Moreover, acquiring 3D training data

(a) View 02 (b) View 03 (c) View 08 (d) View 12

Figure 1.5: Here in the image, each column represents a subset of camera views picked from the NerSemble dataset [Kir+23b] (see Section 2.1). The top row shows ground truth images, the second row displays detailed Canny maps of the subjects, and the third row contains generated images for a fixed seed and prompt to make the person appear happy. As observed, even with structural conditioning using the Canny map, methods like ControlNet [ZRA23] completely change the subject. Additionally, the results exhibit high identity loss, noticeable color loss, spatiotemporal artifacts, and mismatched shadows and wrinkles across views. While other modes exist, they perform similarly and hence are not suitable for our use case.

with rich emotion labels is quite difficult. Commonly used "3D Morphable Models" like FLAME [Li+17] are adept at parameterizing shape and pose, but struggle with micro-level details such as brow wrinkles or partial occlusions. Meanwhile, purely neural 3D representations—like Dynamic Neural Radiance Fields (NeRF) [Gaf+21] and NeRSemble [Kir+23b]—achieve photorealistic quality but often embed geometry and appearance in a black-box manner, making direct expression editing non-trivial unless guided by an additional parametric model or specialized deformation field.

Recent work has begun addressing these gaps. For instance, the Emo3D pipeline [Deh+24]

proposes generating a 52D blendshape vector from textual emotion descriptions, mapping them onto a 3D face geometry. While this approach indicates growing interest in text-driven 3D facial editing, it still relies heavily on blendshape manipulation, offering limited control over texture and lighting. Other pipelines (e.g., GaussianSpeech [Ane+24] or GaussianAvatars [Qia+24a]) show how a 3D Gaussian-splat representation can handle free-viewpoint rendering with high fidelity, but primarily focus on re-enacting existing expressions rather than inventing new emotional states. Indeed, controlling or "editing" those splats to reflect an unseen emotion (e.g., "make the avatar look disappointed" if the person never displayed it during capture) typically requires strong priors or additional 2D references.

Consequently, while 3D emotion editing offers a far richer canvas—allowing the same avatar to appear "sad" from multiple viewpoints under dynamic lighting—practical solutions are stymied by data availability (multi-view captures of emotional expressions), optimization complexity (merging geometry and texture seamlessly), and lack of unsupervised pipelines (few methods exist that do not rely on large, labeled 3D corpora). As illustrated in Figure 1.6, current solutions highlight this gap: geometry-based approaches (left) enable free-viewpoint rendering but lack photorealistic textures, while diffusion-based 2D edits (middle) offer fine detail but fail to provide spatial or temporal consistency across views. Bridging these limitations—achieving photorealistic, geometry-aware emotion editing—remains a key challenge.

Figure 1.6: Bridging 2D and 3D emotion editing: The left represents a geometry-based 3D approach(EMOTE [Dan+23]) for editing emotions, which lacks photo-realism and fine-grain textures. The middle shows a diffusion-based 2D editing [Zha+24], while the right demonstrates the conceptual gap—a bridge toward photorealistic 3D editing with textures rigged to geometry, EMO-GA.

1.3 Conclusion and Thesis Overview

In this thesis, we propose a *photometric-loss-driven* strategy for 3D emotion editing in Gaussian Avatars, avoiding large, labeled 3D emotion datasets. Our pipeline leverages *2D diffusion-based prompts* as guidance signals to modify avatar appearance. By backpropagating 2D photometric errors through a differentiable rasterizer, we jointly adapt FLAME expression parameters and the Gaussian splats' color, preserving both identity and multi-view consistency.

Key challenges include:

1. **Preserving Identity and Details:** Only the FLAME expression parameters and splat color are optimized, ensuring minimal geometric distortion to eyes, lips, and other critical features.
2. **Maintaining Geometry–Texture Coherence:** FLAME controls shape while Gaussian splatting handles texture, preventing view-dependent artifacts.
3. **Capturing Subtle Emotional Nuances:** Localized edits to color and micro-geometry enable fine wrinkles or skin-tone shifts without relying on purely geometric methods or massive labeled data.

Overall, this approach *bridges* 2D editing techniques with controllable 3D geometry, making high-fidelity facial emotion manipulation feasible even with purely self-supervised signals.

Thesis Outline: The remainder of this document is organized as follows:

- **Chapter 2: Dataset** introduces the multi-view capture protocol and data used to build high-fidelity head avatars.
- **Chapter 3: Related Work** surveys existing methods in 3D reconstruction, neural rendering, and emotion editing, framing the context for our proposed approach.
- **Chapter 4: Method** details the proposed pipeline for emotion editing based on Gaussian Avatars, covering the photometric loss design, essentially a diffusion-based optimization, and the specific regularization components that guide emotion manipulation.
- **Chapter 5: Experiments and Results** demonstrates the performance of our method on various subjects, comparing it with alternative approaches.
- **Chapter 6: Conclusion** and **Chapter 7: Future Work** summarize key findings and suggest directions for extending or refining this framework.

Chapter 2

Dataset

2.1 NeRSemble

NeRSemble [Kir+23b] introduced a pivotal Neural Radiance Field (NeRF [Mil+21])-based approach for reconstructing 3D human heads into photorealistic 3D head avatars. This innovative use of multi-view radiance fields enabled highly detailed novel view synthesis, often producing results so realistic that they could be mistaken for actual images. This achievement was made possible through the combination of a deformation field and a multi-resolution ensemble of hash encodings, as well as their high-quality, controlled capture setup.

2.1.1 NeRSemble Dataset

This high-quality, controlled data capture was facilitated by the advanced yet spaese capture rig as shown in Figure 2.1. The subject is positioned in front of 16 machine vision cameras, arranged in a forward-facing configuration and synchronized with sub-microsecond accuracy. This setup provides a 93-degree horizontal and 32-degree vertical field of view. The high resolution of the system is achieved using 7.1-megapixel sensors operating at 73 frames per second, enabling the rig to faithfully record subtle facial movements and intricate details. While the rig lacks cameras positioned behind the subject, this limitation does not impact the Emotion-Driven Editing of Gaussian Avatars [Qia+24a] in our work but influences the back of the generated Gaussian Avatar.

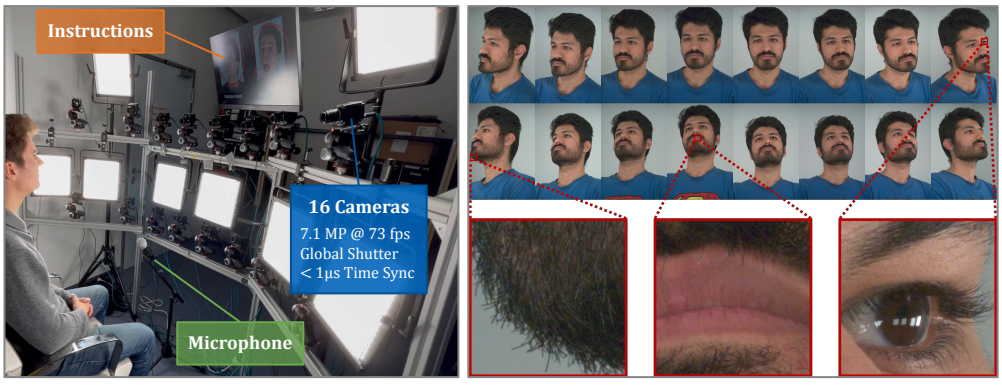


Figure 2.1: NeRSemble's custom-made multi-view video capture setup, capturing 16 distinct yet sparse viewpoints, showcasing the remarkable level of detail achieved from the capture.

This capture rig was then used to generate 4,734 video sequences featuring 222 unique subjects, a number that has since grown significantly at the time of writing. Initially, the

dataset totaled approximately 31.7 million frames. This extensive data collection surpasses previous datasets in both scale and quantity, as demonstrated in Table 2.1.

Dataset	#Subj.	#Cam.	Resolution	Fps
D3DFACS [CKH11]	10	6	1280 x 1024	60
BP4D-Spontaneous [Zha+14]	41	3	1392 x 1040	25
Interdigital Light-Field [Sab+17]	5	16	2048 x 1088	30
4DFAB [Che+18]	180	7	1600 x 1200	60
VOCASET [Cud+19b]	12	12	1600 x 1200	60
MEAD [Wan+20]	48	7	1920 x 1080	30
MultiFace [Wuu+23]	13	150	2048 x 1334	30
NeRSemble	222	16	3208 x 2200	73

Table 2.1: Comparison with other multi-view human face datasets with subject count, camera count, resolution, and frame rate.

2.1.2 Motion and Expression Coverage

The dataset comprises a comprehensive collection of sequences categorized into Expressions, Emotions, Hair, Free, and Sentences, ensuring a wide range of facial dynamics and movements for robust evaluations. Specifically, the dataset includes 9 Expression sequences, 4 Emotion sequences, 1 Hair sequence, 1 Free sequence, and 10 Sentence sequences, as shown in Table 2.2.

Table 2.2: Overview of sequences in NeRSemble data set categorized by type.

Category	Sequence Name
Expression	EXP-1-head
	EXP-2-eyes
	EXP-3-cheeks+nose
	EXP-4-lips
	EXP-5-mouth
	EXP-6-tongue-1
	EXP-7-tongue-2
	EXP-8-jaw-1
	EXP-9-jaw-2
Sentence	SEN-01-cramp_small_danger
	SEN-02-same_phrase_thirty_times
	SEN-03-pluck_bright_rose
	SEN-04-two_plus_seven

Category	Sequence Name
Emotion	EMO-1-shout+laugh
	EMO-2-surprise+fear
	EMO-3-angry+sad
	EMO-4-disgust+happy
Hair	HAIR
Free	FREE
Sentence	SEN-05-glow_eyes_sweet_girl
	SEN-06-problems_wise_chief
	SEN-07-fond_note_friend
	SEN-08-clothes_and_lodging
	SEN-09-frown_events_bad

The Expression and Emotion sequences demonstrate significantly greater head and facial movement compared to the relatively neutral and controlled Sentence sequences. For our work, we specifically utilize the sequences SEN-01, SEN-04, and SEN-06, depending on the subject. This selection ensures predominantly neutral facial inputs for the emotion driven editing of Gaussian Avatars, enabling a fair and consistent comparison across different methods.

2.1.3 Diversity

Each sequence listed in Table 2.2 has been spoken by every participant, offering a diverse pool of subjects to choose from. The dataset composition reflects significant diversity, including approximately 65.5% White, 17% Asian, 8.5% Middle Eastern, and 6.3% Indian participants, to name just a few. Figure 2.2 illustrates a collection of diverse subjects performing the same sequence, further highlighting the diversity of the dataset.

Figure 2.2: Diversity of participants in the NeRSemble dataset, showcasing subjects from various ethnic backgrounds performing the same sequence.

This enables the capture of a wide range of skin textures, hair types, emotion intensities, individuals with glasses, and various beard styles. These attributes make NeRSemble an excellent candidate dataset for testing novel view synthesis and novel expression synthesis on human faces, particularly in scenarios where fine details such as wrinkles and individual hair strands are critical.

2.1.4 Significance for This Work

The NeRSemble dataset, with its high-resolution, diverse multi-view human head data captured in a controlled environment, was a natural choice for this work. Its rich quality and diversity made it ideal for emotion-driven editing of Gaussian Avatars. This dataset allows us to evaluate our edits against high-quality ground truth data and benchmark them against a robust 3D head avatar baseline. Consequently, all subjects used in the following experiments are selected from the NeRSemble dataset.

Chapter 3

Related Works

3.1 3D Gaussian Splatting

Neural Radiance Fields (NeRFs) [Mil+21] introduced in 2020 reshaped novel view synthesis by enabling photorealistic rendering of 3D scenes from sparse multi-view images. They set a new standard in the field, leveraging volumetric ray-marching and position- and view-dependent Multi-Layer Perceptron (MLP) optimization to achieve remarkable realism. However, the computational demands, including long training times and slow rendering speeds, limited their use in real-time applications. To address these challenges, methods like 3D Gaussian Splatting [Ker+23] have emerged. Unlike NeRFs, this approach replaces neural networks with compact, point-based scene representations using anisotropic 3D Gaussians. With adaptive density control and efficient tile-based rasterization, 3D Gaussian Splatting [Ker+23] allowed real-time rendering while at the same time maintaining high visual fidelity, offering a much more practical and efficient alternative for novel view synthesis.

3.1.1 3D Gaussian Splatting Introduction

The 3D Gaussian Splatting pipeline is structured around three key stages as shown in Figure 3.1.1, each of which contributes to its capability for real-time, high-quality novel view synthesis.

The first stage of 3D Gaussian splatting involves initialization: starting with sparse Structure-from-Motion (SfM) points, the pipeline initializes a set of 3D Gaussians. These 3D Gaussians are characterized with properties such as position, anisotropic covariance, and opacity. Additionally, spherical harmonics (SH) are used to model color, allowing the efficient representation of view-dependent effects. The initialization step establishes the foundation for representing the scene with a compact and differentiable volumetric structure.

While SfM points are a natural choice for initializing the scene representation, directly using them can lead to suboptimal results. This is because the sparsity of SfM points and the challenges of reliably estimating normals can hinder optimization. To address this, the pipeline employs a full 3D covariance matrix, Σ, defined in world space and centered at a point's mean position

$$G(\boldsymbol{x}) = e^{-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x}} \tag{3.1}$$

where $\boldsymbol{x}$ is a 3D point, $\boldsymbol{\mu}$ is the mean position, and $\boldsymbol{\Sigma}$ is the covariance matrix. Moreover, this Gaussian is multiplied by α in the blending process.

To render these 3D Gaussians, they must be projected to 2D. This projection is guided by an affine transformation. Following [Zwi+01], the covariance matrix $\boldsymbol{\Sigma}'$ in camera coordinates

is given by:

$$\Sigma' = JW\Sigma W^{\mathrm{T}} J^{\mathrm{T}} \tag{3.2}$$

where J is the Jacobian of the affine approximation of the projective transformation, and W represents the viewing transformation.

Furthermore, for optimization, the covariance matrix Σ is defined using a scaling matrix S and a rotation matrix R, as follows:

$$\Sigma = RSS^{\mathrm{T}} R^{\mathrm{T}} \tag{3.3}$$

This formulation allows independent optimization of scaling and rotation, where the scaling matrix $\mathbf{S}$ describes the size of the ellipsoid and the rotation matrix $\mathbf{R}$ describes its orientation. Using gradient descent, the parameters are optimized to ensure valid covariance matrices throughout the optimization process. This optimization results in a compact and differentiable volumetric representation of the scene, establishing the foundation for further processing.

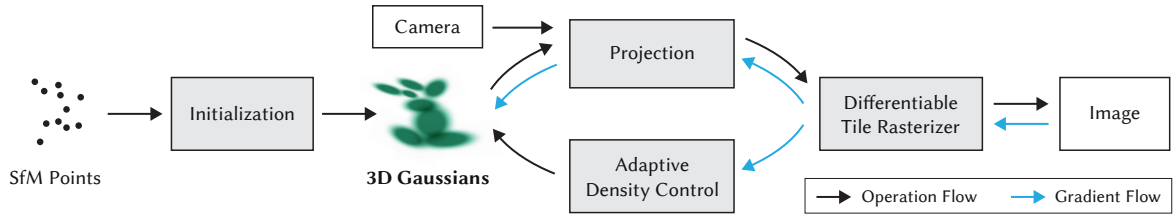


Figure 3.1: Pipeline illustrating the Gaussian splatting process, starting from SfM points initialization to the creation of 3D Gaussian representations. The workflow includes adaptive density control, projection, and rendering using a differentiable tile rasterizer. Operation flow is depicted with black arrows, while gradient flow is shown with blue arrows. [Ker+23]

Building upon this foundation, the second stage concentrates on Projection as well as Adaptive Density Control. During this phase, 3D Gaussians are projected into 2D splats, enabling efficient rendering. The adaptive density control mechanism optimizes the representation by dynamically refining the distribution of Gaussians, splitting larger ones or cloning others to better capture scene geometry and fine details (as illustrated in Figure 3.2).

Figure 3.2: The 3D Gaussians are visualized after optimization by shrinking them by 60% (far right). This visualization highlights the anisotropic shapes of the 3D Gaussians that compactly represent complex geometry post-optimization. The left image shows the actual rendered image, while the center and right images display the original and shrunken Gaussians, respectively [Ker+23].

The third and most critical component is the Differentiable Tile Rasterizer. This Diff Tile Rasterizer controls the rendering process, taking into account visibility ordering and ensuring

efficient backpropagation during optimization. The tile-based design accelerates computations while maintaining visual fidelity, making real-time rendering feasible even for complex scenes with millions of Gaussians to render.

Together, these three stages form the backbone of the 3D Gaussian Splatting pipeline, offering a framework that seamlessly combines compactness, flexibility, and computational efficiency. The following subsections look deeper into the second and third stages of the pipeline.

3.1.2 Projection and Adaptive Density Control

The second stage of the 3D-GS pipeline concentrates on projection and adaptive density control. During this phase, the 3D Gaussians are projected into 2D splats, allowing efficient rendering and accurate scene representation. This stage also incorporates an adaptive density control mechanism that refines the distribution of Gaussians dynamically. Under-reconstructed regions are identified, and new Gaussians are cloned to cover insufficient geometry. Conversely, over-reconstructed regions, characterized by excessively large splats, are optimized by splitting Gaussians into smaller ones to improve scene detail (illustrated in Figure 3.1.2). These processes ensure that the representation remains compact and precise while maintaining high fidelity.

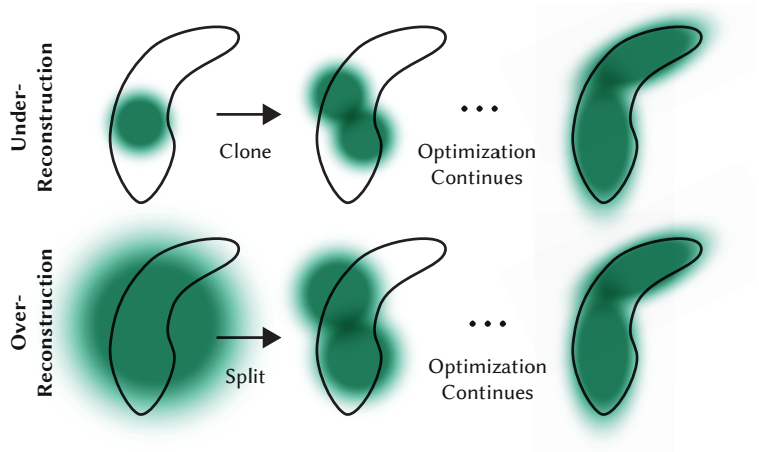


Figure 3.3: Adaptive Gaussian densification strategy. *Top row (under-reconstruction):* Small-scale geometry is insufficiently covered, prompting Gaussian cloning to fill gaps. *Bottom row (over-reconstruction):* Large splats representing small-scale geometry are split into smaller Gaussians, ensuring more precise rendering. Both cases demonstrate the optimization process continuing after densification [Ker+23]

3.1.3 Differentiable Tile Rasterizer

The Differentiable Tile Rasterizer represents the third and final stage of the 3D-GS pipeline. This component is responsible for rendering the 3D Gaussians by projecting them into 2D space, ensuring efficient computation and gradient backpropagation. To achieve this, the rasterizer employs a tile-based sorting strategy, splitting the entire scene into 16x16 tiles. This approach efficiently manages computational overhead and memory usage by processing only the Gaussians relevant to each tile. Parallelization of tile operations further minimizes redundant computations, ensuring scalability for large-scale scenes.

One of the key advantages of this tile rasterizer is its ability to leverage 2D image space loss formulations during optimization. This was made straightforward by projecting 3D Gaussians

into 2D space, the pipeline directly compares rendered outputs with ground truth given images. Hence, pixel-level losses, such as photometric loss or Structural Similarity Index (SSIM), can be used to steer the optimization process, such as:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{\text{photometric}} + \lambda\mathcal{L}_{\text{SSIM}}, \tag{3.4}$$

where λ controls the weighting. This formulation ensures the optimization refines the Gaussian parameters while aligning the rendered results with ground truth 2D observations.

