# GAUSSIAN HEAD & SHOULDERS: HIGH FIDELITY NEURAL UPPER BODY AVATARS WITH ANCHOR GAUSSIAN GUIDED TEXTURE WARPING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The ability to reconstruct realistic and controllable upper body avatars from casual monocular videos is critical for various applications in communication and entertainment. By equipping the most recent 3D Gaussian Splatting representation with head 3D morphable models (3DMM), existing methods manage to create head avatars with high fidelity. However, most existing methods only reconstruct a head without the body, substantially limiting their application scenarios. We found that naively applying Gaussians to model the clothed chest and shoulders tends to result in blurry reconstruction and noisy floaters under novel poses. This is because of the fundamental limitation of Gaussians and point clouds – each Gaussian or point can only have a single directional radiance without spatial variance, therefore an unnecessarily large number of them is required to represent complicated spatially varying texture, even for simple geometry. In contrast, we propose to model the body part with a neural texture that consists of coarse and pose-dependent fine colors. To properly render the body texture for each view and pose without accurate geometry nor UV mapping, we optimize another sparse set of Gaussians as anchors that constrain the neural warping field that maps image plane coordinates to the texture space. We demonstrate that Gaussian Head & Shoulders can fit the high-frequency details on the clothed upper body with high fidelity and potentially improve the accuracy and fidelity of the head region. We evaluate our method with casual phone-captured and internet videos and show our method archives superior reconstruction quality and robustness in both self and cross reenactment tasks. To fully utilize the efficient rendering speed of Gaussian splatting, we additionally propose an accelerated inference method of our trained model without Multi-Layer Perceptron (MLP) queries and reach a stable rendering speed of around 130 FPS for any subjects.

## 1 INTRODUCTION

Personalized and controllable 3D head avatar is a crucial asset for interactive Mixed Reality and metaverse applications. Recent developments in the 3D representations such as 3DMM (Li et al., 2017; Gerig et al., 2017), Neural Radiance Field (Mildenhall et al., 2020), Instant Neural Primitives (Müller et al., 2022), and other implicit representations (Mescheder et al., 2019) have brought rapid advancements in the reconstruction of vivid and controllable neural avatars (Zheng et al., 2022; Grassal et al., 2021; Zielonka et al., 2022; Gao et al., 2022). With the most recent 3D Gaussian Splatting representation (Kerbl et al., 2023), neural avatars can be convincingly reconstructed from a monocular video with impressive fidelity. However, most current methods for creating head avatars concentrate solely on the face and head, discarding other visible parts of the body by using a semantic mask during the training process. Consequently, this results in avatars that appear as heads without bodies, which is not sufficient for many immersive applications, including video conferencing, where a more complete avatar is needed (Shao et al., 2024; Xiang et al., 2024; Zielonka et al., 2022; Gao et al., 2022). Recent techniques aim to create more complete avatars by including visible parts of the body, like shoulders and chest (Zheng et al., 2023; Zhao et al., 2024; Wang et al., 2024; Zheng et al., 2022). However, they are limited to simplified settings where the subject dresses in plain clothing without detailed textures and is instructed to restrict upper body movement. On the other hand, existing full-body avatar methods typically focus on the overall quality of the limbs and torso and fail to faithfully capture the fine details such as high-frequency texture on clothes (Kocabas et al., 2023; Hu et al., 2024b; Li et al., 2024; Lei et al., 2023). Applications of neural avatars that require detailed reconstruction of the upper body area often encounter significant challenges in capturing faithful and intricate details. Overall, current methods still fall short of delivering the level of performance needed for practical, real-world use.

The Gaussian Splatting representation, while being efficient and effective in certain aspects, struggles with accurate modeling of clothed upper bodies. As one of its fundamental limitations, each Gaussian can represent only one color from a specific viewing angle. This heavily limits its capability to handle dynamic objects that have complex textures, such as clothing with intricate patterns. To capture the detailed appearance of such objects, an excessively large number of Gaussians would be needed, increasing memory requirement and slowing down the rendering speed. In addition, the complicated pose-dependent appearances such as brightness changes and cloth wrinkles further increase the difficulty of modeling them with plain Gaussians alone. As a result, when the reconstructed avatar is driven to novel poses, the Gaussians tend to produce several undesirable artifacts such as blurred texture, incorrect colors or floating ellipsoid; see Fig 1.

To address the limitations of existing Gaussian-based avatar methods on clothed upper-body, we argue that the chest and shoulders are expected to have relatively simpler geometry and more intricate deformation compared to the head. Therefore, modeling them with regular and 3DMM-driven Gaussians would be unsuitable and is an over-complication of the problem. Instead, a more appropriate and standard approach would be representing their appearance with a high-frequency texture.

In a traditional texture-based rendering pipeline, the texture is first mapped to mesh geometry in the 3D world space via UV mapping, and then rasterized to the 2D image plane in the view space to obtain the pixel color. However, this approach requires a well-defined UV mapping and accurate mesh geometry, which is challenging to obtain from monocular videos alone due to the lack of multi-view correspondences. Besides, compared to faces that share more common characteristics and stronger priors, the appearance of upper bodies can vary dramatically depending on the texture and tightness of the clothes and they hence contain fewer detectable landmarks. Consequently, body 3DMMs such as SMPL (Loper et al., 2015) fail to provide geometry accurate enough for this purpose.

Hence, we propose to bypass the mapping from texture space to world space, and instead use a sparse set of Gaussians as "anchors" to define a direct neural warping field from a canonical 2D texture space, which consists of a coarse RGB texture and a fine neural texture, to the image plane. As the tracking of body 3DMM tends to be inaccurate due to the lack of landmarks, we only transform anchor Gaussians together with the head Gaussians via a head FLAME 3DMM (Li et al., 2017) through Linear Blend Skinning (LBS). The transformed anchor Gaussians are used as soft constraints of the texture warping represented by a coordinate-based MLP, which is optimized together with the neural texture, regular Gaussians, and the anchor Gaussians. As the resolution of the neural texture is not limited by the number of Gaussians or the density control scheme, we can easily learn the high-frequency textures with sharp details on the clothes and avoid the common artifacts exhibited in Gaussian rendering under novel poses; see Fig 1.

To maintain a competitive rendering speed with Gaussian Splatting and enable real-time interactive applications, we additionally propose a method to remove the neural warping field and neural texture in the model and allow inference of reconstructed avatars at novel poses without any MLP queries. This accelerated inference effectively increases the rendering speed from 70 FPS to around 130 FPS, which surpasses the rendering speed of plain Gaussian Splatting avatars for subjects with high-frequency clothes.

We evaluate the proposed method with various casual monocular videos collected using smartphones or from the Internet. Compared to state-of-the-art methods which incorporate different representations including neural radiance field, Gaussian Splatting, and point clouds, we show that our approach achieves better performance and robustness for both self-reenactment and cross-reenactment tasks. In summary, our contributions are:

- We propose a novel approach that maps intricate texture to the image plane via a sparse set of anchor Gaussians driven by LBS with 3DMM. This allows accurate and robust modeling of high-fidelity clothed chest and shoulders with less number of Gaussians.

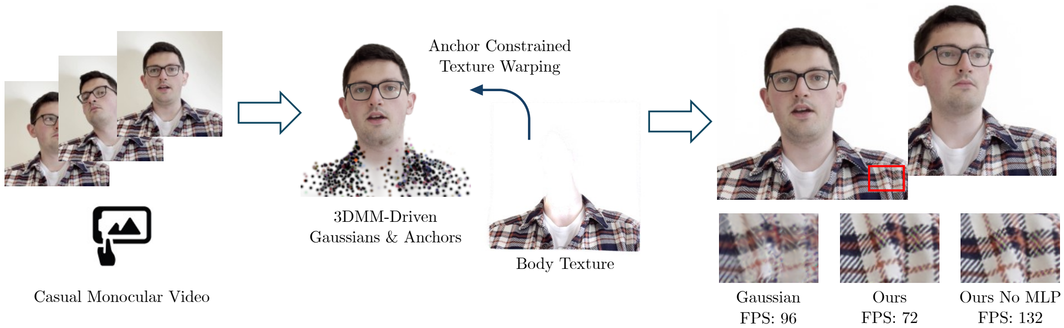


Figure 1: **Gaussian Head & Shoulders** reconstructs 3DMM-driven upper body avatars from casual monocular videos. By utilizing a high-frequency body neural texture which is warped using a neural texture warping field constrained by a set of sparse anchor Gaussians, we can learn sharp details of the cloth texture with highly efficient rendering speed.

- We propose a method to remove the MLP in our method at inference time to prevent any costly queries when rendering with novel poses and expressions and reach a rendering speed of around 130 FPS.

## 2 Related Works

**Neural Head Avatars** The recent advancement in neural 3D implicit and explicit representations has sparked a surge of methodologies within the field of controllable 3D head avatars. Among these approaches, a prominent family of methods involves the reconstruction of a 5D neural radiance field, manifested through various forms such as pure MLP (Gafni et al., 2021; Wang et al., 2021; Kirschstein et al., 2023), hash grid latents (Xu et al., 2023; Gao et al., 2022; Zielonka et al., 2022; Xu et al., 2023; Dhamo et al., 2023; Xiang et al., 2024; Saito et al., 2024; Chen et al., 2023), and 3D Gaussians (Wang et al., 2024; Zhao et al., 2024). Another set of methods utilizes more explicit representations such as deformable meshes with neural textures (Grassal et al., 2021; Zheng et al., 2022; Buehler et al., 2021; Gropp et al., 2020; Khakhulin et al., 2022; Kim et al., 2018) and point clouds (Zheng et al., 2023). With the most recent Gaussian Splatting techniques, the head avatars reconstructed from monocular videos have already reached high fidelities. However, many methods simplify the problem by reconstructing only the head and neck part, resulting in a head-only reconstruction that is not suitable for many applications. Several methods have attempted to also model the chest and shoulders to provide a more immersive user experience (Zheng et al., 2022; Zhao et al., 2024; Wang et al., 2024; Zheng et al., 2023), but they are limited to simple clothes with plain colors, and cannot handle the movements in the upper body in the video.

