LUCAS: Layered Universal Codec Avatars

Di Liu[1,2] Teng Deng[1] Giljoo Nam[1] Yu Rong[1] Stanislav Pidhorskyi[1] Junxuan Li[1]
Jason Saragih[1] Dimitris N. Metaxas[2] Chen Cao[1]
[1]Codec Avatars Lab, Meta [2]Rutgers University

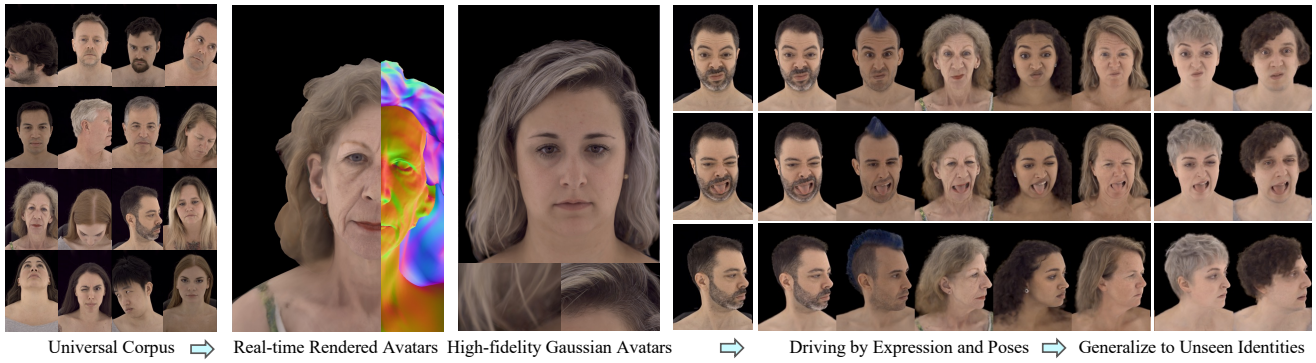

Figure 1. **LUCAS**: A novel approach for high-fidelity **L**ayered **U**niversal **C**odec **A**vatar**s**. We disentangle face and hair into a layered structure, supporting both real-time mesh-based avatar (45 FPS on mobile) and high-fidelity Gaussian avatar generation. Our universal layered prior model also enables accurate expression and pose transfer, even for unseen subjects, while maintaining visual quality.

Abstract

Photorealistic 3D head avatar reconstruction faces critical challenges in modeling dynamic face-hair interactions and achieving cross-identity generalization, particularly during expressions and head movements. We present LUCAS, a novel Universal Prior Model (UPM) for codec avatar modeling that disentangles face and hair through a layered representation. Unlike previous UPMs that treat hair as an integral part of the head, our approach separates the modeling of the hairless head and hair into distinct branches. LUCAS is the first to introduce a mesh-based UPM, facilitating real-time rendering on devices. Our layered representation also improves the anchor geometry for precise and visually appealing Gaussian renderings. Experimental results indicate that LUCAS outperforms existing single-mesh and Gaussian-based avatar models in both quantitative and qualitative assessments, including evaluations on held-out subjects in zero-shot driving scenarios. LUCAS demonstrates superior dynamic performance in managing head pose changes, expression transfer, and hairstyle variations, thereby advancing the state-of-the-art in 3D head avatar reconstruction.

1. Introduction

Photorealistic 3D head avatars are vital for authentic communication in virtual and augmented environments, where capturing subtle expressions and head movements is crucial [33, 44, 55]. High-quality avatars enhance experiences in telecommunications, social VR, virtual training, and healthcare by ensuring accurate geometry and appearance, particularly in dynamic scenarios involving facial and hair deformations [7–9, 11, 12, 15–17, 26–32, 38, 46, 60, 61]. Recent advances in Codec Avatars [22, 35] have achieved remarkable photorealism through sophisticated rendering techniques and volumetric primitives. But these methods often demand significant computational resources, posing challenges for real-time rendering on mobile devices.

Pixel Codec Avatars (PiCA) [37] tackle performance challenges by introducing a pixel-level decoder that dynamically adjusts texture resolution in screen space. This approach enables efficient real-time rendering on mobile devices through direct per-pixel color decoding, eliminating the need for fixed texture maps or vertex-based representations. However, PiCA is a personalized model that requires time-consuming per-identity training, which limits its scalability. Additionally, its single-mesh representation struggles with accurately reconstructing hair, often resulting in artifacts such as hair tails appearing on shoulders or unnatural hair deformation during head movements.

Universal avatar reconstruction approaches [5, 23] have advanced cross-identity generalization, enabling codec avatars to generalize from universal prior models (UPM) trained on data from multiple users. However, these meth-

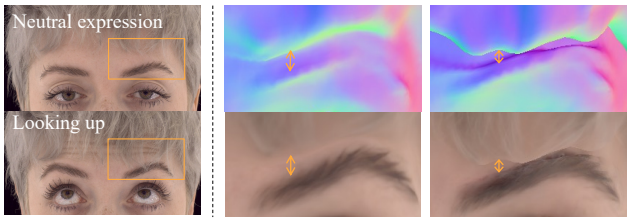

Figure 2. **Layered representation enables adaptive alignment between face and hair.** LUCAS's independent face and hair deformation captures subtle hair movements in response to facial expressions, unlike single-mesh avatars that are globally controlled.

ods encounter significant challenges in hair modeling and dynamic scenarios. Their simplistic representations and inaccurate guide meshes often lead to misaligned geometry and limited hair modeling capabilities, hindering natural deformation during expressions and head movements.

To address these challenges, we propose **LUCAS** (**L**ayered **U**niversal **C**odec **A**vatar**s**), a layered representation that separates face and hair components, allowing them to deform independently while maintaining precise alignment. This design enables more accurate hair dynamics by using shared encoding features but decoding them separately for the face and hair. For instance, as shown in Fig. 2, when the subject gazes upward and frowns, the hair naturally lowers toward the eyebrows. In contrast, single-mesh representations lack the flexibility to disentangle face and hair movements as separate factors, leading to interdependent deformations. Smoothness regularization further exacerbate this issue by enforcing coupled motion. LUCAS overcomes these limitations, enabling realistic and independent deformation for natural movement. LUCAS follows a universal training strategy, training the UPM on data from multiple users, which allows it to generalize easily to unseen users and generate realistic codec avatars. Additionally, we show that our layered mesh design improves the anchor geometry for precise and visually appealing Gaussian renderings [18]. In summary, our contributions are:

- We introduce LUCAS, the first mesh-based Universal Prior Model that enables cross-identity generalization while maintaining real-time rendering on devices.
- The first compositional Universal Prior Model for the head, featuring a layered representation for both the hairless head and hair, which significantly enhances the quality of both face and hair rendering.
- LUCAS demonstrates improved dynamic performance in managing head pose changes, expression transfers, and hairstyle variations, even on unseen subjects in zero-shot driving scenarios.

