FLARE: Fast Learning of Animatable and Relightable Mesh Avatars

SHRISHA BHARADWAJ, Max Planck Institute for Intelligent Systems, Germany
YUFENG ZHENG, ETH Zürich, Switzerland and Max Planck Institute for Intelligent Systems, Germany
OTMAR HILLIGES, ETH Zürich, Switzerland
MICHAEL J. BLACK, Max Planck Institute for Intelligent Systems, Germany
VICTORIA FERNANDEZ ABREVAYA, Max Planck Institute for Intelligent Systems, Germany

Fig. 1. We present FLARE, a method for rapidly building relightable head avatars from monocular videos. Our method estimates a high-fidelity mesh geometry that can be efficiently animated using learned blendshape and linear-blend-skinning-weight fields. Moreover, we model the intrinsic albedo, roughness, specular reflections, and an indirect representation of the light, enabling relighting in novel scenes.

Our goal is to efficiently learn personalized animatable 3D head avatars from videos that are geometrically accurate, realistic, relightable, and compatible with current rendering systems. While 3D meshes enable efficient processing and are highly portable, they lack realism in terms of shape and appearance. Neural representations, on the other hand, are realistic but lack compatibility and are slow to train and render. Our key insight is that it is possible to efficiently learn high-fidelity 3D mesh representations via differentiable rendering by exploiting highly-optimized methods from traditional computer graphics and approximating some of the components with neural networks. To that end, we introduce FLARE, a technique that enables the creation of animatable and relightable mesh avatars from a single monocular video. First, we learn a canonical geometry using a mesh representation, enabling efficient differentiable rasterization and straightforward animation via learned blendshapes and linear blend skinning weights. Second, we follow physically-based rendering and factor observed colors into intrinsic albedo, roughness, and a neural representation of the illumination, allowing the learned avatars to be relit in novel scenes. Since our input videos are captured on a single device with a narrow field of view, modeling the surrounding environment light is non-trivial. Based on the split-sum approximation for modeling specular reflections, we address this by approximating the pre-filtered environment map with a multi-layer perceptron (MLP) modulated by the surface roughness, eliminating the need to explicitly model the light. We demonstrate that our mesh-based avatar formulation, combined with learned deformation, material, and lighting MLPs, produces avatars with high-quality geometry and appearance, while also being efficient to train and render compared to existing approaches.

CCS Concepts: • **Computing methodologies → Machine learning**.

Additional Key Words and Phrases: Neural head avatars, neural rendering, 3D reconstruction, relighting

ACM Reference Format:
Shrisha Bharadwaj, Yufeng Zheng, Otmar Hilliges, Michael J. Black, and Victoria Fernandez Abrevaya. 2023. FLARE: Fast Learning of Animatable and Relightable Mesh Avatars. *ACM Trans. Graph.* 42, 6, Article 204 (December 2023), 15 pages. https://doi.org/10.1145/3618401

Authors' addresses: Shrisha Bharadwaj, Max Planck Institute for Intelligent Systems, Tübingen, Germany, shrisha.bharadwaj@tuebingen.mpg.de; Yufeng Zheng, ETH Zürich, Zürich, Switzerland and Max Planck Institute for Intelligent Systems, Tübingen, Germany, yufeng.zheng@inf.ethz.ch; Otmar Hilliges, ETH Zürich, Zürich, Switzerland, otmar.hilliges@inf.ethz.ch; Michael J. Black, Max Planck Institute for Intelligent Systems, Tübingen, Germany, black@tuebingen.mpg.de; Victoria Fernandez Abrevaya, Max Planck Institute for Intelligent Systems, Tübingen, Germany, victoria.abrevaya@tuebingen.mpg.de.

0730-0301/2023/12-ART204
https://doi.org/10.1145/3618401

1 INTRODUCTION

There has been remarkable progress on learning personalized 3D facial assets, moving from complex and expensive high-end systems [Beeler et al. 2011; Debevec et al. 2000; Ghosh et al. 2011] to using single commodity sensors as input [Grassal et al. 2022; Zheng et al. 2022; Zielonka et al. 2023]. Although a quality gap still exists, it is being gradually bridged by neural methods that leverage implicit or explicit shape representations. In particular, signed distance fields [Zheng et al. 2022] and point clouds [Zheng et al. 2023] have been

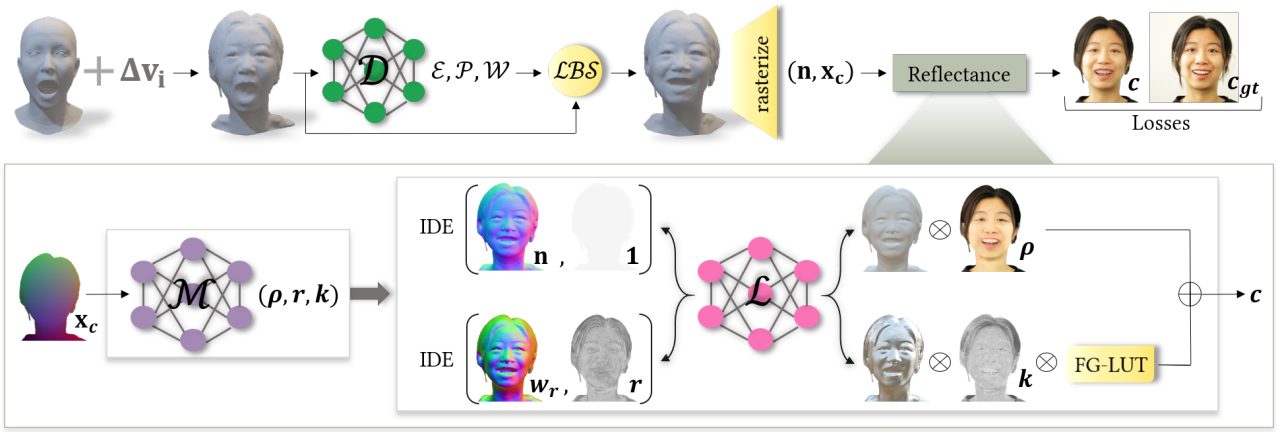


Fig. 2. **Method pipeline.** Top: Given an input video, we optimize for vertex displacements to obtain a canonical geometry. A *deformation* network $\mathcal{D}$ (green) then predicts FLAME [Li et al. 2017] expression blendshapes $\mathcal{E}$, pose correctives $\mathcal{P}$ and blend skinning weights $\mathcal{W}$ given canonical vertices, which are used to deform the mesh into the corresponding expression and pose. The deformed mesh is rasterized following a deferred shading pipeline to obtain per-pixel canonical coordinates $\mathbf{x_c}$ and deformed normals $\mathbf{n}$. Bottom: $\mathbf{x_c}$ is used to query the *material* network $\mathcal{M}$ (purple) to obtain the albedo ρ, roughness r, and specular intensity k. Next, the *lighting* network $\mathcal{L}$ (pink) obtains an estimate of the diffuse shading and specular reflection from the normal and reflection vectors, while taking roughness into account. We use physically-based rendering to compute the final color, which is compared with the ground-truth frame during training.

Table 1. Compared to other methods, FLARE converges rapidly, reconstructs high-fidelity geometry, is compatible with graphics pipelines since it employs a mesh representation, and produces head avatars that can be relighted.

Method	Converges within 15 mins	High-fidelity geometry	Compatible with graphics pipelines	Relightable
IMavatar	✗	✓	✗	✗
NHA	✗	✗	✓	✗
PointAvatar	✗	✓	✗	✓
INSTA	✓	✗	✗	✗
FLARE	✓	✓	✓	✓

used to obtain impressive 3D reconstructions, while NeRF-based [Mildenhall et al. 2020] approaches [Gafni et al. 2021; Zielonka et al. 2023] have shown an outstanding ability to synthesize novel views of the subject. Further, these methods are trained such that the learned avatars can be controlled with novel poses and expressions, making them appropriate for entertainment and telecommunication.

There are several challenges that remain for existing head avatars to be widely applicable in industry. First, the majority of methods are slow to train and/or to render, taking hours [Zheng et al. 2023] or days [Grassal et al. 2022; Zheng et al. 2022] of processing to obtain a single, scene-dependent avatar. This limits the scope of applications and hinders the creation of immersive experiences. Fast approaches have recently been proposed [Gao et al. 2022; Xu et al. 2023; Zielonka et al. 2023], but they suffer from low-quality geometry and often do not generalize well to novel views. Second, to achieve high-quality reconstructions, current methods use shape representations that are not compatible with standard graphics pipelines. Ideally, a mesh representation should enable easy asset extraction and integration. However, recent neural methods that are built on triangulated meshes [Grassal et al. 2022; Khakhulin et al. 2022] do not achieve the same geometric quality as methods based on more flexible representations. Finally, most neural approaches generate avatars that can only be rendered in the same environment in which they were captured since they do not disentangle the light from intrinsic material properties. What is still missing is an efficient method to extract head avatars that have high-fidelity geometry and can be animated and relit.