**Neural Full-Body Avatars** Several works have tried to reconstruct a controllable full-body neural avatar from multi-view or monocular videos (Liu et al., 2024; Shao et al., 2024; Svitov et al., 2024; Li et al., 2024; Lei et al., 2023; Kocabas et al., 2023; Hu et al., 2024b; Jiang et al., 2022). Due to the highly articulated nature of human bodies, they tightly rely on body 3DMMs to deform the neural body representation via LBS and hence fail to faithfully capture subjects with complicated or loose clothing as those cannot be modeled with existing body 3DMMs. Besides, they typically focus on the overall quality of the torso and limbs, and hence tend to present non-trivial artifacts when reconstructing and re-animating an avatar that has a tight focus around the head and shoulder regions.

## 3 Method

Given a monocular video featuring a talking subject with various expressions and head poses, our goal is to reconstruct a high-fidelity and animatable avatar including the head and clothed upper body. As illustrated in Fig 2, our method jointly optimizes 1) a set of standard 3D Gaussians (Kerbl et al., 2023) which tightly follow the transformation of 3DMM via LBS to represent the head region, 2) a set of sparse anchor Gaussians spawning over the clothed body, and 3) a learnable neural texture with pose-dependent neural texture warping field constrained by the anchor Gaussians to represent the clothed body with sharp details and high robustness.

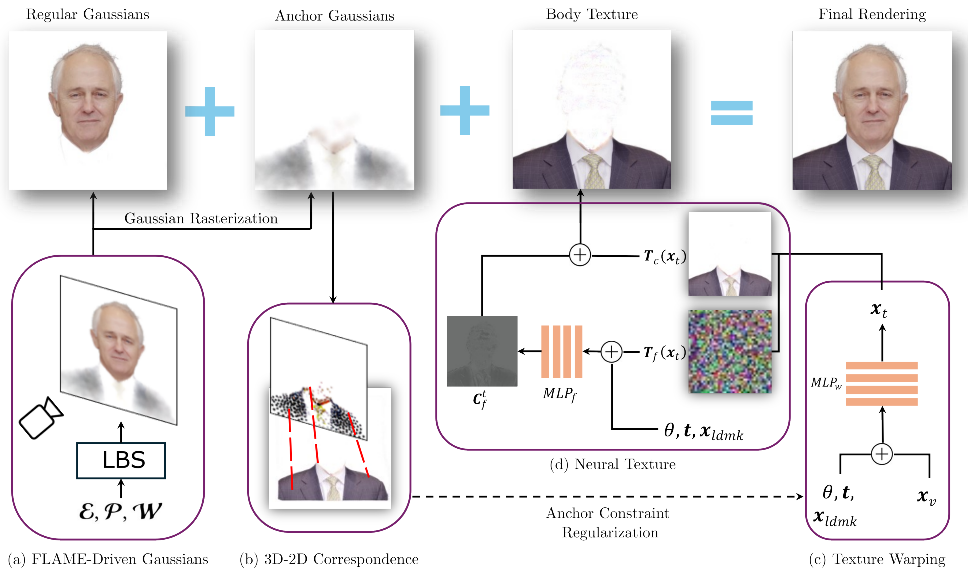


Figure 2: **Method.** (a) We utilize a set of standard head Gaussians and anchor Gaussians driven by LBS with the FLAME model. (b) Anchor Gaussians are initialized with a set of corresponding target coordinates in the texture space. This 3D-2D correspondence is used to constrain (c) a neural texture warping field that maps each pixel on the image plane $\mathbf{x}_v$ to a pixel in the texture space $\mathbf{x}_t$. (d) We then sample in the texture space to fetch the coarse texture $\mathbf{T}_c$ and latent texture $\mathbf{T}_f$, which is parsed by an MLP to obtain pose-dependent fine texture $\mathbf{C}_f^t$. Both coarse and fine textures are then combined to form a body texture, which is blended with other Gaussians through alpha compositing to form the final rendering.

### 3.1 Preliminary- Gaussian Splatting

3D Gaussian Splatting is a volumetric representation that utilizes a dense set of anisotropic Gaussians with varying opacity and view-dependent radiance to represent 3D geometry and appearance. Each Gaussian is described with four parameters: position (Gaussian mean) $\boldsymbol{\mu}$, 3D covariance matrix $\boldsymbol{\Sigma}$, opacity $\alpha$ and Spherical Harmonic (SH) coefficients $\mathbf{SH}$ for computing view-dependent RGB color. For ease of optimization, the covariance matrix is further decomposed into a scaling matrix $\mathbf{S}$, stored as a scaling vector $\mathbf{s}$, and a rotation matrix $\mathbf{R}$, stored as a quaternion vector $\mathbf{q}$. The covariance matrix is obtained as: $\boldsymbol{\Sigma} = \mathbf{RSS}^T\mathbf{R}^T$.

To render 3D Gaussians to RGB images, their means are projected onto 2D image plane with standard projective transformation, while the projected covariance matrix is obtained by $\boldsymbol{\Sigma}' = \boldsymbol{JW\Sigma W}^T\boldsymbol{J}^T$, where $\mathbf{W}$ is the world to camera transformation and $\mathbf{J}$ is the Jacobian approximating the projective transformation (Zwicker et al., 2001). The rendered RGB color at each pixel is then obtained through:

$$\mathbf{C}(\mathbf{x}) = \sum_{i \in N} \mathbf{c}_i \alpha_i^*(\mathbf{x}) \prod_{j=1}^{i-1} (1 - \alpha_j^*(\mathbf{x})), \tag{1}$$

$$\alpha_i^*(\mathbf{x}) = \alpha_i \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i')^T \boldsymbol{\Sigma}'^{-1} (\mathbf{x} - \boldsymbol{\mu}_i')\right), \tag{2}$$

where $\mathbf{x}$ is the 2D pixel coordinate, $\mathbf{c}_i$ is the view-dependent RGB radiance of i-th Gaussian on the ray obtained from SH function, $\alpha_i$ and $\boldsymbol{\mu}_i'$ are the opacity and projected 2D mean of the i-th Gaussian respectively.

### 3.2 FLAME-Driven Head Gaussians

As the face region contains highly distinguishable characteristics and can be described accurately with parametric head 3DMM such as FLAME (Li et al., 2017), we directly utilize standard 3D Gaussians that are deformed with parametric 3DMM via neural LBS to represent the head part (Zheng et al., 2023; Zhao et al., 2024). Specifically, we learn personalized FLAME expression and pose blendshapes and LBS weights through a small 3D coordinate-based MLP for each Gaussian: $\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{W}} = \mathrm{MLP}_d(\boldsymbol{\mu})$, where $\boldsymbol{\mathcal{E}} \in \mathbb{R}^{n_e \times 3}$ are the expression blendshapes, $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{n_p \times 9 \times 3}$ are the pose blendshapes, $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{n_j}$ are the LBS weights corresponding to each of the $n_j$ bones. Following (Hu & Liu, 2023), we use the standard skinning function LBS to obtain the rotation $\boldsymbol{R}$ and translation $\boldsymbol{T}$ for each Gaussian, and apply them to get the Gaussian mean $\boldsymbol{\mu}^d$ and covariance $\boldsymbol{\Sigma}^d$ in the 3D view space:

$$\boldsymbol{R}, \boldsymbol{T} = \mathrm{LBS}(\boldsymbol{B}_{\boldsymbol{\mathcal{P}}}(\theta; \boldsymbol{\mathcal{P}}) + \boldsymbol{B}_{\boldsymbol{\mathcal{E}}}(\psi; \boldsymbol{\mathcal{E}}), \mathbf{J}(\psi), \theta, \boldsymbol{\mathcal{W}}), \tag{3}$$

$$\boldsymbol{\mu}^d = \boldsymbol{R}\boldsymbol{\mu} + \boldsymbol{T},\ \boldsymbol{\Sigma}^d = \boldsymbol{R}\boldsymbol{\Sigma}\boldsymbol{R}^T, \tag{4}$$

where $\mathbf{J}$ is the joint regressor in FLAME, and $\boldsymbol{B}_{\boldsymbol{\mathcal{P}}}$ and $\boldsymbol{B}_{\boldsymbol{\mathcal{E}}}$ are linear combination of blendshapes based on per-frame coefficients $\theta$ and $\psi$ that control the head animation. They can then be rendered with a standard Gaussian rasterization pipeline in Eq 1.

### 3.3 3D-2D Correspondence via Anchor Gaussians

3D Gaussian Splatting has shown promising performance and robustness in reconstructing 3D geometry and appearance from RGB images. However, they suffer from a significant constraint – each individual Gaussian can only represent a spatially invariant color under a fixed viewing direction, hence a vast number of Gaussians is required to represent objects with detailed textures, regardless of the actual complexity of the geometry. A naive application of Gaussian Splatting therefore fails to capture the fine details of the upper body with complex textures and intricate deformation, and results in blurry details and floating artifacts under challenging poses.

We hence propose to learn a high-frequency texture in canonical texture space, and use a sparse set of Gaussians as anchors to guide the warping between texture space and image plane. As such, we only need a small number of Gaussians and a texture with per-pose warping to represent a clothed body with arbitrarily complicated textures. Since anchor Gaussians themselves do not need to exactly represent the high-frequency appearance, we can model them as a simplified version of regular Gaussians: they only use view-independent RGB colors, are isotropic Gaussians with quaternion fixed at $(1, 0, 0, 0)$, and are excluded from the density control and therefore are not split, cloned, or pruned. To prevent them from becoming trivial in rendering, their opacity and size are clamped to be no smaller than $0.05$ and $0.0001$ respectively.

The anchor Gaussians are initialized as follows: after a short warm-up period that only trains plain Gaussian, we first reproject all Gaussian means onto the image plane of a canonical training frame, and filter out Gaussians that are located around the head region based on semantic masks. We then use farthest point sampling (Qi et al., 2017) to select $N_a = 1024$ Gaussians as anchor Gaussians. The first SH basis is converted to RGB values and the anchor scales in three directions are averaged to form a single scale for the anchor Gaussians. We then obtain a sparse set of anchor Gaussians, as well as their projected 2D means $\hat{\mathbf{x}}_i^v$ on the image plane (2D view space) of the canonical frame:

$$\hat{\mathbf{x}}_i^v = \mathbf{P}(\hat{\boldsymbol{\mu}}_i^d), \tag{5}$$

where $\mathbf{P}$ is the camera projective transformation, $\hat{\boldsymbol{\mu}}_i^d$ is the 3D Gaussian mean of the $i$-th anchor Gaussian transformed to 3D view space with LBS. To build the correspondence between anchor Gaussians and texture space coordinates, we assume that the mapping between the 2D image plane of the canonical frame and the texture space is an identity mapping. We can hence define a function $f_{anchor}(i)$ as a fixed correspondence between the $i$-th 3D anchor Gaussian mean and its target 2D pixel coordinate in texture space:

$$f_{anchor}(i) := \mathbf{I}(\hat{\mathbf{x}}_i^v), \tag{6}$$

where $\mathbf{I}$ is the identity function to map 2D image plane coordinates to texture space. Note that $f_{anchor}(i)$ is fixed after initialization and does not update with further optimization of $\hat{\boldsymbol{\mu}}_i$. Such correspondences will later be used to constrain the pose-dependent texture warping, as will be detailed in Sec 3.6.

