D-RayAvatar: Dynamic Ray-tracing based Relightable Gaussian Avatar from Monocular Videos

Zhe Fan, Shi-Sheng Huang, ZhaoChen Li, Hua Huang, *Senior Member, IEEE*

***Abstract*—The success of 3D Gaussian Splatting has inspired a series of Gaussian-based 4D avatars, some of which enable remarkable relighting. However, since the original 3DGS is limited to capture advanced illumination, the relighting quality of the latest 4D Gaussian avatars still need to be improved. The recent Gaussian ray-tracer has shown impressive potential to capture secondary ray-based illumination for *static* scenes, but costs too huge computation to be directly applied for *dynamic* avatars. In this paper, we propose a new Dynamic Ray-tracing based Gaussian Splatting (D-RayGS), which tailors for relightable Gaussian avatar reconstruction to simulate the inter-reflective illumination between Gaussian primitives for high-fidelity avatar inverse rendering, while performing the Gaussian ray-tracing in a fast speed. Based on the D-RayGS, we introduce a highly accurate joint learning of both the dynamic geometry, BRDF materials and lighting, using a *compact* regularization from an extra signed distance field. Benefiting from the D-RayGS and learning, we build a relightable Gaussian avatar reconstruction from monocular videos, i.e., D-RayAvatar, which enables high fidelity relightable rendering quality while performing the dynamic rendering efficiently. Extensive evaluation on public datasets show that our D-RayAvatar can achieve much better Gaussian avatar reconstruction in both geometry reconstruction quality and relightable rendering quality, while achieving fast rendering speed as a new state-of-the-art relightable Gaussian avatar reconstruction approach.**

***Index Terms*—Relightable Gaussian avatar, Gaussian ray-tracing, compact 4D Gaussian avatar reconstruction.**

I. Introduction

The accurate reconstruction of *animatable* head avatars [1]–[3] from monocular videos continues to be an active but challenging research topic in computer graphics and computer vision communities. We have seen remarkable progress made for monocular video based avatar reconstruction [4]–[7] based on NeRF [8]–[10], which enables impressive applications such as novel view synthesis [5], novel expression [11]–[14] and novel pose control [15], [16]. Moreover, the recent success of 3D Gaussian Splatting [17] has inspired many impressive 4D Gaussian avatars [18]–[24] achieving faster avatar rendering. However, most of those NeRF-based or Gaussian-based 4D avatars focus on the accurate appearance modeling while fusing the illumination within the NeRF or Gaussian representation, but couldn't support any relighting applications especially from challenging monocular videos.

The essential challenging for relightable 4D avatar reconstruction comes from the difficulty in joint learning of both dynamic geometry, materials and lighting from limited monocular video observation, while requiring realistic inverse rendering quality in efficient rendering speed simultaneously. PointAvatar [25] probably provides the first relightable 4D avatar solution using efficient differentiable point rendering. FLARE [26] further improves the the relightable rendering quality with differentiable mesh proxy. But the rendering quality for these approaches are still limited mainly due to the limited ability for point or mesh based rasterization. Based on the powerful 3DGS, the recent efforts introduce impressive relightable Gaussian avatar reconstruction from lightstage devices (RGCA [27]) or monocular videos (RGAvatar [28] and HRAvatar [29]). However, since the original 3DGS is limited to capture the advanced lighting illumination [30], the relightable rendering quality for the latest 4D Gaussian avatars still need to be improved. To further simulate advanced lighting effects, the pioneer work of 3DGRT [30] provides a differentiable ray-tracer to capture better secondary ray-based effects such as reflection, shadow etc. The subsequent works [31]–[34] further improve the ray intersection accuracy or ray tracing speed, and have achieved remarkable results for complex illumination reconstruction such as inter-reflection (IRGS [35]) and specular lighting (EnvGS [36]). However, most of these recent advances focus on static objects or scene cases, and often perform pixel-wise ray tracing, which requires too much computation to support fast rendering in dynamic scenarios such as 4D avatars.

Recently, R3DG [37] performs ray tracing on a per-Gaussian level, but doesn't utilize the capability for accurate indirect light estimation. GaussianShader [38] also performs shading at a per-Gaussian level but doesn't support Gaussian tracing for highly accurate lighting modeling.

More importantly, how to accurately optimize the dynamic geometry, BRDF materials and global illumination within the ray-tracing based Gaussian splatting from monocular videos but not multi-view observation [39], remains under-explored and especially challenging, which serves as the key difficulty for relightable Gaussian avatar reconstruction towards much better inverse rendering quality while in efficient manner.

In this paper, we propose a new relightable Gaussian avatar reconstruction, which can simulate more advanced

Manuscript created April, 2026; Zhe Fan is with the School of Computing Science & Technology, Beijing Institute of Technology, Beijing 100081, China. E-mail: fanzhe@bit.edu.cn. Shi-Sheng Huang, ZhaoChen Li, Hua Huang are with the School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China. E-mail: shishenghuang.net@gmail.com, 202211081048@mail.bnu.edu.cn, huahuang@bnu.edu.cn.
Hua Huang is the corresponding author.