2. Related Works

3D Head Avatar Reconstruction. Early approaches to head avatar reconstruction were primarily based on 3D Morphable Face Models (3DMFMs) [4], which used linear combinations of prototype vectors for shape and texture generation, later extended with blendshapes for animation [21]. However, these manual blendshape-based approaches were limited in expressiveness and required significant effort to create. Deep learning has revolutionized this field, introducing non-linear models through VAEs [43] and GANs [48] for more complex facial representations. Lombardi *et al*. [33] pioneered joint modeling of shape and appearance using VAEs, while works like Bagautdinov *et al*. [2] and Ranjan *et al*. [43] employed mesh convolutions for detailed geometry capture. FLAME [24] incorporated linear blend skinning for jaw and neck movements but had difficulty conveying subtle expressions. Recent advances have focused on improving rendering quality and generalization. The Pixel Codec Avatar (PiCA) [37] introduced dynamic texture resolution through pixel-based decoding, departing from traditional fixed texture maps [33] and vertex-based representations [64]. However, PiCA's subject-specific nature and single-mesh representation limit its scalability and hair modeling capabilities. Universal models like LatentAvatar [57] and URAvatar [23] have attempted to address generalization across identities, but often struggle with preserving person-specific details and handling large deformations, particularly in hair regions. Cao *et al*. [5] proposed a shared expression space across identities, but accurate guide meshes remained challenging, affecting reconstruction quality. Our work addresses these limitations by combining PiCA's efficient and accurate pixel-based rendering with a Universal Prior Model for cross-identity generalization. Crucially, we introduce a layered representation that separates face and hair components, enabling better alignment and optimization compared to single-mesh approaches while maintaining high visual fidelity across different identities, expressions and poses.

3D Hair Modeling. Hair modeling in 3D avatar reconstruction has been explored through various approaches. Traditional strand-based methods, whether using multiview stereo [36, 40] or single-view inference [6, 56, 62, 63], focus on explicit strand geometry recovery. While these methods can achieve high geometric accuracy, they are often computationally intensive and impractical for mobile VR applications. Alternative approaches have explored different representations for hair modeling. HeadCraft [47] combines parametric head models with StyleGAN-generated displacement maps for animation control and detail preservation. Volumetric methods like Neural Volumes [34] and MVP [35] have demonstrated impressive results in hair rendering, with HVH [53] and NeuWigs [54] further improving hair animation through layered modeling. However, these person-specific models often struggle with generalization to novel identities. While recent works have attempted to address generalization through pixel-aligned information [42]

Figure 3. **Dehaired Head and Hair Geometries.** Our method precisely disentangles dehaired head from hair for different users.

or cross-identity hypernetworks [5], they either face challenges with complex geometry or depend heavily on precise head mesh tracking. Our method focuses on a universal compositional representation that separately models face and hair components using efficient mesh-based representations, enabling real-time rendering while maintaining visual quality across diverse hairstyles and identities.

Compositional Avatar Representation. Compositional modeling has emerged as a promising direction for improving the quality and controllability of 3D avatars. For face and accessories, MEGANE [22] demonstrated the benefits of compositional modeling by combining surface geometry and volumetric representation for eyeglasses, enabling accurate geometric and photometric interactions with faces. DELTA [10] proposed a hybrid explicit-implicit representation to disentangle face and hair components, but primarily focused on static reconstruction and hairstyle transfer. For head avatars, RGCA [45] and URAvatar [23] have shown the advantages of separately modeling head and eye regions for better eye dynamics and relighting effects. Works like GALA [19] and LayGA [25] have introduced layered representations that decompose body and clothing, showing improved results in clothing dynamics and detail preservation. TECA [58] further extends compositional modeling to text-guided avatar generation. Unlike previous works that treat the head as a single entity or focus on static composition, we introduce a layered representation specifically designed to capture the complex dynamic interactions between face and hair during expressions and pose changes. Our approach uniquely combines compositional modeling with a universal prior model, enabling consistent expression transfer and pose-dependent hair dynamics across different identities.

3. Methods

Our approach starts with generating assets for our captured and tracked datasets (Sec.3.1), which support the universal layered prior model (Sec.3.3), based on a novel mesh-based UPM (Sec.3.2). Our layered design improves anchor geometry for precise Gaussian renderings (Sec.3.4). Loss functions and training details are outlined in Sec. 3.5.

3.1. Dataset and Assets

We use the multi-view capture system from Cao *et al*. [5] to record facial performances of 76 identities captured from 110 distinct cameras. To learn 2D hair segmentation textures for each identity, we add predicted segmentation masks from HRNet [51], trained on our in-house dataset. Notably, 5 identities are bald. Starting with these bald individuals, we iteratively build a linear deformable model [3] of bald head geometry, gradually expanding by "dehairing" the next participant with least amount of hair until covering all 76 identities, see Fig. 3. We learn the linear deformable model by using Expectation Maximization (EM) for factor analysis[13], following Torresani [50], and find that adding Laplacian smoothness loss to the M-step can help regularize shapes yielding better results compared to vanilla PCA [1]. Dehairing is performed by computing the expected values of latent variables similar to the E-step by only using the observed data (excluding hair-covered areas) which in turn is used to infer the hidden bald geometry which we use to inpaint the hair regions, stitched using [49].

3.2. Universal Prior Model for Pixel Codec Avatars

Pixel Codec Avatars (PiCA) [37] offers precise mesh tracking and real-time rendering but limited to personalized models. Inspired by [5], we extend Personalized PiCA with cross-identity capacity, powered by a Universal Prior Model (UPM). We call the new model as **uPiCA**. Similarly, uPiCA adopts a Variational AutoEncoder (VAE) [20] architecture with an expression encoder, and an avatar decoder. Besides, an identity-conditioned hypernetwork [14] is added to generate person-specific avatars.

Identity-conditioned hypernetwork $\mathcal{E}_{\text{id}}$ takes a neutral texture map $\mathbf{T}_{\text{neu}}$ and a neural geometry image (mapping vertex position to texture UV space), $\mathbf{G}_{\text{neu}}$, and generates bias maps Θ_{id} for each level of the avatar decoder $\mathcal{D}$ via a set of skip connections. $\mathcal{E}_{\text{id}}$ also generates a per-identity positional encoding f for pixel decoder and per-identity geometry displacement d for geometry decoder:

$$f, d, \Theta_{\text{id}} = \mathcal{E}_{\text{id}}(\mathbf{T}_{\text{neu}}, \mathbf{G}_{\text{neu}}; \Phi_{\text{id}}). \tag{1}$$

Φ_{id} is the trainable parameters for the identity encoder.

Expression encoder. The expression code z are generated by the expression encoder $\mathcal{E}_{\text{exp}}$, which takes the differences between the current and neutral geometry and texture maps as input: $\Delta\mathbf{G}_{\text{exp}} = \mathbf{G}_{\text{exp}} - \mathbf{G}_{\text{neu}}, \Delta\mathbf{T}_{\text{exp}} = \mathbf{T}_{\text{exp}} - \mathbf{T}_{\text{neu}}$, where $\mathbf{G}_{\text{exp}}$ and $\mathbf{T}_{\text{exp}}$ are the current geometry and texture maps, respectively. $z \in \mathbb{R}^{16\times4\times4}$ is defined as:

$$z = \mathcal{N}(\mu, \sigma);\ \mu, \sigma = \mathcal{E}_{\text{exp}}(\Delta\mathbf{T}_{\text{exp}}, \Delta\mathbf{G}_{\text{exp}}; \Phi_{\text{exp}}), \tag{2}$$

where Φ_{exp} are the trainable parameters of $\mathcal{E}_{\text{exp}}$. Since the model is trained end-to-end on multi-identity data, the same