### 3.4 Neural Texture and Texture Warping

We use a trainable neural texture in canonical space with a pose-dependent neural texture warping field to represent the part of the avatar with relatively simple overall geometry and complicated appearances, i.e., the clothed shoulder and chest. In a traditional textured mesh rendering pipeline, the texture is first mapped to the mesh triangles through a pre-defined UV mapping, and the meshes are then rasterized to find the first intersections with the camera rays. Those first intersections therefore establish a mapping between texture space and image plane. However, this approach is not applicable without accurate surfaces and well-defined UV mapping. We instead propose to bypass the intermediate step and learn a per-pose warping that directly maps pixel coordinates on image plane $\mathbf{x}_v$ to the texture coordinates $\mathbf{x}_t$ for texture fetching. Specifically, the warping field is represented using a coordinate-based MLP:

$$\Delta_{\mathbf{x}} = \mathrm{MLP}_w\left(\gamma(\mathbf{x}_v), \gamma(\theta), \gamma(\mathbf{t}), \gamma(\mathbf{x}_{ldmk})\right), \tag{7}$$

where $\gamma$ is the positional encoding (Mildenhall et al., 2020), $\theta$ is the FLAME pose parameters containing head and neck rotations, $\mathbf{t}$ is the camera position, $\mathbf{x}_{ldmk}$ is 2D body landmarks for neck, left and right shoulders. The corresponding texture coordinate is obtained as $\mathbf{x}_t = \mathbf{x}_v + \Delta_{\mathbf{x}}$.

Our optimizable texture includes a coarse texture $\mathbf{T}_c$ with 3 channels and a latent texture $\mathbf{T}_f$ with $D_t$ channels. Both textures have sizes of $[H + 2P, W + 2P]$, where $H, W$ are the image height and width, $P$ is the padding size which we empirically set to 50 to account for body parts that move in and out in the video sequence. The latent texture $\mathbf{T}_f$ is passed to an MLP to obtain pose-dependent appearances such as brightness changes on the clothes:

$$\mathbf{C}^t_f(\mathbf{x}_t) = \mathrm{MLP}_f\left(\mathbf{T}_f(\mathbf{x}_t), \gamma(\theta), \gamma(\mathbf{t}), \gamma(\mathbf{x}_{ldmk})\right), \tag{8}$$

where $\mathbf{T}_c(\mathbf{x}_t), \mathbf{T}_f(\mathbf{x}_t)$ are coarse and latent texture sampled at 2D coordinate $\mathbf{x}_t$ via bilinear interpolation. The textured pixel color at the coordinate $\mathbf{x}_v$ is therefore obtained as $\mathbf{C}^t(\mathbf{x}_v) = \mathbf{T}_c(\mathbf{x}_t) + \mathbf{C}^t_f(\mathbf{x}_t)$.

By constraining with the correspondences between deformable anchor Gaussians and their fixed projections on 2D texture space, we can learn accurate and effective texture warping for various body movements including translation, rotation, and depth-based (in-and-out) motions; see Fig 3.

### 3.5 Rendering

To this end, we have a hybrid representation that includes 3D regular Gaussians that represent the head of the avatar, 3D anchor Gaussians that sparsely span over the body region, and a 2D neural texture for the body. To render all of them together for joint optimization, we simply use alpha blending:

$$\mathbf{C}^*(\mathbf{x}_v) = \underbrace{\hat{\mathbf{C}}(\mathbf{x}_v)}_{\text{Anchor Gaussians}} + \underbrace{(1 - \hat{\alpha}(\mathbf{x}_v))\mathbf{C}(\mathbf{x}_v)}_{\text{Head Gaussians}} + \underbrace{(1 - \hat{\alpha}(\mathbf{x}_v))(1 - \alpha(\mathbf{x}_v))\mathbf{C}^t(\mathbf{x}_v)}_{\text{Body Texture}}, \tag{9}$$

where $\hat{\mathbf{C}}(\mathbf{x}_v), \mathbf{C}(\mathbf{x}_v)$ are the rendered RGB color of anchor Gaussian and regular Gaussian, $\hat{\alpha}(\mathbf{x}_v), \alpha(\mathbf{x}_v)$ are the total alpha of anchor Gaussian and regular Gaussian at pixel $\mathbf{x}_v$ respectively.

Note that our rendering process always renders anchor Gaussians in front of the regular Gaussians regardless of their actual positions. Though not physically realistic, we designed this rendering order so anchor Gaussians are always non-trivial and never occluded by regular Gaussians.

### 3.6 Optimization

The optimization is split into three different stages: anchor warm-up stage, main optimization stage, and texture refinement stage. In the anchor warm-up stage, neither anchor Gaussians nor body texture is applied, only the regular Gaussians are rendered and optimized. The purpose of this stage is to move Gaussians to roughly spawn over the area of interest including both head and body. At the end of this stage, we initialize anchor Gaussians from regular Gaussians using the method described in Sec 3.3. In the second stage, we render all of the regular Gaussians, anchor Gaussians, and the textured body with alpha compositing described in Eq 9 and jointly optimize them together. In the last stage, to recover faithful appearance for the body texture and enhance its robustness under novel poses, we remove anchor Gaussians from the rendering pipeline, i.e., we set $\hat{\mathbf{C}}$ and $\hat{\alpha}$ to 0 in Eq 9., and freeze everything else except for the neural texture, texture warping field, and opacity and SH of regular Gaussians.

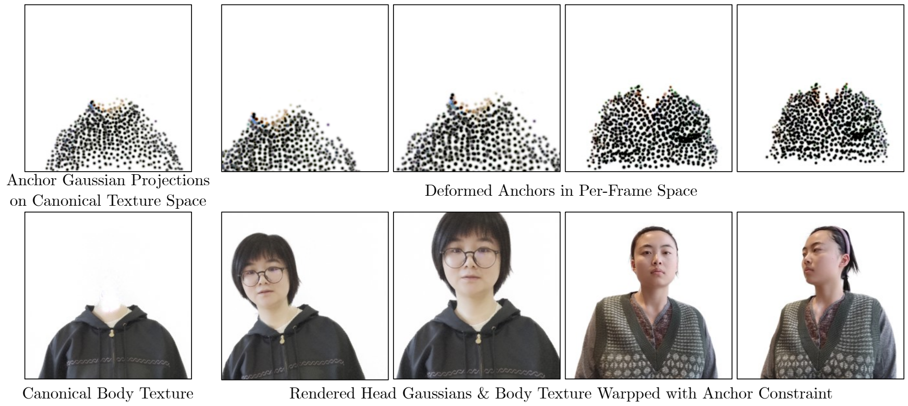


Figure 3: **Anchor Warping.** The anchors are initialized with corresponding projections on canonical texture space. When anchors are deformed via LBS to model the per-frame body movement, they map to the same projections in texture space and hence establish correspondences for body texture warping.

Following (Zheng et al., 2023; 2022), the training losses include standard MSE RGB loss $\mathcal{L}_{\boldsymbol{C}} = MSE(\mathbf{C}^* - \mathbf{C}^{GT})$, and a FLAME regularization that encourages the FLAME blendshapes and LBS weights predicted for each Gaussian stay close to the pseudo ground truth $\tilde{\boldsymbol{\mathcal{E}}}_i, \tilde{\boldsymbol{\mathcal{P}}}_i, \tilde{\boldsymbol{\mathcal{W}}}_i$ obtained from the nearest FLAME vertex:

$$\mathcal{L}_{flame} = \frac{1}{N} \sum_{i=1}^{N+N_a} (\lambda_{\boldsymbol{\mathcal{E}}} |\boldsymbol{\mathcal{E}}_i - \tilde{\boldsymbol{\mathcal{E}}}_i|_2 + \lambda_{\boldsymbol{\mathcal{P}}} |\boldsymbol{\mathcal{P}}_i - \tilde{\boldsymbol{\mathcal{P}}}_i|_2 + \lambda_{\boldsymbol{\mathcal{W}}} |\boldsymbol{\mathcal{W}}_i - \tilde{\boldsymbol{\mathcal{W}}}_i|_2). \tag{10}$$

During main optimization stage, we additionally include a VGG feature loss (Johnson et al., 2016; Simonyan & Zisserman, 2015) $\mathcal{L}_{VGG} = |\mathbf{F}_{vgg}(\mathbf{C}) - \mathbf{F}_{vgg}(\mathbf{C}^{GT})|$, and a head mask regularization to encourage regular Gaussians to stay only within the head region and allow the body texture to be trained properly without being occluded:

$$\mathcal{L}_{head} = MSE(max(\alpha - \alpha_{head}, 0)), \tag{11}$$

where $\alpha_{head}$ is the alpha mask of the head region obtained with matting pre-processing and semantic mask. We also include an L1 regularization on the 2D neural warping field to encourage a clean background to be learned in the neural texture, as well as an L1 loss to slowly decrease the opacity of anchor Gaussians to allow the body texture to be trained properly:

$$\mathcal{L}_{warp} = \frac{1}{HW} \sum_{i=1}^{HW} |\Delta_{\mathbf{x}_i}|, \; \mathcal{L}_{\hat{\alpha}} = \frac{1}{N_a} \sum_{i=1}^{N_a} |\hat{\alpha}_i|. \tag{12}$$

Finally, we include an anchor loss as a soft constraint of the per-pose texture warping:

$$\mathcal{L}_{anchor} = \frac{1}{N_a} \sum_{i=1}^{N_a} (f_{anchor}(i) - (\hat{\boldsymbol{x}}_i^v + \Delta_{\hat{\boldsymbol{x}}_i^v}))^2, \tag{13}$$

i.e., for each anchor Gaussian, we first transform it to 3D view space via LBS, and then project it onto the image plane to obtain its 2D mean $\hat{\boldsymbol{x}}_i^v$ via Eq 5. $\hat{\boldsymbol{x}}_i^v$ is then warped by the neural warping field $\mathrm{MLP}_w$ to obtain the corresponding coordinate in the texture space, which is optimized to match the anchor correspondence defined during initialization.