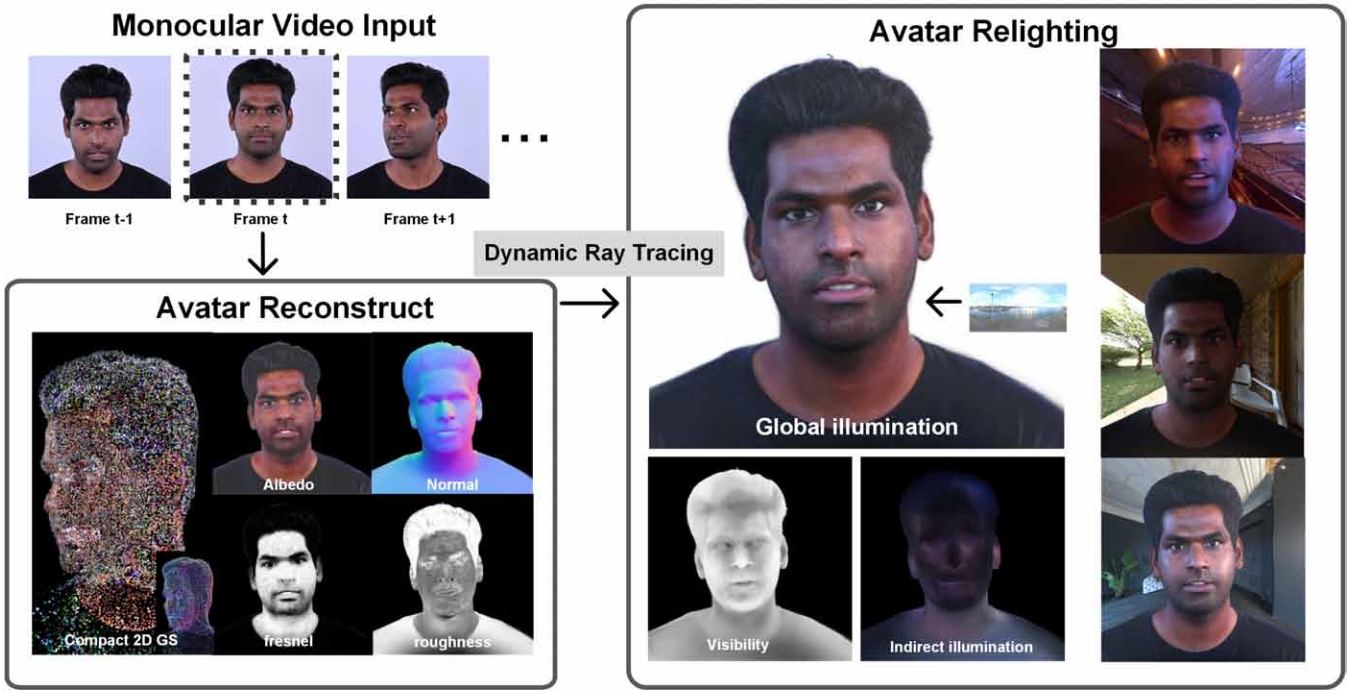

Fig. 1. We propose a new Gaussian avatar reconstruction (D-RayAvatar) from monocular videos based on a novel Dynamic Ray-tracing based Gaussian Splatting (D-RayGS), which can simulate complex global illumination (HDR illumination, middle) for accurate Gaussian avatar reconstruction (left bottom), and realistic relighting under different illumination (right) in a efficient manner.

lighting effects, such as self-occlusion and inter-reflections, by introducing a novel Dynamic Ray-tracing based Gaussian Splatting (D-RayGS). Unlike most previous 3D Gaussian ray-tracer [30], we choose 2D Gaussian primitive as the basic avatar representation due to more accurate ray intersection for better render quality [35]. What's more, different from previous Gaussian ray-tracers [30], [35], [36] performing time-consuming pixel-wise ray-tracing for static scenes, the key observation of our D-RayGS is to directly perform the ray-tracing embedded Physical based Rendering directly on the 2D Gaussian primitives, and then aggregate the final appearance using the efficient Gaussian splatting. This mechanism significantly saves ray-tracing computation, making it especially suitable for the dynamic 4D avatar training and rendering. More importantly, to preserve the high fidelity rendering quality, we further provide an effective D-RayGS learning strategy, which jointly optimizes the *compact* relightable 2D Gaussian primitives from monocular videos, by leveraging extra regularizing from extra SDF field for more accurate decomposition of both the dynamic geometry, BRDF materials and lighting. Benefiting from our efficient D-RayGS and its compact learning strategy, we can achieve more accurate relightable Gaussian avatar reconstruction from monocular videos, enabling more realistic relighting at a fast rendering speed.

To evaluate the effectiveness of our D-RayAvatar, we have conducted extensive evaluation on public datasets by comparing with previous approaches. From the comparison, our D-RayAvatar can achieve better 4D avatar reconstruction than previous 4D avatars such as INSTA [40], PointAvatar [25], FLARE [26] and recent 4D Gaussian avatars such as FlashAvatar [20], SplattingAvatar [22], RGAvatar [28] and HRAvatar [29]. For relighting application, our D-RayAvatar can achieve significant better relighting quality than PointAvatar [25], FLARE [26] and much better than recent RGAvatar [28] and HRAvatar [29] approaches, while maintaining efficient rendering speed.

The main contributions of this paper are summarized as follows:

1) We introduce D-RayGS, a novel rendering approach that performs ray-tracing and shading directly on dynamic 2D Gaussian primitives, enabling efficient simulation of complex global illumination (e.g., self-shadowing and inter-reflections) for dynamic avatar scenes.
2) We provide a compact learning strategy for high-fidelity avatar reconstruction, which enables the robust joint optimization and precise disentanglement of dynamic geometry, materials, and lighting, by incorporating auxiliary SDF regularization and tailored inverse rendering constraints, overcoming the inherent ambiguities of monocular inputs.
3) We propose the first ray-tracing-based relightable Gaussian avatar reconstruction framework from monocular videos. Extensive experiments demonstrate that D-RayAvatar outperforms existing approaches in both reconstruction and relighting quality, in an efficient manner.