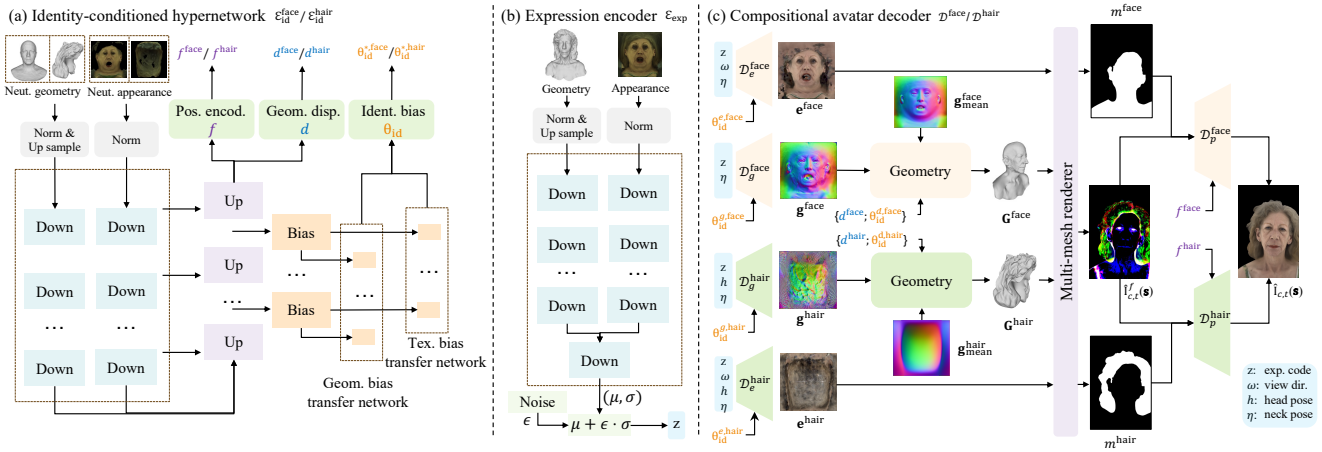

Figure 4. **Overview of LUCAS.** (a) Our identity-conditioned hypernetwork $\mathcal{E}_{\text{id}}^{\text{face}}/\mathcal{E}_{\text{id}}^{\text{hair}}$ generates identity-specific features $\{f, d\}$ and untied biases Θ_{id} from neutral geometry and appearance data. (b) The expression encoder $\mathcal{E}_{\text{exp}}$ learns a unified expression code space that enables consistent expression transfer across identities. (c) Given expression code z, view direction ω, and poses $\{h, \eta\}$, our compositional avatar decoder $\mathcal{D}^{\text{face}}/\mathcal{D}^{\text{hair}}$ produces separate geometry and appearance maps for face and hair. These are combined with mean geometry and geometry displacement for multi-mesh rendering, followed by separate pixel decoders for the final avatar image generation.

expression code can be reused across different identities for driving, ensuring consistent expression transfer.

Avatar decoder. We use a set of multiview images $\mathbf{I}_{c,t}$ (*i.e.*, images from camera c at frame t) with calibrated intrinsics $\mathbf{K}_c$ and extrinsics $\mathbf{R}_c \mid \mathbf{t}_c$. To condition the decoder on the view direction, we compute $\omega = \mathbf{R}_c^\top \mathbf{t}_c$ (approximating the viewing direction based on a head-centered coordinate system). This vector is transformed into a $16 \times 8 \times 8$ grid via a linear layer. Additionally, we enhance the geometry to encompass the shoulder region and use linear blend skinning to model neck pose $\eta \in \mathbb{R}^6$. We use a similar decoder architecture $\mathcal{D}$ as PiCA, which consists of an appearance decoder $\mathcal{D}_e$, a geometry decoder $\mathcal{D}_g$, and a pixel decoder $\mathcal{D}_p$. Noted that the outputs of geometry and appearance decoder are both expression-dependent. The geometry decoder takes the latent code z and neck pose η as input and decodes a head-centered 3D dense position map. The appearance decoder uses the latent code z, viewing direction ω, and neck pose η to decode a low-resolution, view-dependent map of local appearance codes:

$$\mathbf{g} = \mathcal{D}_g\left(z, \eta; \Theta_{\text{id}}^g, \Phi_g\right); \quad \mathbf{e} = \mathcal{D}_e\left(z, \omega, \eta; \Theta_{\text{id}}^e, \Phi_e\right), \quad (3)$$

where $\mathbf{g} \in \mathbb{R}^{256\times256\times3}$ is a map of geometry displacement, and $\mathbf{e} \in \mathbb{R}^{256\times256\times4}$ is a map of appearance codes. Θ_{id}^* are identity-specific biases from Eq. 1 and are related to the corresponding decoders $\mathcal{D}_*$. Φ_* are their corresponding network training parameters. We define the final geometry as $\mathbf{G} = \mathbf{g}_{\text{mean}} + d + \mathbf{g}$, where we apply a Laplacian preconditioning [39] to the gradients of mean geometry $\mathbf{g}_{\text{mean}}$ to bias gradient steps towards smooth solutions. d and $\mathbf{g}$ are the per-identity and expression-dependent geometry displacement, respectively. The final geometry $\mathbf{G}$ is sampled at each vertex's UV coordinates to produce a mesh for rasterization with $\mathbf{e}$. Rasterization assigns to a pixel at screen position $\mathbf{s}$ its corresponding UV coordinates $\mathbf{u}$ and head-centered xyz coordinates $\mathbf{x}$, and produces the feature image $\hat{\mathbf{I}}_{c,t}^f(\mathbf{s})$. The pixel decoder further decodes the color at each pixel to produce the rendered image through:

$$\hat{\mathbf{I}}_{c,t}(\mathbf{s}) = \mathcal{D}_p\left(\hat{\mathbf{I}}_{c,t}^f(\mathbf{s}), f, \mathbf{x}, \mathbf{u}; \Phi_p\right), \quad (4)$$

where Φ_p are the training parameters of $\mathcal{D}_p$. f is the positional encoding from Eq 1. Note that $\mathcal{D}_p$ uses shared weights across subjects, avoiding identity-specific biases from the hypernetwork. Appearance variations are effectively captured by the feature inputs, enhancing network efficiency for runtime deployment and eliminating the need to manage multiple shaders for different users.

3.3. Universal Layered Prior Model

To enable a universal prior model for compositional face and hair avatars across identities, we extend uPiCA to a layered approach, as shown in Fig. 4. In this model, we employ two parallel hypernetworks for face and hair, *i.e.*, $\mathcal{E}_{\text{id}}^{\text{face}}$ and $\mathcal{E}_{\text{id}}^{\text{hair}}$. This separation allows the model to capture intricate details, such as hair deformation due to head movements or facial expressions. We employ a unified expression encoder that extracts shared features from the tracked data, enabling synchronized control of both face and hair deformations through a common expression space. The encoded information is then passed in parallel to two independent decoders, $\mathcal{D}^{\text{face}}$ and $\mathcal{D}^{\text{hair}}$, allowing each part to adapt to its unique geometry and appearance.