In the third stage, we remove the regularization losses including $\mathcal{L}_{head}, \mathcal{L}_{warp}$ and $\mathcal{L}_{\hat{\alpha}}$.

The total training objectives for each of the three stages are as follows:

$$\mathcal{L}_1 = \mathcal{L}_{\boldsymbol{C}} + \mathcal{L}_{flame}, \tag{14}$$

$$\begin{aligned} \mathcal{L}_2 &= \mathcal{L}_1 + \lambda_{VGG}\mathcal{L}_{VGG} + \lambda_{head}\mathcal{L}_{head} + \lambda_{warp}\mathcal{L}_{warp} + \lambda_{\hat{\alpha}}\mathcal{L}_{\hat{\alpha}} + \lambda_{anchor}\mathcal{L}_{anchor}, \\ \mathcal{L}_3 &= \mathcal{L}_1 + \lambda_{VGG}\mathcal{L}_{VGG} + \lambda_{anchor}\mathcal{L}_{anchor}. \end{aligned} \tag{15}$$

### 3.7 Accelerated Rendering with No MLP Queries

One of the main advantages of Gaussian Splatting is its highly efficient rendering speed, which enables many real-time and interactive applications. To take full use of this advantage, we propose an accelerated version of our method that requires no MLP queries at inference time. Specifically, after training the model, we first cache the output of $\mathrm{MLP}_d$ for all head Gaussians and anchor Gaussians, then cache the view-dependent fine texture by querying the fine texture MLP $\mathrm{MLP}_f$ conditioned on the same canonical training frame which was previously used to initialize the anchor Gaussians. The queried fine texture colors are added to the coarse color to make a non-neural RGB texture. To deal with potential noise created by the fine texture MLP at the corners of the texture, we use an off-the-shelf background segmentation network (Chen et al., 2017) to compute a coarse mask and clean all the pixels outside of the mask; we show the necessity of this step in the supplementary. To replace the neural warping field $\mathrm{MLP}_w$ that warps image plane coordinates to texture space, we rely on the correspondence between anchor Gaussians and texture space coordinates to estimate a homography at inference time. Specifically, we first project all anchor Gaussians to the image plane of the canonical training frame, and then remove any anchor Gausians that go beyond the view frustum. To deal with any potential discrepancy between the neural warping field and the anchor correspondences, we update those correspondences based on the prediction of the neural warping field on the current frame:

$$f_{anchor}(i) := \hat{\mathbf{x}}_v^i + \Delta_{\hat{\mathbf{x}}_v^i}. \tag{16}$$

After that, we randomly select 100 training frames and use RANSAC (Fischler & Bolles, 1981) to estimate a homography between the image plane coordinates of anchor Gaussians and their corresponding texture space coordinates, and remove anchor Gaussians that are considered outliers by RANSAC. This effectively removes any anchor deformation that cannot be described by the rigid transformation. Finally, at inference time, we perform LBS on regular head Gaussians and anchor Gaussians. Based on the image plane coordinates of the anchor Gaussians $\hat{\mathbf{x}}_v^i$ and their correspondences $f_{anchor}$, we compute a homography with the least square error via singular value decomposition. The estimated transformation is applied to all pixels on the image plane to find the corresponding non-neural texture, which is then blended with the head Gaussians to form the final rendering. This accelerated inference approach effectively increases the rendering speed from around 70 FPS to 130 FPS.

## 4 Evaluation

**Datasets** We evaluate different methods on 1 mobile phone sequence from PointAvatar (Zheng et al., 2023), 2 internet sequences from Head2Head dataset (Koujan et al., 2020), and 4 sequences captured with mobile phones. All sequences are preprocessed with DECA (Feng et al., 2021) and a slightly modified landmark fitting process from IMAvatar (Zheng et al., 2022). Additionally, we use DWPose (Yang et al., 2023) to predict 2D landmarks for nose, neck and shoulders, which are then smoothed with One Euro Filter (Casiez et al., 2012).

**Baselines** We compare our method with four neural head avatar methods based on various representations, including (1) INSTA (Zielonka et al., 2022), which employs a latent hash grid (Müller et al., 2022) combined with NeRF (Mildenhall et al., 2020), (2) PointAvatar (Zheng et al., 2023), which is based on isotropic point clouds, (3) SplattingAvatar (Shao et al., 2024), which utilizes Gaussian Splatting attached to local space of 3DMM meshes, and (4) GS*, a baseline we implemented by changing the point cloud representation in PointAvatar to Gaussian Splatting, which is similarly deformed via neural LBS.

**Self-Reenactment** We show the quantitative and qualitative results of the self-reenactment task in Tab 1 and Fig 4. Our full version demonstrates superior reconstruction performance compared to existing baselines, especially for subjects with intricate cloth textures. Our No MLP version does not consistently achieve better PSNR when compared to existing baselines, as it is unable to render pose-dependent appearance changes and intricate cloth deformation. However, we note that it consistently achieves better LPIPS, demonstrating that our No MLP version can still generate realistic and faithful renderings. This discrepancy among different metrics arises because of the high sensitivity of PSNR to small misalignments in the cloth texture (Park et al., 2021). As a result, PSNR

| | full | | | head | | |
|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| INSTA | 20.59 | .781 | .236 | 29.80 | .936 | .059 |
| SplattingAvatar | 21.00 | .774 | .274 | 28.29 | .925 | .069 |
| PointAvatar | 25.21 | .851 | .102 | 30.64 | .943 | .042 |
| FlashAvatar | 21.88 | .808 | .127 | 30.33 | .946 | .039 |
| GS* | 25.94 | .854 | .095 | 31.81 | .947 | .036 |
| Ours | 26.80 | .875 | .070 | 33.06 | .959 | .028 |
| Ours (No MLP) | 25.47 | .859 | .074 | 32.54 | .956 | .029 |

Table 1: **Quatitative evaluation of self-reenactment task** We color the best and second-best methods. Our full method achieves much better performance compared to existing baselines. While Ours (No MLP) achieves slightly lower PSNR, which is known to be over-sensitive to small misalignments and prefers blurry results (Park et al., 2021), we show it achieves better LPIPS than existing methods.

| | FPS | #GS | FPS | #GS |
|---|---|---|---|---|
| | 003 | | 004 | |
| FlashAvatar | 134 | 13453 | 137 | 13453 |
| GS* | **141** | 163830 | **159** | 125521 |
| Ours | 70 | 58701 | 71 | 39549 |
| Ours (No MLP) | 129 | 58701 | 134 | 39549 |
| | 005 | | 007 | |
| FlashAvatar | 125 | 13453 | 126 | 13453 |
| GS* | 96 | 317968 | **131** | 191431 |
| Ours | 72 | 50708 | 69 | 52910 |
| Ours (No MLP) | **132** | 50708 | 127 | 52910 |

Table 2: **Performance measure.** We report rendering FPS and the number of Gausssians for each method. The rendering speed of our no MLP version surpasses pure Gaussian implementation for subjects wearing extremely high-frequency cloth.

tends to prefer blurry reconstruction over sharp but slightly misaligned results. Notably, although we did not include specific treatments for the head region, better modeling of the body also leads to better face reconstruction. The qualitative evaluation in Fig 4 demonstrates that both versions of our method can learn sharper and more robust body texture compared to existing methods.

**Cross-Reenactment** For the cross-identity reenactment task, we render the reconstruction of the original identity with FLAME expressions and poses from the source subject. With the full version of our method, we apply an additional Euclidean transformation after warping the image plane coordinates with the MLP. This is to ensure the body texture is always aligned with the head Gaussians under novel poses; see Fig 6. The Euclidean transformation is simply determined by fitting the MLP warped image plane coordinates of the anchor Gaussians and their target coordinates in the texture space. To deal with potential artifacts caused by coordinates warped to unseen corner parts in the texture, we apply the same appearance distillation process and remove the fine texture MLP. The No MLP version is applied the same way as in the self-reenactment task.

In addition to the improvement over the body texture, we observe that avatars reconstructed with our approach often give more accurate and faithful expression control, as shown in Fig 5. We deduce that this is because the 3DMM-driven Gaussians only need to model the head region, leading to a more accurate reconstruction of the head model and more reliable LBS weights and expression and pose blendshapes predicted by the LBS network.

**Ablation** We show the effectiveness of the anchor constraint $\mathcal{L}_{anchor}$, test-time Euclidean transformation and warp loss $\mathcal{L}_{warp}$ in Fig 6. Even for subjects with only slight movement in the upper body, anchor constraint is still needed to learn sharp and accurate cloth texture. Besides, without anchor Gaussians and test time Euclidean transformation, the body texture is unable to align with the head Gaussians under novel poses. The warp loss $\mathcal{L}_{warp}$ is needed to prevent the neural warping field from mapping the background pixel to an arbitrary white pixel in the texture space. As anchor Gaussians only exist within the body region, the additional Euclidean transformation computed from anchor correspondences would significantly distort the background pixels, causing severe artifacts as shown in Fig 6 (b). Additional ablation results can be found in the supplementary.

**Rendering Efficiency** We report the number of Gaussians and the rendering speed for pure Gaussian implementation GS*, Ours, and Ours (No MLP) in Tab 2. The rendering speeds are tested on an RTX4080 Ti. For subjects wearing complicated clothes, the number of Gaussians required to model the high-frequency cloth texture significantly increases for pure Gaussian implementation, hence slowing down the rendering speed, whereas our method only models the head region with Gaussians and hence requires a much fewer number of Gaussians. The rendering speed of our no


Figure 4: **Qualitative comparison of self-reenactment task.** We show that both of our full version and No MLP version can recover a more accurate and robust body texture, even under extreme poses and high-frequency cloth textures. More results in the Supplementary 7.