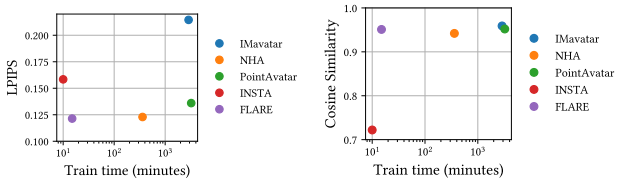
In this work we present a new method, FLARE (Fast Learning of Animatable and RElightable mesh avatars), for building 3D facial avatars from monocular videos that addresses all of these challenges, as shown in Table 1 and Figure 1. We use a mesh representation to allow easy integration as well as fast computation during training and at inference time. We represent the canonical head geometry as a triangular mesh with optimizable vertex locations and learn blendshapes as well as skinning-weight fields to deform the canonical mesh given FLAME [Li et al. 2017] expression and pose parameters. To disentangle the intrinsic material properties and extrinsic light conditions, we leverage physically-based rendering [Cook and Torrance 1982; Walter et al. 2007] where materials and lighting are represented by multi-layer perceptrons (MLPs). Specifically, we use the Disney material model [Burley 2012] and represent albedo, roughness, and specular intensity as hash-encoded spatial MLPs [Müller et al. 2022]. To render color efficiently we adopt the split-sum approximation proposed in [Karis 2013]. However, explicitly computing the environment light is challenging with monocular head videos given their narrow field of view. To address this, we approximate the pre-filtered environment map in the split-sum approximation with a neural network, together with an Integrated Directional Encoding (IDE) [Verbin et al. 2022] that accounts for different roughness levels. Our networks are trained using a two-stage approach, where the first stage is focused on geometry and then the second stage refines the color by leveraging the hash-grid encoding [Müller et al. 2022]. This allows FLARE to control the pace at which geometry and color are learned relative to each other, achieving

detailed results in both areas. While maintaining high accuracy, our method is carefully designed to improve training and rendering efficiency: (1) The canonical material estimation MLPs are fueled by hash-grid encoding [Müller et al. 2022], which effectively represents high-resolution mappings with shallow MLPs, boosting query speed significantly; (2) The neural split-sum approximation reduces the evaluation of expensive integrals into one look-up in the pre-integrated texture [Karis 2013; Munkberg et al. 2022], as well as one forward pass of an MLP; (3) Our morphable mesh representation enables efficient differentiable rasterization with existing tools [Laine et al. 2020], in contrast to implicit representations that require hundreds of queries per pixel. Thanks to the above components, our method reconstructs detailed relightable avatars in around 15 minutes. Our experiments show that our proposed approach achieves high-fidelity geometry as well as realistic renderings, which are on par with, or superior to, existing approaches while being much faster to train as demonstrated in Figure 3. Code is available for research purposes at https://flare.is.tue.mpg.de.

2 RELATED WORK

2.1 3D head avatars from videos

Creating animatable 3D head avatars from videos is a popular research topic because it replaces the need for complex capture equipment [Beeler et al. 2011; Debevec et al. 2000; Ghosh et al. 2011; Riviere et al. 2020] with more easily accessible commodity sensors. Classic approaches [Garrido et al. 2016; Thies et al. 2016] employ statistical models [Blanz and Vetter 1999] to recover the 3D shape and appearance, but only focus on the facial area and produce relatively coarse reconstructions. NerFACE [Gafni et al. 2021] was the first to use dynamic neural radiance fields (NeRF) [Mildenhall et al. 2020] to represent head avatars. IMavatar [Zheng et al. 2022] recovers accurate geometry using implicit surfaces by jointly learning canonical head geometry and expression deformations. However, methods based on implicit representations can be inefficient to train and render. PointAvatar [Zheng et al. 2023] uses a similar deformation model but employs a point cloud representation, enabling faster rasterization and better image quality. Recently, several methods [Gao et al. 2022; Xu et al. 2023; Zielonka et al. 2023] employ InstantNGP [Müller et al. 2022] to speed up radiance field queries and can reconstruct avatars within 5 to 20 minutes. To the best of our knowledge, none of these fast avatar reconstruction methods produce high-quality surface normals. Neural Head Avatar (NHA) [Grassal et al. 2022] reconstructs mesh-based avatars with complete head and hair geometry. However, the reconstructed geometry is relatively coarse, with many details represented in the texture space. None of these recent neural methods factorize light and albedo, with the exception of PointAvatar, which performs a rudimentary factorization using a diffuse shading model. In contrast, our method reconstructs mesh-based avatars with high-quality geometry within 15 minutes, and factorizes lighting into albedo, roughness and extrinsic illumination. This enables our avatars to be readily rendered in new scenes.



(a) Image quality (b) Geometric quality

Fig. 3. Training time vs image quality (a) and geometric quality (b) for SOTA methods. Our method is nearly as quick as INSTA while performing on-par or better than competitors. Lower is better for (a) and higher is better for (b).

2.2 Relightable 3D reconstruction from multi-view images

The ability to learn relightable 3D assets from 2D observations has extensive applications in AR and VR content creation. Several previous methods [Bi et al. 2020; Boss et al. 2021a; Srinivasan et al. 2021; Verbin et al. 2022; Zhang et al. 2021a,b] leverage neural implicit representations such as NeRF [Mildenhall et al. 2020] or neural SDF [Mescheder et al. 2019; Park et al. 2019], which benefit from unconstrained topology but are inefficient to render. Recently, [Munkberg et al. 2022] convert neural SDFs to meshes with differentiable marching tetrahedrons [Shen et al. 2021] and employ physically-based rendering to reconstruct high-quality relightable 3D assets in less than an hour. Neural-PIL [Boss et al. 2021b] leverages a similar idea to us and approximates parts of the split-sum formula [Karis 2013] with neural networks. However, the method requires pre-training on a large dataset, which hinders generalization, and is only tested with multi-view images that have full coverage of the scene.

To obtain 3D head avatars that are both animatable and relightable, recent methods [Dib et al. 2021; Feng et al. 2022] leverage the deformable geometry of pretrained 3DMMs [Li et al. 2017; Paysan et al. 2009], and predict albedo and lighting from a single image. SIRA [Caselles et al. 2023] improves the coarse 3DMM geometry by learning a deformable SDF but requires a large number of 3D scans for training. In contrast, our method reconstructs relightable 3D head avatars from a single monocular video, achieving high-quality geometry without requiring expensive 3D scans for prior training.

3 METHOD

Given a fixed-viewpoint video of a person with frames $\{I_1, \ldots, I_N\}$, foreground masks $\{M_1, \ldots, M_N\}$, and pre-computed FLAME [Li et al. 2017] parameters for shape β, expressions $\{\psi_1, \ldots, \psi_N\}$, and poses $\{\theta_1, \ldots, \theta_N\}$, we jointly train (1) the deforming head geometry, parameterized by a canonical mesh with optimizable vertex locations and expression deformation fields (Sec. 3.1); (2) the intrinsic surface reflectance properties, including albedo, roughness, and specular intensity, represented by an MLP in canonical space (Sec. 3.2), and (3) a lighting MLP that approximates the pre-filtered environment map of the scene. To train these we follow a physically-based rendering approach and rasterize the mesh into images, which are compared with the ground-truth frames (Sec. 4). Figure 2 gives an overview of the method.

3.1 Geometry

To achieve high train- and test-time efficiency, and for compatibility with standard graphics pipelines, we use triangle meshes as the geometric representation. As shown in Figure 2, we learn a single canonical mesh that best explains all views, along with deformation fields that transform the canonical mesh given FLAME pose and expression parameters. We describe each of these below.

3.1.1 Canonical mesh. Given a pre-defined FLAME template mesh $\mathcal{V} = (V, \mathcal{F})$, with the set of vertices $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_M | \mathbf{v}_i \in \mathbb{R}^3\}$, and the set of triangular faces $\mathcal{F}$, we obtain a personalized shape by optimizing $\{\Delta\mathbf{v}_i | i = 1 \ldots M, \Delta\mathbf{v}_i \in \mathbb{R}^3\}$, such that the final canonical mesh vertices are $\{\mathbf{v}_i + \Delta\mathbf{v}_i | i = 1 \ldots M\}$. To facilitate learning we employ a coarse-to-fine approach [Worchel et al. 2022; Zheng et al. 2023], where we upsample the mesh to $\backsim 11k$ vertices vertices during training using the algorithm in [Botsch and Kobbelt 2004].

3.1.2 Deformation field. We deform the canonical geometry using the FLAME parameters computed during the pre-processing step. Specifically, given a canonical vertex $\mathbf{v} \in \mathbb{R}^3$, we deform it as follows:

$$FLAME(\mathbf{v}, \mathcal{P}, \mathcal{E}, \mathcal{W}, \theta, \psi) = LBS(\mathbf{v} + B_P(\theta; \mathcal{P}) + B_E(\psi; \mathcal{E}), J(\beta), \theta, \mathcal{W}), \quad (1)$$

where $J(\beta)$ is the joint regressor, LBS is the standard linear blend-skinning function with blend-skinning weights $\mathcal{W}$, θ and ψ are the FLAME pose and expression parameters, and $B_P(\cdot)$ and $B_E(\cdot)$ compute the pose and expression offsets using the blendshape components $\mathcal{P}$ and $\mathcal{E}$. Similar to IMavatar [Zheng et al. 2022], we train a deformation network $\mathcal{D}$ parameterized by an MLP that, given a canonical vertex location $\mathbf{v}$, returns the expression blendshapes $\mathcal{E} \in \mathbb{R}^{n_e \times 3}$, the pose correctives $\mathcal{P} \in \mathbb{R}^{n_j \times 9 \times 3}$, and the linear blend skinning weights $\mathcal{W} \in \mathbb{R}^{n_j}$ of the vertex (with n_e and n_j the number of expression parameters and bone transformations, respectively),

$$\mathcal{D}(\mathbf{v}) : \mathbb{R}^3 \rightarrow \mathcal{E}, \mathcal{P}, \mathcal{W}. \quad (2)$$

Note that, while IMavatar requires a costly root-finding process to search for canonical correspondences given deformed ray samples, our mesh formulation avoids this by directly deforming the canonical mesh and rasterizing it.