Compositional avatar decoder. We use the same decoder architecture as uPiCA from Sec. 3.2 for the face decoders,

and denote them as $\mathcal{D}_g^{\text{face}}$, $\mathcal{D}_e^{\text{face}}$, and $\mathcal{D}_p^{\text{face}}$. For the hair geometry decoder $\mathcal{D}_g^{\text{hair}}$, we use both the head pose h and neck pose η as inputs since these factors influence hair movement. Additionally, we include the latent code z as the input of $\mathcal{D}_g^{\text{hair}}$, as our experiments reveal that hair deforms with certain facial expressions, such as frowning. This behavior arises because the skin beneath the hair shifts with facial movements, causing the hair to adjust accordingly. For the hair appearance decoder $\mathcal{D}_e^{\text{hair}}$, we take all inputs of $\mathcal{D}_g^{\text{hair}}$ along with the view direction ω to account for view-dependent appearance variations, ensuring that both the geometry and texture adapt seamlessly across viewing angles. The layered hair decoders are formulated as:

$$\begin{aligned}\mathbf{g}^{\text{hair}} &= \mathcal{D}_g^{\text{hair}}(z, \eta, h; \Theta_{\text{id}}^{g,\text{hair}}, \Phi_g^{\text{hair}});\\ \mathbf{e}^{\text{hair}} &= \mathcal{D}_e^{\text{hair}}(z, \omega, \eta, h; \Theta_{\text{id}}^{e,\text{hair}}, \Phi_e^{\text{hair}}),\end{aligned} \tag{5}$$

where $\mathbf{g}^{\text{hair}} \in \mathbb{R}^{256\times256\times3}$ and $\mathbf{e}^{\text{hair}} \in \mathbb{R}^{256\times256\times4}$ are the position and texture map of the hair mesh, respectively.

Multi-mesh joint rendering. After decoding the face and hair components, we obtain two geometry maps: $\mathbf{G}^{\text{face}}$ and $\mathbf{G}^{\text{hair}}$. These maps are concatenated and jointly processed with the texture maps $\mathbf{e}^{\text{face}}$ and $\mathbf{e}^{\text{hair}}$ using a differentiable renderer [41] to produce a unified feature vector for the entire screen image. We further apply the face and hair mask m^{face} and m^{hair} to the rendered feature map and feed the masked feature images $\hat{\mathbf{I}}_{c,t}^{f,\text{face}}(\mathbf{s})$ and $\hat{\mathbf{I}}_{c,t}^{f,\text{hair}}(\mathbf{s})$, along with their corresponding $\mathbf{x}$ and $\mathbf{u}$ into separate pixel decoders $\mathcal{D}_p^{\text{face}}$ and $\mathcal{D}_p^{\text{hair}}$. The final rendered image is given by:

$$\begin{aligned}\hat{\mathbf{I}}_{c,t}^{\text{l}}(\mathbf{s}) = \mathcal{D}_p^{\text{face}}(\hat{\mathbf{I}}_{c,t}^{f,\text{face}}(\mathbf{s}), \mathbf{x}, \mathbf{u}; \Phi_p^{\text{face}}) \odot m^{\text{face}}\\ + \mathcal{D}_p^{\text{hair}}(\hat{\mathbf{I}}_{c,t}^{f,\text{hair}}(\mathbf{s}), \mathbf{x}, \mathbf{u}; \Phi_p^{\text{hair}}) \odot m^{\text{hair}}.\end{aligned} \tag{6}$$

3.4. Layered Meshes for Gaussian Rendering

We show that our layered mesh design improves the anchor geometry for precise and visually appealing Gaussian renderings. Building on prior works [5, 23], we parameterize and anchor Gaussians on the vertices of our layered PiCA guide mesh. We employ parallel face and hair branches for the Gaussian hypernetwork and decoder, which share the same architecture. For simplicity, we denote the hypernetwork and decoder for each branch as $\mathcal{E}_{\text{id}}^{\text{gs}}$ and $\mathcal{D}^{\text{gs}}$. The hypernetwork is formulated as:

$$d_{\text{mean}}^{c}, \Theta_{\text{id}}^{\text{gs}} = \mathcal{E}_{\text{id}}^{\text{gs}}(\mathbf{T}_{\text{neu}}, \mathbf{G}_{\text{neu}}; \Phi_{\text{id}}^{\text{gs}}), \tag{7}$$

where d_{mean}^{c} represents the mean color attribute from neutral appearance data, and $\Theta_{\text{id}}^{\text{gs}}$ is the identity-specific bias map. We denote the vertex positions of the layered PiCA guide mesh as $\{\hat{t}_k\}_{k=1}^{M}$, which serve as anchors for the Gaussians. The Gaussian decoder $\mathcal{D}^{\text{gs}}$ takes the expression code z and neck pose η as input, and is conditioned on the identity untied bias map $\Theta_{\text{id}}^{\text{gs}}$. It outputs the following attributes:

$$\{\delta t_k, q_k, s_k, d_k^c, o_k\}_{k=1}^{M} = \mathcal{D}^{\text{gs}}\left(z, \eta; \Theta_{\text{id}}^{\text{gs}}, \Phi^{\text{gs}}\right), \tag{9}$$

where δt_k is the position delta, q_k is the rotation quaternion, s_k is the scale, d_k^c is the color attribute, and o_k is the opacity. The final Gaussian positions are computed as $t_k = \hat{t}_k + \delta t_k$, and colors as $d_{\text{mean}}^{c} + d_k^c$ for rendering.

3.5. Training and Losses

We jointly optimize all the trainable network parameters Φ using a total loss $\mathcal{L}_{\text{total}}$ consisting of:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{pica}}\mathcal{L}_{\text{pica}} + \lambda_{\text{gs}}\mathcal{L}_{\text{gs}} + \lambda_{\text{dehair}}\mathcal{L}_{\text{dehair}}, \tag{8}$$

where $\mathcal{L}_{\text{pica}}$ and $\mathcal{L}_{\text{gs}}$ are the PiCA reconstruction and Gaussian losses, respectively, and $\mathcal{L}_{\text{dehair}}$ is the dehairing loss. λ_* are their corresponding loss weights. For the dehairing loss $\mathcal{L}_{\text{dehair}}$, a large initial weight is applied with a decay during training to accelerate the convergence of bald geometry, ensuring accurate dehaired geometry without interference from the hair mesh. The dehaired avatar serves as the foundation for adding a hair layer, allowing joint optimization of both face and hair with precise alignment.

PiCA reconstruction loss. We extend the original PiCA losses in [37] to a layered reconstruction loss, defined as:

$$\begin{aligned}\mathcal{L}_{\text{pica}} = \lambda_I\mathcal{L}_I + \lambda_D\mathcal{L}_D + \lambda_N\mathcal{L}_N + \lambda_M\mathcal{L}_M\\ + \lambda_S\mathcal{L}_S + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{seg}}\mathcal{L}_{\text{seg}}.\end{aligned} \tag{9}$$

Here, $\mathcal{L}_I$, $\mathcal{L}_D$, $\mathcal{L}_N$, and $\mathcal{L}_{\text{KL}}$ correspond to photometric, depth, normal, and KL divergence losses, respectively, as defined in the original PiCA paper [37]. The photometric loss $\mathcal{L}_I$ measures the L_1 difference between predicted and ground truth images. The mesh tracking loss $\mathcal{L}_M$ handles hair and face meshes separately by leveraging the tracked hair mesh and the dehaired geometry from avatar dehairing. Smoothness terms $\mathcal{L}_S$, including Laplacian and general smoothness regularization, are applied independently to both hair and face meshes to prevent artifacts caused by noisy depth inputs, incomplete depth supervision, or stochastic gradient descent noise. The segmentation loss $\mathcal{L}_{\text{seg}}$ ensures accurate reconstruction of hair regions, particularly thin strands along the sides of the head, preventing them from blending into the face mesh. Notably, we refine the hair segmentation mask through erosion and dilation, creating a mask that extends beyond exact boundaries to account for inaccuracies, and weights $\mathcal{L}_{\text{seg}}$.