During the backward pass, the accumulated opacity values of the blended splats are reused to compute gradients, bypassing the need to explicitly store intermediate blending states. This significantly reduces memory requirements while preserving gradient accuracy, enabling real-time rendering and optimization even for scenes with millions of Gaussians. The combination of efficient rasterization and the utilization of 2D losses makes this stage critical for the success of the 3D Gaussian Splatting pipeline.

3.2 GaussianAvatars

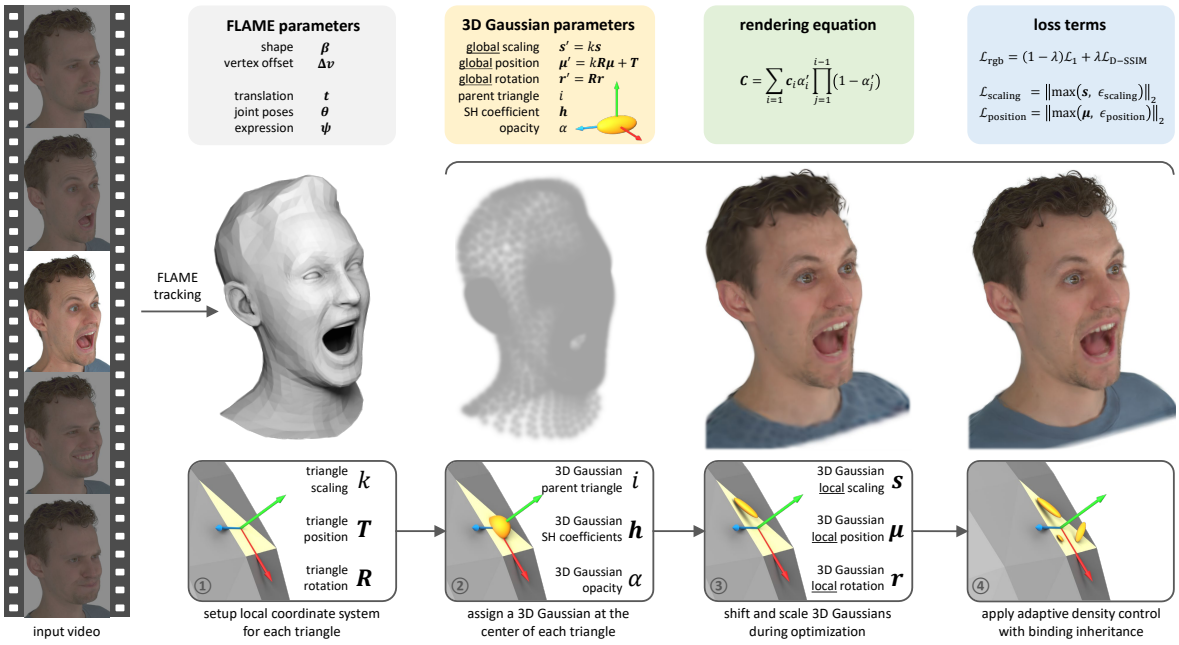


Figure 3.4: Pipeline overview of *GaussianAvatars*. Each 3D Gaussian is assigned to a FLAME triangle in a local coordinate system. During optimization, authors transform these splats to global space, minimize photometric color loss, and update their positions and scales via adaptive density control, ensuring they remain rigged throughout the animation.

Building on the success of 3D Gaussian Splatting [Qia+24a], which demonstrated the effectiveness of representing complex scenes with compact 3D Gaussians for novel-view synthesis, the methodology has naturally matured to address more dynamic and controllable applications. While Gaussian Splatting achieves high-fidelity rendering with real-time performance, it is inherently limited to static or time-consistent dynamic scenes. However, many real-world applications—such as virtual avatars in gaming, immersive telepresence, animation and other virtual reality experiences—demand precise control over pose, expression, and dynamic movements.

To bridge this gap, *GaussianAvatars* [Qia+24a] as shown in Figure 3.4 introduces a significant advancement by extending the principles of Gaussian Splatting into the domain

of photorealistic, controllable head avatars. This approach not only retains the compact and expressive representation of 3D Gaussians but also rigs them to a parametric morphable face model, such as FLAME [Li+17]. This rigging allows seamless manipulation of expressions, poses, and viewpoints, combining geometric accuracy with animation controllability.

One of the key innovations lies in rigging 3D Gaussians to parametric mesh models. By binding Gaussian splats to the triangles of a morphable mesh, GaussianAvatars achieves precise deformation of the radiance field during animation. This enables the representation to adjust to facial expressions and head poses while maintaining high visual fidelity.

The introduction of the binding inheritance strategy addresses a critical limitation of Gaussian Splatting's density control, which, while effective for static scenes, lacked the ability to maintain rigging relationships required for dynamic and animatable elements. Hence, by ensuring that newly added or removed Gaussians inherit their parent triangle's rigging relationship, this innovation allows GaussianAvatars to achieve adaptive density control while preserving the precise controllability necessary for realistic animations.

The consequence is a robust pipeline(illustrated in Figure 3.4 that excels in photorealistic rendering and novel-expression synthesis and provides a practical pathway to create animatable, high-quality head avatars with fine control over dynamic features. The following sections look in-depth into transforming static 3D Gaussian Splatting into dynamic GaussianAvatars, highlighting the technical modifications and novel contributions that enable this.

3.2.1 Pipeline Overview

The method as shown in Figure 3.2.1 takes as input a multi-view video recording of a human head, sourced from the NeRSemble Dataset (Section 2.1). This video provides the raw data necessary for reconstructing a high-fidelity 3D representation. One of the first steps involves extracting FLAME parameters (Chapter 1.1.2), such as shape (β), pose (θ and $\mathbf{t}$), expression (ψ), and vertex offsets ($\Delta\mathbf{v}$), using a photometric head tracker (Chapter 4.3). These FLAME parameters are optimized to fit the observed multi-view video frames, producing a consistent mesh topology with per-frame variations.

Once the FLAME mesh is created, the method initializes 3D Gaussians. Each triangle in the FLAME mesh is paired with a 3D Gaussian, positioned at the center of the triangle. The local coordinate system for each triangle is defined using its position ($\mathbf{T}$), orientation ($\mathbf{R}$), and scale (k), ensuring that the Gaussians are dynamically aligned with the mesh's deformations during animation. This pairing ensures a consistent connection between the Gaussian splats and the underlying geometry, establishing what is often referred to as "rigging."

During optimization, the 3D Gaussians are transformed from the local coordinate space of their parent triangles into a global space. This transformation allows the 3D Gaussians to adapt seamlessly to changes in expression and pose. Unlike static Gaussian Splatting, which focuses on scene representation, this dynamic application requires a mechanism to define the relationship between newly added Gaussians and the existing structure. To achieve this, the method introduces a binding inheritance mechanism, where newly spawned Gaussians inherit the relationships of their parent triangles. This ensures the structural and visual integrity of the avatar, avoiding any floating artifacts or inconsistencies during animation.

Finally, the optimization process utilizes a differentiable tile rasterizer (Subsection 3.1.3) to render the 3D Gaussians into 2D space. The rendering process is supervised using a combination of photometric and structural similarity (SSIM) losses, ensuring that the generated avatar closely matches the input sequence. Moreover, regularization terms are applied to enforce consistency in the position and scaling of the 3D Gaussians, mitigating artifacts during animation. This pipeline culminates in photorealistic, animatable head avatars that faithfully reproduce complex expressions and subtle movements with remarkable visual fidelity.

3.2.2 Inherited Formulations from Gaussian Splatting

GaussianAvatars builds upon the foundational mathematical formulations introduced in 3D Gaussian Splatting. The representation of 3D Gaussians as anisotropic ellipsoids, defined by their mean position $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ (refer to Equation 3.1), remains central to this method. Similarly, the construction of $\boldsymbol{\Sigma}$ using the scaling and rotation matrices, $\boldsymbol{S}$ and $\boldsymbol{R}$ respectively (refer to Equation 3.3), is retained to capture the size and orientation of each Gaussian.

Color Modeling and Blending: The modeling of color, shared with Gaussian Splatting, is inspired by earlier point-based rendering techniques that employ blending equations to combine contributions from multiple primitives. In this work, the rendered pixel color C is determined by blending the contributions from all 3D Gaussians overlapping the pixel, calculated as:

$$C = \sum_{i=1}^{n} c_i \alpha_i' \prod_{j=1}^{i-1} (1 - \alpha_j'), \tag{3.5}$$

where c_i represents the color of the i-th Gaussian, modeled using spherical harmonics, and α_i' is the blending weight, incorporating the Gaussian's opacity α while respecting visibility order by sorting Gaussians by depth.

Gaussian Splatting incorporated and extended this blending equation, integrating spherical harmonics to achieve high-fidelity, view-dependent effects. This formulation is retained here, enabling photorealistic rendering that aligns seamlessly with both the underlying geometry and the dynamic appearance of the scene.

3.2.3 Rigging and Binding Inheritance

Rigging, being one of the crucial elements in GaussianAvatars, establishes a consistent connection between the FLAME mesh triangles and their corresponding 3D Gaussian splats. Each 3D Gaussian is initialized in the local coordinate space of its parent triangle. This setup ensures that the splat's local position ($\boldsymbol{\mu}$), rotation ($\boldsymbol{r}$), and scaling (s) are dynamically updated in response to deformations in the FLAME mesh during animation. At rendering time, the local Gaussian properties are transformed into the global coordinate space using:

$$\boldsymbol{r}' = \boldsymbol{R}\boldsymbol{r}, \tag{3.6}$$

$$\boldsymbol{\mu}' = s\boldsymbol{R}\boldsymbol{\mu} + \boldsymbol{T}, \tag{3.7}$$

$$s' = ks. \tag{3.8}$$

Here, $\boldsymbol{R}$ (triangle rotation), $\boldsymbol{T}$ (triangle position), and k (triangle scaling) represent the parent triangle's rotation, position, and scaling, respectively. These transformations ensure seamless alignment of the splats with the animated mesh.

Binding inheritance further enhances the dynamic capabilities of the pipeline. When new Gaussians are added or existing ones are removed during the adaptive density control process, the binding inheritance mechanism ensures that the new splats inherit the rigging relationship of their parent triangles, preventing floating artifacts and maintaining the structural integrity of the animated avatar and animations.

3.2.4 Optimization

The optimization process in GaussianAvatars builds upon the principles established in Gaussian Splatting, while introducing additional loss terms to address the challenges of dynamic avatar creation. The process begins with the initialization of 3D Gaussians, one for each triangle of the FLAME mesh, based on the first frame of the video sequence, ensuring a consistent starting point for optimization. Subsequently, the system incrementally adds new Gaussians to under-reconstructed regions while refining existing splats over the remaining frames. This adaptive strategy enables high-fidelity modeling without overloading the initial representation.

The primary loss function used for optimization combines a photometric term ($\mathcal{L}_{\text{rgb}}$), which itself is a combination of an $\mathcal{L}_1$ loss and a Structural Similarity Index (SSIM) loss, and two regularization terms: position loss ($\mathcal{L}_{\text{position}}$) and scaling loss ($\mathcal{L}_{\text{scaling}}$). Together, these terms ensure that the Gaussians remain well-aligned with their parent triangles while avoiding artifacts such as floating or displacement of splats, as well as excessive shrinking. The complete loss function is expressed as:

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{position}}\mathcal{L}_{\text{position}} + \lambda_{\text{scaling}}\mathcal{L}_{\text{scaling}}, \tag{3.9}$$

where the regularization terms are weighted by $\lambda_{\text{position}}$ and λ_{scaling}, respectively.

Position Loss: The position loss enforces a close alignment between the local Gaussian position and its parent triangle, ensuring that splats remain in plausible locations relative to the animated mesh. This is formalized as:

$$\mathcal{L}_{\text{position}} = \| \max(\boldsymbol{\mu}, \epsilon_{\text{position}}) \|_2, \tag{3.10}$$

where $\epsilon_{\text{position}}$ is a tolerance threshold that allows small deviations without penalty.

Scaling Loss: The scaling loss regularizes the scale of each Gaussian relative to its parent triangle. This prevents issues such as oversized splats that might jitter during animation or undersized splats that degrade rendering quality. It is defined as:

$$\mathcal{L}_{\text{scaling}} = \| \max(s, \epsilon_{\text{scaling}}) \|_2, \tag{3.11}$$

where $\epsilon_{\text{scaling}}$ sets a lower limit to disable penalties for very small scales.

To further enhance fidelity, adaptive density control dynamically adjusts the distribution of Gaussians.

By balancing reconstruction quality with structural integrity through these loss formulations, and leveraging an incremental optimization strategy across the video sequence, GaussianAvatars achieves photorealistic rendering with a dynamic, high-fidelity avatar capable of realistic poses and expressions.

3.2.5 Reconstruction and Animation

Gaussian Avatar reconstructions with Ground Truth images in both self-reenactment and novel-view synthesis tasks, showcased in Figure 3.8, highlights the fidelity of Gaussian Avatars in capturing intricate details such as hair strands, teeth, and facial expressions. The results emphasize the method's ability to closely approximate the Ground Truth across various subjects.

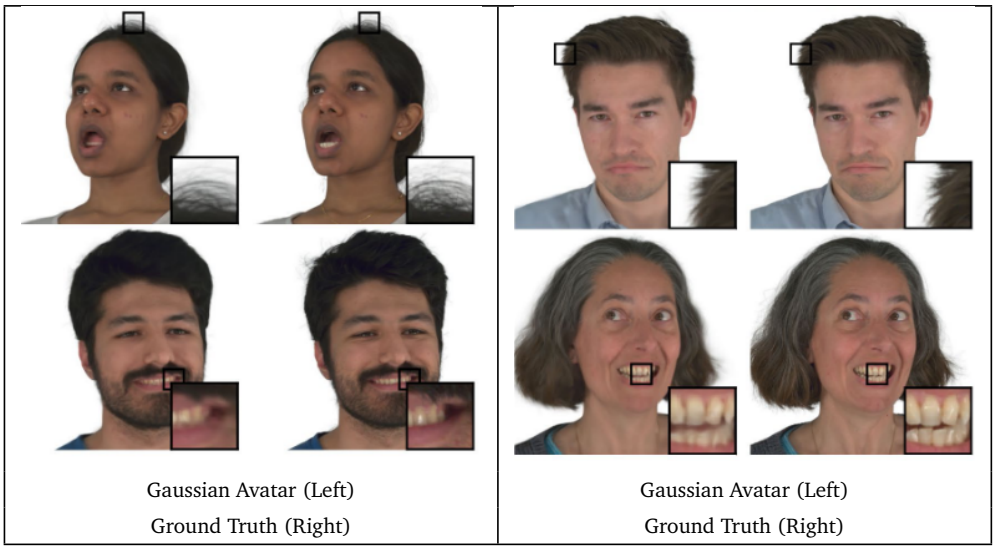


Figure 3.8: Qualitative comparison of Gaussian Avatar renderings and Ground Truth images, showcasing self-reenactment (left stack) and novel-view synthesis (right stack). Each stack emphasizes detailed regions such as hair strands, teeth, and facial expressions [Qia+24a].

3.3 2D Frameworks for Emotion Editing

3.3.1 Audio-Driven Emotional Video Portraits

Although the primary focus of this thesis lies in altering emotional expressions within a 3D framework, a number of methods have explored emotion manipulation in 2D video portraits. One standout example is the work of Ji [Ji+21], often cited as *Audio-Driven Emotional Video Portraits* (EVP). This approach targets two main challenges in 2D facial animation: capturing emotional nuances alongside lip synchronization, and adapting those emotions to a target video without losing the subject's identity or natural head motions.

Core Idea and Motivation

Early 2D talking-head methods often emphasize speech synchronization and mouth movements [Son+22; Che+19], leaving emotional expressions as an afterthought. By contrast, EVP aims to introduce explicit emotion control, Figure 3.9 provides an overview of the EVP pipeline, illustrating how the system extracts emotional and linguistic features from audio, manipulates facial landmarks, and synthesizes realistic talking-head animations. The authors posit that an emotionally expressive portrait must move beyond merely matching lip shapes to speech; it should convey affective states that reflect the nuances of the spoken audio.

Cross-Reconstructed Emotion Disentanglement

To separate emotion from linguistic content, EVP employs a cross-reconstruction pipeline. First, the system gathers audio clips that contain the same uttered phrases but differ in emotional tone. By applying Dynamic Time Warping (DTW) [SC78] to align Mel-Frequency Cepstral Coefficients (MFCC), EVP obtains pairs of audio signals sharing identical content but varying emotional delivery. Two encoder networks learn:

1. A *duration-independent* embedding focusing on emotional style.
2. A *duration-dependent* embedding capturing the phonetic content.

Through this design, the network can recompose emotional states independently from what is actually being said.

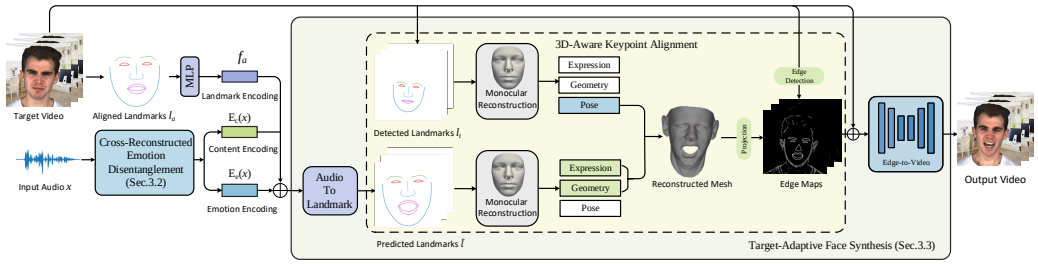


Figure 3.9: Schematic depiction of Ji [Ji+21] (EVP). Their system extracts emotional and linguistic information from audio, projects manipulated facial landmarks onto a target subject through 3D-aware modeling, then synthesizes the final video using image-to-image translation.

3D-Aware Landmark Alignment

Once the system extracts emotional style, it predicts 2D landmarks representing the desired facial expression. Aligning these landmarks directly on a target video, however, could falter when head rotation or pose differs significantly from the source. To counteract this, EVP employs a 3D morphable face model [BV99] to factor out pose and re-project the manipulated landmarks into the target's coordinate frame. This *3D-aware* step preserves realistic head orientation and mitigates spatial distortion in the final video.

Target-Adaptive Face Synthesis

After landmark alignment, EVP merges the resultant edge map with emotional cues to drive a face-synthesis network based on image-to-image translation [Wan+18]. This step produces a photorealistic video sequence where the subject's identity, head pose, and lip movements remain coherent, while the emotional expression reflects the newly imposed style.

Key Strengths and Potential Shortcomings

The flexibility to interpolate smoothly between emotions (e.g., transitions from *sad* to *happy*) is a marked strength of EVP. Unlike many prior works that rely on discrete emotion categories, this method allows a more nuanced control, as illustrated in Figure 3.10, where smooth transitions between emotions are demonstrated. However, the 2D nature of the pipeline relies heavily on data coverage: if the target poses or emotional states are insufficiently represented in the training set, the synthesized frames may exhibit facial distortions. Moreover, although the system can convincingly edit expressions within the existing camera view, it does not inherently provide free-viewpoint or full 3D consistency.

In contrast, the emotion-editing framework proposed in this thesis does not rely on a large emotion-labeled dataset, nor does it require methods to handle significant head-pose deviations in 2D space. By leveraging 3D representations, the approach sidesteps many limitations of purely 2D pipelines, inherently separating pose from emotional expression and enabling more robust animation under varied viewpoints.