3.2 Reflectance

To make our avatars relightable, we factorize the observed colors into learned albedo, roughness, and specular intensity, as well as a neural representation of the environment illumination. We adopt the Disney shading model [Burley 2012] and modify it to better suit our input data. We elaborate on the appearance model in the following.

3.2.1 Physically-based rendering. According to the classic rendering equation [Kajiya 1986], the radiance $L_o(\mathbf{x}, \omega_o) \in \mathbb{R}^3$ leaving from a surface point $\mathbf{x} \in \mathbb{R}^3$ with normal vector $\mathbf{n}$ in the direction ω_o is modeled as

$$L_o(\mathbf{x}, \omega_o) = \int_\Omega f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\omega_i)(\mathbf{n} \cdot \omega_i) d\omega_i, \quad (3)$$

where the integral is over the hemisphere $\Omega = \{\omega_i | (\omega_i \cdot \mathbf{n}) > 0\}$, $f_r(\mathbf{x}, \omega_i, \omega_o)$ is the bi-directional reflectance distribution function (BRDF), and L_i is the incoming light intensity from direction ω_i.

Following the dichromatic reflection model [Shafer 1985], the BRDF is decomposed into a diffuse term f_d and a specular term f_s, and the total reflectance is calculated as $f_r(\mathbf{x}, \omega_i, \omega_o) = f_d(\mathbf{x}, \omega_i, \omega_o) + k(\mathbf{x}) f_s(\mathbf{x}, \omega_i, \omega_o)$, where $k(\mathbf{x})$ is a spatially-varying specular intensity factor that weighs the contribution of the specular BRDF, similar to [Riviere et al. 2020]. The rendering equation then becomes:

$$L_o(\mathbf{x}, \omega_o) = \underbrace{\int_\Omega f_d(\mathbf{x}, \omega_i, \omega_o) L_i(\omega_i)(\mathbf{n} \cdot \omega_i) d\omega_i}_{L_o^{diff}} + \underbrace{k(\mathbf{x}) \int_\Omega f_s(\mathbf{x}, \omega_i, \omega_o) L_i(\omega_i)(\mathbf{n} \cdot \omega_i) d\omega_i}_{L_o^{spec}}. \quad (4)$$

We use a simple Lambertian model for the diffuse term:

$$L_o^{diff} = \frac{\rho(\mathbf{x})}{\pi} \int_\Omega L_i(\omega_i)(\mathbf{n} \cdot \omega_i) d\omega_i, \quad (5)$$

where ρ is the spatially-varying RGB albedo. For the specular term we use the Cook-Torrance microfacet model [Cook and Torrance 1982]:

$$L_o^{spec} = \int_\Omega \frac{D(\mathbf{n}, \omega_o, \omega_i, r) G(\omega_o, \omega_i, r) F(\omega_o, \omega_i, F_0)}{4(\omega_o \cdot \mathbf{n})(\omega_i \cdot \mathbf{n})} L_i(\omega_i)(\mathbf{n} \cdot \omega_i) d\omega_i. \quad (6)$$

Here, the surface roughness r modulates the microfacet normal distribution function D, and the geometry attenuation function G that accounts for self-shadowing. F denotes the Fresnel equation that describes the proportion of light reflected at different surface angles. We follow the Disney material model for the specific choice of D, G and F functions, see [Burley 2012; Karis 2013].

3.2.2 Estimating intrinsic materials. We optimize the albedo and roughness of our head model, as well as specular intensity values. These properties are canonical properties of the surface and remain constant during facial deformation. Therefore, we employ an MLP, $\mathcal{M}$, that receives canonical surface points $\mathbf{x_c}$ as input and predicts albedo ρ, roughness r and specular intensity k:

$$\mathcal{M}(\mathbf{x_c}) : \mathbb{R}^3 \rightarrow \rho, r, k. \quad (7)$$

3.2.3 Split-sum approximation. The split-sum approximation [Karis 2013] was proposed to efficiently evaluate the specular reflectance by splitting it into two integrals that can be pre-computed:

$$L_o^{spec} \approx \int_\Omega L_i(\omega_i) D(\mathbf{n}, \omega_i, \omega_o, r)(\omega_i \cdot \mathbf{n}) d\omega_i \int_\Omega f(\omega_i, \omega_o)(\omega_i \cdot \mathbf{n}) d\omega_i. \quad (8)$$

The first term corresponds to a *pre-filtered environment map*, where the environment light L_i is convolved with the normal distribution function D. This term is pre-computed for a set of roughness values and stored as a series of 2D look-up textures (LUT), where each mipmap level is selected based on roughness, and each texture is indexed by the reflection vector $\omega_r = 2(\omega_o \mathbf{n})\mathbf{n} - \omega_o$. The second integral, known as the *BRDF integration map* contains the rest of the terms, and it is equivalent to integrating Equation 6 with a white environment map ($L_i(\omega_i) = 1$) [Karis 2013]. This term depends on

the roughness r and the cosine angle $(\omega_o \cdot \mathbf{n})$, and it is also stored as an LUT, which will be referred to as $FG-LUT$.

3.2.4 Neural split-sum approximation. The split-sum approximation can help to efficiently learn a rich model of illumination, and has been used to disentangle the light and materials from multi-view images [Munkberg et al. 2022]. However, our setting considers as input a fixed viewpoint video, which is a more challenging scenario for light disentanglement. We found through experiments that optimizing environment maps directly often leads to sub-optimal results (See Figure 10). To address this, we approximate the pre-filtered environment map in Eq. 8 with a neural network:

$$\mathcal{L}(\omega_r, r) \approx \int_\Omega L_i(\omega_i) D(\mathbf{n}, \omega_i, \omega_o, r)(\omega_i \cdot \mathbf{n}) d\omega_i. \tag{9}$$

To design this neural network, we observe that the roughness parameter r influences the output via the normal distribution function D, i.e., a larger roughness corresponds to a wider distribution and leads to blurrier filtered light maps. In 2D LUTs, the pre-filtered environment maps for different roughness values are represented as mipmaps. A key challenge for the neural split-sum approximation is to model this behavior for different roughness levels. To address this, we propose to adapt the Integrated Directional Encoding (IDE) [Verbin et al. 2022] to represent different mip levels of neural fields. The IDE encodes the input reflection vector ω_r through the expected value of a set of spherical harmonics under a von Mises-Fischer (vMF) distribution centered at ω_r, where the concentration parameter κ is defined as the inverse roughness $1/r$:

$$IDE(\omega_r, r) = \mathbb{E}_{\omega \sim vMF(\omega_r, 1/r)}[Y_l^m | (l,m) \in \mathcal{M}_L], \tag{10}$$

with $\mathcal{M}_L = \{(l,m) : l = 1 \dots 2^L, m = 0 \dots l\}$, Y_l^m the spherical harmonics basis functions, and $L = 4$. In practice, this positional encoding limits the representational power of the neural network when using larger roughness values, which essentially mimics the behavior of mipmap levels in a continuous manner. Note that the incident illumination L_i, now represented as part of the pre-filtered light MLP $\mathcal{L}$, also determines the diffuse shading. We observe that setting the roughness to its maximum value $r = 1$ within the GGX distribution for D (employed by the Disney material model) equates to $1/\pi$, and the pre-filtered environment map term becomes the diffuse shading of Equation 5. Hence, we can use $\mathcal{L}$ to represent both the diffuse shading and the specular pre-filtered environment map:

$$L_o^{diff} = \frac{\rho(\mathbf{x})}{\pi} \cdot \mathcal{L}(IDE(\mathbf{n}, 1)) \tag{11}$$

$$L_o^{spec} = \mathcal{L}(IDE(\omega_r, r)) \cdot FG - LUT(r, \omega_o \cdot \mathbf{n}). \tag{12}$$

We thus replace the explicit integration of a scene environment map with a single query over the pre-filtered light MLP, while still grounding the formulation on a physics-based model via the $FG{-}LUT$ term. At test time we relight the avatar by simply replacing $\mathcal{L}$ with a pre-filtered environment map.

3.2.5 Color prediction. The final outgoing radiance is calculated as

$$L_o(\mathbf{x}, \omega_o) = \frac{\rho(\mathbf{x})}{\pi} \cdot \mathcal{L}(IDE(\mathbf{n}, 1)) + k(\mathbf{x}) \mathcal{L}(IDE(\omega_r, r)) \cdot FG - LUT(r(\mathbf{x}), \omega_o \cdot \mathbf{n}). \tag{13}$$

4 TRAINING

4.1 Loss Functions

In this section, we discuss the loss functions employed by FLARE, grouped by image-related losses, geometry-related losses, deformation-related losses, and material regularizers.