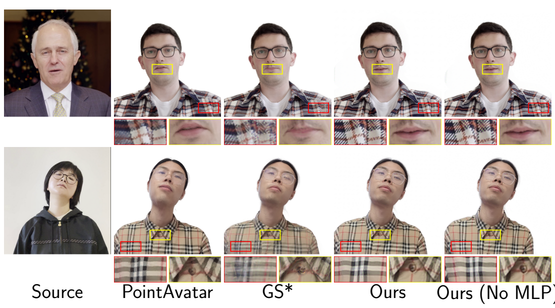


Figure 5: **Qualitative evaluation of cross-identity reenactment.** Our method leads to both better cloth texture and more accurate expression, as LBS network only focuses on the head region in our approach. More results in the Supplementary 8.

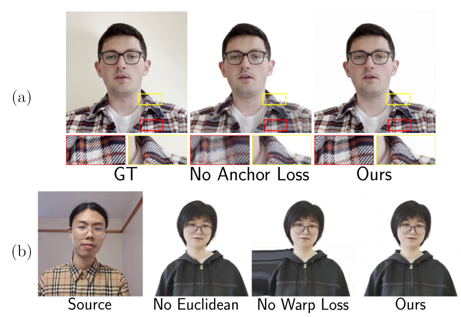


Figure 6: **Qualitative ablation for self-reenactment (a) and cross-reenactment (b).**

MLP version even surpasses pure Gaussian implementation for subject 005, who wears cloth with a very high-frequency texture.

## 5 Conclusion

We present Gaussian Head & Shoulders, a method that reconstructs high-quality and animatable upper body avatars including head, chest and shoulders. By utilizing high-frequency neural texture to represent the clothed body, we are able to model sharp and robust cloth details and significantly reduce the number of Gaussians needed to represent a subject. By constraining the texture warping with a sparse set of anchor Gaussians, the body texture is accurately mapped to the correct position even under unseen poses. By caching the neural texture and replacing the neural warping field with a projective transformation estimated using anchor correspondences, we significantly improve rendering speed and reach over 130 FPS at novel poses, surpassing the rendering speed of pure Gaussian implementation for subjects with complicated cloth textures.

**Limitation.** Although our method can learn faithful texture for the shoulder and chest, it cannot handle arm and hand motions, which would require specific prior and representation such as SMPLX (Loper et al., 2015).

## REFERENCES


Marcel C. Buehler, Abhimitra Meka, Gengyan Li, Thabo Beeler, and Otmar Hilliges. Varitex: Variational neural face textures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 2527–2530, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450310154. doi: 10.1145/2207676.2208639. URL `https://doi.org/10.1145/2207676.2208639`.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. 06 2017.

Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. Monogaussianavatar: Monocular gaussian point-based head avatar. *arXiv*, 2023.

Helisa Dhamo, Yinyu Nie, Arthur Moreau, Jifei Song, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Headgas: Real-time animatable head avatars via 3d gaussian splatting, 2023.

Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. volume 40, 2021. URL `https://doi.org/10.1145/3450626.3459936`.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL `https://doi.org/10.1145/358669.358692`.

Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8649–8658, June 2021.

Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. Reconstructing personalized semantic facial nerf models from monocular video. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 41(6), 2022. doi: 10.1145/3550454.3555501.

Thomas Gerig, Andreas Forster, Clemens Blumer, Bernhard Egger, Marcel Lüthi, Sandro Schönborn, and Thomas Vetter. Morphable face models - an open framework. *CoRR*, abs/1709.08398, 2017. URL `http://arxiv.org/abs/1709.08398`.

Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. *arXiv preprint arXiv:2112.01554*, 2021.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of Machine Learning and Systems 2020*, pp. 3569–3579. 2020.

Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a.

Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Shoukang Hu and Ziwei Liu. Gauhuman: Articulated gaussian splatting from monocular human videos. *arXiv preprint arXiv:*, 2023.

Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *Proceedings of the European conference on computer vision (ECCV)*, 2022.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 694–711, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46475-6.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. Realistic one-shot mesh-based head avatars. In *European Conference of Computer vision (ECCV)*, 2022.

Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018.

Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Trans. Graph.*, 42(4), jul 2023. ISSN 0730-0301. doi: 10.1145/3592455. URL https://doi.org/10.1145/3592455.

Muhammed Kocabas, Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. Hugs: Human gaussian splats, 2023. URL https://arxiv.org/abs/2311.17910.

M. Koujan, M. Doukas, A. Roussos, and S. Zafeiriou. Head2head: Video-based neural head synthesis. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020) (FG)*, pp. 319–326, Los Alamitos, CA, USA, may 2020. IEEE Computer Society. doi: 10.1109/FG47880.2020.00048. URL https://doi.ieeecomputersociety.org/10.1109/FG47880.2020.00048.

Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models, 2023.

Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. URL https://doi.org/10.1145/3130800.3130813.

Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jing Lin, Ailing Zeng, Haoqian Wang, Lei Zhang, and Yu Li. One-stage 3d whole-body mesh recovery with component aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21159–21168, 2023.

Xinqi Liu, Chenming Wu, Jialun Liu, Xing Liu, Chen Zhao, Haocheng Feng, Errui Ding, and Jingdong Wang. Gva: Reconstructing vivid 3d gaussian avatars from monocular videos. *Arxiv*, 2024.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4455–4465, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society. doi: 10.1109/CVPR.2019.00459. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00459.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL `https://doi.org/10.1145/3528223.3530127`.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), dec 2021.

Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. In *CVPR*, 2024.

Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. In *Computer Vision and Pattern Recognition (CVPR)*, 2024.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1409.1556`.

Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. Monocular, One-stage, Regression of Multiple 3D People. In *ICCV*, 2021.

David Svitov, Pietro Morerio, Lourdes Agapito, and Alessio Del Bue. Haha: Highly articulated gaussian human avatars with textured mesh prior, 2024.

Jie Wang, Jiu-Cheng Xie, Xianyan Li, Feng Xu, Chi-Man Pun, and Hao Gao. Gaussianhead: High-fidelity head avatars with learnable gaussian derivation, 2024.

Lizhen Wang, Xiaochen Zhao, Jingxiang Sun, Yuxiang Zhang, Hongwen Zhang, Tao Yu, and Yebin Liu. Styleavatar: Real-time photo-realistic portrait avatar from a single video. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023.

Ziyan Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael Zollh¨ofer. Learning compositional radiance fields of dynamic human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5704–5713, 2021.

Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. Avatarmav: Fast 3d head avatar reconstruction using motion-aware neural voxels. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023.

Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4210–4220, 2023.

Zhenhui Ye, Tianyun Zhong, Yi Ren, Jiaqi Yang, Weichuang Li, Jiawei Huang, Ziyue Jiang, Jinzheng He, Rongjie Huang, Jinglin Liu, et al. Real3d-portrait: One-shot realistic 3d talking portrait synthesis. *arXiv preprint arXiv:2401.08503*, 2024.

Zhongyuan Zhao, Zhenyu Bao, Qing Li, Guoping Qiu, and Kanglin Liu. Psavatar: A point-based morphable shape model for real-time head avatar animation with 3d gaussian splatting, 2024.

Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. I M Avatar: Implicit morphable head avatars from videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J. Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4574–4584, 2022. URL `https://api.semanticscholar.org/CorpusID:253761096`.

M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS '01.*, pp. 29–538, 2001. doi: 10.1109/VISUAL.2001.964490.

# Gaussian Head & Shoulders: High Fidelity Neural Upper Body Avatars with Anchor Gaussian Guided Texture Warping

## Supplementary Material

In this supplementary material, we provide additional implementation and evaluation details in Sec A, as well as extended results including additional ablation studies, limitations, and a comparison with SMPL-driven body avatar in Sec B. Ethic discussions are in Sec C. We also highly recommend the readers to view our supplementary video.

## A Implementation Details

### A.1 Preprocessing

Our data preprocessing pipeline for extracting FLAME parameters, camera parameters and body landmarks is modified from (Zheng et al., 2022). After obtaining rough FLAME parameters from DECA (Feng et al., 2021), we further optimize the FLAME parameters to minimize the 68 facial landmarks for 3000 iterations. For subject 001, we keep the original training and test split used by PointAvatar (Zheng et al., 2023). For other subjects, we use the last 500 or 1000 frames as test frames, depending on the total frame count in the video. For all subjects, we simply use the first frame as the canonical training frame for initializing anchor Gaussians and updating the anchor correspondences. We use DWpose (Yang et al., 2023) to detach the noise, neck and shoulder landmarks, which are illustrated in Fig 9.

### A.2 Network Architecture

We have three MLPs in total: $\mathrm{MLP}_d$ which predicts the expression blendshapes $\boldsymbol{\mathcal{E}}$, pose blendshapes $\boldsymbol{\mathcal{P}}$ and LBS weights $\boldsymbol{\mathcal{W}}$ for each regular Gaussian and anchor Gaussian; $\mathrm{MLP}_f$ which predicts pose-dependent fine texture; $\mathrm{MLP}_w$ which warps view space coordinates to texture space coordinates. All three MLPs have 4 hidden layers and 128 neurons in each hidden layer. The standard Fourier frequency positional encoding (Mildenhall et al., 2020) is applied to the pixel coordinate, FLAME head rotation, camera translation and 2D landmarks before inputting to $\mathrm{MLP}_f$ and $\mathrm{MLP}_w$. The pixel coordinate and 2D landmarks are encoded with a frequency of 10, and camera translation and FLAME head rotation are encoded with a frequency of 2. All three MLPs are initialized to predict 0s at the beginning by setting the weights and bias of the output layer to 0. All MLPs use ReLU as the intermediate activations. Tanh is used as the final activation for $\mathrm{MLP}_f$, no final activation is used for $\mathrm{MLP}_w$, and the final activation for $\mathrm{MLP}_d$ are the same as (Zheng et al., 2023).

We use a latent dimension $D_t = 32$ for the latent texture $\mathbf{T}_f$. The coarse texture $\mathbf{T}_c$ is initialized to be the same as the white background, while the fine latent $\mathbf{T}_f$ is initialized and a random and uniform distribution between $[0, 1]$.

### A.3 Training Details

For all subjects, we use $\lambda_{head} = 1$, $\lambda_{anchor} = 1$, $\lambda_{warp} = 0.025$, $\lambda_{\hat{\alpha}} = 0.15$. For VGG loss weight $\lambda_{VGG}$, we set it to 0 for the first 10K iterations, and then $0.1$ for the rest of the training. This is needed as we empirically observe that training the neural texture and warping field with a strong VGG loss from the beginning severely harms their stability. The weights of FLAME regularization are initially set to $\lambda_{\boldsymbol{\mathcal{E}}} = 1000, \lambda_{\boldsymbol{\mathcal{P}}} = 1000, \lambda_{\boldsymbol{\mathcal{W}}} = 1$ and are reduced by half at 15k, 30k, 45k iteration respectively.