Implications for This Thesis

EVP demonstrates that realistic facial animation often benefits from separating the emotional cues from the strictly linguistic elements of speech. However, in this thesis, emotion editing is

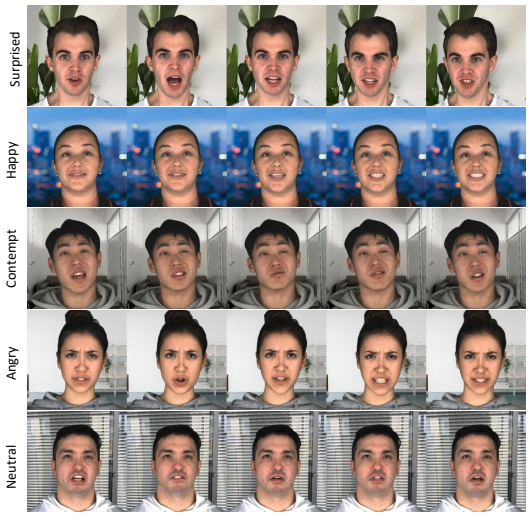


Figure 3.10: Visualization of the results from Ji [Ji+21] demonstrating their EVP pipeline. The figure showcases generated video portraits conditioned on the same speech content but rendered with different emotions: *Surprised, Happy, Contempt, Angry, and Neutral*. These results highlight the ability of EVP to produce expressive talking-head animations with fine-grained emotional control.

not driven by audio-based disentanglement. Instead, a diffusion model serves as the primary mechanism for adjusting emotional expressions in a 3D setting. By working with neutral or minimally expressive speech tracks, the emphasis shifts away from detailed content-emotion separation and focuses on photometric guidance signals to modify expressions directly in 3D. This design inherently bypasses many of the constraints found in 2D pipelines, such as handling complex head poses, and opens the door to free-viewpoint rendering where emotional states can be flexibly and accurately controlled without relying on extensive emotion-labeled datasets.

3.3.2 UltraEdit: Instruction-Based Fine-Grained Image Editing at Scale

Text-guided diffusion methods have led to a new class of image-editing frameworks that can alter or synthesize visual attributes using high-level textual prompts. *UltraEdit*[Zha+24] is one such approach, which stands out for its scale (over 4 million instruction-based editing samples) and its ability to process and refine user instructions with larger language models. Built on top of a *Stable Diffusion XL* backbone, UltraEdit natively supports two key editing paradigms:

- **Free-form Editing:** Users provide a text instruction (e.g., *"Turn the cat into a robot"*), and UltraEdit attempts to modify *any* relevant parts of the image based on the description.
- **Region-Based Editing:** Users optionally provide a mask indicating which part(s) of the image to edit (e.g., *"Change the face to a lion's face, but preserve the rest"*), enabling more precise, localized changes while preserving other regions.

As shown conceptually in Figure 3.11, UltraEdit begins with a source image and an instruction (plus an optional region mask). It then leverages a specialized `U-Net` that denoises in a guided manner, focusing on the portion of the latent space specified by the prompt. For region-based edits, UltraEdit carefully inpaints only the masked region, leaving unmasked areas untouched. The authors compile these large-scale editing samples by pairing a variety

of human- and LLM-generated textual instructions with real or generated images, thereby covering a wide spectrum of possible changes (e.g., color shifts, object additions, style modifications, and global or local transformations).

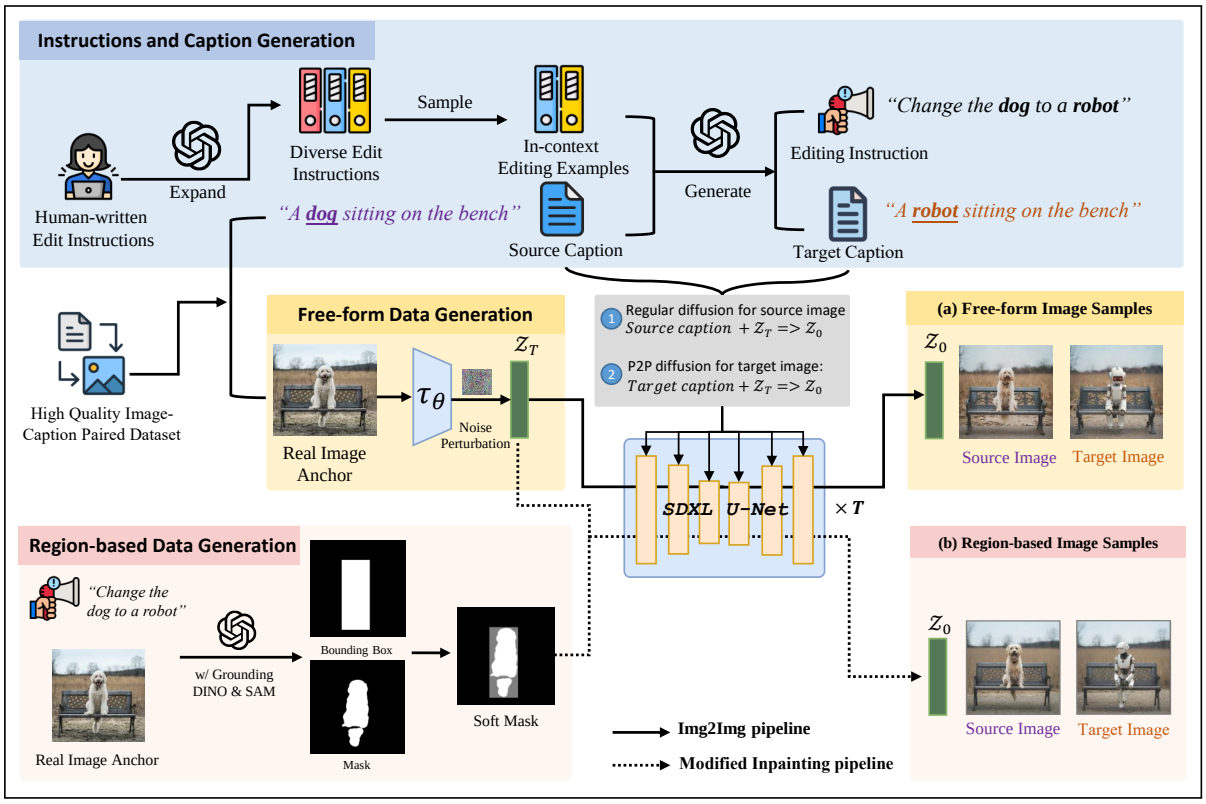


Figure 3.11: Schematic overview of UltraEdit's editing pipeline [Zha+24]. The user provides a *source image*, an *editing instruction*, and optionally a *mask*. UltraEdit then applies a modified Stable Diffusion XL-based inpainting or prompt-to-prompt method to produce the *target image*.

Additionally, UltraEdit demonstrates visually convincing edits across a wide range of source images and instructions. Figure 3.12 highlights some of these edits, including both free-form changes (e.g., "change her expression to one of joy and excitement") and region-based transformations (e.g., "turn the cat into a robot" while preserving the surrounding book page). This extent of instruction coverage, alongside robust masking and inpainting, illustrates UltraEdit's versatility for generating localized or global edits as needed.

Key Technical Details of the UltraEdit Pipeline

While UltraEdit's usage can be summarized as "provide an instruction and mask, then edit the image accordingly," the underlying pipeline is more extensive, designed to handle the challenges of large-scale, instruction-based editing. Zhao [Zha+24] outline three main stages (see Figure 3.11 for a conceptual depiction) that jointly produce massive, high-quality edited samples, supporting both free-form and region-based scenarios:

1. **Instruction and Caption Generation:** The pipeline begins by collecting real or synthetic images with associated captions. Next, a large language model (LLM) expands these captions, generating diverse editing instructions. For example, given a caption *"a dog sitting on a bench,"* the LLM might produce *"turn the dog into a robot"* or *"turn the bench into a rainbow bench."* These expanded instructions form the raw textual variety that UltraEdit can eventually handle.

Figure 3.12: Examples of free-form (*left*) and region-based (*right*) image editing produced by UltraEdit [Zha+24]. Prompts range from "Replace the tie with a superhero cape" to "Change her expression to one of joy and excitement," illustrating the system's ability to handle diverse transformations. Notably, the unmasked regions remain largely consistent, preserving identity or background context.

2. **Free-Form Editing Data:** UltraEdit then synthesizes new images based on the source images, instruction texts, and a diffusion-based model derived from *Stable Diffusion XL*. A specialized denoising pipeline incorporates *prompt-to-prompt* (P2P) control, guiding generation via partial embeddings (e.g., conditioning on certain latent features). This stage produces "before" and "after" pairs for training: the *source image* plus *instruction* lead to the *edited (target) image*, encompassing diverse modifications (e.g., stylistic changes, object insertions).

3. **Region-Based Editing Data:** Beyond free-form edits, UltraEdit supports precisely localized changes. By applying object detection (e.g., *GroundingDINO* [Liu+23]) and segmentation (e.g., *SAM* [Kir+23a]) on the source images, UltraEdit automatically identifies relevant objects or regions for editing. A modified inpainting diffusion routine then updates only the masked parts, preserving the remaining pixels. For instance, with the prompt *"change the cat's face into a lion's face,"* bounding-box detection finds the cat's face, and inpainting modifies only that region to become lion-like. This approach is critical for high spatial fidelity and minimal identity changes.

These three stages enable UltraEdit to scale to over four million *instruction–image* pairs. By exposing the model to a vast and varied training set of real and synthetic data, UltraEdit learns to robustly interpret textual prompts and generate consistent image modifications, including challenging cases such as subtle facial expression changes or partial scene transformations.

Implications for This Thesis

In our pipeline (Chapter 4), we rely on UltraEdit to generate *pseudo-ground-truth* images that depict new emotional expressions. Specifically, we leverage UltraEdit's region-based editing to preserve a subject's identity (e.g., hair, eyes, lip region) while refining localized areas (e.g., skin texture, eyebrows, cheek geometry) to convey specific emotions. This yields consistent

emotion cues (e.g., *"sad," "angry," "surprised"*) that drive the optimization of our 3D Gaussian avatars. Unlike text-to-image methods that generate entirely novel images, UltraEdit operates at a *per-pixel* level, ensuring crucial identity details remain intact—a key requirement for avatar-based reconstruction (Section 4).

Moreover, by using soft and hard masks (Subsection 4.2.2), we contain the random tendencies of diffusion-based editing to targeted facial regions, preventing unwanted modifications to hair, or other sensitive identity preserving regions. Trained on a large corpus of *real* and *generated* image-instruction pairs, UltraEdit exhibits robustness to diverse appearances and geometries, supporting richer or more subtle emotional changes (e.g., slight smiles vs. intense frowns). Thus, UltraEdit forms the backbone of our self-supervised emotion-editing approach: its masked, view-specific edits become the reference *pseudo-ground-truth* images that our 3D pipeline aims to match.

3.4 3D Frameworks for Emotion Editing

While significant progress has been made in altering facial expressions in 2D (as explored in Section 3.3), the direct manipulation of *3D* avatars to convey varied emotional states remains comparatively underexplored. One reason for this scarcity of research is the difficulty of acquiring suitable 3D training data that is both accurately labeled for emotion and sufficiently diverse to model subtle or extreme facial expressions. Unlike 2D video editing, where large-scale datasets such as VoxCeleb [NCZ17] or LRW [ZC16] can be repurposed to some extent for emotion tasks, the 3D domain requires specialized capture setups (e.g., multi-view rigs, 3D scans, or volumetric studios). Furthermore, the spatiotemporal complexity of full facial geometry and texture often demands more intricate modeling and supervision strategies.

3.4.1 Challenges in 3D Emotion Manipulation

The limited exploration of 3D emotion editing can be attributed to the following key challenges:

- **Data Acquisition:** Existing 3D datasets (e.g., VOCASET [Cud+19a] or MEAD [Wan+20]) typically center on speech-driven animation with limited or highly constrained emotional annotations. Capturing extensive emotional variations (including subtle or mixed states) often requires expensive, actor-driven motion capture sessions.

- **Expressive Deformations:** Emotions are not confined to small regions of the face; they can involve movements of the brows, cheeks, mouth, jaw and eyes. Achieving realistic 3D deformations while retaining identity and lip synchronization to speech can be far more complex than simpler 2D landmarks or warp fields.

- **Generalization and Free-Viewpoint Rendering:** A fully 3D representation must generalize across different poses, camera viewpoints, and lighting conditions. This raises the bar for model capacity and training data requirements, particularly if one aims for high fidelity in open-world scenarios.

Given these obstacles, only a handful of methods have aimed to achieve direct 3D emotion editing for avatars. Many existing solutions rely on parametric face models—for instance, FLAME [Li+17], SMPL-X [Pav+19], the Basel Face Model (BFM) [Pay+09a], or FACEWARE [25]—which provide expression blendshapes or latent codes for controlling facial geometry. While such models effectively capture a wide range of deformations, they often

require extensive labeled training datasets to map specific emotion categories onto the parametric space. Other approaches adapt 2D methods to partial 3D rigs, combining geometry constraints with image-based inpainting [Thi+18; Kim+18; IBP15], but these can still be limited by the underlying requirement for 2D-labeled emotion data.

It is important to note that leveraging a parametric model like FLAME [Li+17] or BFM [Pay+09a] does not itself constitute a shortcoming. Indeed, the methodology presented in this thesis also employs FLAME for rigging (see Section 3.2.3), as it provides a convenient and well-established way to anchor 3D Gaussians to a deformable mesh. However, relying on a parametric approach *solely* for high-level expression blendshapes typically demands supervised training signals or discrete emotion annotations to achieve robust emotion editing. Hence, bridging the gap between purely parametric solutions and more flexible 3D editing paradigms continues to be a research challenge, motivating novel strategies that either exploit limited data more effectively or altogether circumvent the need for large-scale emotion labels.

3.4.2 EMOTE: Emotional Speech-Driven 3D Animation

A recent step forward in 3D-based emotion manipulation is introduced by Danček [Dan+23] in their work, *EMOTE* (Expressive Model Optimized for Talking with Emotion). EMOTE targets the fundamental problem of synchronizing 3D facial animations with speech while explicitly controlling the displayed emotion. Figure 3.13 provides a conceptual overview of their pipeline, in which audio signals, one-hot emotion labels, and a parametric face model are combined to produce lip-synced 3D avatars capable of exhibiting various affective states.

Overview of the EMOTE Approach

Goal and Inputs. EMOTE operates by synchronizing 3-D facial animations with speech while explicitly controlling emotional expressions. The pipeline takes as input:

1. **Raw Speech Audio:** Processed via Wav2Vec 2.0 [Bae+20] to extract phonetic features.
2. **Emotion Labels:** One-hot vectors indicating discrete emotion types (e.g., neutral, happy, sad, anger). The researcher can manually set these labels or retrieve them from a separate emotion recognition pipeline.
3. **Parametric Face Model:** A deformable 3D mesh representation, FLAME [Li+17] (see (§1.1.2)), is used as the underlying geometry. The authors specifically learn transformations in the parameter space of FLAME to animate the face over time.

By merging these three data streams, the pipeline is designed to output a per-frame sequence of FLAME parameters that define both lip articulation and emotional expression in a coherent, speech-synchronized manner.

Content-Emotion Disentanglement: The crux of EMOTE is to disentangle the speech content (which drives rapid lip movements tied to phonetics) from longer-scale, more global emotional expressions such as smiling, brow-lowering, or frowning. Danček employ a *temporal variational autoencoder* (called FLINT in their text) that encodes sequences of 3D face parameters to a latent space. This latent space is then conditioned on emotion features, aggregated by a lightweight transformer from per-frame emotion inferences. The authors also introduce a cross-batch *disentanglement loss*: by swapping the emotion conditions between two sequences but keeping the same phonetic content, they enforce the network to maintain lip-sync consistency even as the emotional state changes.

Speech-Driven 3D Animation: Once trained, EMOTE's encoder-decoder stack maps raw audio features and the user-selected emotion label to a time series of FLAME parameters, animating the mouth region in sync with speech while also modulating the rest of the face to reflect the chosen emotion. The authors highlight this synergy between accurate lip-reading consistency (per-frame alignment with the input speech) and the more holistic emotional expression, achieved via multi-scale losses (including a lip-reading loss and a video-based emotion loss) that encourage plausible 3D geometry under varied conditions.

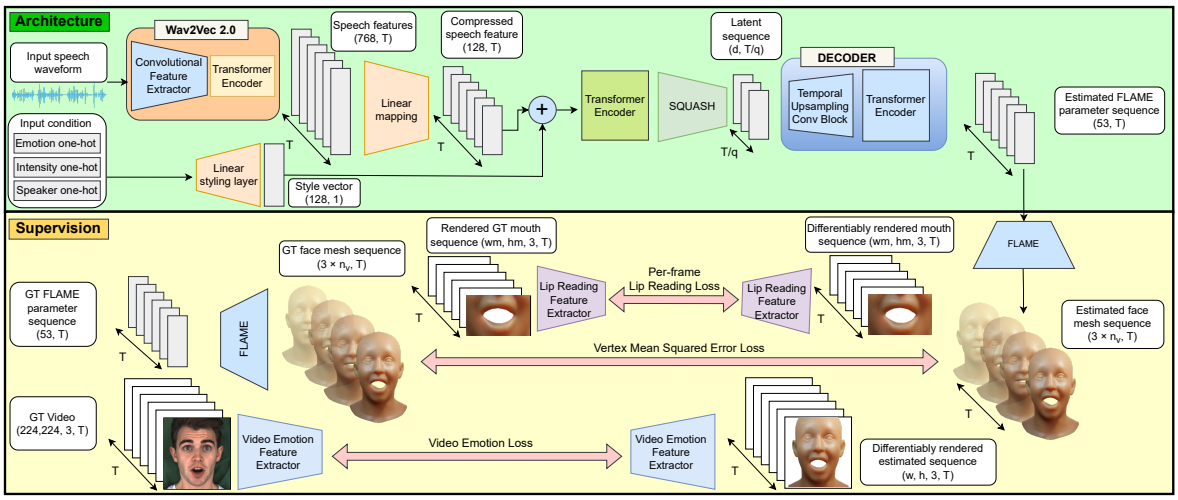


Figure 3.13: EMOTE Pipeline Overview. Speech inputs are encoded by Wav2Vec 2.0 to capture phonetic content, while a separate branch extracts or receives emotion embeddings. A temporal VAE (FLINT) predicts sequences of FLAME parameters conditioned on both speech and emotion. Finally, the FLAME model is rendered or visually evaluated in 3D. Figure courtesy of Danček [Dan+23].

Results and Conclusions: Qualitatively, EMOTE demonstrates high-quality 3D avatars with improved lip synchronization compared to baseline methods on pseudo-3D ground-truth data. Its ability to explicitly assign and interpolate between emotions at test time enables a user-driven pipeline for expressing affective states such as *happy*, *sad*, and *anger*. However, while certain emotions—like *anger* and *happy*—produce visibly distinct changes in facial geometry, others, such as *sad* and *fear*, often share similar structural features, making them difficult to distinguish based on geometry alone. In such cases, texture variations (e.g., wrinkles, shading, and skin tone changes) play a crucial role in enhancing perceptual clarity. This highlights a fundamental challenge: without detailed textural cues, some emotional expressions become ambiguous, complicating both recognition and interpolation.

Figure 3.14 illustrates EMOTE's ability to modify facial expressions across different one-hot emotion labels, demonstrating both its strengths in generating expressive 3D avatars and the importance of texture in distinguishing similar emotions.

Key Strengths and Potential Shortcomings

Although EMOTE constitutes a substantial advancement for 3D avatar animation, it inherits certain limitations from its reliance on labeled 3D emotion data:

- **Discrete Emotion Labels:** While categorical emotions (e.g., anger vs. happy) are well captured through geometry, subtle expressions like a faint smile or hesitation require texture-based cues (e.g., micro-expressions, wrinkles) for accurate recognition.