Gaussian loss. The parameters of the Gaussian branch are optimized using the following loss:

$$\mathcal{L}_{\text{gs}} = \lambda_{\text{render}}\mathcal{L}_{\text{render}} + \lambda_{\text{scale}}\mathcal{L}_{\text{scale}} + \lambda_{\Delta}\mathcal{L}_{\Delta}. \tag{10}$$

The Gaussian render loss $\mathcal{L}_{\text{render}}$ applies the L_1 loss on the rendered image, following the original 3DGS paper [18]. To regularize the scale of the Gaussian primitives, we define a

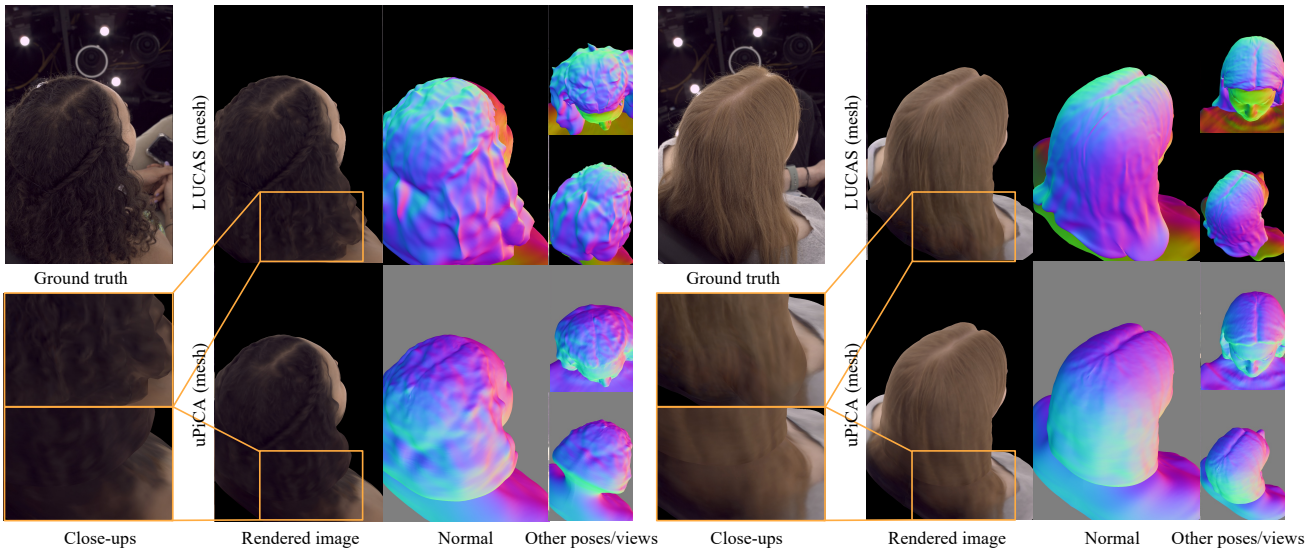

Figure 5. **Qualitative comparison (mesh).** Our layered representation enables better reconstruction of long hair compared to uPiCA's single-mesh approach. While uPiCA struggles with hair-shoulder intersections and loses hair tail details during head movement, our method maintains clean geometry with accurate hair shape and positioning across different head poses.

pre-clamped scale regularization loss as:

$$\mathcal{L}_{\text{scale}} = \frac{1}{M}\sum_{k=1}^{M}\left(\frac{1}{\max(r_{\text{min}}, s_k)}\cdot\mathbb{I}(s_k < r_{\text{min}}) + \left(\max(0, s_k - r_{\text{max}})\right)^2\right), \tag{11}$$

where s_k is the scale value of the k-th Gaussian primitive along any axis, and M is the total number of primitives. The variables r_{min} and r_{max} represent the lower and upper bounds of the primitive scale, set to 0.1 and 5.0 in our experiments. Note that the regularization loss is computed on the original, unclamped scales s_k to penalize deviations effectively. We clamp the primitive scale values to the range $[r_{\text{min}}, r_{\text{max}}]$ before passing them to the Gaussian renderer, ensuring the rendered Gaussians remain within a controlled range. $\mathbb{I}(\cdot)$ denotes the indicator function, which equals 1 if the condition is true and 0 otherwise. This formulation ensures stable optimization by keeping the Gaussian scales within appropriate bounds. Moreover, we apply a delta position loss $\mathcal{L}_\Delta$ to both the hair and face Gaussians to prevent them from drifting too far from their guide mesh. This loss ensures that hair Gaussians stay within the hair area, and Gaussians on the bald regions of the face mesh do not migrate into the hair region. Specifically, the loss penalizes position deviations as follows:

$$\mathcal{L}_\Delta = \mathbb{E}\left[\left(\delta t^{\text{hair}}\right)^2\right] + \mathbb{E}\left[\left(\delta t^{\text{face}} \odot (1 - m^{\text{face}})\right)^2\right], \tag{12}$$

where δt^{hair} and δt^{face} represent the position deltas of the hair and face Gaussians, respectively, and m^{face} is the face mask used to ensure that the delta loss is only applied to the bald head region. More training details are given in *suppl.*

4. Experiments

Evaluation protocols. We adopt three widely-used metrics for evaluation: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [52], and Learned Perceptual Image Patch Similarity (LPIPS) [59]. We restrict the evaluation to the foreground regions, as defined by masks derived from the reconstructed geometry.

Baselines. For mesh-based methods, we primarily compare with Universal PiCA (uPiCA), which extends Pixel Codec Avatars (PiCA)[37] by incorporating our proposed Universal Prior Model (UPM), as detailed in Sec. 3.2. Additionally, we perform per-identity comparisons with PiCA to evaluate personalized reconstruction performance. For Gaussian-based methods, our main comparison is with URAvatar [23], benchmarking our model's ability to capture fine-grained visual details with Gaussian splatting.

4.1. Evaluation of the layered representation

Disentangled representation enhances mesh quality. A key contribution of our work is the compositional representation of the face and hair as two separate meshes. This design addresses a fundamental limitation of single-mesh avatars: their constrained UV space allocation, where hair is restricted to a small portion of the UV map while the face dominates. By allowing separate UV maps for face and hair, our approach enables more accurate representation of complex hairstyles, particularly for long hair. As shown in Fig. 5, the comparison across various head poses demon-