We train our model with Adam optimizer for 70k iterations in total, where the three stages of our training take 4k, 46k and 20k iterations respectively. The learning rate for blendshapes and LBS weight MLP $\mathrm{MLP}_d$, neural texture, anchor Gaussian parameters and neural warping field are set to $10^{-3}$, which is halved at 30k-th and 60k-th iterations respectively. The learning rate and density control hyperparameters for regular Gaussians are the same as proposed by the original paper (Kerbl et al., 2023), except that we use a density gradient threshold of $2.5 \times 10^{-4}$ before we start applying VGG loss, and $8 \times 10^{-3}$ afterward. For every 10k iterations during the training, we also re-project all

| | 001 | | | 002 | | | 003 | | | 004 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| INSTA | 18.58 | 0.751 | 0.269 | 22.90 | 0.880 | 0.177 | 22.24 | 0.809 | 0.175 | 19.45 | 0.784 | 0.310 |
| SplattingAvatar | 18.49 | 0.737 | 0.307 | 25.34 | 0.876 | 0.171 | 21.34 | 0.790 | 0.220 | 19.83 | 0.765 | 0.351 |
| PointAvatar | 22.83 | 0.822 | 0.100 | 30.61 | 0.924 | 0.062 | 28.12 | 0.874 | 0.077 | 23.99 | 0.837 | 0.133 |
| FlashAvatar | 19.87 | 0.782 | 0.133 | 25.44 | 0.894 | 0.082 | 24.79 | 0.869 | 0.063 | 20.42 | 0.795 | 0.216 |
| GS* | 23.26 | 0.814 | 0.082 | 32.99 | 0.937 | 0.046 | 29.85 | 0.888 | 0.054 | 24.18 | 0.836 | 0.139 |
| Ours | 25.95 | 0.856 | 0.064 | 31.98 | 0.949 | 0.042 | 31.26 | 0.917 | 0.042 | 24.68 | 0.839 | 0.120 |
| Ours No MLP | 24.48 | 0.840 | 0.070 | 31.44 | 0.942 | 0.042 | 28.85 | 0.892 | 0.051 | 24.61 | 0.837 | 0.120 |
| | 005 | | | 006 | | | 007 | | | 008 | | |
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| INSTA | 19.47 | 0.757 | 0.251 | 23.44 | 0.861 | 0.165 | 18.68 | 0.733 | 0.291 | 19.97 | 0.675 | 0.246 |
| SplattingAvatar | 20.06 | 0.763 | 0.250 | 22.78 | 0.838 | 0.201 | 20.15 | 0.754 | 0.257 | 19.97 | 0.665 | 0.432 |
| PointAvatar | 22.82 | 0.847 | 0.142 | 29.42 | 0.929 | 0.043 | 22.30 | 0.826 | 0.088 | 21.61 | 0.748 | 0.174 |
| FlashAvatar | 19.65 | 0.789 | 0.152 | 24.25 | 0.871 | 0.060 | 20.02 | 0.770 | 0.116 | 20.56 | 0.691 | 0.197 |
| GS* | 22.80 | 0.847 | 0.129 | 29.56 | 0.924 | 0.039 | 22.31 | 0.820 | 0.099 | 22.60 | 0.762 | 0.173 |
| Ours | 24.48 | 0.895 | 0.074 | 30.97 | 0.943 | 0.033 | 23.26 | 0.856 | 0.074 | 21.47 | 0.726 | 0.111 |
| Ours No MLP | 22.19 | 0.860 | 0.078 | 28.71 | 0.912 | 0.037 | 21.49 | 0.827 | 0.081 | 22.02 | 0.765 | 0.116 |

Table 3: **Quatitative evaluation of full self-reenactment task** We report PSNR↑, SSIM↑, and LPIPS↓ ,and color the best and second-best methods for each subject respectively.

anchor Gaussians to the image plane of the canonical image plane, and remove the anchor Guassians that are out of the view frustum. This is to prevent unconstrained anchor Gaussians from applying noisy regularization on the texture warping field.

Following (Zheng et al., 2023) and (Zheng et al., 2022), we also add a static bone, which does not take any transformation with the FLAME expression and poses.

As our preprocessing pipeline does not track eye movement, for subjects with significant eye movements in the training frames, i.e., subjects 002 and 005, we do not update the opacity and SH of regular Gaussians in the third stage to prevent undesirable view-dependent artifacts. For subjects where the semantic mask fails, i.e., subject 003, the No MLP texture may contain significant noise in the head region. We hence manually define a rough bounding box for this subject to clean the No MLP texture for self-reenactment and cross-reenactment tasks.

The training takes around 2 hours for each subject on an RTX4080 Ti.

### A.4 Evaluation Details

Following (Zheng et al., 2023) and (Grassal et al., 2021), we also fine-tune the pre-tracked FLAME expression, pose parameters, camera translation and body landmarks during the training to account for inaccuracies in the preprocessing pipeline. We use Adam optimizer with a learning rate of $10^{-4}$ and optimize them from the 30k-th iteration. For test-time tracking optimization, we only use L2 RGB loss. Since we do not have a direct gradient flowing back from the body texture to the FLAME parameters, we also optimize a translation and rotation offset for the body texture mapping.

## B Additional Results

### B.1 Videos

We strongly encourage the readers to watch the videos containing self-reenactment and cross-reenactment results in the supplementary.

As shown in the videos, existing methods either fail to model the body properly (INSTA (Zielonka et al., 2022), SplattingAvatar (Shao et al., 2024)), or fail to learn the details on head and body (PointAvatar (Zheng et al., 2023)). While the pure Gaussian Splatting baseline (GS*) could learn the face and body with much better details, it still learns blurry textures and presents severe artifacts

| | 001 | | | 002 | | | 003 | | | 004 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| INSTA | 26.62 | 0.898 | 0.082 | 34.89 | 0.963 | 0.032 | 29.80 | 0.922 | 0.071 | 28.35 | 0.941 | 0.069 |
| SplattingAvatar | 24.29 | 0.876 | 0.109 | 32.66 | 0.958 | 0.034 | 25.06 | 0.881 | 0.098 | 27.13 | 0.932 | 0.073 |
| PointAvatar | 26.17 | 0.904 | 0.079 | 34.93 | 0.968 | 0.021 | 30.90 | 0.923 | 0.053 | 29.65 | 0.948 | 0.045 |
| FlashAvatar | 27.44 | 0.911 | 0.069 | 35.61 | 0.973 | 0.021 | 30.30 | 0.939 | 0.037 | 28.09 | 0.942 | 0.046 |
| GS* | 27.10 | 0.906 | 0.062 | 37.61 | 0.975 | 0.015 | 32.26 | 0.928 | 0.038 | 30.55 | 0.950 | 0.042 |
| Ours | 29.31 | 0.926 | 0.047 | 36.91 | 0.981 | 0.013 | 33.36 | 0.943 | 0.030 | 31.58 | 0.957 | 0.039 |
| Ours No MLP | 29.16 | 0.924 | 0.048 | 36.89 | 0.981 | 0.013 | 32.06 | 0.939 | 0.034 | 31.45 | 0.956 | 0.041 |
| | 005 | | | 006 | | | 007 | | | 008 | | |
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| INSTA | 29.15 | 0.940 | 0.054 | 33.43 | 0.977 | 0.022 | 22.98 | 0.871 | 0.119 | 33.18 | 0.975 | 0.022 |
| SplattingAvatar | 28.75 | 0.938 | 0.061 | 31.93 | 0.967 | 0.030 | 23.56 | 0.873 | 0.121 | 32.92 | 0.976 | 0.024 |
| PointAvatar | 31.39 | 0.952 | 0.036 | 34.94 | 0.981 | 0.016 | 24.85 | 0.893 | 0.062 | 32.32 | 0.977 | 0.025 |
| FlashAvatar | 31.03 | 0.957 | 0.030 | 34.00 | 0.982 | 0.017 | 23.14 | 0.881 | 0.073 | 33.03 | 0.980 | 0.018 |
| GS* | 32.36 | 0.959 | 0.030 | 35.62 | 0.983 | 0.014 | 25.00 | 0.892 | 0.064 | 33.99 | 0.980 | 0.020 |
| Ours | 33.90 | 0.967 | 0.027 | 36.90 | 0.987 | 0.012 | 26.35 | 0.921 | 0.045 | 36.14 | 0.988 | 0.013 |
| Ours No MLP | 33.74 | 0.967 | 0.026 | 36.77 | 0.987 | 0.012 | 25.00 | 0.909 | 0.048 | 35.27 | 0.988 | 0.012 |

Table 4: **Quatitative evaluation of head-only self-reenactment task.** We report the metrics with the body region masked out. Note that the body region is still used during the training.

| | 002 | | | 005 | | | 007 | | |
|---|---|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| No Anchor Loss | 24.96 | .910 | .088 | 22.91 | .854 | .117 | 19.30 | .773 | .134 |
| No Warp Loss | 32.86 | .949 | .041 | 24.19 | .891 | .081 | 22.74 | .848 | .076 |
| Ours | 31.98 | .949 | .042 | 24.48 | .895 | .074 | 23.26 | .856 | .074 |

Table 5: **Quatitative ablation.** We show the anchor constraint is necessary for learning sharp and correct body texture. While the warp loss might not necessarily improve the performance for the self-reenactment task, it is needed for cross-reenactment with out-of-distribution poses.

when the subject is moving in extreme head rotation. It is most obvious for the self-reenactment and cross-reenactment videos of subject 005 – many Gaussians modeling the cloth texture are not well-aligned with each other, as a result, they cannot move naturally with the head motion. In comparison, our method can learn extremely sharp textures with robust performance under novel poses and motions.

### B.2 Ablation

Additional ablation results are presented in Table 5 and Figure 10, demonstrating the critical role of the anchor loss in achieving sharp and precise textures. Although the warp loss $\mathcal{L}_{warp}$ does not necessarily improve the numerical metrics for the self-reenactment task, Fig **??** illustrates its importance in preventing the significant failure when combining neural warping with additional Euclidean transformation.