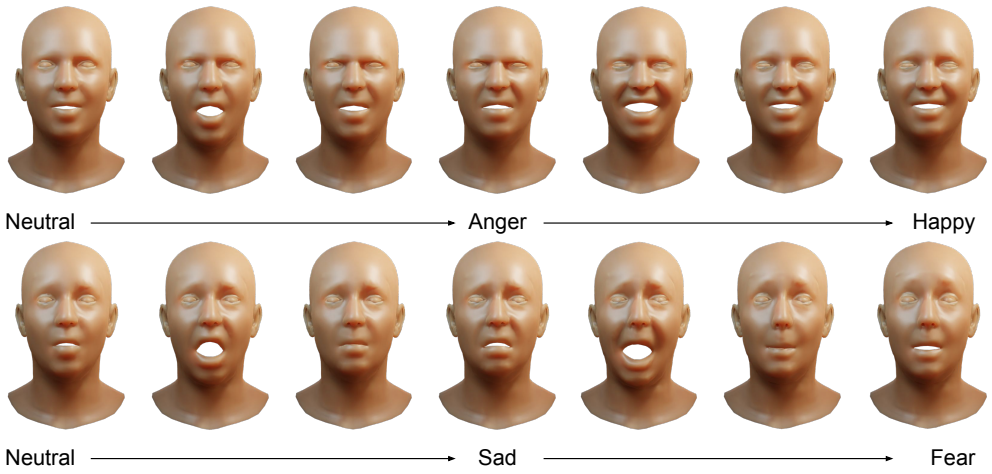


Figure 3.14: Qualitative EMOTE [Dan+23] Results. 3D avatars animated by EMOTE under different one-hot emotion labels (*Neutral*, *Anger*, *Happy*, *Sad*, *Fear*). While EMOTE successfully modifies facial expressions to reflect distinct emotions, some emotions, such as *Sad* and *Fear*, exhibit highly similar geometric structures, making them difficult to differentiate without texture-based cues. This underscores the role of detailed facial textures in improving emotional clarity, especially for subtle affective states. Meanwhile, EMOTE preserves lip synchronization while enabling explicit emotional control.

- **Dataset Dependencies:** Datasets like MEAD [Wan+20] contain specific speakers and scripted emotional expressions, which may generalize poorly to in-the-wild applications or new identity shapes. Furthermore, obtaining more extensive 3D scans with meticulously annotated emotional states remains logistically challenging.

- **Rigid Coupling of Speech and Emotion:** Because the method is primarily speech-driven, certain emotional expressions that occur independently of speech articulation (e.g., raised eyebrows or a suspicious glance while silent) are less directly addressed.

Many of these challenges underscore the broader difficulty of editing emotional expressions in 3D. Unlike in 2D, establishing free-viewpoint, robustly animated faces with limited supervision is more complex due to the higher dimensionality of geometry and appearance. Although this thesis does demonstrate transformations for several discrete emotions (e.g., happy, sad, surprised, angry), the underlying pipeline does not rely on large-scale, emotion-labeled 3D datasets like VOCASET [Cud+19a] or MEAD [Wan+20]. Instead, a diffusion-based guidance mechanism is used to refine expressions and even modify appearance (e.g., adding makeup). This approach avoids rigid dependence on predefined emotion categories, offering a more adaptable framework that can capture subtle or mixed affective states. By moving beyond strictly labeled data, it becomes feasible to explore a broader continuum of human emotions while still accommodating discrete emotional edits as needed.

Conclusion and Future Directions:

In summary, while 3D-based emotion editing is gradually gaining traction, many existing methods—including EMOTE—rely heavily on annotated datasets or pre-defined emotion embeddings. This dependency can constrain the range of expressions and make it challenging to incorporate more subtle affective states. A key limitation of such approaches is their reliance on geometry alone, which can lead to ambiguity in differentiating visually similar but subtle emotions. In such cases, texture-based cues—such as wrinkles, shading, and subtle skin deformations—are essential for conveying emotional depth and improving perceptual clarity.

The approach in this thesis aims to alleviate these limitations by introducing a stable diffusion-driven editing mechanism that operates within a discrete, one-hot-based emotion framework while benefiting from the flexibility of diffusion models. Although the pipeline supports predefined emotion categories (e.g., happy, sad, surprised), its design allows adaptation to evolving diffusion techniques, enabling richer affective variations. By leveraging a generic image-based guidance signal, the proposed method maintains fine-grained textural details crucial for distinguishing complex emotional expressions, enhancing both realism and emotional expressiveness in 3D facial animation.

Chapter 4

Method

4.1 Method Overview

In this chapter, we present the complete workflow behind our *Emotion-Driven Editing of Gaussian Avatars (EMO-GA)*. The process is visualized in a compact diagram, as illustrated in Figure 4.1, that consists of three main phases:

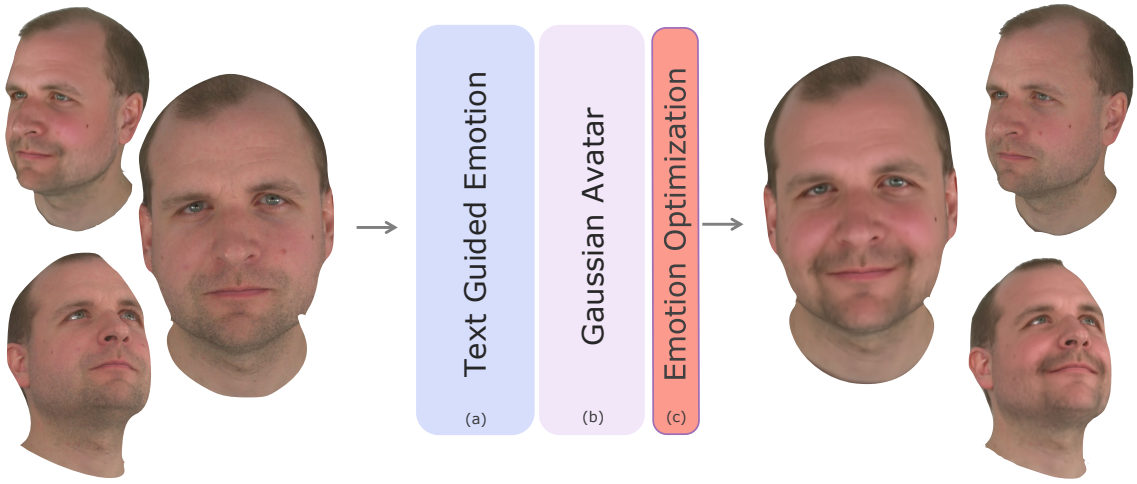


Figure 4.1: High-level overview of our EMO-GA pipeline illustrating self-supervision with text guided emotion (a), Gaussian Avatars, and spatiotemporal consistent emotion optimization (b,c). On the left, the input is a multiview sequence, while on the right, the output is an emotion-driven, view-consistent 3D avatar.

- **Phase (a) in Lavender**: The diffusion-based component, providing 2D photometric cues for expression changes.
- **Phase (b) in Lilac**: The Gaussian Avatars component, where we adapt the core pipeline for robust, accelerated emotion editing.
- **Phase (c) in Coral**: The optimization stage that refines both geometry and texture according to the desired emotional conditioning.

Initially, we explain how the *diffusion model* (Lavender region) is used to generate guiding images through prompts. This section covers how we extract 2D signals that represent the target emotion, whether using a clear label (e.g., *happy*) or a subtle textual prompt (e.g., “slight smile, no brow raise”).

Next, in the *Gaussian Avatars* (lilac region), we discuss the *FLAME tracking* process, focusing on how we obtain pose and expression parameters from the input multi-view sequences. We also detail the changes we introduce into the baseline pipeline to ensure effective expression manipulation.

In the *optimization strategy* (coral region), We describe the photometric and regularization losses that drive emotion manipulation while preserving the subject's identity. We also explain the constraints that ensure stability and spatiotemporal consistency over entire the sequence.

Finally, We give an overview of the hyper-parameter choices, including learning rates, iteration counts, and parameter ranges for each stage. We clarify how we balance spatiotemporal consistency with visual fidelity when applying our pipeline to different subjects and target expressions.

4.2 Self-Supervision

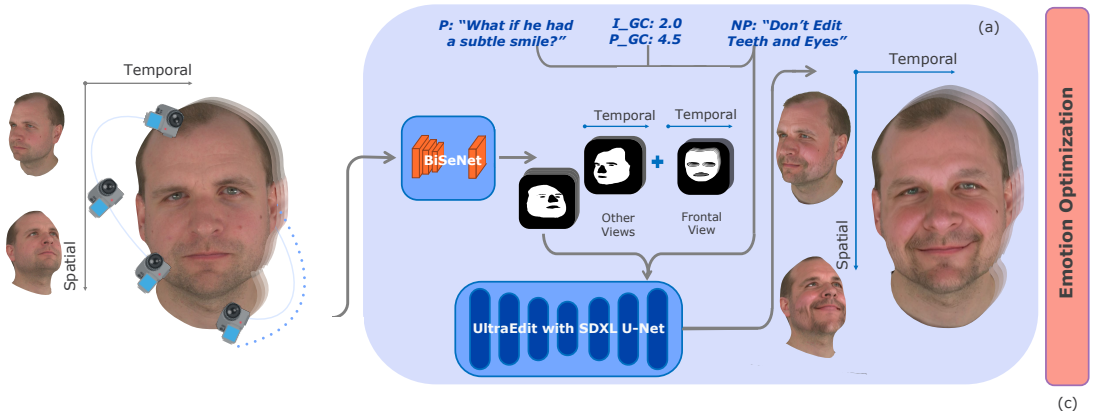
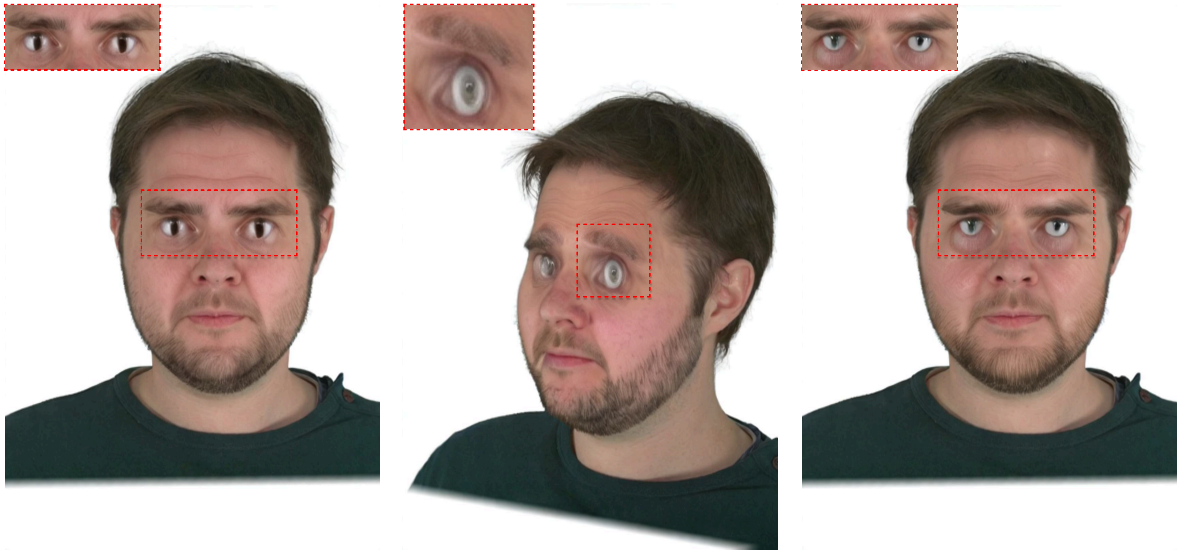


Figure 4.2: Overview of pseudo-ground-truth generation using UltraEdit, a Stable Diffusion-based model [Zha+24]. The input consists of a multi-view sequence (left), and the output is the edited pseudo-ground-truth (right). The figure highlights the hyperparameters (e.g., prompt weights, generation constraints) tuned during the process, alongside the temporal and spatial considerations applied to ensure consistency. UltraEdit processes ground truth sequences with masks and guiding prompts, enabling controlled, emotion-driven edits while preserving facial identity to some degree.

A major challenge in emotion-driven 3D editing is the scarcity of labeled 3D datasets with explicit emotion annotations. If we had access to fully supervised 3D data, we could optimize expressions and textures directly in 3D, but collecting such emotion labels at scale is not feasible. Consequently, we adopt a self-supervised strategy: We begin with nearly neutral facial expressions and generate *pseudo* ground-truth images that guide the avatar toward specific emotions.

However, this approach involves significant complexity because diffusion-based methods inherently lack temporal consistency. A small change in input pixels can drastically alter the resulting image, producing artifacts reminiscent of "ants" or random noise patterns, as illustrated in Figure 4.3. These artifacts complicate the editing process and often disrupt frame-to-frame consistency. To mitigate this, we apply strong regularization constraints, discussed in the optimization strategy (Section 4.5), which counteracts the instability of the pseudo supervision.

We experimented with multiple Stable Diffusion XL-based image-to-image solutions, leveraging their extensive training data to handle a wide range of facial attributes. Ultimately, we chose *UltraEdit* as our primary self-supervised diffusion method due to its ability to incorporate masked regions and produce higher-quality, more localized edits compared to other stable diffusion variants (see Figure 4.2). UltraEdit also offers efficient region-based control for fine-grained adjustments, and these guidance settings tend to preserve identity more consistently than alternative editing frameworks. This setup provides me with sufficiently diverse pseudo emotion exemplars that drive geometry and texture changes in the Gaussian Avatars pipeline.



Timestep 0, View 8 | Timestep 0, View 2 | Timestep 1, View 8

Figure 4.3: Over the sequence, the temporal inconsistency of diffusion gives rise to ant-like artifacts, particularly visible in high-frequency regions like the eyes and beard. These artifacts highlight how diffusion models are temporally inconsistent. Furthermore, when the viewpoint changes for the same timestep, the diffusion results vary dramatically, demonstrating spatial inconsistency. To generate this sequence, a mask was used to preserve identity; otherwise, the denoising process results in a complete identity change.

4.2.1 UltraEdit Pipeline Overview

UltraEdit is an instruction-driven image-editing framework that builds on top of Stable Diffusion technology while incorporating large-scale, high-quality image pairs and carefully designed region-based editing logic. Figure 4.2 provides an overview of its data flow: the pipeline accepts a source image, a user prompt describing the desired change, and an optional mask that isolates the region to be edited. A specialized U-Net refines the output through iterative diffusion steps, ensuring localized edits and preserving essential details in unmasked areas.

This design meets our requirements for self-supervision as it facilitates selective alterations (e.g., adjusting only the key facial features for a smile), maintains better coherence outside the masked region, and enables more reliable pseudo ground-truth creation for emotion editing. These advantages make Ultra-Edit particularly valuable for generating the subtle or targeted expressions required by our Emotion-Driven Editing of Gaussian Avatars pipeline.

4.2.2 Mask vs. No Mask

One of the main reasons we chose UltraEdit for our self-supervised editing is its mask-based approach. In this setup, white regions in a mask indicate where UltraEdit should perform edits, while black regions remain untouched, preserving identity in areas we do not wish to alter. These masks are generated using the Bilateral Segmentation Network (BiSeNet) [Yu+18], which produces what we refer to as "hard masks." These masks typically focus on regions such as the lips, eyeballs, nose, and eyebrows, ensuring localized adjustments to expressions while leaving the overall face shape and other critical features unchanged. Figure 4.2.2 illustrates this process, demonstrating how masks effectively confine edits to the desired areas.

In contrast, processing without masks leads to significant spatial inconsistencies and identity drift. As shown in Figure 4.2.2, the absence of masks causes the diffusion model to alter unintended areas, resulting in dramatic modifications to features like the beard, eyes, and facial contours. This makes the "no mask" approach unsuitable for generating consistent Gaussian avatars, as the subject's identity would drift irreversibly during emotion optimization. The mask-based approach, therefore, is essential for maintaining both coherence and identity throughout the editing process.

Hard masks are effective for most viewpoints, but our main optimization is over the front view, these hard masks can cause problems for the frontal view eight. In that frontal perspective, the region between the nose and the lips, as well as the eyebrows and eyes, is extremely sensitive. If we rely solely on hard masks, UltraEdit might produce unnatural changes around these areas. To address this issue, we create "soft masks" specifically for the frontal view: we enlarge the masked region slightly around the lips and also include the eye sockets, rather than just the eyeballs. This helps preserve the subject's identity by maintaining continuity across subtle facial contours. Figure 4.2.2 compares one such edit using a hard mask.

We apply these soft masks only to the frontal camera view, while BiSeNet's hard masks are used for all other viewpoints. Even if a mask is imperfect in some frames or if UltraEdit occasionally generates unexpected results, the overall pipeline remains stable. The key point is that the masks are generally accurate enough to guide localized edits, ensuring that facial features remain recognizable and that emotion changes target only the desired regions. Hence, from here on, we will show results using masking.

4.2.3 View Dependence

As discussed in Section 4.2.2, the diffusion model is highly sensitive to viewpoint changes, causing significant temporal shifts in denoised results for non-frontal views (see Figure 4.5). This sensitivity prompted me to assign higher learning weights to the front camera during optimization, while reducing contributions from more extreme angles. Doing so lightens the burden on the optimizer, since the frontal view provides a more stable reference for facial features and is also the viewpoint in which emotional expressions are generally most salient.

Figure 4.5 illustrates this behavior, where the ground truth image is shown alongside edited results across ten views. The edits for frontal views maintain better consistency compared to oblique angles, where artifacts and identity drift become more pronounced. This demonstrates the need for selective camera weighting, to improve both convergence speed and final visual quality.

Furthermore, we found that although the dataset (NerSemble 2) offers sixteen camera views, we only needed about ten of them for reliable emotion editing. Some cameras capture angles that are too oblique or uninformative for detailed facial edits, and including them can introduce noise into the optimization. We explore the specific advantages of this selective

camera weighting in the optimization discussion (Section 4.5), where we show how focusing on key viewpoints improves both convergence speed and final visual quality.

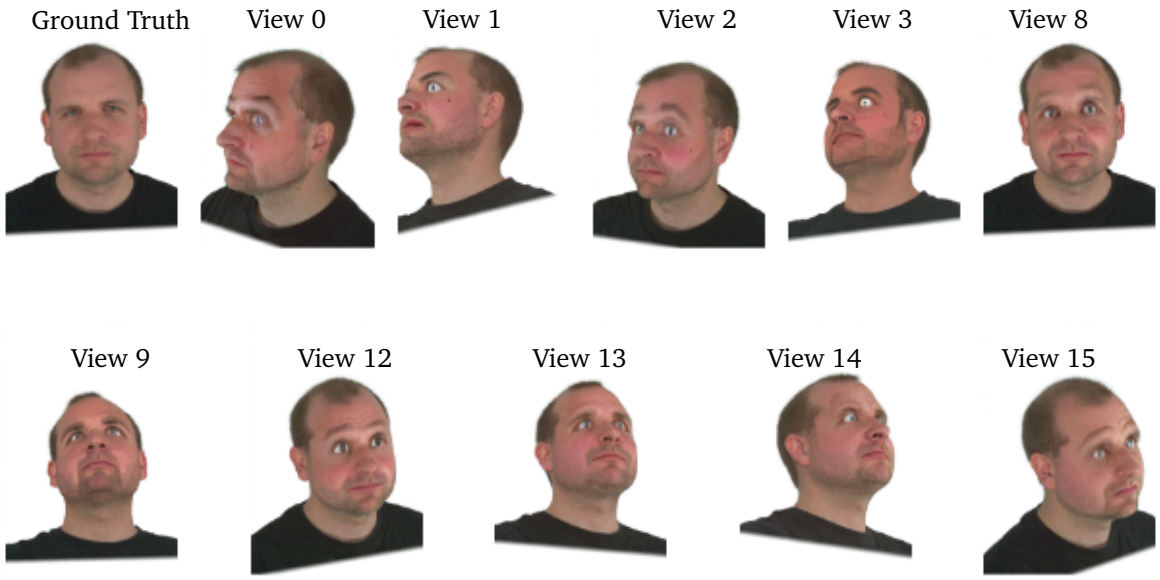


Figure 4.5: Illustration of view dependence in diffusion-based editing. The ground truth image (top left) is shown alongside ten edited outputs (row order) from UltraEdit, corresponding to different camera views. While the edits align well for frontal and near-frontal views (e.g., Views 1–3), significant identity and texture inconsistencies arise at more oblique angles (e.g., Views 1, 2, and 10). This highlights the importance of selective weighting for camera contributions during optimization. **Note:** Mask-based mode was used for these visualizations, as non-masked denoised images are not usable for emotion editing.