II. Related Works

Animatable Head Avatars. The animatable head avatar reconstruction has achieve much progress since the pioneering work of 3DMM [41], with lots of subsequent efforts made to

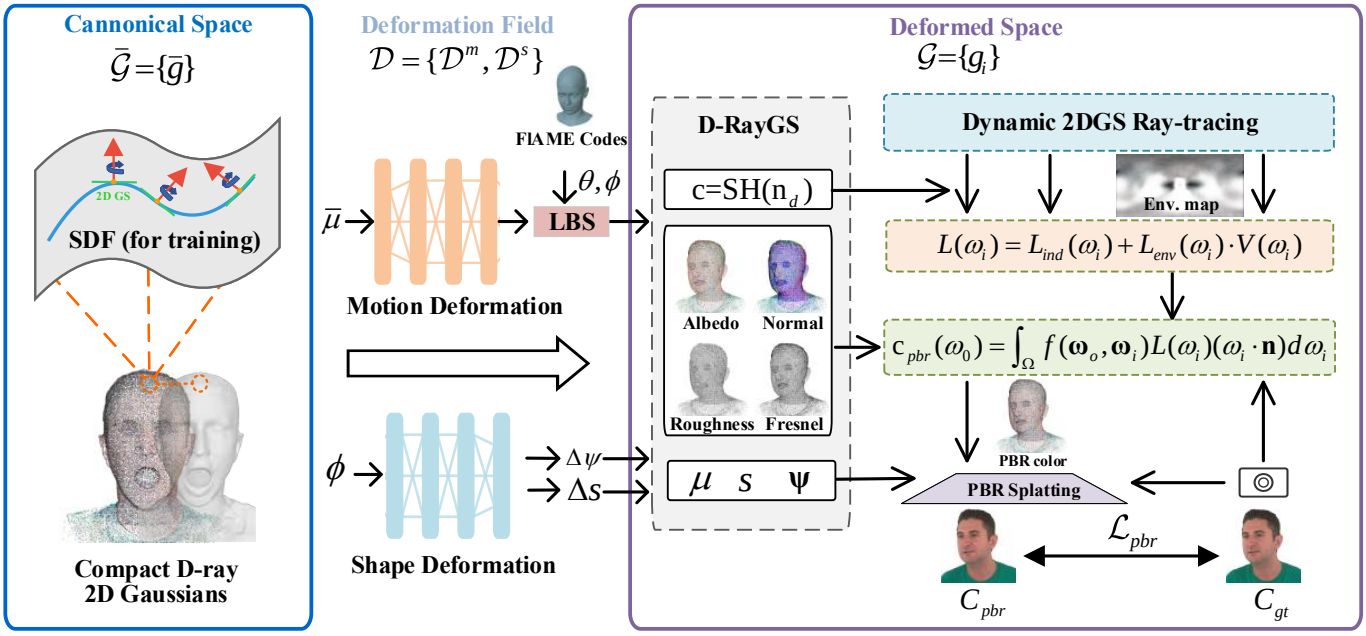

Fig. 2. The main pipeline of D-RayAvatar. Given monocular video input, we propose to reconstruct the Gaussian avatar represented by relightable 2D Gaussian primitives, which enables efficient inverse rendering using dynamic ray-tracing based Gaussian splatting (D-RayGS). We deform the relightable canonical Gaussian attributes $\bar{g}$ to deformed g using motion deformation $\mathcal{D}^m$ and shape deformation $\mathcal{D}^s$, then compute its Gaussian-level PBR color $c_{pbr}(\omega_0)$ by performing ray-tracing embedded Physically Based Rendering directly on each g, and aggregate the PBR color image by splatting all the Gaussians using a PBR splatting. The D-RayGS is learnt to be compact with the geometric regularization from an extra SDF field for high quality relightable rendering.

generate realistic avatars using blendshapes priors from subject identity [42], pose and expression [43], emotion [44], and even bones and muscle [45], and also boost up various tasks including face reconstruction [46]–[48], photo-realistic face synthesis [49], [50], and face reenactment [51], [52]. But the 3DMM based approaches often use topology-fixed template mesh, which limits the final rendering quality even on the facial part.

The success of NeRF [8]–[10] has inspired many implicit head avatar reconstruction like NerFACE [4], HeadNeRF [6], IMAvatar [7], HAvatar [13] and LatentAvatar [53], and further reduce the NeRF training and inferencing time within 20 minutes (NeRFBlendShape [11]) or even minutes by decoupling the motion and appearance with motion-aware neural voxels as AvatarMAV [12]. On the other hand, some approaches adopt to use the 3D-aware GAN to generate high fidelity controllable 3D avatars, such as Next3D [54] and AniPortraitGAN [55]. Very recently, the success of 3D Gaussian Splatting has inspired a serials of 4D Gaussian avatars [18]–[24], [56], which achieves impressive head avatar creation results. The latest work of SOAP [3] proposed a style-omniscient framework to generate rigged facial avatar with remarkable driving accuracy by leveraging multiview diffusion model [57]. However, most of previous neural 4D avatars focus on the avatar reconstruction but didn't support relightable rendering. In contrast, our paper follows the latest Gaussian avatar framework but aims at high fidelity relightable rendering at very efficient speed.