### B.3 Texture Cleaning

When distilling the pose-dependent fine texture into the coarse texture for our no MLP version, we utilized DeepLabV3 (Chen et al., 2017) to obtain a coarse mask of the background and set the values of those pixels to 1. This is needed because the body texture contains a padding region to account for the body part that is moving in and out during the video. A majority section of the padding, especially the padding region on the top the left and right sides, are rarely used and trained during optimization. As a result, the fine texture colors obtained in those regions can produce noisy artifacts; see Fig 12.


Figure 7: **Qualitative evaluation of self-identity reenactment.**


Figure 8: **Qualitative evaluation of cross-identity reenactment.**

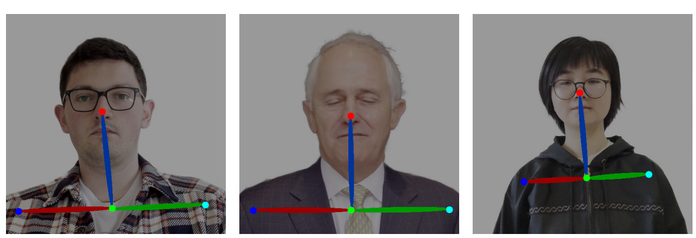

Figure 9: **Landmarks.** We use DWPose (Yang et al., 2023) to detect nose, neck and shoulder landmarks to use as input to $\mathrm{MLP}_f$ and $\mathrm{MLP}_w$.

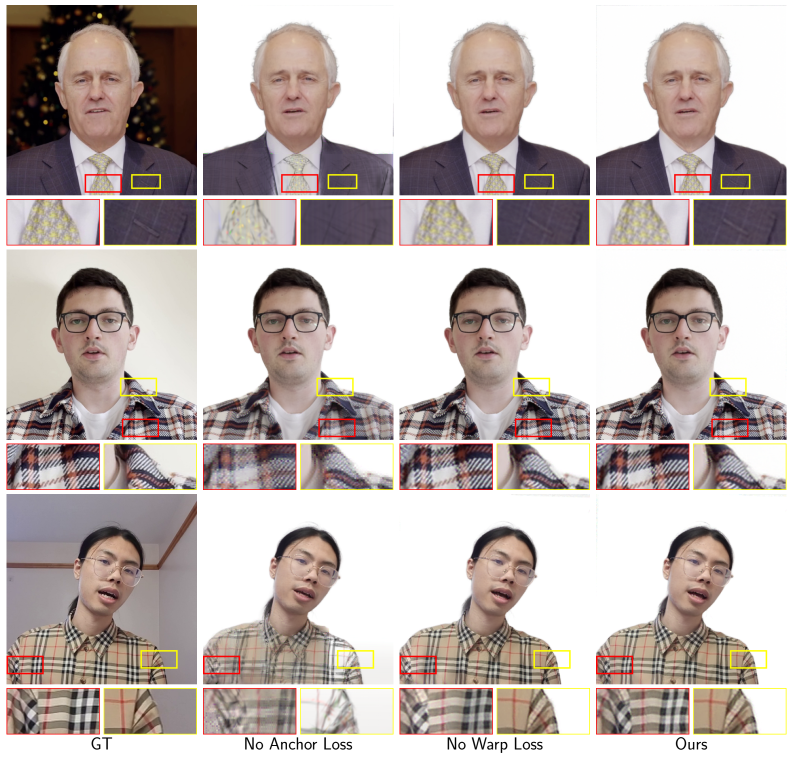


Figure 10: **Qualitative Ablation.**

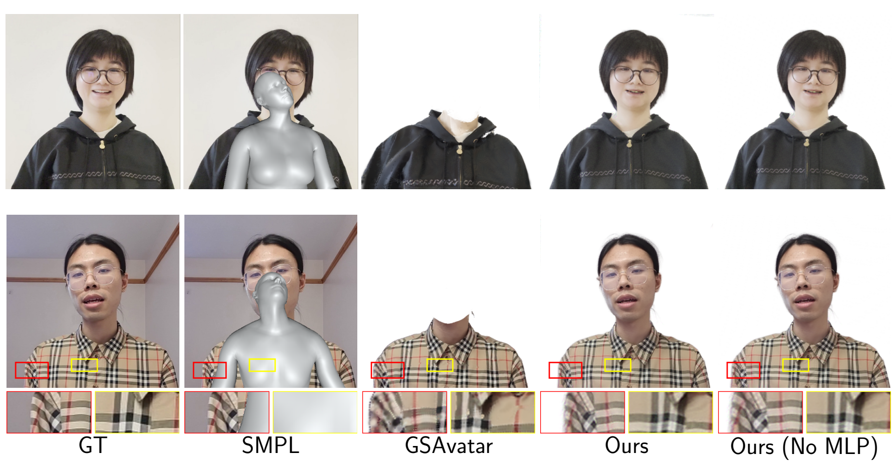


Figure 11: **Qualitative comparison with full body avatar methods.** Due to the limited landmarks available on the shoulders and chest, existing SMPL tracking methods fail to obtain correct SMPL parameters. Fully body neural avatars that rely on SMPL hence fail to learn accurate and robust body. While our method does not include SMPL 3DMM, the use of static virtual bone and neural texture warping allow us to learn the body texture accurately.

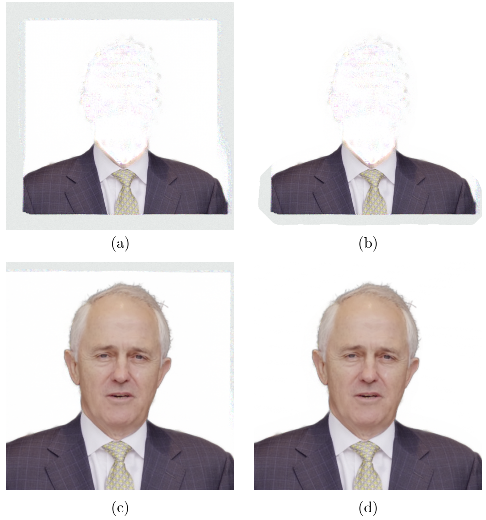


Figure 12: **Texture cleaning.** We show the body texture without masking (a) and with cleaning (b), as well as the rendering without texture cleaning (c) and with texture cleaning (d).

| | 004 | | | 007 | | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| GSAvatar | 17.08 | .811 | .178 | 16.64 | .744 | .143 |
| Ours | 26.82 | .887 | .094 | 23.87 | .885 | .052 |
| Ours No MLP | 26.70 | .885 | .094 | 22.43 | .861 | .056 |

Table 6: **Body Only Quantitative Comparison with Full Body Avatars.** We show that existing full body neural avatar methods that rely on SMPL deformation perform significantly worse than our methods. Metrics are computed after masking out the background and head regions.

### B.4 Comparison with Full Body Avatars

To verify our choice of driving anchor Gaussians only with head 3DMM (FLAME), we select two subjects that show a larger portion of the upper body and compare our method with GSAvatar, a Gaussian Splatting based full body neural avatar methods that deform the representation based on SMPL (Hu et al., 2024b). As the code release of GSAvatar only supports SMPL instead of SMPLX, we simply use semantic masks to remove the head region during the training and compare only the reconstruction quality of the body part. As shown in Tab 6 and Fig 11, since the existing SMPL tracking methods for monocular videos are developed only for views that include the whole body, the fitted SMPL is significantly misaligned with the GT (Sun et al., 2021), even after fine-tuning during Gaussian optimization. As a result, the clothed body reconstructed by GSAvatar presents several artifacts under novel poses and are significantly misaligned the GT. Our method is able to reconstruct the chest and shoulders with much better quality and accuracy. We would also like to note that, although we do not include body 3DMM in our method, due to the usage of virtual static bone, technically speaking, the effect is exactly the same as have a SMPLX 3DMM where the body and hand parts (SMPLX and MANO) are kept static during the whole sequences.

### B.5 Novel View Synthesis

We show novel view synthesis results of our method in Fig. . Typically, because our method modeled the body as 2D texture, it would be difficult to render it from novel views, just as StyleAvatar Wang et al. (2023). However, one key novelty of our method is the use of Anchor Gaussians as a constraint between 3D and 2D, and we can hence effectively utilize them to achieve a certain extent of novel view rendering. Specifically, we render the head Gaussians and the Anchor Gaussians at each novel view, reproject the Anchor Gaussians back to the image plane to obtain their 2D coordinates, and


Figure 13: **Pose-Dependent Appearance Modeling.** By utilizing the pose-dependent fine texture, we can model pose-dependent appearance changes on clothes such as wrinkles or lighting changes. However, the fine texture is mainly used to deal with those appearance changes as noise in the training frames, modeling them exactly the same as the ground truth at test time remains a challenging problem. We additionally visualize the difference in fine texture in the last column.

Figure 14: **SMPLX Estimation via OSX Lin et al. (2023).** Some latest SMPLX prediction methods such as OSX Lin et al. (2023) are capable of predicting more accurate body 3DMM annotation than landmark optimization pipeline used in Hu et al. (2024b). However, as they are still mainly trained and optimized on frames with full-body or upper-body portraits with arms visible, their performance can be degraded with our tight framing setting: they tend to struggle with shoulders and can fail to detect any body with extreme poses such as the one shown in last column. Regardless, please note that we do not incorporate SMPLX not only because the annotation accuracy is not guaranteed, but also to keep a fair comparison with our baselines, where only FLAME 3DMM is used for LBS.


Figure 15: **Novel View Synthesis Results.** Since our method is trained only with monocular video where only limited view angles are included for the body, we can only render novel views with small displacement to the training views, similar to all other monocular neural avatar methods.

| | 005 | | | 007 | | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| GaussianAvatar | 17.61 | .772 | .211 | 19.74 | .770 | .189 |
| Ours | 24.48 | .895 | .074 | 23.26 | .856 | .074 |
| Ours No MLP | 22.19 | .860 | .078 | 21.49 | .827 | .081 |
| GaussianAvatar (Head) | 30.27 | .956 | .040 | 22.89 | .879 | .098 |
| Ours (Head) | 33.90 | .967 | .027 | 26.35 | .921 | .045 |
| Ours No MLP (Head) | 33.74 | .967 | .026 | 25.00 | .909 | .048 |

Table 7: **Quantitative Comparisons with GaussianAvatar Hu et al. (2024a) in Self-Reenactment Task.**

further compute a homography that minimizes the anchor constraint loss $\mathcal{L}_{anchor}$. This will ensure the body to move properly with the head and they always stay connected. Please note that similar to the existing neural avatar reconstruction method using monocular view, we can only render novel views with small displacement to the training views, as extrapolated views significantly degrade the results.