4.2.4 Diffusion Control

A distinguishing feature of our workflow is the way we utilize diffusion hyperparameters to guide expression edits without relying on direct identity losses. Specifically, we adjust parameters such as *prompt guidance scale* and *image guidance scale*, as well as use *negative prompts*, to fine-tune how faithfully UltraEdit adheres to the source image versus how aggressively it enforces the new expression. A higher prompt guidance scale, for instance, places stronger emphasis on the textual prompt (e.g., *give the person a subtle smile*), potentially magnifying target attributes (like a broader smile). Conversely, a lower image guidance scale makes the editing process less constrained by the original image, which can lead to noticeable alterations in the subject's identity if taken to the extreme.

Negative prompts help exclude unwanted traits or artifacts by instructing UltraEdit not to introduce, for example, extra teeth or unwanted facial changes. This mechanism complements the masks and fosters more predictable edits. By balancing these diffusion controls, we ensure our pseudo ground-truth images remain coherent with the subject's identity and only incorporate the emotion-specific features we aim to edit (see Figure 4.6 for an exhaustive hyperparameter comparison).

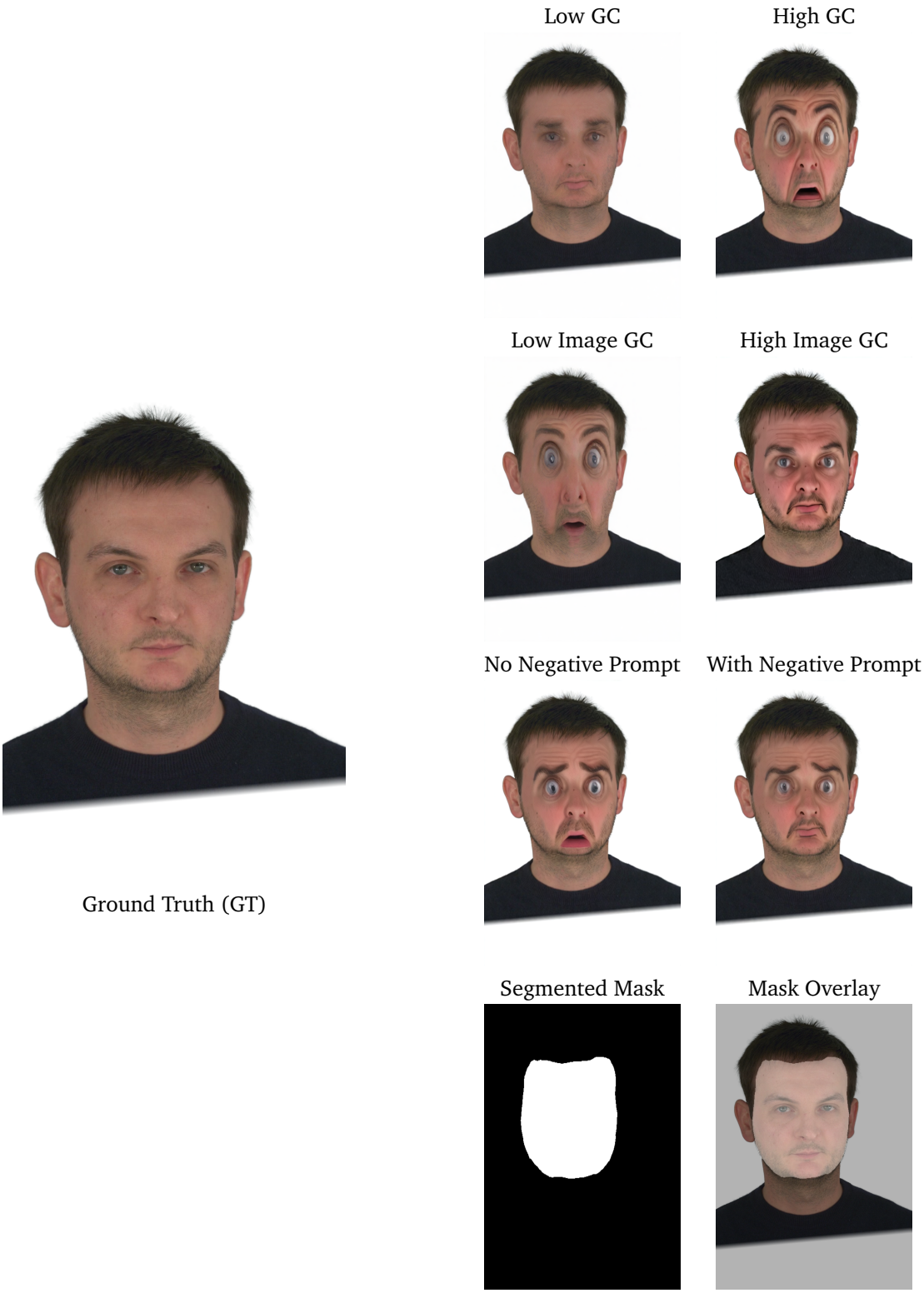


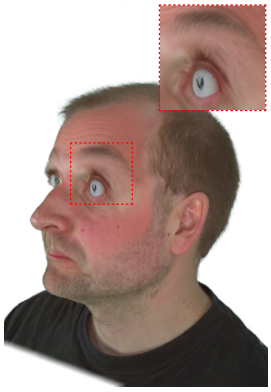
Figure 4.6: Illustration of diffusion hyperparameter effects on generations. The prompt used was: *"What if the person had a surprised face with wide open eyes and raised eyebrows?"* with a fixed seed (4039213507). The top row demonstrates the effect of **guidance control (GC)**: low GC (1) produces less pronounced edits, while high GC (10) exaggerates the expression; a safe GC value is 5. The second row shows the impact of **image guidance control (IGC)**: low IGC (1) leads to identity loss, whereas high IGC (3) respects the input image more; a safe IGC value is 2.0, as anything above this makes the generation overly red and animated. The third row evaluates the effect of **negative prompts**: the absence of a negative prompt produces artifacts in eyes and lips, while using the negative prompt *"do not edit the eyes and lips"* helps preserve their structure. The final row showcases the masks used during generation. The left shows the segmentation mask applied, and the right shows the mask overlaid on the subject. Note that all generations used masking, as non-masked modes were not employed in this thesis.

Figure 4.4: Illustration of spatial inconsistencies in diffusion-based editing. The top row shows the ground truth images for different views. The second row shows results generated using UltraEdit with a mask applied to preserve identity, focusing on critical regions like the lips, eyes, and nose. The third row shows the binary masks used for identity preservation in the second row. The fourth row demonstrates results without any mask input to UltraEdit, where spatial inconsistencies increase drastically across views, leading to noticeable identity drifts.

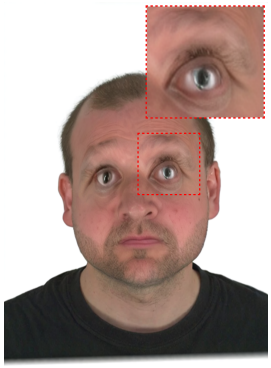
Ground Truth, View 0

Ground Truth, View 8

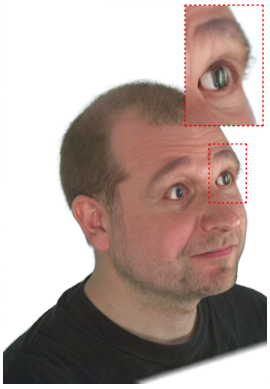
Ground Truth, View 15



Timestep 0, View 0



Timestep 0, View 8



Timestep 1, View 15

Binary Mask, View 0

Binary Mask, View 8

Binary Mask, View 15

No Mask, View 0

No Mask, View 8

No Mask, View 15

4.3 FLAME Tracking via Photometric Optimization

To accurately drive 3D emotion edits, we need per-frame head geometry that captures the subject's true expressions and poses. We obtain this geometry using the VHAP tracker (Versatile Head Alignment with Adaptive Appearance Priors)[Qia24], which aligns a deformable FLAME model to each frame of a multi-view sequence. By employing differentiable mesh rasterization and adaptive appearance priors, VHAP can handle challenging regions like hair, ears, and neck—areas not covered by simpler landmark-based methods.

Figure 4.7 presents an example of the tracker's output for *three views* of the same time step (top, middle, and bottom rows). Some of the important columns to note are:

- **Column 1 (Leftmost)**: The original ground-truth image, showing the subject's raw appearance for each view.
- **Color-Coded Normals** (rainbow-shaded render): Encodes each pixel's 3D orientation.
- **White-Shaded Geometry**: A plain render of the tracked FLAME mesh without texture.
- **Black Silhouette**: The extracted head outline or foreground mask.
- **Blue Overlay with Landmarks** (rightmost columns): Highlights the FLAME mesh overlaid in blue, plus facial landmarks (red or green points).

Once tracking completes its photometric optimization, we obtain a sequence of *per-frame* FLAME meshes reflecting accurate shape, expression, and pose across all viewpoints. These tracked meshes ensure that rigging happens as closely as possible to the subject's original face shape, preserving the unique facial structure. As a result, subsequent edits—such as adding a smile or a frown—remain consistent with the subject's real head motion and identity, maintaining both spatial and temporal coherence throughout the sequence.

Figure 4.7: Example tracking outputs from VHAP for three different *camera views* of the same time step (top, middle, bottom rows). The first column (leftmost) shows the ground-truth image, intermediate columns include color-coded normals and white-shaded geometry, and the rightmost columns show the overlay plus facial landmarks. These results confirm that the FLAME model is fitted consistently, facilitating view-consistent 3D editing in subsequent stages.

4.4 Gaussian Avatars

As mentioned earlier, there is only limited prior work on emotion editing in 3D head models, and most approaches rely on mesh-based representations. We opted to build on the state-of-the-art Gaussian Avatars framework due to its ability to combine high-quality texture representation (via splatted Gaussians) with an underlying FLAME mesh for motion and expression control.

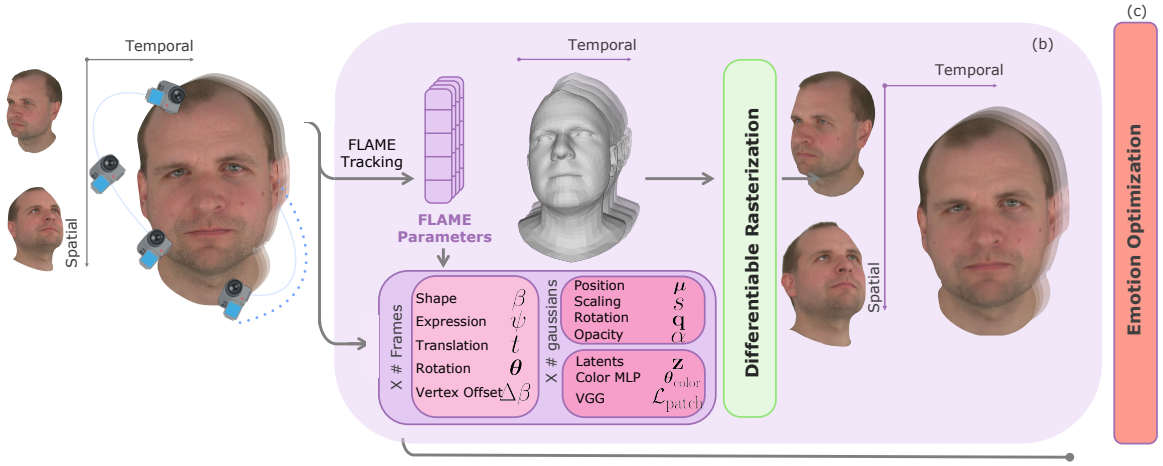


Figure 4.8: Overview of the modified Gaussian Avatars pipeline (b). On the left, a multi-view sequence serves as the input, which is processed to generate a 3D head avatar on the right. FLAME parameters, including shape, expression, translation, and rotation, are preserved from the vanilla Gaussian Avatars framework. Similarly, Gaussian splat parameters remain unchanged except for color. A color MLP is introduced to represent colors conditioned on one-hot emotion vectors and per-Gaussian features (denoted as **z**). Additionally, a perceptual patch loss is incorporated to capture fine-grained local details.

4.4.1 Building on Vanilla Gaussian Avatars

The core Gaussian Avatars technique (Section 3.2) estimates a set of splatted 3D Gaussians bound to a FLAME mesh. Each frame of a multi-view sequence provides updated FLAME parameters—i.e., shape, expression, pose, and vertex offsets—and associates each Gaussian with position, scaling, rotation, opacity, and color. In the original formulation, color is encoded via spherical harmonics to capture view dependency.

4.4.2 Replacing Spherical Harmonics with a Color MLP

In our modified pipeline, we replace the spherical harmonics with a *Color MLP* [Ane+24] conditioned on a one-hot emotion vector and per-Gaussian latent features. This approach allows for better flexibility and accuracy in color representation by replacing the fixed, spherical harmonics-based color model with an emotion-driven architecture. Inspired by *GaussianSpeech* [Ane+24], which utilizes MLP-based color prediction for Gaussian Avatars, we adopt a similar formulation to ensure smooth transitions and better adaptation to emotion-driven changes in our Gaussian Avatars.

Figure 4.9 and Figure 4.10 illustrate the two MLP blocks used in this approach:

1. **Emotion Embedding MLP** (Figure 4.9): This module maps a 5D one-hot emotion vector (e.g., [happy, sad, neutral, etc.]) into a 32-dimensional latent space using two linear layers and a ReLU activation. This embedding captures high-level emotional cues in a compact form.

2. **Color MLP** (Figure 4.10): This block takes as input a concatenation of (a) the 128-dimensional per-Gaussian feature vector and (b) the 32-dimensional emotion embedding. The output is a (16,3) spherical harmonic color representation for rendering. By incorporating emotion information, the Color MLP is capable of capturing subtle expression-driven color variations generated via diffusion.

Figure 4.9: Emotion Embedding MLP. The input neutral vector `[0, 0, 0, 1, 0]` represents the neutral emotion. It uses two linear layers and a ReLU activation to produce a 32D emotion embedding.

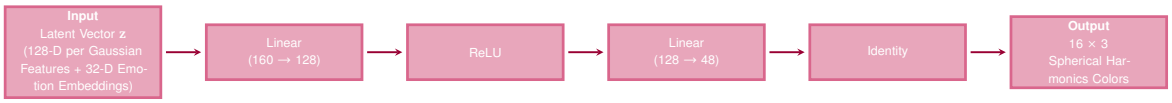


Figure 4.10: Color MLP: The input is a latent vector **z** consisting of 128-D Gaussian features and 32-D emotion embeddings. The output is a 48-dimensional representation of spherical harmonics colors (16 × 3 base + view dependent colors). The MLP employs two linear layers, a ReLU activation, and an Identity for final color prediction.

By utilizing per Gaussian local feature vector and conditioning our MLP on an embedded emotion representation, the pipeline effectively adapts to the original color distribution of the training data. This distinction allows the pipeline to handle the inherent differences between the diffusion-generated data and the original training distribution. By replacing spherical harmonics with the Color MLP, the method achieves sharper detail in regions like the eyes and less observed areas such as the inner mouth. The Color MLP not only ensures smoother transitions across views and significantly reduces artifacts, particularly at oblique angles, but also enforces more robust view consistency while accelerating the overall training process for emotion optimization. We will see the results of this improvement in the next chapter.

4.4.3 Patch-Based Losses for High-Frequency Details

Finally, we incorporate two patch-based loss terms for high-frequency detail:

$$\mathcal{L}_{\text{VGG\_patch}} = \frac{1}{J}\sum_{j=1}^{J}\sum_{k=1}^{K}\left\|\zeta_k\left(I_{\text{render}}^{(j)}\right) - \zeta_k\left(I_{\text{gt}}^{(j)}\right)\right\|_1, \tag{4.1}$$

Here, J is the number of patches, and K is the number of layers in each feature extractor. The operator ζ_k denotes VGG-based perceptual extraction. As illustrated in Figure 4.11, this captures our patching process, where 16 random patches are generated on the face-segmented region of the subject for capturing high-frequency details. For the coming sections:

$$\mathcal{L}_{\text{patches}} = \mathcal{L}_{\text{VGG\_patch}}. \tag{4.2}$$

These loss terms emphasize fine-grained visual details, with $\mathcal{L}_{\text{VGG\_patch}}$ capturing perceptual differences using a VGG-based feature extractor [Sim14]. Both terms focus on all the views, where subtle details (e.g., moles, fine skin texture) are visible in various views, to ensure high-quality Gaussian Avatar.

Overall Loss Composition. In addition to these new patch-based terms, we retain the standard photometric loss $\mathcal{L}_{\text{rgb}}$ from Gaussian Avatars, which itself combines an $\mathcal{L}_1$ term and a Structural Similarity Index (SSIM) term. Formally, the complete objective combines both the baseline loss and the patch-based additions as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{patches}}. \tag{4.3}$$

By enforcing fine-grained consistency on only the most crucial facial regions, this combined loss produces sharper eyes and other unobserved regions, all while preserving the global quality ensured by $\mathcal{L}_{\text{rgb}}$.

(a) Input frame of the subject (b) Face segmented and overlaid (c) Generated patches

Figure 4.11: Illustration of the patching process. Left: Input frame of the subject. Middle: Face-segmented region overlayed. Right: Randomly generated patches on the face region, used for capturing high-frequency details.

4.5 Emotion Optimization

Having established how we generate pseudo ground-truth images via diffusion (Phase (a) in lavender) and how we represent a 3D head avatar with Gaussian splats (Phase (b) in lilac), we now describe the final optimization procedure (Phase (c) in coral) that *bridges* these components to produce the desired emotional edits. Figure 4.12 gives an overview of the entire pipeline: the noisy diffusion outputs serve as target images, driving changes in both FLAME expression parameters and the per-Gaussian color latents to embed the configured emotion.

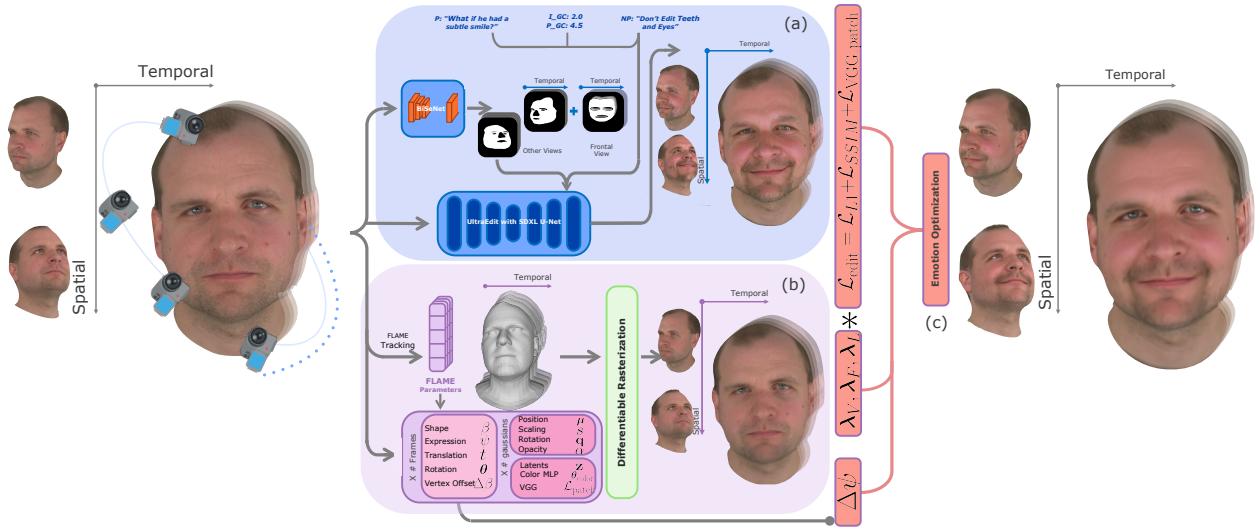


Figure 4.12: Complete EMO-GA pipeline. Phase (a) in lavender (top) produces pseudo-ground-truth images via diffusion-based editing. These are often spatially and temporally inconsistent but carry strong emotional cues. Phase (b) in lilac constructs and renders the Gaussian avatar using FLAME parameters and input sequence. Finally, Phase (c) in coral optimizes both geometry (FLAME expression offsets) and color (per-Gaussian latents) to align the avatar with the noisy, emotion-edited diffusion images as references. For clarity, we illustrate only two views here along the spatial axis, but in practice, we utilize all 15 camera views for Gaussian Avatars, while selecting 10 for emotion optimization

4.5.1 Challenges of Noisy pseudo-ground-truth

As discussed in Section 4.2.2 and Section 4.2.3, the diffusion-generated images introduce considerable temporal and spatial inconsistencies. Although these images carry strong emotional cues, they differ from frame to frame in wrinkles, shading, or contour alignment, and can even exhibit "ant-like" artifacts when played back sequentially. Moreover, they lack the controlled capture conditions of the NeRSemble dataset [Kir+23b], making them an imprecise match for the multi-view geometry of Gaussian Avatars.