Inverse Rendering and Relighting. Inverse rendering is an essential technical to support relighting application, which has achieved much progress for approaches using lightstage devices [58]–[64], with impressive relighting results. The recent neural inverse rendering [65]–[68] adopts to learn relightable 3D assets from multi-view images or spare-view/monocular videos, which gets rid of the usage of lightstage. Munkberg et al. [69] adopts neural SDFs to reconstruct high quality relightable 3D assets. SwitchLight [70] proposes a 2D relighting approach for impressive human portrait relighting results by combining a physics-guided architecture with a pre-training framework. IC-Light [71] proposes to impose consistent light transport into diffusion process, which achieves impressive in-the-wild image relighting and editing. Meanwhile, most of those neural inverse rendering approaches focus on 2D image-based relighting, but is not suitable for animatable 3D head avatar relighting tasks.

To enable relightable 3D head avatars reconstruction, some early works [72], [73] leverage deformable priors from 3DMM [41] to learn neural materials and lighting from image or videos. Xu et al. [74] propose to learn animatable neural avatar from sparse-view video by modeling appearance color with spherical harmonics (SH) diffuse color and spherical Gaussian (SG) specular color. However, most of those previous approaches need very time-consuming light visibility tracing operations, which are limited for dynamic relightable rendering applications.

Some recent works adopts to use explicit representation with differential rendering, such as points (PointAvatar [25]) and mesh (FLARE [26]), for efficient relightable 4D avatar reconstruction. However, their rendering quality are still limited mainly due to the limited ability of point or mesh based rasterization. URAvatar [75] is capable of creating ultra-photorealistic and relightable head avatars from phone scans. However, the method relies on expensive light stage data

and substantial computational resources to construct its prior model, without enabling inter-reflective illumination modeling as like Gaussian tracing approaches. Very recently, RGAvatar [28] and HRAvatar [29] propose to learning animatable Gaussian avatars from monocular videos, which enable remarkable inverse rendering results. However, since the 3DGS is limited to simulate advanced lighting illumination, the relightable rendering quality still needs to be improved. On the other hand, the recent efforts proposed a few remarkable Gaussian ray-tracers [30]–[34] to capture secondary ray-based lighting effects for more realistic static scene rendering, but often need huge pixel-wise Gaussian ray-tracing even under object level cases [35], which is challenging to be applied for dynamic 3D avatar reconstruction.

Our work is inspired by those previous relightable 3D avatar works, but provides a new dynamic ray-tracing based Gaussian splatting which is quite suitable for high fidelity Gaussian avatar reconstruction to support more better realistic relightable rendering, while in a more efficient manner.

III. D-RayAvatar Reconstruction

Given a person's monocular portrait video with N image frames $\mathcal{I} = \{I_1, ..., I_N\}$, we reconstruct an animatable Gaussian avatar, i.e., D-RayAvatar $\mathcal{G}(\phi, \theta, L_{env})$ represented by a set of dynamic relightable 2D Gaussians [76] , which can be animated by varying the expression ϕ and head pose θ parameters as like FLAME model [43], and perform relighting under different global illumination L_{env}. Our D-RayAvatar's core rendering framework is a new Dyanmic Ray-tracing based Gaussian Splatting (D-RayGS) (Sec. III-A), which performs the ray-tracing embedded Physically Based Rendering (PBR) [77] directly on the 2D Gaussian primitives for *efficient* dynamic rendering, and preserves the high fidelity realistic rendering quality using a *compact* D-RayGS optimization (Sec. III-B) with a two stage coarse-to-fine learning (Sec. III-C). The system overview of our D-RayAvatar is shown in Fig. 2.

A. Dynamic Ray-tracing based Gaussian Splatting

Different from previous time-consuming pixel-wise Gaussian ray-tracer [30], [35], [36], we introduce a more efficient dynamic ray-tracing based Gaussian splatting (D-RayGS), which perform ray-tracing embedded PBR directly on the 2D Gaussian primitives and aggregate the final appearance color following the Gaussian Splatting.

Relightable 2D Gaussian Primitive. To make accurate ray-tracing intersection calculation, we adopt 2D Gaussians but not 3D Gaussians [30] as the basic representation, and extend the 2D Gaussians to relightable 2D Gaussian primitives to enable Gaussian-based PBR.

Specifically, we define a relightable 2D Gaussian primitive $g = \{\mu, \mathbf{\Psi}, \mathbf{s}, \alpha, SH, b, r, f_0, v\}$, where $\mu \in R^3$ is the centroid position, $\mathbf{s} = \{s_u, s_v\}$ is 2D scale, $\mathbf{\Psi} = \{\psi_u, \psi_v\}$ are rotation angles for two rotation axis vectors, α is the density, SH is the spherical harmonics parameters, b, r, f_0 representing the BRDF material parameters albedo, roughness and fresnel reflectivity at normal incidence respectively, and v representing the light visibility. As shown in Fig. 2, we start to deform a canonical $\bar{g}$ to g using a deformation field $\mathcal{D}(\phi, \theta)$, and compute its Gaussian-level PBR color $c_{pbr}(\omega_0)$ by performing ray-tracing embedded Physically Based Rendering directly on each g for each given view ray ω_0, and aggregate the PBR color image by splatting all the relightable 2D Gaussian primitives $\mathcal{G}(\mathcal{D}(\phi, \theta), L_{env}) = \{g_i | i \in \mathbf{M}\}$ using a PBR splatting.