4.1.1 Image. Given a ground-truth frame I_j and a predicted image $\tilde{I}_j$, we compute (1) the $L2$ loss in log space between the masked ground-truth and the predicted image following [Munkberg et al. 2022]:

$$\mathcal{L}_{RGB} = ||log(I_j) - log(\tilde{I}_j)||_2^2, \tag{14}$$

(2) an L2 loss between ground-truth and predicted binary masks:

$$\mathcal{L}_{mask} = ||M_j - \tilde{M}_j||_2^2, \tag{15}$$

and (3) a perceptual loss [Johnson et al. 2016] given as:

$$\mathcal{L}_{vgg} = ||F_{vgg}(I_j) - F_{vgg}(\tilde{I}_j)||_2^2, \tag{16}$$

where F_{vgg} represents the extracted features from the first four layers of a pre-trained VGG [Simonyan and Zisserman 2015] network.

4.1.2 Geometry. During the optimization of mesh vertices, it is necessary to constrain them in order to avoid self-intersections and to obtain a coherent shape. We follow [Luan et al. 2021; Worchel et al. 2022] and use a Laplacian smoothness regularizer where the magnitude of the differential coordinates of each vertex is minimized. For canonical vertices given by $\{\mathbf{v}_i + \Delta\mathbf{v}_i | i = 1 \dots M\}$, the regularizer is defined as:

$$\mathcal{L}_{laplacian} = \frac{1}{M} \sum_1^M ||\delta_i||_2^2 \tag{17}$$

where $\delta_i = (LV)_i$ are the differential coordinates of the i-th vertex, and $L \in \mathbb{R}^{M \times M}$ the graph Laplacian of the mesh [Sorkine 2005]. Additionally, we apply a normal consistency term [Luan et al. 2021; Worchel et al. 2022] that enforces cosine similarity between neighboring face normals and is given by:

$$\mathcal{L}_{normal} = \frac{1}{|\mathcal{F}|} \sum_{(i,j) \in \mathcal{F}} (1 - \mathbf{n}_i \cdot \mathbf{n}_j)^2. \tag{18}$$

$\mathcal{F}$ is the set of triangle pairs that share an edge, and n_i is the normal of triangle i.

4.1.3 Deformation. We regularize the blendshapes and skinning weights similar to [Zheng et al. 2022] as follows:

$$\mathcal{L}_{flame} = \frac{1}{M} \sum_1^M (\lambda_e ||\mathcal{E}_i - \hat{\mathcal{E}}_i||_2 + \lambda_p ||\mathcal{P}_i - \hat{\mathcal{P}}_i||_2 + \lambda_w ||\mathcal{W}_i - \hat{\mathcal{W}}_i||_2), \tag{19}$$

where $\mathcal{L}_{flame}$ regularizes $\mathcal{E}$, $\mathcal{P}$ and $\mathcal{W}$ with pseudo ground-truth $\hat{\mathcal{E}}$, $\hat{\mathcal{P}}$ and $\hat{\mathcal{W}}$, obtained from the nearest vertex of the FLAME [Li et al. 2017] template. Here, $\lambda_e = 50, \lambda_p = 50, \lambda_w = 2.5$.

4.1.4 Material Regularization. We apply a white light regularization over the diffuse shading as in [Munkberg et al. 2022]:

$$\mathcal{L}_{light} = \frac{1}{3}\sum_{i=0}^{3}|\bar{c}_i - \frac{1}{3}\sum_{i=0}^{3}\bar{c}_i|, \tag{20}$$

where $\bar{c}_i$ is the per-channel average intensity. Additionally, we regularize the specular intensity k by computing the z-score of our predicted specular intensities relative to a Gaussian distribution based on the MERL / ETH Skin Reflectance Database [Weyrich et al. 2006]. The dataset provides specular intensity measurements for 156 faces, with a mean value of 0.3753 and a standard deviation of 0.1655. The regularization is defined as:

$$\mathcal{L}_{spec}(\mathbf{x_c}) = \frac{\mathbf{x_c} - 0.3753}{0.1655}. \tag{21}$$

We employ a similar strategy to regularise the roughness. However, since the statistics reported in [Weyrich et al. 2006] are computed for the Torrance-Sparrow model, we empirically set the mean to $\mu_{rough} = 0.5$ and standard deviation to $\sigma_{rough} = 0.1$ through visual evaluation. We provide an ablation study in Section 5.4.1 to support the choice of this hyper-parameter. The loss is defined as:

$$\mathcal{L}_r(\mathbf{x_c}) = \frac{\mathbf{x_c} - \mu_{rough}}{\sigma_{rough}}. \tag{22}$$

Finally, we enforce a smoothness constraint for both albedo and roughness similar to [Munkberg et al. 2022], with an additional robust term [Barron 2019] that helps preserve high-frequency details. Specifically, for each canonical point $\mathbf{x_c} \in \mathbb{R}^3$ we compute a random displacement vector $\epsilon \in \mathbb{R}^3$ sampled from a Gaussian distribution, and compute the albedo (ρ) and roughness (r) for both points. We apply an L1 loss between these two to enforce smoothness within neighboring points as follows:

$$\mathcal{L}_{smooth}(\mathbf{x_c}) = f_{robust}(||\rho(\mathbf{x_c}) - \rho(\mathbf{x_c} + \epsilon)||_1) \tag{23}$$

$$+f_{robust}(||r(\mathbf{x_c}) - r(\mathbf{x_c} + \epsilon)||_1) \tag{24}$$

where f_{robust} is the adaptive robust loss function of [Barron 2019].

4.1.5 Loss function. The full loss function is as follows:

$$\begin{aligned}\mathcal{L} = \lambda_{RGB}\mathcal{L}_{RGB} + \lambda_{vgg}\mathcal{L}_{vgg} + \lambda_{mask}\mathcal{L}_{mask} + \\ \lambda_{flame}\mathcal{L}_{flame} + \lambda_{laplacian}\mathcal{L}_{laplacian} + \lambda_{normal}\mathcal{L}_{normal} + \\ \lambda_{smooth}\mathcal{L}_{smooth} + \lambda_r\mathcal{L}_r + \lambda_{spec}\mathcal{L}_{spec} + \lambda_{light}\mathcal{L}_{light}\end{aligned} \tag{25}$$

where $\{\lambda_i \in \mathbb{R}\}$ weigh the importance of the corresponding terms and we empirically determined them as: $\lambda_{RGB} = 1.0, \lambda_{vgg} = 0.1, \lambda_{mask} = 2.0, \lambda_{flame} = 5.0, \lambda_{laplacian} = 60.0, \lambda_{normal} = 0.1, \lambda_{smooth} = 0.01, \lambda_r = 0.01, \lambda_{spec} = 0.01, \lambda_{light} = 0.01$.

4.2 Training Details

We train FLARE using differentiable rendering to compare the predicted images with ground-truth frames. Given the current canonical mesh $\{\mathbf{v}_i + \Delta\mathbf{v}_i | i = 1 \dots M\}$, we first estimate expression blendshapes $\mathcal{E}$, pose correctives $\mathcal{P}$ and blend skinning weights $\mathcal{W}$ through a forward pass of the deformation network, $\mathcal{D}(\mathbf{v}_i + \Delta\mathbf{v}_i) \rightarrow (\mathcal{E}, \mathcal{P}, \mathcal{W})$. With the expression and pose parameters ψ, θ, we obtain deformed vertex positions $\tilde{\mathbf{v}}_i$ using the FLAME function in Equation 1, $\tilde{\mathbf{v}}_i = FLAME(\mathbf{v}_i + \Delta\mathbf{v}_i, \mathcal{E}, \mathcal{P}, \mathcal{W}, \theta, \psi)$. Following a deferred shading pipeline, the deformed vertices are rasterized to obtain triangle indices and barycentric coordinates per pixel. We then interpolate and obtain the corresponding canonical point locations $\mathbf{x_c}$, deformed point locations $\mathbf{x_d}$ (used to compute ω_o), and deformed normals $\mathbf{n_d}$ for each pixel. Next, we compute material properties by querying the material network $\mathcal{M}(\mathbf{x_c}) \rightarrow (\rho, r, k)$ (Equation 7). Finally, we query the lighting MLP $\mathcal{L}(\omega_r, r)$ using the deformed normals $\mathbf{n_d}$ to obtain the left-hand side of Equation 8 and the diffuse shading of Equation 5. The final color for the pixel is computed using Equation 13. Our framework is implemented in PyTorch and trained using a single A100 Nvidia GPU with 80GB of memory and a batch size of 4 images per iteration.

4.2.1 Two-stage training. To learn high-frequency facial features and to enable fast rendering, we incorporate hash-grid encoding [Müller et al. 2022] for the material MLP, $\mathcal{M}$. However, we found that this approach overfits to colors quickly, learning texture much faster than the geometry, resulting in smoother shapes of lower quality. To address this, we design a two-stage training approach. During the first stage, we equip the material MLP with positional encoding [Mildenhall et al. 2020] and jointly optimize the geometry $\Delta\mathbf{v}_i$, deformation $\mathcal{D}$, material $\mathcal{M}$, and lighting $\mathcal{L}$. The first stage can achieve detailed geometry but often learns blurry texture. During the second stage of training, we leverage the pre-trained mesh geometry and deformation from the previous stage and re-optimize both material and lighting MLPs, where $\mathcal{M}$ is now equipped with high-resolution hash-grid encoding [Müller et al. 2022]. With the proposed two-stage training, our method can achieve both high-fidelity geometry and realistic texture (See Fig. 11).