A key challenge, then, is to reconcile the stability of the avatar's geometry (informed by the real multi-view data) with the noisy emotional references from diffusion. The inconsistency is not merely a cosmetic issue: if we try to naively fit each frame's geometry and color to a single, potentially flawed diffusion output, we risk local minima that distort identity or produce jitter across frames. Additionally, certain emotional nuances may be misrepresented or "overcooked" by the diffusion model's random variations, making purely geometric or purely color-based edits insufficient.

In light of these issues, our emotion-optimization procedure must:

- **Guard against flicker and overfitting:** We introduce shared expression offsets in FLAME rather than per-frame modifications, preventing frame-by-frame jitter.
- **Balance geometry and color updates:** Subtle shading or lip color changes require texture edits, whereas dramatic expressions necessitate geometry tweaks (e.g., a bigger smile). An iterative scheme updates both in tandem.
- **Focus on robust viewpoints:** By weighting frontal cameras more strongly, we reduce the undue influence of oblique, often noisier, pseudo references.

These strategies collectively address the mismatch between a rigorous multi-view capture and a temporally unconstrained diffusion output, enabling the final pipeline to retain the best of both worlds: a high-fidelity initial 3D avatar derived from real data and vivid, plausible emotion changes guided by the noisy diffusion edits.

4.5.2 Parameters to Optimize

We optimize three main sets of parameters during this emotion-editing phase:

1. **FLAME Expression Offsets:** Rather than adjusting per-frame expression codes independently—risking jitter or overfitting—We introduce a shared offset for all timesteps in the sequence. This offset modifies the neutral FLAME expression parameters in a consistent manner, helping the mesh geometry lean toward the desired emotion without frame-to-frame flicker.
2. **Per-Gaussian Latent Vectors and Emotion Embedding:** Each Gaussian is associated with a 128-dimensional feature vector $\mathbf{z}$, which, together with a one-hot *emotion embedding*, feeds into the Color MLP (Section 4.4.2). By optimizing these vectors and the embedding jointly, the system can modify fine-grained texture details (e.g., wrinkles or subtle shading) to more closely align with the emotional cues present in the pseudo-ground-truth.

4.5.3 Balancing Geometry and Color

Relying on geometry (i.e., FLAME expressions) alone cannot reproduce the subtle color and shading changes observed in the pseudo-ground-truth—particularly when the diffusion model modifies features like lip color, skin tone, or shadowing around the eyes. Conversely, modifying texture alone (via the color latent vector $\mathbf{z}$) often proves insufficient to convey an exaggerated emotion, such as a wide grin or furrowed brow. Consequently, the optimization must balance both expression offsets and color latents, guided by the patch-based and photometric losses, which we will detail in the next section (Section 4.5.5).

To stabilize training, we begin with a *warm-up* stage in which we allow only moderate changes to the FLAME offsets, ensuring the avatar's geometry shifts gradually toward the new expression. Once the avatar has partially matched the diffusion references, we alternate updates between geometry and color parameters—an iterative scheme that prevents runaway solutions in either domain.

As illustrated in Figure 4.13, the difference between optimizing only for geometry and optimizing for both geometry and texture is striking. While a geometry-only approach captures the overall structure, it fails to represent finer details such as lighting variations, skin tone shifts, or lip color changes introduced by diffusion edits. Incorporating photometric losses ensures a more complete reproduction of the target emotion.

Figure 4.13: Comparison of different optimization strategies. The first image from the left shows the original ground truth; the second shows results when optimizing only for geometry; the third demonstrates improvement when optimizing for both geometry and texture; and the fourth shows a geometry deformation heatmap. Without photometric losses, expressions may be acceptable but lack finer texture details like shading and realism. Here, the driving emotion was set to anger.

4.5.4 Regularization and Spatiotemporal Consistency

Because the diffusion-based references can be erratic, we imposed additional constraints to preserve identity and avoid overfitting to misleading artifacts. For instance:

- **Front-View Concentration:** As in Section 4.2.3, we assign higher weights to the frontal cameras during optimization, since the pseudo-ground-truth is typically most reliable for front-facing renders. Masks are more apparent and better defined due to the use of soft masking rather than hard masking, while diffusion generates comparatively more stable results across different views, improving overall consistency. This ensures that the primary emotional cues are learned from views where the expression is most distinguishable.

- **Frame-Wise Regularization:** When optimizing each frame's FLAME expression and shared colors becomes challenging, we apply key frame-based weighting to mitigate diffusion inconsistencies. For certain subjects and extreme expressions—such as surprise with wide-open eyes—diffusion generations can become unconstrained even with soft masking, making reliable optimization difficult. In such cases, the pseudo-ground truth becomes unreliable for emotion editing. To mitigate this, we apply frame-wise weighting, prioritizing key frames where the diffusion edits remain realistic while reducing the weights of more inconsistent frames. This ensures that the optimization process focuses on stable reference points, improving overall emotion transfer and preventing optimization from being driven by unreliable generations.

- **View-Wise Weighting:** While front-facing views provide more reliable pseudo-ground-truth, oblique camera angles are more prone to noise and shape distortions due to inconsistencies in diffusion generations. To prevent such noisy views from degrading the optimization, we weigh them less in the final loss. This ensures that view-dependent artifacts do not propagate across the entire solution, leading to a more stable and coherent reconstruction. However, these oblique views remain necessary—without them, the Gaussian splats tend to overfit to the front view, leading to artifacts in less visible regions, such as random coloration on the neck or unnatural shading in side angles.

Combined with the baseline photometric term $\mathcal{L}_{\text{rgb}}$ and the patch-based losses $\mathcal{L}_{\text{patches}}$, these regularizations ensure the final edited avatar remains visually realistic while still reflecting the emotional cues from the noisy diffusion outputs.

4.5.5 Loss Functions in EMO-GA

To guide the 3D head avatar toward the target emotion while preserving identity and multi-view consistency, we combine multiple loss terms into a single objective, $\mathcal{L}_{\text{edit}}$. These terms balance traditional photometric alignment (via $\mathcal{L}_{\text{L1}}$ and $\mathcal{L}_{\text{SSIM}}$) with perceptual patch-based loss ($\mathcal{L}_{\text{VGG\_patch}}$). We also leverage per-frame ($\lambda^{(f)}_{\text{frame}}$) and per-view ($\lambda^{(v)}_{\text{view}}$) weights to emphasize more reliable frames and viewpoints, as discussed in Section 4.5.4.

L1 Photometric Loss

$$\mathcal{L}_{\text{L1}} = \sum_{f=1}^{F} \lambda^{(f)}_{\text{frame}} \Big(\sum_{v=1}^{V} \lambda^{(v)}_{\text{view}} \left\| I^{(v,f)}_{\text{render}} - I^{(v,f)}_{\text{edit}} \right\|_1 \Big). \tag{4.4}$$

This term measures pixelwise intensity differences between the rendered image $I^{(v,f)}_{\text{render}}$ and the diffusion-edited image $I^{(v,f)}_{\text{edit}}$. We weight each frame f by $\lambda^{(f)}_{\text{frame}}$ and each view v by $\lambda^{(v)}_{\text{view}}$, enabling me to concentrate on specific frames (e.g., with more reliable diffusion edits) or specific views (e.g., frontal) as needed.

SSIM Loss

$$\mathcal{L}_{\text{SSIM}} = \sum_{f=1}^{F} \lambda^{(f)}_{\text{frame}} \Big(\sum_{v=1}^{V} \lambda^{(v)}_{\text{view}} \big(1 - \text{SSIM}\big(I^{(v,f)}_{\text{render}}, I^{(v,f)}_{\text{edit}} \big) \big) \Big). \tag{4.5}$$

The Structural Similarity Index (SSIM) focuses on structural and luminance similarities between the rendered and edited images. Subtracting SSIM from 1 transforms it into a loss. Like $\mathcal{L}_{\text{L1}}$, it leverages the same per-frame and per-view weights to ensure spatiotemporal consistency.

VGG Patch Loss

$$\mathcal{L}_{\text{VGG\_patch}} = \lambda_{\text{view,frontal}} \sum_{f=1}^{F} \lambda^{(f)}_{\text{frame}} \frac{1}{J} \sum_{j=1}^{J} \sum_{k=1}^{K} \left\| \zeta_k \big(I^{(j,f,\text{frontal})}_{\text{render}} \big) - \zeta_k \big(I^{(j,f,\text{frontal})}_{\text{gt}} \big) \right\|_1. \tag{4.6}$$

Here, ζ_k denotes the k-th layer's feature map from a VGG-based network [Sim14], and J specifies the number of patches sampled in the frontal region for each frame f. The operator $I^{(j,f,\text{frontal})}_{\text{render}}$ extracts a *patch* from the rendered image in the frontal camera, and $I^{(j,f,\text{frontal})}_{\text{gt}}$ is the corresponding patch from the diffusion-edited pseudo-ground-truth. The scalar $\lambda_{\text{view,frontal}}$ further emphasizes consistency with the frontal perspective.

Patch Sampling Strategy: To ensure meaningful local feature supervision, we sample 16 random patches from the frontal face segmentation per frame. Each patch has a size of 128×128 pixels, covering approximately 95% of the face region. This ensures that critical areas such as the eyes, nose, and mouth are consistently included while avoiding background distractions. Our patch loss approach is inspired by GaussianSpeech [Ane+24], which leverages local patch-based supervision to refine details in vanilla Gaussian Avatar.

Combined Editing Loss

By summing these four terms, we obtain the total editing loss:

$$\mathcal{L}_{\text{edit}} = \mathcal{L}_{\text{L1}} + \mathcal{L}_{\text{SSIM}} + \mathcal{L}_{\text{VGG\_patch}}. \tag{4.7}$$

Hence, finally, we get:

$$\begin{aligned}\mathcal{L}_{\text{edit}} = &\sum_{f=1}^{F} \lambda_{\text{frame}}^{(f)} \left(\sum_{v=1}^{V} \lambda_{\text{view}}^{(v)} \left\| I_{\text{render}}^{(v,f)} - I_{\text{edit}}^{(v,f)} \right\|_1 + \sum_{v=1}^{V} \lambda_{\text{view}}^{(v)} \left(1 - \text{SSIM}\left(I_{\text{render}}^{(v,f)}, I_{\text{edit}}^{(v,f)}\right)\right) \right) \\ &+ \lambda_{\text{view,frontal}} \sum_{f=1}^{F} \lambda_{\text{frame}}^{(f)} \frac{1}{J} \sum_{j=1}^{J} \sum_{k=1}^{K} \left\| \zeta_k\left(I_{\text{render}}^{(j,f,\text{frontal})}\right) - \zeta_k\left(I_{\text{gt}}^{(j,f,\text{frontal})}\right) \right\|_1.\end{aligned} \tag{4.8}$$

4.5.6 Optimization Conclusion

In summary, the *Emotion Optimization* phase (c) merges the noisy but emotion-rich pseudo ground-truth images from diffusion (phase a) with the stable, multi-view geometry of Gaussian Avatars (phase b). By carefully choosing which parameters to optimize (FLAME offsets, color latents, and emotion embeddings) and restricting the optimization through spatiotemporal weighting and shared expression offsets, the pipeline attains realistic, compelling emotional edits.

4.6 Implementation Details

Our overall implementation combines three separate phases: (a) generating *pseudo* ground-truth edits with UltraEdit, (b) constructing the *Gaussian Avatars* from the NeRSemble dataset, and (c) carrying out the *emotion optimization* that aligns the avatar to the noisy diffusion references. This section provides practical insights into the resolutions, hyperparameters, hardware configurations, and training protocols for each phase.

Phase (a): UltraEdit (Lavender)

- **Hardware and VRAM Usage:** UltraEdit requires large memory capacity to run stable diffusion-based editing. We used an NVIDIA RTX 4090 with 24 GB of VRAM to generate all pseudo ground-truth images. Jointly optimizing UltraEdit and Gaussian Avatars was not feasible due to VRAM constraints, so we instead pre-generated the edited images and stored them in a pickle file.
- **Diffusion Hyperparameters:** For each subject, we explored image guidance scales in the range of 1.5–2.5 and text prompt guidance scales between 4.5–6.0. Through experimentation, we noticed that going beyond 6.0 made the edits deviate too much from the underlying identity. Negative prompts (e.g., "do not add extra teeth," "do not heavily alter the eyes") helped maintain fidelity to the original subject. Example textual prompts include:
  - *"What if the person had a subtle smiling face?"*
  - *"Make the person angry"*

  - *"What if the person had a very sad face?"*
  - *"What if the person had surprised face with wide open eyes and raised eyebrows?"*

- **Soft and Hard Masks:** As described in Section 4.2.2, we used soft masks for frontal views and hard masks for all other views. From these edited images, we typically selected 10 views (out of the 15 available in NeRSemble) for the subsequent optimization step, discarding extremely oblique angles.

- **Final Output Storage:** UltraEdit-generated diffusion images were stored at a resolution of 512×512. To ensure consistency in loss computations during emotion optimization, we downsampled the Gaussian Avatar renders from their original optimized resolution of 1100×1604 to 512×512. This allowed direct comparison between the rendered outputs and the pseudo ground-truth while maintaining computational efficiency. Otherwise, fitting large numbers of Gaussian splats and latent vectors—even for a batch size of 8—on an NVIDIA RTX 4090 would not be feasible due to memory constraints.

Phase (b): Gaussian Avatars (Lilac)

- **Photometric Tracking Resolution:** To obtain per-frame FLAME parameters, we used the VHAP tracker at $\frac{1}{4}$-scale (550×802) of the original 2200×3208 NerSemble frames. Tracking at full resolution was impractical due to excessive VRAM requirements (> 24GB). However, for actually *building* the Gaussian Avatar, we used higher-resolution frames (1100×1602), since Gaussian Avatars are sensitive to image quality and benefit strongly from higher fidelity sequences. Once downsampled to $\frac{1}{4}$-scale, the tracking process required less than 8GB of VRAM, allowing it to run efficiently on an NVIDIA 3070Ti mobile GPU. On this hardware, tracking a subject with 16 sequences and 16 views from the NerSemble dataset took approximately one day to complete.

- **Gaussian Avatars Training:** We largely followed the default settings from the vanilla Gaussian Avatars pipeline, except we shortened the total training from 600k iterations down to 120k. Our experiments indicated that beyond 120k, improvements in color or geometry were minimal—especially given that the Color MLP converges more quickly than the spherical harmonics approach used in vanilla Gaussian Avatars.

- **Color MLP Hyperparameters:** The learning rate for the color MLP began at 1×10^{-3} and decayed exponentially throughout training by 1×10^{-2}. Similarly, the *emotion embedding* was learned with an initial rate of 1×10^{-3}. Inspired by GaussianSpeech [Ane+24], we set $\lambda_{\text{vgg}} = 0.001$ for this additional loss term in the vanilla Gaussian Avatars pipeline.

- **Hardware for Gaussian Avatars:** We used an NVIDIA RTX 2080 Ti (12 GB VRAM), on which it took roughly 7 hours to arrive at a stable Gaussian Avatar sharp enough for further emotion optimization.

Phase (c): Emotion Optimization (Coral)

- **Batched Training Setup:** For each subject, we loaded 10 diffusion-edited views per frame, giving a batch size of ≈ 8 frames (for memory reasons). Therefore, each training step processed about 80 images (10 views × 8 frames). We used Adam [Kin14] for stochastic optimization.

- **Learning Rates and Exponential Decay:** Each learnable parameter group was assigned a distinct initial learning rate and followed an exponential decay schedule for stable convergence. The learning rates were set as follows:

- lr_expr $= 5 \times 10^{-3}$: Controls the FLAME expression offsets.
- lr_gs_feature $= 1 \times 10^{-4}$: Governs the Gaussian per-splat feature vectors ($\mathbf{z}$), where gs_features stands for Gaussian Features.
- lr_color_mlp $= 1 \times 10^{-4}$: Used for updating the color MLP weights.
- lr_emotion_embed $= 1 \times 10^{-4}$: Corresponds to the emotion embedding parameters.

After a short *warm-up* period (Section 4.5.4), each group's learning rate underwent exponential decay, with decay factors falling within the range $[10^{-1}, 1]$. This ensured that the optimization process remained stable after emotion alignment was achieved.

- **Oscillating Learning Strategy:** We divided training into cycles (`-cycle_oscillation` = 2 by default), each lasting about 25 epochs. In one cycle, we focused solely on FLAME offsets, allowing the geometry to gradually adapt to the emotion-driven targets while keeping color MLP learning rates zero. In the alternate cycle, we optimized both FLAME geometry and the color MLP together, ensuring that the FLAME geometry received additional updates while incorporating subtle color shifts. This periodic oscillation strategy allowed the FLAME expression offsets to stabilize and better capture emotion-driven geometric deformations before fine-tuning the appearance features.

- **Expression Offset Multiplier:** We introduced a scale factor (`-expr_offset_multiplier`) in the 10–30 range to globally modulate how far the FLAME expression could deviate from neutrality. This helps expedite reaching a sufficiently exaggerated expression (e.g., frowns) without requiring hundreds of thousands of updates on a plateaued loss landscape.

- **Loss Balancing:** We employed the following recommended defaults:
  - $\lambda_{\text{vgg}} = 0.001$
  - $\lambda_{\text{L1\_SSIM}} = 0.8$ vs. 0.2 ratio for L1 vs. SSIM

  Additionally, the per-view weighting $\lambda^{(v)}_{\text{view}}$ was set to:

$$\lambda^{(v)}_{\text{view}} = \begin{bmatrix} 0.4 & 0.4 & 0.4 & 0.4 & 1.0 & 0.2 & 0.4 & 0.4 & 0.4 & 0.4 \end{bmatrix}, \quad \text{where}$$

$$\begin{aligned} v \in \{0,1,2,3\} &\longmapsto \text{Cameras } \{0,1,2,3\}, \\ v \in \{8,9\} &\longmapsto \text{Cameras } \{8,9\}, \\ v \in \{12,13,14,15\} &\longmapsto \text{Cameras } \{12,13,14,15\}. \end{aligned}$$

  This weighting emphasizes a particular camera if needed, with the frontal camera ($v = 4$) assigned the highest weight ($\lambda^{(v)}_{\text{view}} = 1.0$) for easier optimization. This can also be set to 2.0 for nosier pseudo-ground-truth emotion edited images. Per-frame weights $\lambda^{(f)}_{\text{frame}}$ were selected in the range 0.1–30 based on the diffusion output quality for that frame/emotion.

- **Number of Epochs:** We trained for a total of about 800 epochs, with 150 of them being FLAME expression code warm-ups, gradually ramping up FLAME expression offsets. During warm-up, the geometry does change, however the biggest drop in loss curves comes from our photometric losses.

4.7 Method Conclusion

In this chapter, we introduced a three-phase pipeline—depicted in Figure 4.12—for achieving emotion-driven edits of Gaussian Avatars. Through Phase (a), we generated pseudo ground-truth images using a diffusion-based setup (UltraEdit) that, despite its strong emotional cues, remained noisy and inconsistent. Phase (b) tackled the creation of a robust Gaussian Avatar from the NeRSemble dataset via FLAME tracking and a modified Color MLP, ensuring high-quality geometry, texture, and motion rigging. Finally, Phase (c) bridged these elements, optimizing the avatar to match the noisy yet expressive edited images while respecting multi-view consistency.