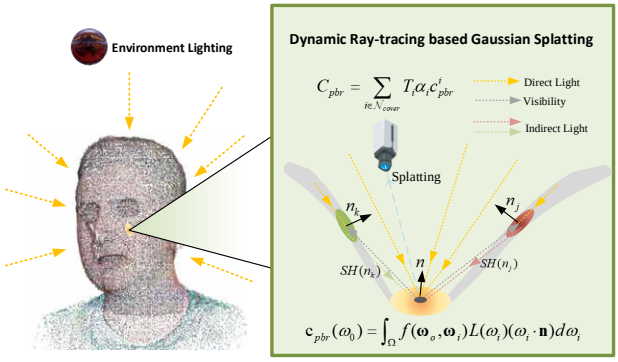

Fig. 3. The illustration of ray-tracing based PBR rendering. We perform PBR rendering directly on each 2D Gaussian primitives, where the visibility and indirect lighting are also traced to simulate highly accurate final PBR splatting.

Dynamic Deformation Field. We build up the deformation field $\mathcal{D} = \{\mathcal{D}^m, \mathcal{D}^s\}$ with two components including the motion deformation $\mathcal{D}^m$ and shape deformation $\mathcal{D}^s$, where $\mathcal{D}^m$ deforms the centroid position μ and $\mathcal{D}^s$ deforms the shape related parameters ($\mathbf{s}$ and $\mathbf{\Psi}$). Specifically, we follow PointAvatar [25] to construct $\mathcal{D}^m$, which deforms the centroid position $\bar{\mu}$ from canonical state to dynamic μ, i.e., $\mu = \mathcal{D}^m(\bar{\mu})$, by combining linear blend skinning (LBS) and an extra position offset field $\Delta\mu$ as:

$$\mathcal{D}^m(\bar{\mu}) = LBS(\bar{\mu} + \Delta\mu + \mathcal{B}_P(\theta, \mathcal{P}) + \mathcal{B}_E(\phi, \mathcal{E}), \theta, \mathcal{W}), \quad (1)$$

where $\mathcal{B}_P(\cdot), \mathcal{B}_E(\cdot)$ compute the pose and expression offsets using the blendshape components $\mathcal{P}, \mathcal{E}$ respectively, $\mathcal{W}$ is the blend-skinning weights, θ, ϕ are the pose and expression parameters. We train a MLP-based network $\mathcal{F}^m$ which, given canonical centroid position $\bar{\mu}$ as input, returns the pose blendshapes $\mathcal{P} \in R^{4\times9\times3}$, expression blendshapes $\mathcal{E} \in R^{100\times3}$ and blend skinning weights $\mathcal{W}$ respectively.

To construct the *shape* deformation $\mathcal{D}^s$, motivated by the intuition that the scale and local rotation (i.e., rotation around the normal) of each Gaussian are independent of the global head pose, we train a MLP-based network $\mathcal{F}^s$ that, given the expression parameter ϕ as input, returns the scale offset Δs and rotation offset $\Delta\psi$ to obtain the deformed scale s and and rotation ψ as:

$$(\Delta s, \Delta\psi) = \mathcal{F}^s(\phi), \Longrightarrow s = \bar{s} + \Delta s, \quad \psi = \bar{\psi} \otimes \Delta\psi. \quad (2)$$

Ray-tracing Embedded PBR Rendering. For efficient rendering, we calculate each primitive g's color c_{pbr} by performing the Physical Based Rendering equation directly on the primitive. Moreover, we execute ray tracing to simulate

advanced lighting such as inter-reflection when calculating the PBR equation directly on the primitive level, but not the time-consuming pixel-wise based Gaussian ray-tracer [30], [35], [36]. Specifically, given any outgoing light direction ω_0 as shown in Fig. 3, we compute the outgoing light radiance for each primitive g to get its PBR color $c_{pbr}(\omega_0)$ as:

$$c_{pbr}(\omega_0) = \int_{\Omega} f(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) L(\omega_i) \Big(\omega_i \cdot \mathbf{n}\Big) d\omega_i, \tag{3}$$

where $f(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i)$ models the BRDF properties of each g, $L(\omega_i)$ represents each 4D Gaussian g's incident light along the incoming light direction ω_i, n is the normal vector of g. We adopt the simplified Disney BRDF model [78] and divide the BRDF function $f(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i)$ into diffuse term f_d and specular term f_s, i.e., $f(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) = f_d + f_s$ as $f_d = \frac{\boldsymbol{b}}{\pi}, f_s(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) = \frac{D(\boldsymbol{h};r) \cdot F(\boldsymbol{\omega}_o, \boldsymbol{h};f_0) \cdot G(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, h;r)}{(\boldsymbol{n} \cdot \boldsymbol{\omega}_i) \cdot (\boldsymbol{n} \cdot \boldsymbol{\omega}_o)}$, where D is the microfacet distribution function, h is the half-vector, F is the Fresnel term, and G is the geometric attenuation factor. We approximate the rendering equation using Monte Carlo integration and sample incident light directions ω_i over the hemisphere of each Gaussian. These ray directions are then used to jointly querying the environment map, trace visibility and indirect light transport.