5 EVALUATION

In this section, we present qualitative and quantitative results of FLARE. First, we show qualitative examples of the individual components, including geometry, albedo, roughness, diffuse and specular shading, as well as relit images (Sec. 5.2). Next, we compare our results with the state-of-the-art (SOTA) baselines in terms of image quality and albedo, as well as geometric accuracy (Sec. 5.3). Finally, we conduct an ablation study to evaluate our design choices in Sec. 5.4. All the results in this section are generated using frames from the test set. For each test frame, we obtain FLAME parameters (pose and expression) from the pre-processing step and animate the personalized canonical representation (geometry) estimated by each baseline method. These animated renderings are relit with novel environment maps. We include additional results in the supplementary video.

5.1 Dataset

We use 2 subjects released by [Zheng et al. 2022], 2 by [Zheng et al. 2023] (where 1 subject is captured by a webcam), and 1 by [Zielonka et al. 2023]. We additionally capture 15 subjects with a smartphone to demonstrate the robustness of FLARE for diverse skin tones and shapes. We follow the protocol of [Zheng et al. 2022, 2023] for the capture and obtain an average of 3000-4000 frames for training and around 1000-3000 frames for testing. These new subjects gave prior informed written consent for their data to be used for academic research purposes. In total, we conduct the evaluations for 20 subjects. To measure geometric accuracy we use a dataset

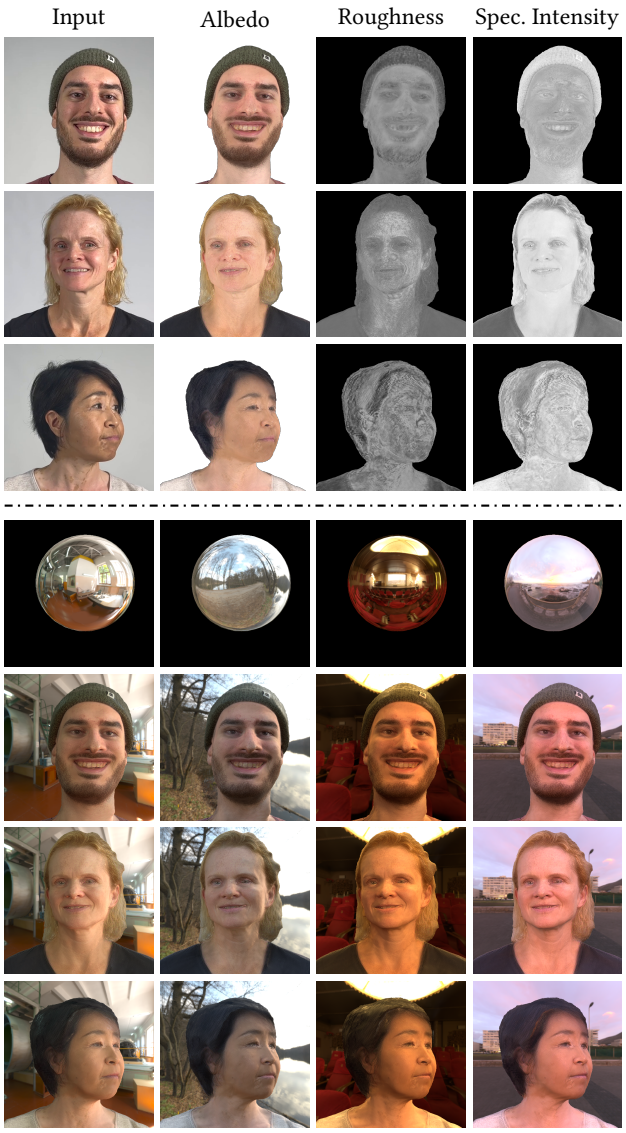


Fig. 4. Qualitative results. The first three rows illustrate our intrinsic material estimates (albedo, roughness, and specular intensity) for three different subjects. The next three rows show the above subjects in the same pose and expression under 4 different environment maps.

of synthetic heads [Briceno and Paul 2019; Grassal et al. 2022], in which each head has 200 frames for training and 200 frames for testing.

5.2 Qualitative Evaluation: Intrinsic Materials and Relighting

The intrinsic material properties (albedo, roughness, and specular intensity) and relit faces are visualized in Figures 4 and 5. The rendered albedo images in Figure 4 illustrate that FLARE is capable of removing evident shadows and specular highlights in the face region; e.g., see the subject in the second row. The influence of the predicted roughness values can be visualized in the relit images: the teeth of the subject in the first row and the hair of the subject in the second row are correctly predicted as shiny surfaces (lower roughness values), which results in realistic reflections when relit with new environment lighting. Finally, the robustness of FLARE is demonstrated across different skin tones, skin textures, hair types, hair styles, facial hair, and even accessories such as a cap. Despite having a single monocular video as input, FLARE computes geometries and materials that enable realistic and plausible relighting.

5.2.1 Comparison with PointAvatar. To the best of our knowledge, PointAvatar [Zheng et al. 2023] is the only other neural avatar method trained from a monocular video that disentangles diffuse shading from albedo. Thus, we qualitatively evaluate the albedo and shading of FLARE in Figure 5 by comparing it with PointAvatar. We relight the renderings of PointAvatar using a Lambertian shading model, where we use the predicted surface normals to obtain diffuse shading. From Figure 5, we observe that the albedo estimated by PointAvatar is biased towards light skin tones and fails to capture the skin color of the subjects. In comparison, the albedo estimated by FLARE resembles the color of the subject, and much of the shading is removed.

Relighting FLARE 's estimated materials results in natural looking images. This is due, in part, to the estimated specular highlights, which are absent in PointAvatar's formulation. The specular highlights are visible in the 5th row of Figure 5, on the left and right cheeks of the first subject from the left, and in the rightmost subject, who has smooth and shiny hair that reflects the environment's light.

5.3 Comparisons with State-of-the-Art

In this section, we compare the results of FLARE with the following state-of-the-art (SOTA) methods for neural head avatar estimation from videos: (1) IMavatar [Zheng et al. 2022] and (2) PointAvatar [Zheng et al. 2023], which use a deformation module similar to ours, with a signed distance function (SDF) and point cloud representation for geometry, respectively; (3) NHA [Grassal et al. 2022], which employs a mesh representation along with an alternating training strategy between geometry and color; and (4) INSTA [Zielonka et al. 2023], which learns an animatable avatar using a NeRF [Mildenhall et al. 2020] representation and leverages the InstantNGP framework [Müller et al. 2022] for faster optimization. NHA and PointAvatar employ test-time optimization of the expression and pose parameters due to noisy pre-processing estimates. Hence, we report the optimized quantitative evaluations for both NHA and PointAvatar to retain their best performance. However, it must be noted that the reported results of FLARE, IMavatar, and INSTA are *not* optimized at test time.

5.3.1 Image quality. First, we compare FLARE with SOTA methods with respect to image quality. We use the same FLAME parameters sampled from the test set on all baselines[1] and measure the accuracy against the ground-truth frames by using mean absolute error (L1 distance), PSNR, structural similarity index measure (SSIM) [Wang

[1]INSTA uses a different pre-processing pipeline and the estimated FLAME parameters are different, see Appendix B. However, the test frames and conveyed expressions remain the same.

Fig. 5. Qualitative comparision with PointAvatar. The first two rows show albedo and diffuse shading estimated by FLARE compared with PointAvatar [Zheng et al. 2023]. The next row shows the roughness and specular intensity (Spec. Intensity) estimated by FLARE for the same subjects as above. The bottom two rows contain relighting results of FLARE and PointAvatar for the same subjects, animated with test poses and expressions.

et al. 2004], and perceptual loss (LPIPS) [Zhang et al. 2018]. The evaluations are computed only on the masked regions for all the methods. Qualitative results can be found in Figure 6, and quantitative results are shown in Table 2. We make the following observations in comparison to prior art: (1) In terms of image quality, the baselines perform approximately on par with each other, with FLARE obtaining the highest score over half the metrics and second highest over the other half. (2) Despite PointAvatar's ability to capture high-frequency texture details, the point cloud representation, containing approximately 400k points, is susceptible to sparsity at extreme jaw or neck poses. From Figure 6, 4th row and 5th column, we can observe the artifacts that occur at extreme poses, producing a salt-and-pepper-like noise. In comparison, our mesh representation inherently solves the sparsity issue with approximately 11k vertices (we evaluate mesh resolution in Sec. 5.4.4). (3) FLARE is able to capture high-frequency texture details better than IMavatar and this is evidenced qualitatively as well as quantitatively, where IMavatar has the weakest LPIPS score. (4) INSTA can converge quickly and produces visually convincing expressions and high-quality texture with forward-facing poses. However, at extreme neck poses we observe noisy texture, possibly due to the volumetric representation that fails to extrapolate well. (5) NHA also employs a mesh-based representation to learn the geometry. However, the predicted mesh is unable to capture high-fidelity details as well as the baselines and produces an over-smoothed representation. We believe that this is a result of their training strategy in which the geometry is primarily supervised with pseudo-normals from [Abrevaya et al. 2020], unlike the rest of the baselines, which learn geometry exclusively via inverse rendering. Instead, we carefully consider how fast the texture network is trained compared to the geometry network, and we observe that this was helpful in achieving high-fidelity geometry. We evaluate our training strategy further in Sec. 5.4.