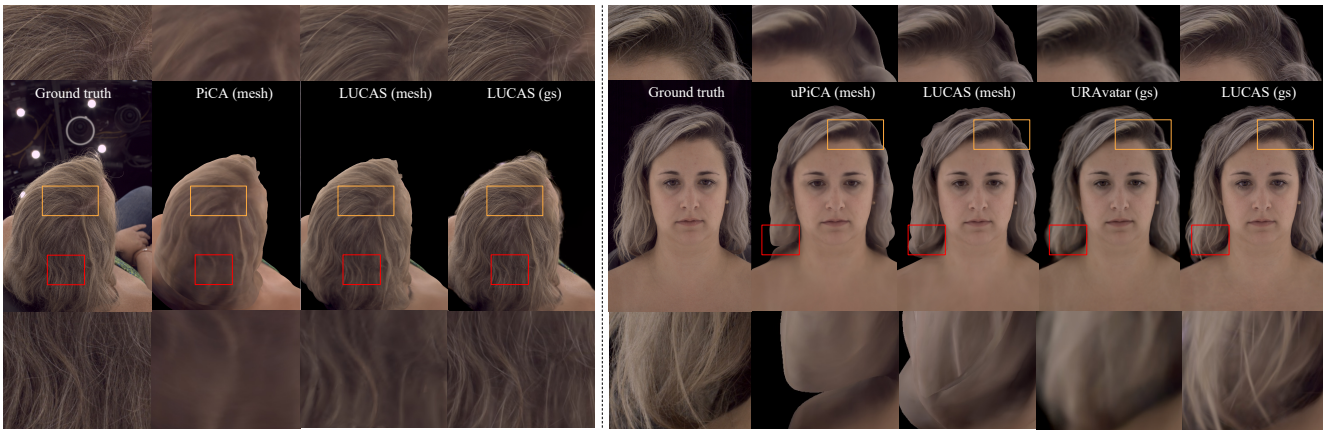

Figure 6. **Qualitative comparison.** Left: Comparison with personalized models shows our method achieves more precise hair reconstruction than PiCA's mesh results. Right: In comparison with universal models, while uPiCA exhibits artifacts such as hair growing from shoulders, our LUCAS (mesh) achieves cohesive reconstruction. When rendered with Gaussian splatting, LUCAS (gs) demonstrates superior detail preservation compared to URAvatar, particularly in complex hairstyles.

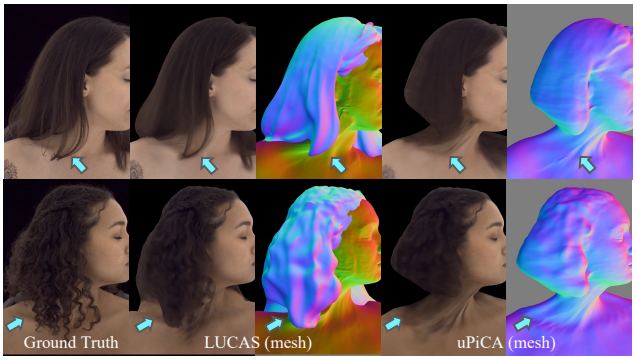

Figure 7. **Comparison on dynamic hair animation.** Our LUCAS mesh tracks hair strand deformation and aligns with head and neck movements, outperforming uPiCA in dynamic scenarios.

strates our method's superior capability in reconstructing long hair details. While uPiCA struggles with hair reconstruction, especially during head movement, our approach maintains precise geometry and reduces common artifacts such as hair color bleeding onto shoulders. This improvement becomes particularly evident when the avatar tilts or lowers its head, where our layered representation ensures the hair remains correctly positioned, resulting in visually coherent and realistic renderings.

Improved hair deformation during animation. In Fig. 7, we demonstrate the advantage of our method in more dynamic scenarios. The examples show how our LUCAS mesh deforms to match hair strand movement in response to head and neck poses, accurately tracking the motion of long hair. In contrast, uPiCA struggles to adapt the hair strands to the changing head positions, resulting in less natural deformations. This comparison highlights the benefit of our layered approach, which provides better control over hair dynamics and improves realism during animation.

Table 1. **Quantitative comparisons** on per-subject (†) and cross-subject (*) optimization against the state-of-the-art. The top three techniques are highlighted in red, orange, and yellow, respectively.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
†PiCA (mesh) [37]	32.0512	0.8895	0.2678
†LUCAS (mesh)	33.5211	0.9044	0.2479
†LUCAS (gs)	35.2027	0.9286	0.2407
*uPiCA (mesh)	32.5623	0.8971	0.2594
*LUCAS (mesh)	33.0254	0.9073	0.2537
*URAvatar (gs) [23]	33.1227	0.9034	0.2464
*LUCAS (gs)	34.5579	0.9201	0.2394

Enhanced Gaussian avatars through better meshes. The improved mesh structure strengthens the foundation for Gaussian avatars, as the Gaussian splatting process relies heavily on the underlying mesh geometry. While Gaussian splatting can mitigate some errors inherent in single-mesh models, our layered approach further enhances visual fidelity, particularly for intricate hairstyles. As shown in Fig. 6, we demonstrate improvements over both personalized and universal models. Compared to PiCA, our approach achieves more detailed hair reconstruction even at the mesh level, with Gaussian splatting further enhancing the visual fidelity. In the universal model comparison, while uPiCA suffers from artifacts like disconnected hair growing from shoulders, LUCAS's mesh representation achieves more cohesive reconstruction. When comparing Gaussian-based methods, LUCAS (gs) demonstrates clear advantages over URAvatar in preserving fine details. These visual improvements are quantitatively validated in Table 1.

4.2. Ablation study

Impact of expression code. In Fig. 8(a), we compare results with and without the expression code for hair. With-

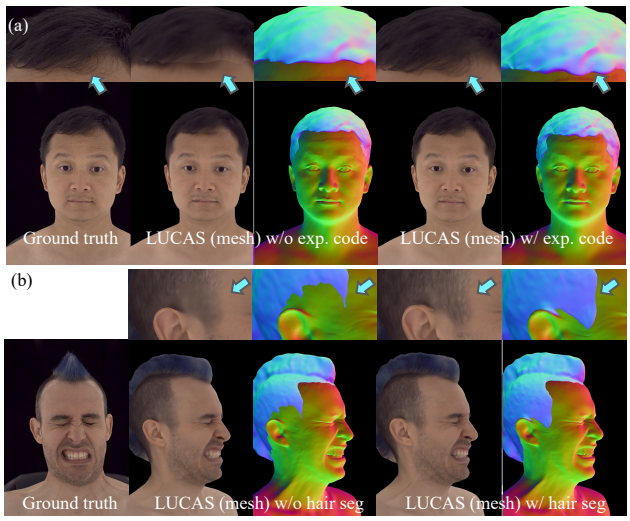

Figure 8. **Ablation study.** (a) Expression code improves face-hair synchronization during expressions. (b) Hair segmentation regularization preserves fine hair details.

Table 2. **Ablation study** on expression code and hair segmentation regularization, evaluated on both training and unseen subjects. The top two techniques are highlighted in red and yellow, respectively.

	Training subjects			Unseen subjects		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
w/o exp. code	34.1014	0.9129	0.2498	31.9128	0.8874	0.2601
w/o hair seg	34.0285	0.9140	0.2485	31.7964	0.9098	0.2554
Full model	34.4981	0.9189	0.2402	32.5847	0.9087	0.2496

out the expression code, the hair mesh fails to move naturally with facial movements, particularly during expressions like frowning. This observation aligns with the findings in Fig. 2, where a subject looks upward and frowns, the hair should lower slightly toward the eyebrows. Our layered representation enables this natural movement by sharing the same expression code z but decoding it separately for face and hair, allowing each component to deform independently and precisely. This advantage is further validated by the quantitative improvements shown in Table 2.

Impact of segmentation regularization. In Fig. 8(b), we assess the effect of hair segmentation regularization. This component is particularly crucial for reconstructing thin hair, as seen in the example, where the hair on both sides is quite fine. Without segmentation regularization, the mesh struggles to capture these thin strands, resulting in blurred renderings. Adding the segmentation term significantly improves the mesh reconstruction, allowing the fine hair to appear correctly in the final render. Quantitative results are also shown in Table 2.