Gaussian-based Indirect Light Ray-tracing. For more accurate lighting effect, we divide the incident light $L(\omega_i)$ into ambient L_{env} and indirect L_{ind} parts as $L(\omega_i) = V(\omega_i) \cdot L_{env}(\omega_i) + L_{ind}(\omega_i)$, where $V(\omega_i)$ is the light visibility term of each 4D Gaussian g, $L_{env}(\omega_i)$ is the global direct environmental light (modeled as an HDR environment map) to be learnt, $L_{ind}(\omega_i)$ is traced among Gaussian primitives. As shown in Fig. 3, for each primitive g, we sample M rays $\{r_j = \mu + d * w_j | j = 1, ..., M\}$ on the hemispherical domain centered at g's centroid position μ, and perform ray-tracing for each ray r_j to intersection neighbor 2D Gaussian primitives [35], i.e., $(L^j_{ind}(r_j), 1 - V(\omega_i)) \leftarrow Trace(r_j)$, to trace the indirect light during training as:

$$L_{ind}(\omega_i) = \sum_{j=1}^{M} L^j_{ind}(r_j) \Longleftarrow L^j_{ind}(r_j) = SH(n_j), \tag{4}$$

where n_j is the normal vector of the traced primitive g_j which is intersected with ray r_j, and $SH(\cdot)$ is the SH color queried from the SH parameter of g_j using the normal vector n_j. Please note that since each primitive g_j keeps moving every frame, for more accurate calculation of $SH(g_j)$, we adopt to query the SH color using g_j's normal direction n_j but not the incident ray r_j or the view direction.

PBR Splatting. After we compute the PBR color $c_{pbr}(\omega_o)$ for each g, we render the PBR image C_{pbr} through a PBR splatting, which performs an alpha-blending of the PBR color $c_{pbr}(\omega_o)$ splatted on the image as: $C_{pbr} = \sum_{i \in \mathcal{N}_{cover}} T_i \alpha_i c^i_{pbr}$, $T_i = \prod_{j=1}^{i-1}(1 - \alpha_j)$, where $\mathcal{N}_{cover}$ is the set of primitives which are splatted on the image covering the same image pixels.

B. Compact D-RayGS Optimization

To improve the rendering quality of our D-RayGS as like previous Gaussian ray-tracers [35], [36], we build up an optimization by introducing effective regularization from an extra SDF field to force the *compactness* of D-RayGS in the training stage, which leads to highly accurate joint decomposition of the dynamic geometry, BRDF materials and environment lighting, thus enabling high fidelity relighting for our efficient D-RayGS.

SDF Field. Within the avatar's 3D canonical space, we build a MLP-based network $\mathcal{F}_{sdf}$ that predicts the signed distance s for every position $x \in R^3$, i.e., $s = \mathcal{F}_{sdf}(x) \in R$. Using the SDF field $\mathcal{F}_{sdf}$, we propose to force the centroid position μ of each $\bar{g}$ to be located on the underlying surface $\mathcal{G}_s$ of the SDF field. The normal vector n_s for each on-surface point $x_s \in \mathcal{G}_s$ can be computed by the back propagation of the geometry network $\mathcal{F}_{sdf}$, i.e., $n_s = \frac{\partial \mathcal{F}_{sdf}}{\partial x_s}$.

On-Surface Regularization. With the aid of SDF field, we can regularize the primitive g to be on-surface points with zero level set of SDF and Eikonal regularization. Specifically, we use a SDF loss L_{sdf} performed on the centroid positions of all primitives, and a Eikonal loss L_{eik} performed on the sampled positions as:

$$\mathcal{L}_{\text{sdf}} = \left\| \text{SDF}\left(\bar{\mu}\right) \right\|^2, \mathcal{L}_{\text{eik}} = \left(\left\| \nabla_{\mathbf{x_e}} \text{SDF}\left(\mathbf{x_e}\right) \right\| - 1 \right)^2. \tag{5}$$

Normal Regularization. On the other hand, during the dynamic deformation process, we also force the normal vector n of each g equivalent to deformed normal n_μ at μ measured by the deformation of SDF field, , i.e., $n = n_\mu$. Following PointAvatar [25] and Eq. (1), we can compute the deformed normal from the n_μ using $n_\mu = l_\mu \frac{\partial \mathcal{F}_{sdf}}{\partial \bar{\mu}} (\frac{\partial \mathcal{D}^m(\bar{\mu})}{\partial \bar{\mu}})^{-1}$, where l_μ is a normalizing scalar to ensure the normal is of unit length.

One benefit of our 2D Gaussian primitive representation over the previous 3D Gaussian based representation [28], [29], is that we have more clear normal definition, i.e., $n = \psi_u \times \psi_v$, than the 3D Gaussians. Therefore, we enforce $\psi_u \times \psi_v = n_\mu$ and compute ψ_u and ψ_v via an axis-angle rotation (n_μ, ω). Applying the on-surface regularization and normal regularization together amounts to force the 2D Gaussian primitives tightly locating in the tangential space of SDF field's on-surface mesh, which can lead to more compact D-RayGS optimization.

C. Coarse-to-Fine Learning

We use a coarse-to-fine learning strategy to learn the compact D-RayGS representations $\mathcal{G}(\mathcal{D}(\phi, \theta), L_{env})$ from monocular video frames $\mathcal{I}$. Except from the on-surface regularization and normal regularization mentioned above, we also use other priors cues including:

Albedo Regularization. It is defined as $\mathcal{L}_{albedo} = |\mathbf{B} - \mathbf{B}_{prior}|$, where B, B_{prior} are the splatted albedo map and referential albedo map estimated and adjusted from a prior model [79], respectively.