Table 2. Quantitative comparisons in terms of image quality on real data. The evaluations are performed only on the face region. Red color indicates the best value, yellow second best, and light yellow is the third best.

	L1 ↓	LPIPS ↓	SSIM ↑	PSNR ↑
IMavatar	0.0290	0.2091	0.8491	23.0975
NHA	0.0265	0.1243	0.8390	22.7071
PointAvatar	0.0234	0.1400	0.8391	24.6520
INSTA	0.0290	0.1607	0.8379	23.6279
Ours	0.0245	0.1225	0.8421	24.7845

Fig. 6. Qualitative comparison between FLARE and state-of-the-art methods. The canonical representation of each baseline method is animated using test poses and expressions. Odd columns: generated images. Even columns: generated normals.

5.3.2 Geometric Accuracy. We quantitatively evaluate the geometry quality on synthetic heads using the renderings generated by the authors of NHA with the open source *MakeHuman* project [Briceno and Paul 2019]. Geometric accuracy is measured using the cosine similarity between the ground truth and predicted normals. Results are shown in Table 3 and Figure 7. IMavatar and FLARE exhibit high-fidelity normals that resemble the input identity and obtain

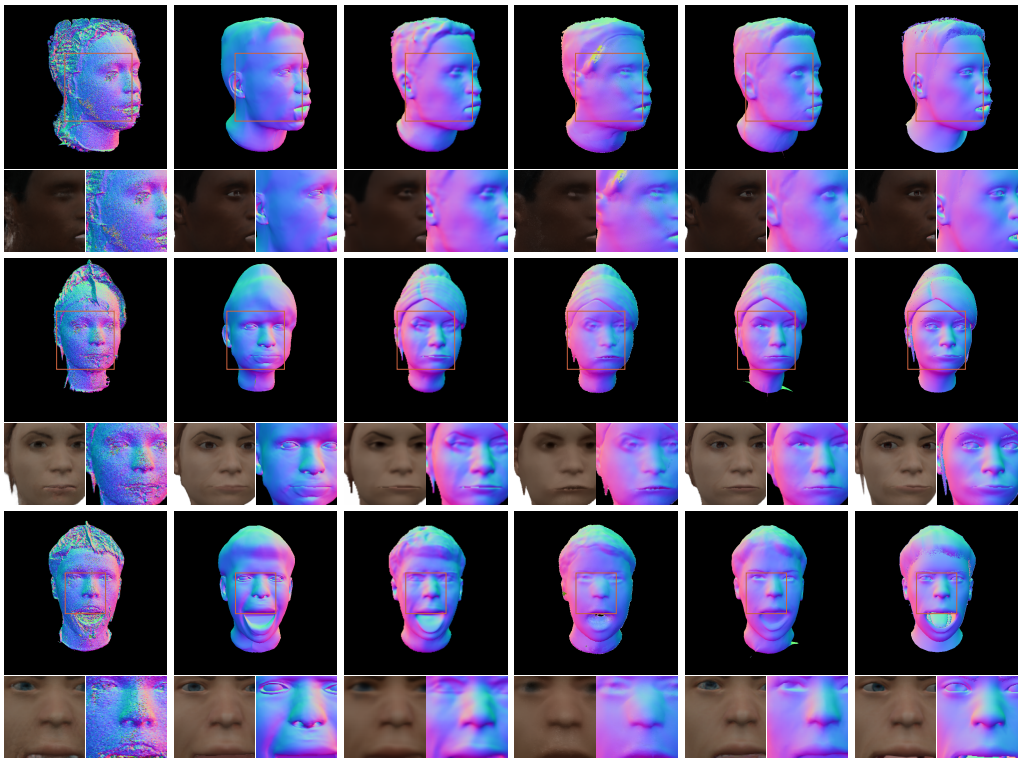


Fig. 7. Qualitative comparisons on synthetic data. The estimated surface normals and texture on synthetic data are compared with state-of-the-art methods by animating the canonical representation using test poses and expressions. Our method can capture high-fidelity geometry as well as color. GT = Ground Truth.

Table 3. Quantitative comparisons in terms of geometric accuracy on a synthetic dataset. Showing cosine similarity compared to ground-truth normals (higher is better). Red color indicates the highest value, yellow second highest and light yellow is third.

	Female 1	Female 2	Male 1	Male 2
IMavatar	0.961	0.966	0.954	0.955
NHA	0.94	0.95	0.94	0.94
PointAvatar	0.954	0.954	0.944	0.958
INSTA	0.665	0.751	0.757	0.713
Ours	0.950	0.955	0.948	0.953

relatively close scores quantitatively. However, IMavatar is approximately 200 times slower to train than FLARE, mainly due to the root-finding step during ray tracing between deformed and canonical points. Moreover, it uses an SDF representation that requires a post-processing step to obtain a mesh, while FLARE can be trained in approximately 15 minutes and directly produces a canonical mesh that can be animated. INSTA, on the other hand, exhibits noisy shapes that can be observed in both Figure 7 and Figure 6, and the normals of NHA do not completely capture the identity. Note that the synthetic heads have smooth geometry and, consequently, most methods do well with only small numerical differences between methods.

5.3.3 Training Time. Figure 3 plots the training time of each method against image quality (LPIPS) and geometric quality (cosine similarity). The plot is measured over the same data as Tables 2 and 3. We find that FLARE can be trained almost as quickly as INSTA but with better performance in terms of image quality and state-of-the-art results in terms of geometry.

5.4 Ablation Study

5.4.1 Loss Functions. We evaluate the contribution of the terms in the loss function that are not adopted by prior avatar methods but are crucial in our setting.

Specular Intensity Regularization. $\mathcal{L}_{spec}$: The specular intensity k controls the intensity of the specular highlights. In Figure 8 we qualitatively evaluate the effectiveness of using the regularizer and show relighting results for one subject with and without the specular intensity regularization. We observe the occurrence of unnaturally sharp highlights that have high intensity around the subject's lower lip and cheek regions when specular intensity is left unconstrained. Constraining it with $\mathcal{L}_{spec}$ makes the non-Lambertian effects more subtle and natural.

Roughness Regularization $\mathcal{L}_r$*:* To regularize the roughness we employ a statistical approach similar to specular intensity. However, we know of no suitable database of statistical values for roughness that can be used to regularize the appearance model. Hence, we

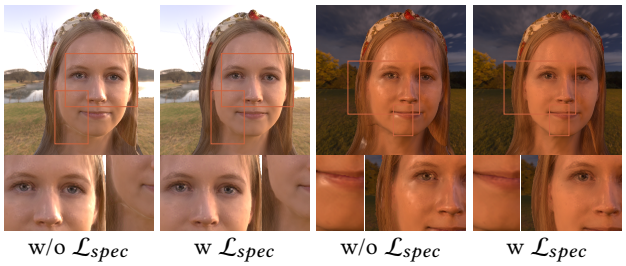


Fig. 8. Ablation of $\mathcal{L}_{spec}$. Qualitative comparison of relighting results with and without the specular intensity regularizer. Results indicate that it is necessary to constrain the specular intensity statistically to avoid unrealistically sharp highlights.

Table 4. Ablation of $\mathcal{L}_r$. Influence of $\mathcal{L}_r$ on image quality when using different mean roughness values.

Mean Roughness	L1 ↓	LPIPS ↓	SSIM ↑	PSNR ↑
0.3	0.0257	0.1014	0.8772	24.341
0.4	0.0250	0.1031	0.8744	24.268
0.5	**0.0239**	**0.0941**	0.8834	24.847
0.6	0.0225	0.0954	**0.8824**	**25.023**
0.7	0.0247	0.0985	0.8790	24.471
w/o regularization	0.0265	0.1028	0.8738	23.984

Fig. 9. Ablation of $\mathcal{L}_r$. Qualitative comparison of relighting results with and without the roughness regularizer. Results indicate that it is necessary to constrain the roughness to ensure non-Lambertian reflections on the skin look plausible.

employ an empirical mean with a fixed standard deviation of 0.1, and evaluate the results of using different mean values in Table 4. Additionally, we evaluate the results of not using this regularization and show qualitative results in Figure 9. We observe a similar behavior as with specular intensity when roughness is left unconstrained. The final numerical prediction of each subject is not affected by a large margin since the network learns to compensate for wrong predictions with other estimations. However, Figure 9 reveals that the regularizer helps produce visually realistic renderings.

5.4.2 Standard PBR vs. Lighting MLP. We compare our proposed approach with a method that estimates a standard texture-based environment map for training, which will be referred to as "Standard PBR". For this experiment we use the same hyper-parameters, loss functions, training protocol, and geometric representation of our method, and replace the pre-filtered light MLP $\mathcal{L}$ with a learnable texture of the environment map, where the integral is solved with the approach proposed by [Munkberg et al. 2022]. In Figure 10 we visualize an example of geometry and relighting obtained with both methods. We observe that the standard PBR results in noisy texture and geometry predictions that are evident after relighting the subject. This is probably due to the redundant calculations of the regions in the environment map that are never observed in our monocular setting, creating instability in the optimization process. Further, we can observe that the input image in Figure 10 (bottom left) is captured such that the main light source is from the right of the subject. However, the environment maps have the main light source coming from the left. Here, PBR exhibits shadows in the texture that are retained from the original input data; for instance, see the shadowing on the nose. This is also observed in the estimated albedo, and it is not prominent in our results.