### B.6 Additional Baselines

We include comparisons with additional baselines including Real3DPortrait Ye et al. (2024), GaussianAvatar Hu et al. (2024a); see Fig 16 and Fig 17. We included the quantitative results for self-reenactment evaluations in Tab 7. StyleAvatar Wang et al. (2023) unfortunately degenerates and fails on our dataset; see Fig 18.

In Fig 19 and Table 8, we included comparison with Real3D-Portrait trained on single identity video. We trained the motion adapter for 100,000 steps on a single A100 GPU, which takes around 80 hours. We then trained the HTB-SR model for 80,000 steps, which takes around 30 hours. The inference speed of Real3D-Portrait is around 20 FPS on a single GPU. Note that in comparison, our method only requires less than 3 hours to train and can infer with around 130 FPS. It can be seen that our method is able to generate the head and cloth with much better quality. In Real3D-Portrait, a torso model is used to predict 2D warping from body keypoints to deform the latent image for fused body generation. While this approach can effectively learn to correctly connect the head to the body, without 3D-2D constraints from anchor Gaussian, it fails to learn sharp textures on the clothes. This result also matches our No Anchor Loss ablation in Fig 10.

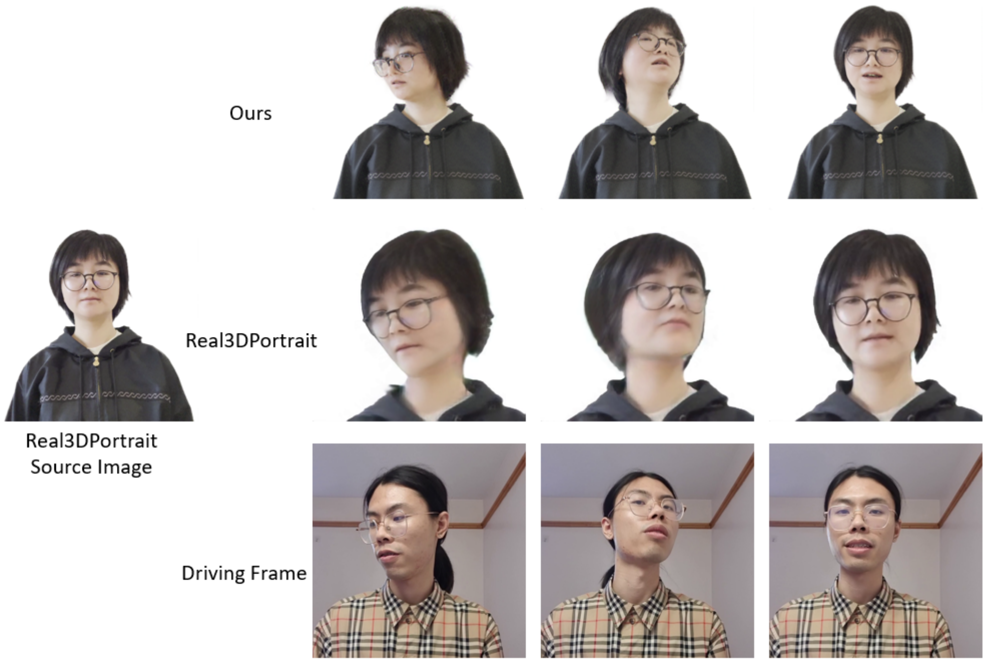


Figure 16: **Comparison with Real3DPortrait Ye et al. (2024) in Cross-Reenactment Task.** Although Real3DPortrait is trained with multi-identity datasets with rich facial prior extracted from the training, it fails to produce high-quality reenactment with extreme poses and cannot render shoulders and chest due to fixed tight framing in the training. Our method generates more faithful and accurate results in comparison.

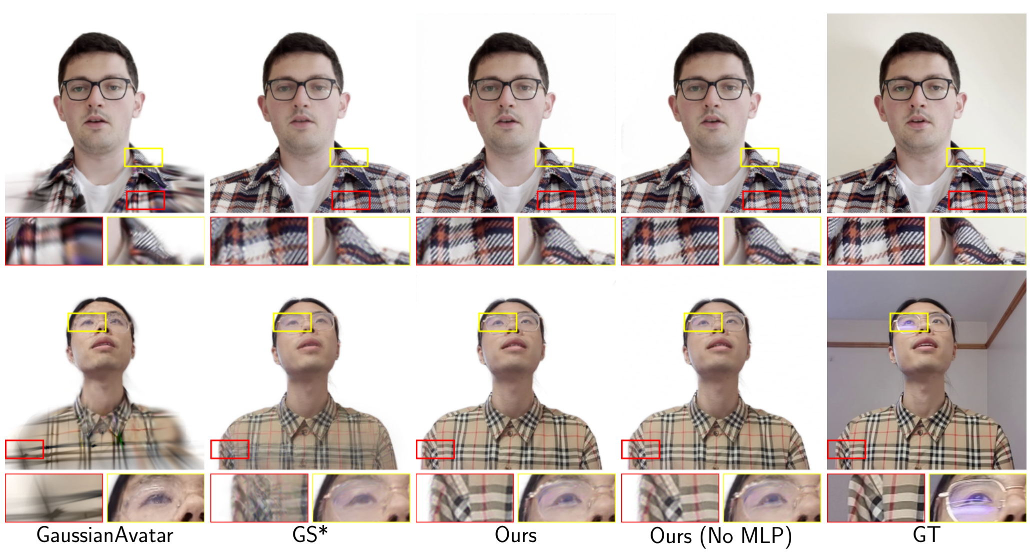


Figure 17: **Additional Comparisons with GaussianAvatar Hu et al. (2024a) in Self-Reenactment Task.**


Figure 18: **StyleAvatar Wang et al. (2023). Results**. We attempted to evaluate StyleAvatar on our dataset with the original framing. However, it seems that StyleAvatar quickly degenerates and fails after training for 10K iterations.

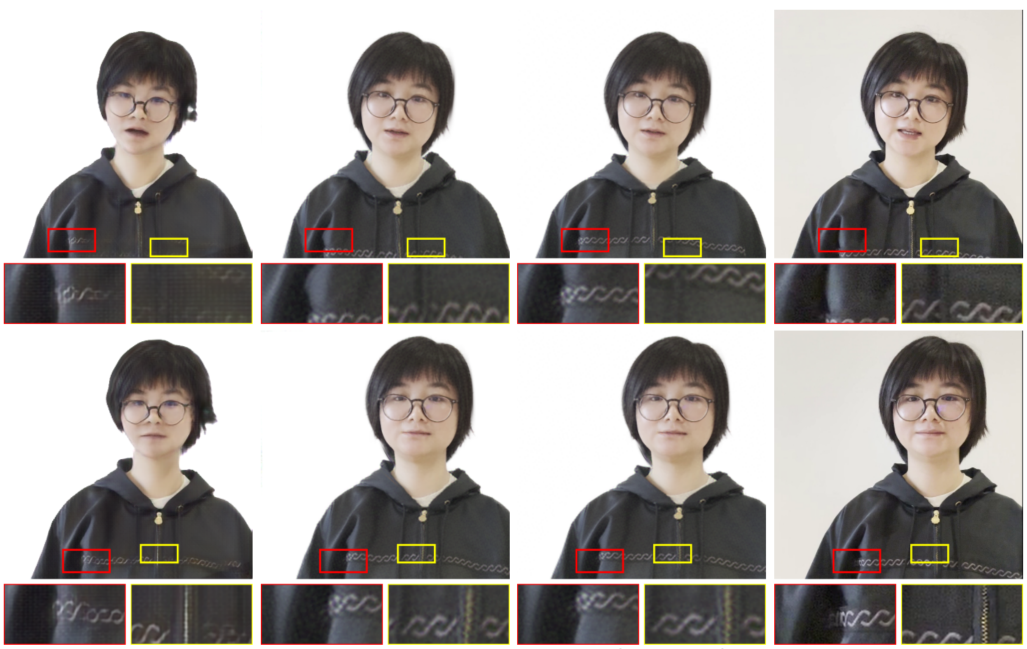


Figure 19: **Additional Comparison with Real3D-Portrait Ye et al. (2024)**. We re-trained Real3D-Portrait on our single identity video to generate fair comparisons. We trained the motion adapter for 100,000 steps on a single A100 GPU, which takes around 80 hours. We then trained the HTB-SR model for 80,000 steps, which takes around 30 hours. The comparison shows that our method is able to reconstruct both the head and the cloth texture with much better quality.

| | 004 | | |
|---|---|---|---|
| | PSNR | SSIM | LPIPS |
| Real3D-Portrait | 16.07 | .739 | .209 |
| Ours | 24.68 | .839 | .120 |
| Ours No MLP | 24.61 | .837 | .120 |
| Real3D-Portrait (Head) | 23.01 | .895 | .083 |
| Ours (Head) | 31.58 | .957 | .039 |
| Ours No MLP (Head) | 31.45 | .956 | .041 |

Table 8: **Quantitative Comparisons with Real3D-Protrait Hu et al. (2024a) in Self-Reenactment Task.**

### B.7 LIMITATIONS

Although we propose a no MLP version that is able to render at novel poses with 130 FPS, as it completely relies on rigid homography transformation to map body texture to the view space, it is unable to model any non-rigid deformation in the body. In addition, for sequences with extreme head rotations, it might move the body in a way that is not exactly aligned with the ground truth, as shown in the supplementary videos. However, we observe that the results produced with this no MLP version still present a faithful rendering. For cases where the non-rigid body deformation is important, we recommend the use of the full version, whose rendering speed is around 70 FPS and can be further optimized by caching the fine texture only.

## C ETHICS

We captured 4 human subjects with mobile phones for our experiments. All subjects have signed consent forms for using the captured video in this research and publication. We will release the data for subjects with permission.

Our method constructs faithful and animatable head avatars and can be used to generate videos of real people performing synthetic poses and expressions. We do not condone any misuse of our work to generate fake content of any person with the intent of spreading misinformation or tarnishing their reputation.