We fine-tuned the prior model on a mini-batch of PBR 3D head assets. However, the estimated albedo still exhibited some inconsistencies: not only with the skin tone, but also across different frames temporally. To address this, we leverage the assumption that the illumination in the training videos is approximately white. Based on this prior, we propose an albedo correction method to compute the final prior albedo B_{prior}. Details of this calibration method and the fine-tuning process for prior model are provided in the Appendix.

PBR Rendering Loss. It is defined as $\mathcal{L}_{pbr} = \mathcal{L}_1(C_{pbr},C_{gt}) + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}}(C_{pbr}, C_{gt})$, where we perform the PBR splatting to achieve the PBR rendering image C_{pbr}, and use the input image C_{gt} as supervision.

SH Rendering Loss. Formally, $\mathcal{L}_C = \mathcal{L}_1(C_{app}, C_{gt}) + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}}(C_{app}, C_{gt})$, where C_{app} is the SH rendering.

White Light Regularization. $\mathcal{L}_{light} = \sum_c(\mathcal{S}_c - \frac{1}{3}\sum_i \mathcal{S}_i), i, c \in \{\mathcal{R}, \mathcal{G}, \mathcal{B}\}$, where we adopt a white light regularization over the shading component with a regularization loss $\mathcal{L}_{light}$ and $\mathcal{S}_c$ is one channel of the diffuse shading.

We also adopt the BRDF smooth loss $\mathcal{L}_{smooth}$ following R3DG [37] and the Flame Deformation Regularization $\mathcal{L}_{\text{flame}}$ to regularize the $\mathcal{D}^m$ similar with previous avatars [7], [25], [28]. Uniquely, we employ the referential albedo map, not the original training image conventionally, as the guidance image for BRDF smooth loss to enforce spatial consistency of materials.

During the learning, we follow a coarse-to-fine learning strategy. In the coarse stage we mainly learn the dynamic deformation field $\mathcal{D}(\phi, \theta)$, SDF and the non-relightable parts of the 2D Gaussian primitives guided by:

$$\mathcal{L}_{co} = \mathcal{L}_C + \lambda_{flame}\mathcal{L}_{\text{flame}} + \lambda_{sdf}\mathcal{L}_{\text{sdf}} + \lambda_{eik}\mathcal{L}_{\text{eik}}, \quad (6)$$

which serves a basic 2D Gaussian reconstruction to obtain rough geometry and dynamic deformation. In the fine stage, we jointly learning both the relightable parts of the 2D Gaussian primitives and environment lighting L_{env}, and also slightly fine-tune the non-relightable 2D Gaussian primitives parts and dynamic deformation field, to achieve accurate compact geometry, BRDF materials and lighting decomposition simultaneously. The overall loss function for the fine stage is defined as:

$$\mathcal{L}_{fine} = \mathcal{L}_{co} + \mathcal{L}_{pbr} + \lambda_l\mathcal{L}_{light} + \lambda_{albedo}\mathcal{L}_{albedo} + \lambda_s\mathcal{L}_{smooth}. \quad (7)$$

IV. Experiments

A. Implementation Details

Data Preprocessing. As like previous Gaussian avatars, we adopt MICA [80] to extract the pose and expression parameters for preprocessing of each frame. Besides, we also extract eye-blinking parameters by MICA, and perform the linear blend skinning (LBS) in the motion deformation field by combining the pose blendshapes, expression blendshapes, blend skinning weights from FLAME model [43] and eye-blinking blendshapes from MICA.

Lightweight Networks. We use a lightweight MLP to construct $\mathcal{F}^m$ in the motion deformation field with network structure as $[256, 256, 256, 256]$, and use softplus as activation function to predict the shape, pose, expression blendshape components. Similarly, we also use a four layers MLP to construct $\mathcal{F}^s$ with network structure is $[256, 256, 256, 256]$ to predict the shape deformation respectively. For the geometry network $\mathcal{F}_{sdf}$ in the SDF field, we use the similar MLP provided by PointAvatar [25] to predict signed distance.

Datasets. To perform training and evaluation of RGAvatar, we randomly collect a dataset with individual persons' fixed viewport monocular videos from publicly released datasets, i.e., NeRFace [4], HDTF [81], NeRFBlendShape [11], PointAvatar [25], and INSTA [40], which count for 12 subjects. Besides, we also captured extra 3 subjects using a fixed viewport webcam, thus making in total a dataset with 15 subjects in our collected dataset to perform the training and evaluation. In average, each subject's monocular video has around 3000 RGB frames, with image resolution set as 512×512. During the network training, we sample half of the frames for training, and the left frames for evaluation.

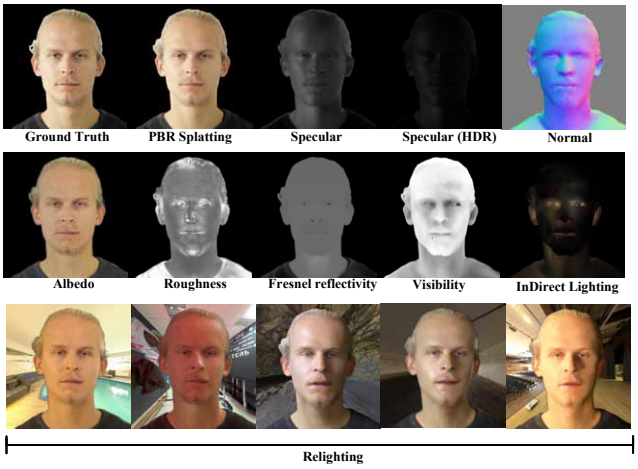

Fig. 4. The visual results of our D-RayAvatar reconstruction. Specifically, We show the specular components, normal, BRDFs, traced results(visibility and indirect lighting), optimized lighting and relighting results under different changing environmental lights.