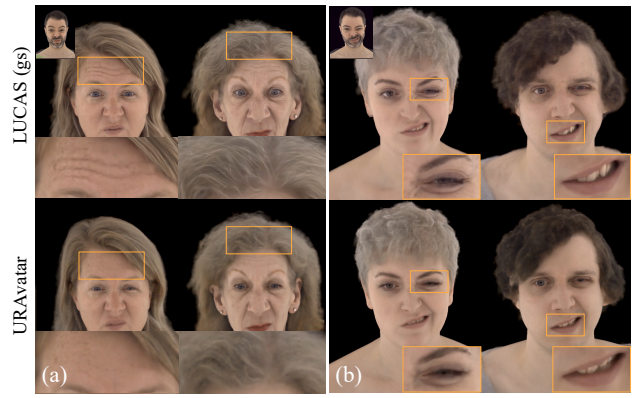

Figure 9. **Visualization of avatar driving.** (a) Expression retargeting from a source identity (top left) to multiple avatars demonstrates precise transfer of facial annd hair details. (b) Zero-shot driving on *unseen subjects* shows accurate preservation of fine details around eyes and mouth regions.

4.3. Evaluation of avatar driving

Driving avatars with diverse inputs. Our universal model demonstrates versatile driving capabilities across different types of inputs as shown in Fig.1. More specifically, in Fig. 9(a), expressions from a source identity (top left) are accurately transferred to multiple personalized avatars, preserving fine details in both wrinkles and hair. This precision stems from our universal layered prior model, where separate decoding of face and hair enables better reconstruction of intricate details.

Testing on zero-shot driving. To further evaluate generalization, we test our model on unseen subjects through zero-shot driving. Fig. 9(b) demonstrates that our model successfully transfers novel expressions to untrained identities while maintaining precise facial features, particularly around the eyes and mouth regions.

5. Conclusion

We present LUCAS, the first universal compositional representation for 3D head avatars that disentangles face and hair components. This separation allows independent deformation, resolving issues like misplaced hair and misaligned dynamics. It also improves the anchor geometry for precise and visually appealing Gaussian ren- derings. Our Universal Layered Prior Model enables effective cross-identity generalization and avatar driving, even for unseen subjects.

Limitation and future work. Our layered approach improves face and hair reconstruction but struggles with extreme hair deformations. Unseen poses during driving can degrade long hair deformation, especially in zero-shot scenarios. Future work will focus on relighting, training with a broader range of hairstyles for a more robust universal prior, and fine-tuning on real-world data to enhance applicability.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010. 3

[2] Timur Bagautdinov, Chenglei Wu, Jason Saragih, Pascal Fua, and Yaser Sheikh. Modeling facial geometry using compositional vaes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3877–3886, 2018. 2

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 1999. 3

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 157–164. 2023. 2

[5] Chen Cao, Tomas Simon, Jin Kyu Kim, Gabriel Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, et al. Authentic volumetric avatars from a phone scan. 2022. 1, 2, 3, 5

[6] Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. Single-view hair modeling for portrait manipulation. *ACM Transactions on Graphics (TOG)*, 31(4): 1–8, 2012. 2

[7] Qi Chang, Zhennan Yan, Mu Zhou, Di Liu, Khalid Sawalha, Meng Ye, Qilong Zhangli, Mikael Kanski, Subhi Al'Aref, Leon Axel, et al. Deeprecon: Joint 2d cardiac segmentation and 3d volume reconstruction via a structure-specific generative method. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 567–577. Springer, 2022. 1

[8] Hang Chu, Shugao Ma, Fernando De la Torre, Sanja Fidler, and Yaser Sheikh. Expressive telepresence via modular codec avatars. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 330–345. Springer, 2020.

[9] Quan Dao, Khanh Doan, Di Liu, Trung Le, and Dimitris Metaxas. Improved training technique for latent consistency models. *arXiv preprint arXiv:2502.01441*, 2025. 1

[10] Yao Feng, Weiyang Liu, Timo Bolkart, Jinlong Yang, Marc Pollefeys, and Michael J Black. Learning disentangled avatars with hybrid 3d representations. *arXiv preprint arXiv:2309.06441*, 2023. 3

[11] Yunhe Gao, Mu Zhou, Di Liu, Zhennan Yan, Shaoting Zhang, and Dimitris N Metaxas. A data-scalable transformer for medical image segmentation: architecture, model efficiency, and benchmark. *arXiv preprint arXiv:2203.00131*, 2022. 1

[12] Yunhe Gao, Zhuowei Li, Di Liu, Mu Zhou, Shaoting Zhang, and Dimitris N Metaxas. Training like a medical resident: Context-prior learning toward universal medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11194–11204, 2024. 1

[13] Zoubin Ghahramani and Geoffrey E. Hinton. The em algorithm for mixtures of factor analyzers. 1996. 3

[14] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 3

[15] Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Anastasis Stathopoulos, Xiaoxiao He, Yuxiao Chen, et al. Proxedit: Improving tuning-free real image editing with proximal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4291–4301, 2024. 1

[16] Xiaoxiao He, Chaowei Tan, Bo Liu, Liping Si, Weiwu Yao, Liang Zhao, Di Liu, Qilong Zhangli, Qi Chang, Kang Li, et al. Dealing with heterogeneous 3d mr knee images: A federated few-shot learning method with dual knowledge distillation. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2023.

[17] Xiaoxiao He, Ligong Han, Quan Dao, Song Wen, Minhao Bai, Di Liu, Han Zhang, Martin Renqiang Min, Felix Juefei-Xu, Chaowei Tan, et al. Dice: Discrete inversion enabling controllable editing for multinomial diffusion and masked generative models. *arXiv preprint arXiv:2410.08207*, 2024. 1

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 5

[19] Taeksoo Kim, Byungjun Kim, Shunsuke Saito, and Hanbyul Joo. Gala: Generating animatable layered assets from a single scan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1535–1545, 2024. 3

[20] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[21] John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)*, 1(8):2, 2014. 2

[22] Junxuan Li, Shunsuke Saito, Tomas Simon, Stephen Lombardi, Hongdong Li, and Jason Saragih. Megane: Morphable eyeglass and avatar network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12769–12779, 2023. 1, 3

[23] Junxuan Li, Chen Cao, Gabriel Schwartz, Rawal Khirodkar, Christian Richardt, Tomas Simon, Yaser Sheikh, and Shunsuke Saito. Uravatar: Universal relightable gaussian codec avatars. *arXiv preprint arXiv:2410.24223*, 2024. 1, 2, 3, 5, 6, 7

[24] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 2

[25] Siyou Lin, Zhe Li, Zhaoqi Su, Zerong Zheng, Hongwen Zhang, and Yebin Liu. Layga: Layered gaussian avatars for animatable clothing transfer. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3

[26] Di Liu, Jiang Liu, Yihao Liu, Ran Tao, Jerry L Prince, and Aaron Carass. Label super resolution for 3d magnetic resonance images using deformable u-net. In *Medical Imaging 2021: Image Processing*, pages 606–611. SPIE, 2021. 1

[27] Di Liu, Zhennan Yan, Qi Chang, Leon Axel, and Dimitris N Metaxas. Refined deep layer aggregation for multi-disease, multi-view & multi-center cardiac mr segmentation. In *Inter-*

national Workshop on Statistical Atlases and Computational Models of the Heart, pages 315–322. Springer, 2021.

[28] Di Liu, Yunhe Gao, Qilong Zhangli, Ligong Han, Xiaoxiao He, Zhaoyang Xia, Song Wen, Qi Chang, Zhennan Yan, Mu Zhou, et al. Transfusion: multi-view divergent fusion for medical image segmentation with transformers. In *International conference on medical image computing and computer-assisted intervention*, pages 485–495. Springer, 2022.