A few critical takeaways include:

- **Self-supervision with Masks, guidance control and Weighted Views**: By employing both soft and hard masking strategies as well as guidance control and weighing the views, we maintained each subject's identity under various camera viewpoints while minimizing diffusion artifacts. Additionally, weighting specific frames helped optimization process and foster spatio-temporal stability.
- **Replacing Spherical Harmonics with a Color MLP**: This architectural enhancement allowed more precise color control and faster emotion optimization.
- **Patch-Based Losses for Fine-Grained Detail**: Beyond global photometric consistency, we leveraged perceptual VGG loss to preserve delicate features like high-frequency features and shading, enhancing realism especially around the eyes and the forehead.
- **Iterative Optimization of Geometry and Texture**: We found that a *shared* expression offset across frames mitigated frame-by-frame jitter. Combined with iterative oscillating learning strategy of geometry and color, the pipeline converges to plausible emotional expressions without sacrificing the initial subject identity.

Although this pipeline required careful balancing of losses, weighted cameras, and masking, the final results confirm that stable, 3-D emotion edits can be achieved—even when guided only by the noisy outputs of a diffusion model. In the next chapter, we present qualitative and quantitative evaluations of our approach, showcasing how it produces realistic, view-consistent facial animations across different emotional expressions.

Chapter 5

Results and Ablations

In this chapter, we evaluate our EMO-GA pipeline through a combination of *quantitative metrics* and *qualitative analysis*. Our goal is to assess how effectively EMO-GA modifies facial expressions while maintaining identity consistency and visual realism.

For quantitative evaluation, we use:

1. **Frechet Inception Distance (FID) & Kernel Inception Distance (KID):** These metrics measure how closely the edited results resemble real images in a learned feature space.

2. **Emotion Detection Accuracy (RMN):** We use an emotion recognition network (RMN [PVT21]) to quantify how accurately each method (EMO-GA, EMOTE[Dan+23], Diffusion[Zha+24], and Ground-Truth) conveys the intended expression.

3. **Identity Preservation (DeepFace[SO21]):** We compute face verification distances using **VGG-Face** and **ArcFace** to ensure that edited avatars remain recognizable.

These metrics jointly evaluate fidelity, expression correctness, and identity stability, all of which are critical for realistic and personalized 3D avatars. After presenting these numerical results, we follow up with qualitative comparisons, highlighting where each method excels or struggles in capturing fine-grained emotional nuances.

5.1 Quantitative Comparisons

5.1.1 Frontal-View FID and KID Metrics

To quantify the perceptual quality of generated expressions, we compute two popular metrics: *Fréchet Inception Distance* (FID) and *Kernel Inception Distance* (KID). Here, we focus on frontal views to facilitate direct comparisons among the 3D baselines (EMO-GA and EMOTE) and the 2D Diffusion edits, which naturally have a single viewpoint.

Since EMOTE uses only geometric deformation without retexturing, we obtain a fair comparison by applying EMOTE's expression parameters to our 3D Gaussian Avatar (GA). This produces outputs with identical textures to EMO-GA and ensures that any differences in FID/KID reflect expression quality rather than missing texture information.

Analysis: In Table 5.1, the left two blocks of columns measure alignment against 2D Diffusion-based edits, while the right two blocks measure alignment against the ground-truth (GT) neutral avatar. We *do not cross-compare* these two groups because they represent

Run	EMOGA vs. Diff		EMOTE vs. Diff		EMOGA vs. GT		EMOTE vs. GT	
	FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓
1) Subj 306 (happy)	40.30	**0.0661**	46.31	0.0788	16.07	**0.0248**	22.54	0.0354
2) Subj 306 (angry)	38.58	**0.0600**	51.40	0.0866	16.26	**0.0253**	19.45	0.0297
3) Subj 306 (sad)	35.76	**0.0576**	49.92	0.0862	13.98	**0.0205**	21.06	0.0328
4) Subj 306 (surprised)	29.01	**0.0388**	47.32	0.0741	17.00	0.0262	14.15	**0.0183**
5) Subj 229 (angry)	49.59	**0.0792**	104.92	0.1929	56.19	0.0994	49.98	**0.0857**
6) Subj 229 (happy)	44.69	**0.0725**	86.15	0.1579	61.63	0.1059	25.26	**0.0362**
7) Subj 229 (sad)	52.03	**0.0879**	88.67	0.1604	43.64	0.0758	31.54	**0.0475**
8) Subj 229 (surprised)	50.26	**0.0746**	75.95	0.1283	70.82	0.1256	19.36	**0.0236**
9) Subj 326 (angry)	78.92	**0.1378**	86.64	0.1526	12.47	0.0191	9.23	**0.0126**
10) Subj 326 (sad)	76.10	**0.1320**	79.27	0.1393	10.17	0.0150	7.15	**0.0074**
11) Subj 326 (surprised)	66.00	**0.1093**	71.23	0.1185	16.44	0.0248	11.86	**0.0154**
12) Subj 326 (happy)	64.29	**0.1106**	68.57	0.1185	15.32	0.0241	12.07	**0.0162**

Table 5.1: Frontal-view FID/KID results for each subject and emotion. Lower values indicate better perceptual quality (↓). For each comparison group, the best (lowest) FID is underlined and the best (lowest) KID is in bold.

qualitatively different baselines: (1) matching the appearance of Diffusion outputs that already embed the intended emotion; and (2) matching the subject's neutral GT avatar.

From the Diffusion comparisons (left group), EMO-GA consistently achieves lower (better) FID and KID than the geometry-only EMOTE baseline. This improvement is often by a large margin, suggesting that our approach to optimizing textures—guided by the pseudo-ground-truth Diffusion outputs—yields significantly better perceptual alignment with the edited (emotional) images.

In the **GT** comparisons (right group), EMOTE tends to outperform EMO-GA because EMOTE retains the subject's original textures and only alters the expression. However, EMO-GA is surprisingly competitive, even surpassing EMOTE in a few cases. This indicates that while EMOTE naturally aligns closely to the original texture distribution, EMO-GA's optimized textures are still able to track the subject's appearance with relatively low perceptual error.

It is also interesting to note that, although *visually* our model produces very compelling "angry" expressions, the "sad" emotion sometimes shows an even *better* numerical match when comparing to diffusion edits. One possible explanation is that "sad" edits from Diffusion tend to have softer and more gradual texture changes, making it easier for our optimization process to align with them. In contrast, "angry" expressions often involve sharper wrinkles and more intense shading variations, which may not fully converge to the Diffusion-edited textures, leading to slightly higher FID/KID scores. Thus, while "angry" may be visually striking, the optimization process aligns more effectively with the smoother, less texture-intensive "sad" emotion.

In summary, FID and KID both confirm that EMO-GA aligns more faithfully to the Diffusion-based emotional edits than EMOTE does, reflecting our texture with geometry-optimization advantage. Meanwhile, EMOTE generally aligns more closely with the original avatar, though EMO-GA remains surprisingly close or even better in several cases. These results indicate that controlling textures explicitly (as in EMO-GA) can simultaneously achieve strong alignment with external emotional references (Diffusion outputs) while remaining near the neutral avatar's distribution, underscoring the flexibility and robustness of our method.

5.1.2 Frontal-View Emotion Detection (RMN)

To evaluate whether our edited avatars convey the intended facial expressions, we use an off-the-shelf facial expression recognizer (RMN [PVT21]) that automatically detects faces and classifies them into several emotion labels (e.g., *Angry*, *Happy*, *Sad*, *Surprised*, etc.). While our pipeline also targets *Sad* and *Surprised* expressions, in practice we observed inconsistent bounding-box detections and spurious multi-face outputs for those emotions in certain frames. For clarity, we therefore report RMN-based metrics *only for Angry and Happy*, where the classifier produced stable, single-face detections. We emphasize that automated emotion recognition is only a surrogate measure; a more comprehensive user study could better capture subjective nuances across the full spectrum of expressions.

Analysis: Table 5.2 presents precision and recall scores for *Angry* and *Happy* predictions across EMO-GA (ours), EMOTE [Dan+23], and the Diffusion-based edits [Zha+24]. Both EMO-GA and Diffusion excel at rendering *Happy* expressions, while EMOTE sometimes produces neutral-like expressions instead. For *Angry*, EMO-GA demonstrates high precision but occasionally fails to generate some *Angry* expressions, whereas EMOTE exhibits greater variability in its outputs. These discrepancies suggest that certain fine-grained cues, such as sharper wrinkles or subtle texture variations, is more challenging to consistently model solely via Geometry and vanilla Gaussian Avatars. While RMN confirms that EMO-GA and Diffusion effectively convey *Happy* expressions, EMOTE appears to introduce more neutral tendencies. However, a user study is planned to provide deeper insights and a more comprehensive evaluation across multiple emotions.

Method	Angry		Happy		Sad		Surprised	
	Precision (↑)	Recall (↑)	Precision (↑)	Recall (↑)	Precision (↑)	Recall (↑)	Precision (↑)	Recall (↑)
EMOTE	**1.00**	0.40	0.00	0.00	0.00	0.00	0.00	0.00
Diffusion	**1.00**	0.20	**1.00**	**1.00**	0.00	0.00	**0.67**	**1.00**
EMOGA	**1.00**	**0.60**	**1.00**	**1.00**	0.00	0.00	0.00	0.00

Table 5.2: RMN-based precision and recall for edited emotions (of *Angry* and *Happy*) on EMO-GA, EMOTE, and 2D Diffusion edits.

5.1.3 Frontal-View Identity Preservation

While capturing expressive emotions is crucial, maintaining the subject's identity remains equally important. If an avatar's facial details deviate too far from the original person—losing subtle cues such as the shape of the lips, eye spacing, or small moles—then our pipeline fails its ultimate goal of creating a personalized 3D model. Moreover, diffusion-based 2D edits often introduce noticeable changes to identity due to their denoising process, sometimes resulting in inconsistent or "off-subject" faces. To quantify how well each method balances expression editing with identity fidelity, we measure face verification distances against the neutral (initial) avatar for the entire sequence. Specifically, we adopt VGG-Face and ArcFace[SO21], two widely used face recognition networks that learn complementary deep feature representations. Low average distance indicates stronger identity retention; conversely, higher distance signifies greater risk of drifting away from the subject's original appearance. In the following, we present the identity scores for EMO-GA, EMOTE, and the Diffusion baselines, highlighting the ways in which EMO-GA preserves fine-grained characteristics while achieving strong emotion edits.

Subject	VGG-Face Dist (mean) (↓)			ArcFace Dist (mean) (↓)		
	EMOTE	Diffusion	EMOGA	EMOTE	Diffusion	EMOGA
306	**0.238**	0.375	0.417	**0.255**	0.411	0.398
229	**0.252**	0.320	0.266	**0.234**	0.409	0.324
326	**0.196**	0.317	0.219	**0.212**	0.376	0.241

Table 5.3: Average Identity Distances per subject (rows) and method (columns). We calculate VGG-Face and ArcFace distances from each edited sequence to that subject's neutral avatar, then average across the four emotions for each subject. Lower values mean closer resemblance to the original identity.

Analysis: From Table 5.3, EMO-GA shows moderately higher identity deflections than EMOTE which is expected since EMOTE retains the exact original texture (altering only geometry) and thus remains very close to the neutral avatar. Nevertheless, EMO-GA still consistently outperforms Diffusion, underscoring how soft masking and multi-view constraints help our pipeline avoid excessive "off-subject" drift. Crucially, *all* reported values lie below the commonly cited thresholds (ArcFace distances $\lesssim 0.5$ and VGG-Face distances $\lesssim 0.7$–1.0) that typically indicate the same identity [SO21]. This confirms that, despite EMO-GA's more extensive texture edits, each subject's defining facial cues (e.g., eyes, mouth, or moles) remain largely intact. Furthermore, whereas EMOTE's minimal texture changes naturally yield lower distances, it also conveys more subdued emotions. In contrast, EMO-GA introduces stronger and more diverse expressions while still preserving identity within acceptable limits. We conclude that our pipeline strikes a balanced trade-off between expressive edits and faithful resemblance to the original subject, even compared to a simpler geometry-only alternative.

5.2 Qualitative Comparisons

Although quantitative metrics offer a partial glimpse into our pipeline's performance, they cannot fully convey the richness of 3D facial expressions—particularly when accounting for nuances like subtle skin shading, wrinkles, or high-frequency textural cues. Consequently, we now present qualitative results that highlight how EMO-GA captures and synthesizes diverse emotions on Gaussian splats in ways no table of numbers can adequately depict.

In the following examples, we showcase various subjects from our dataset, spanning multiple viewpoints, timesteps, and expressions (e.g., *happy*, *sad*, *angry*, *surprised*). While an unbounded number of combinations is theoretically possible, we focus on select key frames that illustrate how EMO-GA's texture–geometry synergy yields convincing emotion edits. Figure 5.2 below exemplifies one such comparison: The rows show different subjects, each rendered from multiple views, and the columns depict input frames vs. final avatar states for each target emotion.

Notably, emotions like *happiness* require more than a mere geometric mesh offset (e.g., raising the corners of the lips). True smiles depend on texture changes reflecting creases around the eyes or subtle lighting variations on the cheeks—nuances that geometry alone cannot reproduce. Our pipeline's combined texture-and-geometry optimization proves essential in capturing these high-frequency details. Furthermore, though our diffusion-based "pseudo ground-truth" often suffers from noise and inconsistencies, the final 3D avatars remain stable and view-consistent across frames, thanks to the Color MLP and the multi-view constraints embedded in EMO-GA.

Prompts Used: In generating these pseudo ground-truth edits, we supplied the diffusion

Figure 5.1: Qualitative examples of EMO-GA's emotion edits. Each row corresponds to a single subject at a fixed timestep, although the actual timesteps vary across rows to better illustrate a range of expressions. From left to right, we show the input frame (neutral) followed by EMO-GA outputs under four target emotions: *Happy*, *Sad*, *Angry*, and *Surprised*. Despite being driven by noisy 2D diffusion edits, EMO-GA achieves coherent, view-consistent changes in both geometry (e.g., raised brows, denser cheek regions) and texture (e.g., subtle shading around the eyes, shifts in skin tone for anger). This synergy underscores how combining color and geometry optimization captures nuanced expressions—far beyond what simple mesh deformations alone could convey.

model with succinct textual queries—e.g., `"What if the person had a subtle smiling face?"` or `"Make the person angry"`. For *sad* and *surprised*, we used prompts such as `"What if the person had a very sad face?"` and `"What if the person had a surprised face with wide open eyes and raised eyebrows?"`. We also included a single negative prompt (e.g., `"do not alter the hair or ears"`) to discourage the model from making distracting changes to those areas. By tuning the guidance scale and limiting extraneous facial modifications, we arrived at strong emotional cues that EMO-GA could then adapt into realistic 3D avatars.

In Figure 5.3, we compare two time steps for each of two subjects across four target emotions. Rows (a) and (c) show our EMO-GA results, while rows (b) and (d) depict the diffusion-based edits used to guide EMO-GA.

Colored stars highlight typical diffusion inconsistencies:

- The **red star** shows an instance where the diffusion model stays too close to the neutral avatar, resulting in an under-pronounced emotion.
- The **blue star** indicates problematic "crooked teeth," a known artifact that complicates color alignment in our 3D optimization.
- The **pink star** points to shifting forehead wrinkles between frames—an example of how small changes in the input frame can lead to highly inconsistent high-frequency details.

Despite such noise, masking and hyperparameter tuning enable diffusion to preserve enough identity cues to guide EMO-GA.

Figure 5.2: Additional frontal-view comparisons of EMO-GA's emotion edits. Each row represents a distinct subject, shown under a single camera viewpoint for clarity. From left to right, we have the input frame (close to neutral) followed by EMO-GA's outputs across four target emotions: *Happy*, *Sad*, *Angry*, and *Surprised*. Although our pipeline is inherently multi-view, we present these frontal snapshots to emphasize how each subject's unique facial features (e.g., cheeks, eyebrows, lip shape) transform in a view-consistent manner. Subtle texture cues—like shading near the eyes for *sad* expressions or slightly altered skin tone for *angry*—highlight the combined influence of geometry and color optimization. The results mirror the same prompts described above, demonstrating that EMO-GA can reliably produce expressive variations while preserving each subject's identity from one emotion to another.

Our 3D approach then synthesizes these emotional edits into a stable, view-consistent avatar—free from the drastic frame-to-frame variations seen in raw diffusion outputs.

Figure 5.3: Inconsistencies in Diffusion Edits Across Frames. Rows (a) and (c) depict our EMO-GA outputs at two timesteps (left-to-right: *Happy*, *Sad*, *Angry*, *Surprised*), while rows (b) and (d) are the corresponding diffusion-based references. Colored stars emphasize common issues: a **red star** shows when the model underplays an expression, a **blue star** flags undesirable "crooked teeth," and a **pink star** highlights the inconsistent intensity of forehead wrinkles across frames. These artifacts necessitate careful masking and parameter tuning but still provide enough cues to drive EMO-GA's stable, multi-view emotion edits.

In Figure 5.4, rows (a) and (c) present our EMO-GA emotion edits at a pair of representative timesteps, whereas rows (b) and (d) showcase an EMOTE-based approach (i.e., geometry-only). Both methods attempt to portray four target emotions (*Happy*, *Sad*, *Angry*, and *Surprised*), with the leftmost column providing the original "Input Avatar" for reference. Visually, one can observe that EMO-GA's combined texture–geometry adjustments yield more expressive results, while EMOTE's output sometimes distorts or mutes the intended emotion (as indicated by the red star for *Happy* and the blue star for *Angry*). Meanwhile, EMOTE occasionally fares better in simpler expressions like *Sad* or *Surprised*, confirming that geometry alone cannot capture certain high-frequency or shading-based emotion cues that EMO-GA seamlessly integrates.

5.3 Ablations

We examine two key ablations that highlight the flexibility of our pipeline. **First**, replacing the default spherical-harmonics color model in Gaussian Avatars with our Color MLP accelerates training and preserves better view-consistency at extreme angles (Figure 5.7). **Second**, because EMO-GA relies on strong photometric losses (often from a single high-quality diffusion frame), it can sometimes "overfit" to color modifications well beyond pure expression changes—such as blush or makeup (Figure 5.8). Although this effect is not our primary objective, it illustrates how the pipeline can tackle even inconsistent diffusion artifacts by overfitting on a frame and hence allows a coherent 3D edit when a single frame's cues are especially pronounced.

Figure 5.4: Comparisons of EMO-GA (rows (a) and (c)) vs. EMOTE (rows (b) and (d)) at two timesteps for four emotions. The left column shows the input (neutral) avatar, while the next four columns show *Happy*, *Sad*, *Angry*, and *Surprised* expressions. EMO-GA benefits from texture edits (e.g., subtle lighting shifts in cheeks and lips), whereas EMOTE relies solely on geometric deformation, leading to occasional distortions (red star indicates a curved-down lip for *Happy*; blue star flags a misaligned brow for *Angry*). Audio-based for EMOTE can influence expressions, but purely geometry-driven approaches often fail to reproduce the richer emotional cues that EMO-GA achieves by also updating texture.

Figure 5.5: Multi-frame baseline comparisons for a *Happy* expression. Each row is four timesteps apart. The top row shows the subject's original input sequence, which is theoretically neutral but exhibits slight resting expression variations. The second row (EMOTE) applies only geometric blendshapes for "happy," the third row (Diffusion Edits) uses a fixed random seed for 2D "happy" pseudo-ground truths, and the bottom row (EMO-GA) combines geometry and texture optimization. EMO-GA remains temporally stable.