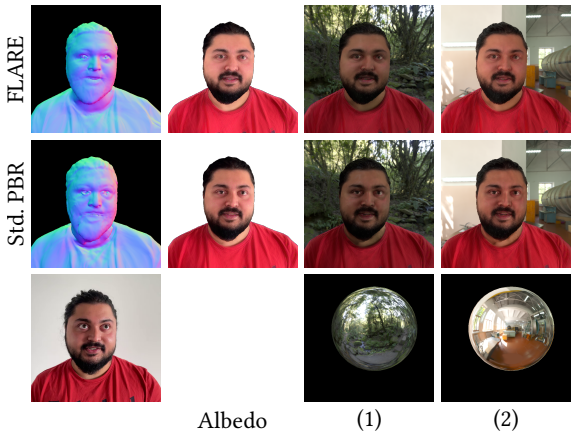


Fig. 10. Ablation Study, Comparison against learning a full environment map ("standard PBR"), as in [Munkberg et al. 2022]. Using this representation typically results in noisier geometry and color. From left to right: predicted geometry, predicted albedo, relighting under two different environment maps. The bottom row shows the input test image, and the two environment maps.

5.4.3 Two-stage training. Through the course of our experiments, we noticed that it is necessary to control the speed at which the texture is learned in order to obtain both good geometry and albedo. In particular, using a hash-grid positional encoding [Müller et al. 2022] results in better image quality, and the method converges very fast. However, this results in noisier geometries since there is not enough gradient signal coming from the color supervision. This behavior can be observed in the first column of Figure 11, where a high-quality rendered image corresponds to a relatively noisy geometry. On the other hand, using a standard positional encoding [Mildenhall et al. 2020] (second column in Figure 11) converges slower and leads to blurry textures, but learns geometric details from the observed images. Our two-stage training approach achieves the best of both options, as shown in the last column of Figure 11.

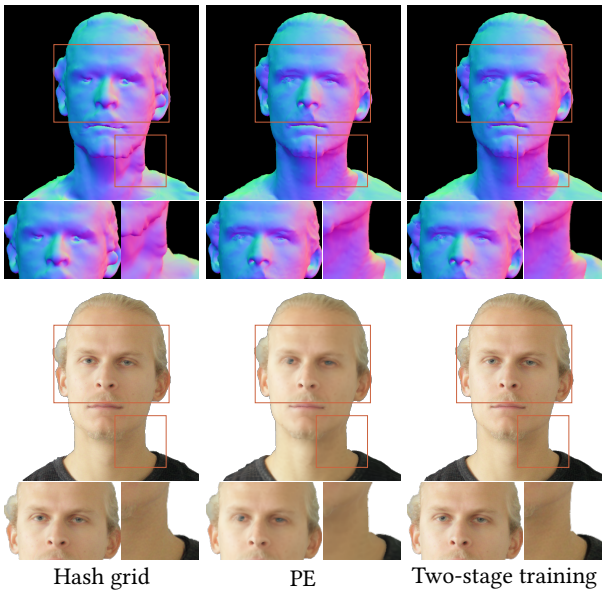


Fig. 11. Ablation Study. Qualitative comparison between the hash-grid encoding of [Müller et al. 2022], the positional encoding of [Mildenhall et al. 2020] ("PE"), and our two-stage approach. Top row: estimated normals; bottom row: estimated rendering.

Fig. 12. Ablation Study: Mesh resolution. Qualitative comparison between surface normals of two subjects with and without upsampling the mesh. This figure demonstrates that upsampling the FLAME mesh to roughly 11K vertices helps capture high-fidelity geometry.

5.4.4 Mesh Upsampling. The FLAME mesh contains 5023 vertices that model the face and neck region, without hair or shoulders. In our setting, we learn the geometry of the subjects including diverse hair types and hairstyles, facial hair, head accessories, and part of the shoulder. However, optimizing with only 5023 vertices results in a smooth coarse geometry, as illustrated in Figure 12. The output mesh appears smooth as the vertices around the shoulder and hair region are stretched out to form triangles occupying a large area. To capture the high-fidelity geometric details of the subject, we increase the resolution by upsampling the mesh [Botsch and Kobbelt 2004] to around 11k vertices. This improves the quality in the hair, neck, and shoulder regions. Note that our resolution is lower than the roughly 16K vertices used by NHA, yet our geometric quality is higher.

Fig. 13. Limitations. Modeling the mouth interior and eyes are challenging due to their complex material properties, variation in appearance (e.g. subjects 1, 2, and 3 have different-sized teeth), and the fact that we do not model eye blinks (e.g. subject 3). Capturing sharp specular highlights is also challenging due to the approximations made by our lighting model (subjects 3 and 4).

6 LIMITATIONS AND FUTURE WORK

FLARE can be trained in around 15 minutes and produces competitive results compared to methods that generate high-fidelity geometry at the expense of longer training times (on the order of days). However, there are still limitations, as shown in Figure 13. Firstly, the quality of the eyes and mouth interior needs improvement. These are challenging areas due to their complex material properties, and most neural avatar methods currently struggle with modeling these. For the mouth interior, an additional challenge comes from the fact that the teeth are exposed to varying degrees during training and this varies significantly between subjects. When a person does not smile with their teeth or does not articulate sufficiently, then the model does not have enough information to correctly reproduce the tooth color and geometry. Further, the FLAME mesh does not have vertices in the mouth interior and thus, during rasterization, there are no vertices projected onto the image of the mouth, resulting in no gradient being propagated there. Our remeshing step partly addresses this problem and, for some subjects, there are vertices formed around the teeth. However, modeling the teeth remains a challenging task due to the constant motion of the lips and limited supervision. Similarly, the eye area exhibits challenging photometric properties that are not always captured by our method. In addition, our pre-processing step does not track eye blinking, resulting in inevitable errors during optimization that yield a noisy geometry around the eyes. Future work should develop techniques that can enhance the estimation of the mouth and eye area, in both photometric and geometric respects.

Second, capturing harsh neck shadows, self-shadows, and sharp specular highlights is difficult as demonstrated in Figure 13. We can remove shadows cast on the face region as the subject moves their head in various directions. However, the shoulder and neck areas remain mostly static and shadows are baked in. Additionally, although non-Lambertian reflections that look plausible can be captured by our method during relighting due to the estimated materials, we miss reproducing the sharp specular highlights of the ground truth. This is due to the several approximations that we make to model the pre-filtering of the environment and to simplify

the integral of the rendering equation. Finally, our method does not model more subtle skin properties such as sub-surface scattering, or time-dependent appearance changes. We hypothesize that this could enhance realism and, consequently, the estimated geometry. We believe this is an interesting direction to pursue in the future.

7 ETHICS

The goal of FLARE is to enable fast, subject-specific, avatar creation that can be used to generate novel expressions and to place the avatars in different scenes. This capability, however, opens the door to potential misuse, where new malicious content of the training subject can be generated without their consent. Although the quality of FLARE still exhibits identifiable artifacts signaling its AI origin, the rapid progression of the field suggests these cues may diminish over time. Addressing this remains an important technical and legal challenge.

8 CONCLUSION

In this work we presented FLARE, a new method for building animatable and relightable head avatars from monocular video in 15 minutes. Our approach directly produces a mesh representation that can be efficiently rendered and animated, along with material parameters that allow the avatars to be placed in scenes under novel illumination. This is achieved by combining traditional computer graphics methods for rendering with neural networks that approximate some of the components. More specifically, we optimize a canonical mesh geometry while approximating the expression deformations, albedo, roughness and specular intensity values using coordinate-based MLPs. Further, we avoid explicitly computing an environment map from a narrow field of view by approximating the pre-filtered environment map in the split-sum formulation with a neural network. Finally, we propose a two-stage approach designed to control the pace at which geometry and texture are learned relative to one another. Our experimental results show that we can obtain mesh avatars of high geometric and image fidelity. Once learned, the avatars can be readily inserted and rendered in arbitrary scenes using standard graphics pipelines, enabling downstream applications in gaming, film production and telepresence.

ACKNOWLEDGMENTS

We thank Jacob Munkberg, Jon Hasselgren, and Pramod Rao for fruitful discussions and Wojciech Zielonka for discussions regarding the baseline. We thank Asuka Bertler, Claudia Gallatz, Taylor McConnell, and Markus Höschle for their support with data collection and thank all the participants for their time. We thank Haoran Yun, Peter Kultis, and Nikos Athanasiou for additional support.

Disclosure: MJB has received research gift funds from Adobe, Intel, Nvidia, Meta/Facebook, and Amazon. MJB has financial interests in Amazon, Datagen Technologies, and Meshcapade GmbH. While MJB is a consultant for Meshcapade, his research in this project was performed solely at, and funded solely by, the Max Planck Society.

REFERENCES

Victoria Fernandez Abrevaya, Adnane Boukhayma, Philip H.S. Torr, and Edmond Boyer. 2020. Cross-Modal Deep Face Normals With Deactivable Skip Connections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jonathan T Barron. 2019. A General and Adaptive Robust Loss Function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4331–4339.

Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W Sumner, and Markus Gross. 2011. High-Quality Passive Facial Performance Capture using Anchor Frames. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*. 1–10.

Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. 2020. Neural Reflectance Fields for Appearance Acquisition. *arXiv preprint arXiv:2008.03824* (2020).

Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*. 187–194.