[29] Di Liu, Xiang Yu, Meng Ye, Qilong Zhangli, Zhuowei Li, Zhixing Zhang, and Dimitris N Metaxas. Deformer: Integrating transformers with deformable models for 3d shape abstraction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14236–14246, 2023.

[30] Di Liu, Long Zhao, Qilong Zhangli, Yunhe Gao, Ting Liu, and Dimitris N Metaxas. Deep deformable models: Learning 3d shape abstractions with part consistency. *arXiv preprint arXiv:2309.01035*, 2023.

[31] Di Liu, Qilong Zhangli, Yunhe Gao, and Dimitris Metaxas. Lepard: Learning explicit part discovery for 3d articulated shape reconstruction. *Advances in Neural Information Processing Systems*, 36, 2024.

[32] Di Liu, Bingbing Zhuang, Dimitris N. Metaxas, and Manmohan Chandraker. Instantaneous perception of moving objects in 3d. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024. 1

[33] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (ToG)*, 37(4):1–13, 2018. 1, 2

[34] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 2

[35] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 1, 2

[36] Linjie Luo, Hao Li, Sylvain Paris, Thibaut Weise, Mark Pauly, and Szymon Rusinkiewicz. Multi-view hair capture using orientation fields. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1490–1497. IEEE, 2012. 2

[37] Shugao Ma, Tomas Simon, Jason Saragih, Dawei Wang, Yuecheng Li, Fernando De La Torre, and Yaser Sheikh. Pixel codec avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 64–73, 2021. 1, 2, 3, 5, 6, 7

[38] Carlos Martín-Isla, Víctor M Campello, Cristian Izquierdo, Kaisar Kushibar, Carla Sendra-Balcells, Polyxeni Gkontra, Alireza Sojoudi, Mitchell J Fulton, Tewodros Weldebirhan Arega, Kumaradevan Punithakumar, et al. Deep learning segmentation of the right ventricle in cardiac mri: the m&ms challenge. *IEEE Journal of Biomedical and Health Informatics*, 27(7):3302–3313, 2023. 1

[39] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large steps in inverse rendering of geometry. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 40(6), 2021. 4

[40] Sylvain Paris, Will Chang, Oleg I Kozhushnyan, Wojciech Jarosz, Wojciech Matusik, Matthias Zwicker, and Frédo Durand. Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph.*, 27(3):30, 2008. 2

[41] Stanislav Pidhorskyi, Tomas Simon, Gabriel Schwartz, He Wen, Yaser Sheikh, and Jason Saragih. Rasterized edge gradients: Handling discontinuities differentiably. In *Computer Vision – ECCV 2024*, pages 335–352, Cham, 2025. Springer Nature Switzerland. 5

[42] Amit Raj, Michael Zollhofer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pixel-aligned volumetric avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11733–11742, 2021. 2

[43] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European conference on computer vision (ECCV)*, pages 704–720, 2018. 2

[44] Alexander Richard, Colin Lea, Shugao Ma, Jurgen Gall, Fernando De la Torre, and Yaser Sheikh. Audio-and gaze-driven facial animation of codec avatars. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 41–50, 2021. 1

[45] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2024. 3

[46] Gabriel Schwartz, Shih-En Wei, Te-Li Wang, Stephen Lombardi, Tomas Simon, Jason Saragih, and Yaser Sheikh. The eyes have it: An integrated eye and face model for photorealistic facial animation. *ACM Transactions on Graphics (TOG)*, 39(4):91–1, 2020. 1

[47] Artem Sevastopolsky, Philip-William Grassal, Simon Giebenhain, ShahRukh Athar, Luisa Verdoliva, and Matthias Neissner. Headcraft: Modeling high-detail shape variations for animated 3dmms. 2025. 2

[48] Gil Shamai, Ron Slossberg, and Ron Kimmel. Synthesizing facial photometries and corresponding geometries using generative adversarial networks. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 15(3s):1–24, 2019. 2

[49] Olga Sorkine-Hornung and Marc Alexa. As-rigid-as-possible surface modeling. In *Eurographics Symposium on Geometry Processing*, 2007. 3

[50] Lorenzo Torresani, Aaron Hertzmann, and Christoph Bregler. Learning non-rigid 3d shape from 2d motion. In *Advances in Neural Information Processing Systems*. MIT Press, 2003. 3

[51] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3349–3364, 2019. 3

[52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[53] Ziyan Wang, Giljoo Nam, Tuur Stuyck, Stephen Lombardi, Michael Zollhöfer, Jessica Hodgins, and Christoph Lassner. Hvh: Learning a hybrid neural volumetric representation for dynamic hair performance capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6143–6154, 2022. 2

[54] Ziyan Wang, Giljoo Nam, Tuur Stuyck, Stephen Lombardi, Chen Cao, Jason Saragih, Michael Zollhöfer, Jessica Hodgins, and Christoph Lassner. Neuwigs: A neural dynamic model for volumetric hair capture and animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8641–8651, 2023. 2

[55] Shih-En Wei, Jason Saragih, Tomas Simon, Adam W Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernan Badino, and Yaser Sheikh. Vr facial animation via multiview image translation. *ACM Transactions on Graphics (ToG)*, 38(4):1–16, 2019. 1

[56] Keyu Wu, Lingchen Yang, Zhiyi Kuang, Yao Feng, Xutao Han, Yuefan Shen, Hongbo Fu, Kun Zhou, and Youyi Zheng. Monohair: High-fidelity hair modeling from a monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24164–24173, 2024. 2

[57] Yuelang Xu, Hongwen Zhang, Lizhen Wang, Xiaochen Zhao, Han Huang, Guojun Qi, and Yebin Liu. Latentavatar: Learning latent expression code for expressive neural head avatar. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 2

[58] Hao Zhang, Yao Feng, Peter Kulits, Yandong Wen, Justus Thies, and Michael J Black. Teca: Text-guided generation and editing of compositional 3d avatars. In *2024 International Conference on 3D Vision (3DV)*, pages 1520–1530. IEEE, 2024. 3

[59] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[60] Qilong Zhangli, Jingru Yi, Di Liu, Xiaoxiao He, Zhaoyang Xia, Qi Chang, Ligong Han, Yunhe Gao, Song Wen, Haiming Tang, et al. Region proposal rectification towards robust instance segmentation of biological images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 129–139. Springer, 2022. 1

[61] Qilong Zhangli, Jindong Jiang, Di Liu, Licheng Yu, Xiaoliang Dai, Ankit Ramchandani, Guan Pang, Dimitris N Metaxas, and Praveen Krishnan. Layout-agnostic scene text image synthesis with diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7496–7506. IEEE Computer Society, 2024. 1

[62] Yujian Zheng, Zirong Jin, Moran Li, Haibin Huang, Chongyang Ma, Shuguang Cui, and Xiaoguang Han. Hairstep: Transfer synthetic to real using strand and depth maps for single-view 3d hair modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12726–12735, 2023. 2

[63] Yi Zhou, Liwen Hu, Jun Xing, Weikai Chen, Han-Wei Kung, Xin Tong, and Hao Li. Hairnet: Single-view hair reconstruction using convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 235–251, 2018. 2

[64] Yuxiang Zhou, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Dense 3d face decoding over 2500fps: Joint texture & shape convolutional mesh decoders. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1097–1106, 2019. 2