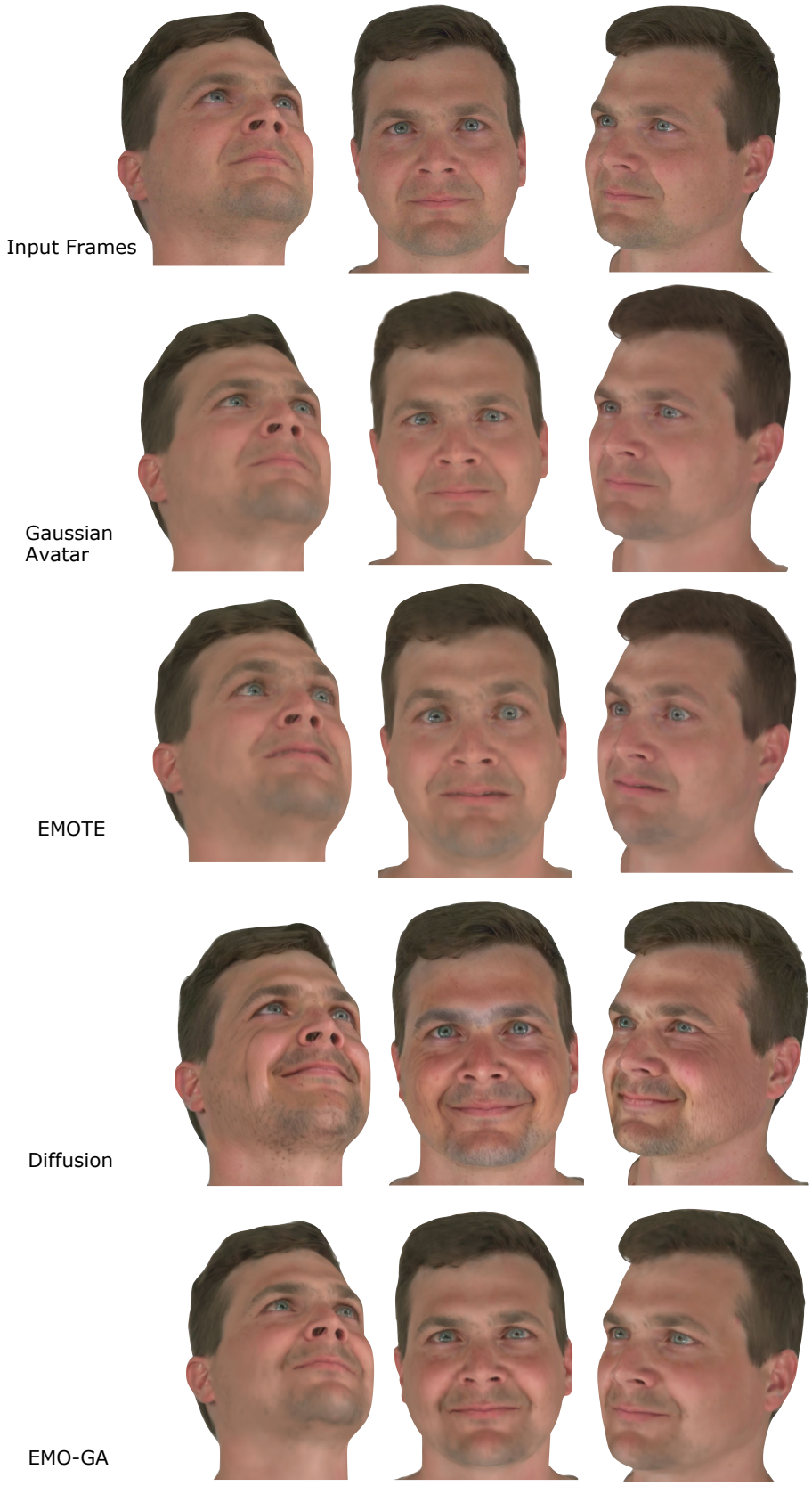


Figure 5.6: Multi-view baseline comparison for a *Happy* expression. The top row shows the high-resolution ground-truth frames (three different camera views) of the subject at the same time-step. Below that, we see: (a) the reconstructed Gaussian Avatar with no emotion edits, (b) EMOTE driving the same avatar purely via geometric blendshapes for *happy*, (c) the 2D diffusion "happy" edits, and finally (d) EMO-GA's texture–geometry outputs. By examining these rows across multiple viewpoints, we can observe how EMO-GA maintains consistent geometry and richer textural details (e.g., shading, cheeks, subtle wrinkles) compared to geometry-only baselines and noisy 2D diffusion references.

Figure 5.7: Color MLP vs. Spherical Harmonics. (a) Vanilla Gaussian Avatar using spherical harmonics for color, (b) Our Color MLP approach. Note how the MLP maintains crisper details at oblique views and converges faster in emotion-optimization.

Figure 5.8: Additional effect of photometric color matching. (a) Baseline avatar, (b) EMO-GA capturing blush from diffusion when prompted for makeup on the forehead and cheeks. Although not our focus, this shows how photometric cues in diffusion images can yield simple makeup-like texture edits.

Chapter 6

Conclusion

In this thesis, we showed that collecting a large, fully labeled 3D dataset is no longer the only path to creating expressive, view-consistent avatars. Instead, we harnessed noisy but emotion-rich 2D diffusion outputs as a form of pseudo supervision, mapping them onto a 3D framework. Despite the inherent instability of diffusion prompts—where a small input change can radically alter the generated face—our pipeline addresses these inconsistencies through careful masking, weighted camera contributions, and spatiotemporal regularization.

Gaussian Avatars proved essential here, combining a high-quality FLAME mesh for rigging with splatted Gaussians high quality textures. By adding a Color MLP (rather than spherical harmonics), our method seamlessly captures subtle shading, wrinkles, or lip coloration that geometry-only approaches often miss. Even when the diffusion references showed mismatched shading or "ants" artifacts, iterative photo-loss optimization allowed the system to reconcile noise and converge on stable, realistic expressions. These results suggest that so long as a 3D pipeline is differentiable and robustly regularized, even imperfect 2D "ground truths" can guide plausible 3D edits.

As we look ahead, spatial computing devices (like the Apple Vision Pro) highlight the value of dynamic, highly personalized 3D avatars in immersive applications. Our EMO-GA approach demonstrates a clear way to incorporate next-generation 2D generative models into multi-view 3D reconstruction without the need for specialized 3D emotion datasets. Future diffusion advances—both in quality and temporal stability—will further reduce the noise that must be filtered out, opening new possibilities for virtual communication, creative expression, and realistic face editing. Ultimately, bridging 2D generative power and 3D geometric consistency stands as a promising route for emotionally expressive avatars in emerging AR/VR ecosystems.

Chapter 7

Future Work

Although our method leverages diffusion outputs for emotion edits, it currently operates at a resolution of 512×512. This has clear limitations for modern high-resolution standards, especially as 4K or even 8K imagery becomes more common. A natural extension would involve integrating more advanced image-generation models (e.g., those offering 1K–2K outputs) into the pipeline. By reducing or eliminating the noise in pseudo ground truths, one might transform the optimization into a more direct mapping task, potentially achieving sharper textures and finer facial details.

Another avenue concerns geometry supervision. In our setup, FLAME expression parameters shift solely through photometric constraints, which is indirect and sometimes struggles with large or subtle deformations. An optional extension is to impose direct geometric losses—perhaps derived from a tool such as DECA[Fen+21], which accurately estimates FLAME expression codes from single images. If each diffusion frame provides consistent FLAME constraints, the optimization might converge more rapidly or with fewer artifacts. The challenge is to reconcile potentially noisy per-frame expressions with the stable multi-view dataset, avoiding uncanny valley effects caused by overshoot in geometry.

A third direction involves moving beyond discrete emotion labels entirely. Instead of targeting specific categories (*happy*, *sad*, and so forth), one could represent emotion along continuous axes (e.g., arousal or valence). Such an approach might permit fine control over subtle expressions, blending or shifting between emotional states smoothly. Training a model in this continuous space might require more sophisticated embedding strategies but could yield more nuanced results for applications ranging from film production to interactive avatars in virtual reality.

Additionally, one can consider a UV-based approach for texture representation, rather than working exclusively in screen space. Recent work such as Texture-GS [Xu+24] projects 3D Gaussian splats onto a 2D UV domain, disentangling geometry and texture more effectively. Incorporating such a strategy into EMO-GA could further improve the stability of texture optimization and allow higher-resolution or more localized edits. This approach might also facilitate downstream processes that expect or benefit from a UV map format (e.g., specialized shading or user-driven painting).

Finally, merging these ideas—high-resolution diffusion references, geometry-aware expression constraints, continuous emotion modeling, and a UV-centric representation—would likely produce a more versatile pipeline. As 3D avatars are set to feature prominently in emerging spatial computing devices, addressing each of these enhancements will ensure that future emotion-editing methods are both visually compelling and flexible across a broad range of deployment scenarios.

Bibliography

[Ane+24] Aneja, S., Sevastopolsky, A., Kirschstein, T., Thies, J., Dai, A., and Nießner, M. "GaussianSpeech: Audio-Driven Gaussian Avatars". In: *arXiv preprint arXiv:2411.18675* (2024).

[AL24] Azari, B. and Lim, A. "EmoStyle: One-Shot Facial Expression Editing Using Continuous Emotion Parameters". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 6385–6394.

[Bae+20] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. "wav2vec 2.0: A framework for self-supervised learning of speech representations". In: *Advances in neural information processing systems* 33 (2020), pp. 12449–12460.

[BV99] Blanz, V. and Vetter, T. "A Morphable Model for the Synthesis of 3D Faces". In: *Proceedings of SIGGRAPH 99*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194. DOI: 10.1145/311535.311556.

[Che+19] Chen, L., Maddox, R. K., Duan, Z., and Xu, C. "Hierarchical cross-modal talking face generation with dynamic pixel-wise loss". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 7832–7841.

[Che+18] Cheng, S., Kotsia, I., Pantic, M., and Zafeiriou, S. "4DFAB: A Large Scale 4D Database for Facial Expression Analysis and Biometric Applications". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[CKH11] Cosker, D., Krumhuber, E., and Hilton, A. "A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling". In: *2011 International Conference on Computer Vision*. 2011, pp. 2296–2303. DOI: 10.1109/ICCV.2011.6126510.

[Cud+19a] Cudeiro, D., Bolkart, T., Laidlaw, C., Ranjan, A., and Black, M. J. "Capture, learning, and synthesis of 3D speaking styles". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 10101–10111.

[Cud+19b] Cudeiro, D., Bolkart, T., Laidlaw, C., Ranjan, A., and Black, M. J. "Capture, Learning, and Synthesis of 3D Speaking Styles". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10093–10103. DOI: 10.1109/CVPR.2019.01034.

[Dan+23] Daněček, R., Chhatre, K., Tripathi, S., Wen, Y., Black, M., and Bolkart, T. "Emotional Speech-Driven Animation with Content-Emotion Disentanglement". In: ACM, Dec. 2023. DOI: 10.1145/3610548.3618183. URL: https://emote.is.tue.mpg.de/index.html.

[Deh+24] Dehghani, M., Shafiee, A., Shafiei, A., Fallah, N., Alizadeh, F., Gholinejad, M. M., Behroozi, H., Habibi, J., and Asgari, E. "Emo3D: Metric and Benchmarking Dataset for 3D Facial Expression Generation from Emotion Description". In: arXiv, Oct. 2024. URL: https://arxiv.org/abs/2410.02049.

[25] *Faceware: Facial Motion Capture and Animation Tools*. Online: https://facewaretech.com/. Accessed: February 1, 2025. 2025.

[Fen+21] Feng, Y., Feng, H., Black, M. J., and Bolkart, T. "Learning an Animatable Detailed 3D Face Model from In-The-Wild Images". In: vol. 40. 8. 2021. URL: https://doi.org/10.1145/3450626.3459936.

[Gaf+21] Gafni, G., Thies, J., Zollhofer, M., and Nießner, M. "Dynamic neural radiance fields for monocular 4d facial avatar reconstruction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8649–8658.

[Goo+14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2014, pp. 2672–2680.

[IBP15] Ichim, A. E., Bouaziz, S., and Pauly, M. "Dynamic 3D avatar creation from hand-held video input". In: *ACM Transactions on Graphics (ToG)* 34.4 (2015), pp. 1–14.

[Iso+17] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. "Image-to-Image Translation with Conditional Adversarial Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1125–1134.

[Ji+21] Ji, X., Zhou, H., Wang, K., Wu, W., Loy, C. C., Cao, X., and Xu, F. "Audio-driven emotional video portraits". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 14080–14089.

[Ker+23] Kerbl, B., Kopanas, G., Leimkühler, T., and Drettakis, G. "3d gaussian splatting for real-time radiance field rendering." In: *ACM Trans. Graph.* 42.4 (2023), pp. 139–1.

[Kim+18] Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Niessner, M., Pérez, P., Richardt, C., Zollhöfer, M., and Theobalt, C. "Deep video portraits". In: *ACM transactions on graphics (TOG)* 37.4 (2018), pp. 1–14.

[Kin14] Kingma, D. P. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[Kir+23a] Kirillov, A., Mintun, E., Ravi, N., Shapovalov, R., Andrade, P. W., Yu, R., Pang, B., Dollar, P., and Girshick, R. *Segment Anything*. arXiv preprint arXiv:2304.02643. https://arxiv.org/abs/2304.02643. 2023.

[Kir+23b] Kirschstein, T., Qian, S., Giebenhain, S., Walter, T., and Nießner, M. "NeRSemble: Multi-View Radiance Field Reconstruction of Human Heads". In: *ACM Transactions on Graphics (ACM Trans. Graph.)* 42.4 (July 2023), 161:1–161:14. DOI: 10.1145/3592455. URL: https://doi.org/10.1145/3592455.

[Li+17] Li, T., Bolkart, T., Black, M. J., Li, H., and Romero, J. "Learning a model of facial shape and expression from 4D scans." In: *ACM Trans. Graph.* 36.6 (2017), pp. 194–1.

[Liu+23] Liu, S., Li, F., Zhang, H., Li, C., Qiu, L., Lu, L., Lu, T., Zhang, X., Zhang, L., Qiu, W., Yuan, B., Zhang, J., and Hu, H. *Grounding DINO: Marrying DINO with Grounded Text Queries for Zero-Shot Object Detection*. arXiv preprint arXiv:2303.05495. https://arxiv.org/abs/2303.05495. 2023.

[Mar+25] Marrie, J., Menegaux, R., Arbel, M., Larlus, D., and Mairal, J. *LUDVIG: Learning-free Uplifting of 2D Visual features to Gaussian Splatting scenes*. 2025. arXiv: 2410.14462 [cs.CV]. URL: https://arxiv.org/abs/2410.14462.

[Mil+21] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[NCZ17] Nagrani, A., Chung, J. S., and Zisserman, A. “Voxceleb: a large-scale speaker identification dataset”. In: *arXiv preprint arXiv:1706.08612* (2017).

[Ope25] OpenAI. *OpenAI Sora*. https://openai.com/. Accessed: 2025-01-02. 2025.

[Pav+19] Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A., Tzionas, D., and Black, M. J. “Expressive body capture: 3d hands, face, and body from a single image”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 10975–10985.

[Pay+09a] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. “A 3D face model for pose and illumination invariant face recognition”. In: *2009 sixth IEEE international conference on advanced video and signal based surveillance*. Ieee. 2009, pp. 296–301.

[Pay+09b] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. “A 3D Face Model for Pose and Illumination Invariant Face Recognition”. In: *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2009, pp. 296–301. DOI: 10.1109/AVSS.2009.58.

[PVT21] Pham, L., Vu, T. H., and Tran, T. A. “Facial expression recognition using residual masking network”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 4513–4519.

[Qia24] Qian, S. *VHAP: Versatile Head Alignment with Adaptive Appearance Priors*. Sept. 2024. URL: https://github.com/ShenhanQian/VHAP.

[Qia+24a] Qian, S., Kirschstein, T., Schoneveld, L., Davoli, D., Giebenhain, S., and Nießner, M. “Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 20299–20309.

[Qia+24b] Qian, Z., Wang, S., Mihajlovic, M., Geiger, A., and Tang, S. “3DGS-Avatar: Animatable Avatars via Deformable 3D Gaussian Splatting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2024, pp. 5020–5030.

[Sab+17] Sabater, N., Boisson, G., Vandame, B., Kerbiriou, P., Babon, F., Hog, M., Gendrot, R., Langlois, T., Bureller, O., Schubert, A., and Allie, V. “Dataset and Pipeline for Multi-View Light-Field Video”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. July 2017.

[SC78] Sakoe, H. and Chiba, S. “Dynamic programming algorithm optimization for spoken word recognition”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.1 (1978), pp. 43–49. DOI: 10.1109/TASSP.1978.1163055.

[SO21] Serengil, S. I. and Ozpinar, A. “HyperExtended LightFace: A Facial Attribute Analysis Framework”. In: *2021 International Conference on Engineering and Emerging Technologies (ICEET)*. IEEE. 2021, pp. 1–4. DOI: 10.1109/ICEET53442.2021.9659697. URL: https://ieeexplore.ieee.org/document/9659697.

[Sim14] Simonyan, K. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[Son+22] Song, L., Wu, W., Qian, C., He, R., and Loy, C. C. “Everybody’s talkin’: Let me talk as you want”. In: *IEEE Transactions on Information Forensics and Security* 17 (2022), pp. 585–598.

[Thi+18] Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., and Nießner, M. "FaceVR: Real-Time Gaze-Aware Facial Reenactment in Virtual Reality". In: *ACM Transactions on Graphics 2018 (TOG)* (2018).

[Wan+20] Wang, K., Wu, Q., Song, L., Yang, Z., Wu, W., Qian, C., He, R., Qiao, Y., and Loy, C. C. "Mead: A large-scale audio-visual dataset for emotional talking-face generation". In: *European Conference on Computer Vision*. Springer. 2020, pp. 700–717.

[Wan+18] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., and Catanzaro, B. "Video-to-video synthesis". In: *arXiv preprint arXiv:1808.06601* (2018).

[Wuu+23] Wuu, C.-h., Zheng, N., Ardisson, S., Bali, R., Belko, D., Brockmeyer, E., Evans, L., Godisart, T., Ha, H., Huang, X., Hypes, A., Koska, T., Krenn, S., Lombardi, S., Luo, X., McPhail, K., Millerschoen, L., Perdoch, M., Pitts, M., Richard, A., Saragih, J., Saragih, J., Shiratori, T., Simon, T., Stewart, M., Trimble, A., Weng, X., Whitewolf, D., Wu, C., Yu, S.-I., and Sheikh, Y. *Multiface: A Dataset for Neural Face Rendering*. 2023. arXiv: 2207.11243 [cs.CV]. URL: https://arxiv.org/abs/2207.11243.

[Xu+24] Xu, T.-X., Hu, W., Lai, Y.-K., Shan, Y., and Zhang, S.-H. *Texture-GS: Disentangling the Geometry and Texture for 3D Gaussian Splatting Editing*. 2024. arXiv: 2403.10050 [cs.CV]. URL: https://arxiv.org/abs/2403.10050.

[Yu+18] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., and Sang, N. "Bisenet: Bilateral segmentation network for real-time semantic segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 325–341.

[ZRA23] Zhang, L., Rao, A., and Agrawala, M. *Adding Conditional Control to Text-to-Image Diffusion Models*. 2023. arXiv: 2302.05543 [cs.CV]. URL: https://arxiv.org/abs/2302.05543.

[Zha+14] Zhang, X., Yin, L., Cohn, J. F., Canavan, S., Reale, M., Horowitz, A., Liu, P., and Girard, J. M. "BP4D-Spontaneous: a high-resolution spontaneous 3D dynamic facial expression database". In: *Image and Vision Computing* 32.10 (2014). Best of Automatic Face and Gesture Recognition 2013, pp. 692–706. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2014.06.002. URL: https://www.sciencedirect.com/science/article/pii/S0262885614001012.

[Zha+24] Zhao, H., Ma, X., Chen, L., Si, S., Wu, R., An, K., Yu, P., Zhang, M., Li, Q., and Chang, B. "UltraEdit: Instruction-based Fine-Grained Image Editing at Scale". In: *arXiv preprint arXiv:2407.05282* (2024).

[Zho+23] Zhong, Y., Wei, H., Yang, P., and Wang, Z. *ExpCLIP: Bridging Text and Facial Expressions via Semantic Alignment*. 2023. arXiv: 2308.14448 [cs.CV]. URL: https://arxiv.org/abs/2308.14448.

[ZC16] Zisserman, A. and Chung, J. "Lip reading in the wild". In: *Asian Conference on Computer Vision (ACCV)*. 2016.

[Zwi+01] Zwicker, M., Pfister, H., Van Baar, J., and Gross, M. "EWA volume splatting". In: *Proceedings Visualization, 2001. VIS'01.* IEEE. 2001, pp. 29–538.

List of Figures

List of Tables