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. 2021a. NeRD: Neural Reflectance Decomposition from Image Collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 12684–12694.

Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. 2021b. Neural-PIL: Neural Pre-integrated Lighting for Reflectance Decomposition. In *Conference on Neural Information Processing Systems (NeurIPS)*, Vol. 34. 10691–10704.

Mario Botsch and Leif P. Kobbelt. 2004. A Remeshing Approach to Multiresolution Modeling. In *Eurographics Symposium on Geometry Processing*.

Leyde Briceno and Gunther Paul. 2019. MakeHuman: A Review of the Modelling Framework. In *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)*. Springer International Publishing, Cham, 224–232.

Adrian Bulat and Georgios Tzimiropoulos. 2017. How Far Are We from Solving the 2D & 3D Face Alignment Problem? (And a Dataset of 230,000 3D Facial Landmarks). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. https://doi.org/10.1109/iccv.2017.116

Brent Burley. 2012. Physically-Based Shading at Disney.

Pol Caselles, Eduard Ramon, Jaime Garcia, Xavier Giro-i Nieto, Francesc Moreno-Noguer, and Gil Triginer. 2023. SIRA: Relightable Avatars from a Single Image. In *Winter Conference on Applications of Computer Vision (WACV)*.

Robert L Cook and Kenneth E. Torrance. 1982. A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics (TOG)* 1, 1 (1982), 7–24.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the Reflectance Field of a Human Face. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*. 145–156.

Abdallah Dib, Cedric Thebault, Junghyun Ahn, Philippe-Henri Gosselin, Christian Theobalt, and Louis Chevallier. 2021. Towards High Fidelity Monocular Face Reconstruction with Rich Reflectance using Self-Supervised Learning and Ray Tracing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 12819–12829.

Haiwen Feng, Timo Bolkart, Joachim Tesch, Michael J. Black, and Victoria Abrevaya. 2022. Towards Racially Unbiased Skin Tone Estimation via Scene Disambiguation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. 2021. Learning an Animatable Detailed 3D Face Model from In-The-Wild Images. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 40, 4, 1–13.

Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2021. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8649–8658.

Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. 2022. Reconstructing Personalized Semantic Facial NeRF Models From Monocular Video. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 41, 6 (2022).

Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016. Reconstruction of Personalized 3D Face Rigs from Monocular Video. *ACM Transactions on Graphics (TOG)* 35, 3 (2016), 1–15.

Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec. 2011. Multiview Face Capture using Polarized Spherical Gradient Illumination. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 1–10.

Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. 2022. Neural Head Avatars from Monocular RGB Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

James T Kajiya. 1986. The Rendering Equation. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*. 143–150.

Brian Karis. 2013. *Real Shading in Unreal Engine 4.* Technical Report. Epic Games.

Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson W.H. Lau. 2022. MODNet: Real-Time Trimap-Free Portrait Matting via Objective Decomposition. In *AAAI*

Conference on Artificial Intelligence.
Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. 2022. Realistic One-shot Mesh-Based Head Avatars. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
Diederik P Kingma and Jimmy Ba. 2015. ADAM: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.
Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. 2020. Modular Primitives for High-Performance Differentiable Rendering. *ACM Transactions on Graphics* 39, 6 (2020).
Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. 2017. Learning a Model of Facial Shape and Expression from 4D Scans. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 36, 6 (2017), 194:1–194:17. https://doi.org/10.1145/3130800.3130813
Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. 2021. Unified Shape and SVBRDF Recovery using Differentiable Monte Carlo Rendering. In *Computer Graphics Forum (Eurographics Symposium on Rendering)*.
Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4460–4470.
Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 41, 4, Article 102 (July 2022), 15 pages.
Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Mueller, and Sanja Fidler. 2022. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 165–174.
Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. 2009. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 296–301.
Jérémy Riviere, Paulo Gotardo, Derek Bradley, Abhijeet Ghosh, and Thabo Beeler. 2020. Single-Shot High-Quality Facial Geometry and Skin Appearance Capture. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*. Association for Computing Machinery (ACM).
Steven A Shafer. 1985. Using Color to Separate Reflection Components. *Color Research & Application* 10, 4 (1985), 210–218.
Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. 2021. Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*.
Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*.
Olga Sorkine. 2005. Laplacian Mesh Processing. *Eurographics (State of the Art Reports)* 4, 4 (2005).
Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. 2021. NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7495–7504.
Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2387–2395.
Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. 2022. Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).
Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. 2007. Microfacet Models for Refraction Through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*. 195–206.
Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861
Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, et al. 2006. Analysis of Human Faces Using a Measurement-Based Skin Reflectance Model. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 1013–1024.
Markus Worchel, Rodrigo Diaz, Weiwen Hu, Oliver Schreer, Ingo Feldmann, and Peter Eisert. 2022. Multi-View Mesh Reconstruction with Neural Deferred Shading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6187–6197.
Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. 2023. AvatarMAV: Fast 3D Head Avatar Reconstruction Using Motion-Aware Neural Voxels. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*.
Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. 2021a. PhySG: Inverse Rendering with Spherical Gaussians for Physics-Based Material Editing and Relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 586–595.
Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. 2021b. NeRFactor: Neural Factorization of Shape and Reflectance under an Unknown Illumination. In *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*.
Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. 2022. I M Avatar: Implicit Morphable Head Avatars from Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J. Black, and Otmar Hilliges. 2023. PointAvatar: Deformable Point-based Head Avatars from Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Wojciech Zielonka, Timo Bolkart, and Justus Thies. 2022. Towards Metrical Reconstruction of Human Faces. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer International Publishing.
Wojciech Zielonka, Timo Bolkart, and Justus Thies. 2023. Instant Volumetric Head Avatars. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

A IMPLEMENTATION DETAILS

FLAME Mesh and Deformation Network. We manually add mesh faces to the FLAME template mesh between the upper and lower lips to close the mouth cavity, similar to NHA [Grassal et al. 2022]. Additionally, we also simplify the tessellated eye region of the FLAME template following [Zielonka et al. 2023]. Similar to PointAvatar [Zheng et al. 2023], during training, we map the optimized mesh vertices to a canonical pose with jaw open and a neutral expression and then proceed to perform LBS to deform the mesh. This additional step encourages the canonical mesh to have an open-mouth expression, which facilitates the learning of mouth movements. We train the deformation MLP $\mathcal{D}$ only during the first stage with a learning rate of 10^{-3} and use the Adam optimizer [Kingma and Ba 2015]. We adopt the network architecture of PointAvatar which is similar to [Zheng et al. 2022], except we do not predict additional vertex displacements (only skinning weights, expression and pose blendshapes). During the second stage, we freeze the deformation network and use the weights from the first stage.

Optimization of Mesh Vertices. The canonical mesh is upsampled once during training, resulting in a final mesh of approximately 11K vertices. During the first stage of training, when the number of vertices increases, we reduce the learning rate of the vertex offsets from 10^{-3} to $10^{-3} * 0.75$ and increase the weight of the Laplacian and normal regularizer by 4 times following [Worchel et al. 2022]. This helps in learning a smoother mesh and prevents the vertices from diverging after the upsampling step. During the second stage, we set the learning rate of the vertex offsets to a very small value (10^{-5}) and initialize the training with the canonical mesh from the previous stage.

Texture Estimation. For the Material MLP $\mathcal{M}$ during the first stage, we use a ReLU MLP of 4 hidden layers with 128 neurons each. For the final layer, we use the sigmoid activation function. For the second stage, since we use the hash-grid positional encoding, we adopt a smaller network architecture of 2 hidden layers of 64 neurons each with the same activation functions as before. We set the learning rate at 10^{-3} and use the Adam optimizer. The Fresnel coefficient is set to $F_0 = 0.047$ during the first stage following [Karis 2013]. During the second stage, since the geometry is well-estimated and the rendered shape aligns well with the ground truth, we use a segmentation mask and explicitly set the Fresnel coefficient of the skin region to $F_0 = 0.028$, while the rest is set to a constant of $F_0 = 0.047$ (Fresnel coefficient for hair). For the lighting MLP $\mathcal{S}_\psi$, we use a ReLU MLP with 2 hidden layers with 64 neurons each for both training stages. However, we do not use an activation function for the output layer as we learn these computations in the sRGB log space of tone-mapped RGB colors.

B DATA PRE-PROCESSING

We use the pre-processing pipeline of IMavatar [Zheng et al. 2022] where the segmentation masks are obtained from MODNet [Ke et al. 2022], the FLAME parameters (shape, pose and expression) and the camera parameters are estimated using DECA [Feng et al. 2021] and later refined by fitting to 2D facial keypoints [Bulat and Tzimiropoulos 2017]. For IMavatar, NHA, PointAvatar, and our method, we use the same pre-processed data. We refrain from using the same pre-processed data for INSTA [Zielonka et al. 2023] because the method does not optimize for pose parameters and instead translates the camera. Moreover, it has a depth-supervision loss where the depth maps are generated from the shape parameter of FLAME, and the preprocessing used by INSTA (MICA: [Zielonka et al. 2022]) is quantitatively better than the shape estimation of DECA [Feng et al. 2021]. Thus, to get the best results for INSTA, we use the pre-processing released by the authors. Additionally, the synthetic dataset [Briceno and Paul 2019] contains dynamic neck motions, and INSTA's pre-processing fails to estimate the camera pose correctly for many frames. Hence, for INSTA, we only evaluate the frames that have a mask overlap of more than 90%.