B. Avatar Reconstruction Evaluation

We first evaluate the Gaussian avatar reconstruction quality including the D-RayAvatar's geometry, BRDF materials (albedo, roughness and Fresnel reflectivity), and the estimated environment map respectively. Fig. 4 shows the our D-RayAvatar reconstruction results of one subject from our collected dataset. Benefiting from compact D-RayGS learning, we can see that our approach can accurately reconstruct the 2D Gaussians compactly approximate the underlying surface of the avatar's dynamic geometry, and the splatted normal image also demonstrates accurate prediction of the normals, which are coherent to the geometry.

Relighting Evaluation. Based on the accurate D-RayGS learning, we then evaluate the relighting quality by giving various changing environmental lights. As shown in Fig. 4, we demonstrate the relighting results when given some environmental lights, from soft lights to brighter lights. We can see that our approach can achieve high realistic relighting quality given different environmental lighting robustly. Besides, we also perform relighting under different expression synthesis. As shown in Fig. 4, when the avatar is animated using novel expression parameters, our approach can also achieve high-fidelity relighting results with natural soft shadows, which is benefit from the accurate D-RayGS learning by our approach.

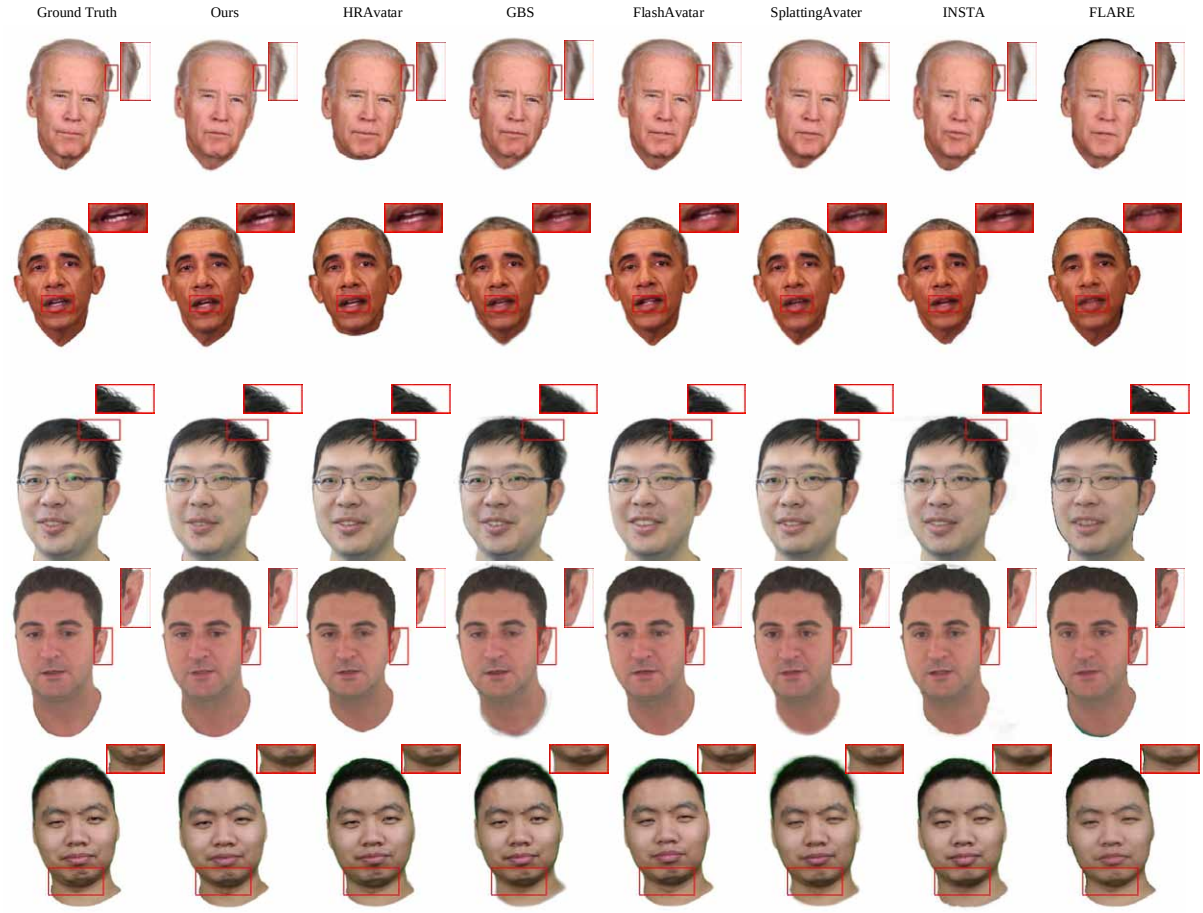

Fig. 5. Visual comparison results for different 4D avatars including INSTA, FlashAvatar, SplattingAvatar, FLARE, GBS, HRAvatar and Ours.

C. Reconstruction Comparison

We evaluate the reconstruction quality(self-reenactment) of our method by comparing it with recent state-of-the-art monocular 4D avatar reconstruction methods:FLARE [26], INSTA [40], FlashAvatar [20], SplattingAvatar [22], GaussianBlendShapes [82] and HRAvatar [29].

Quantitative Comparison. We systematically evaluate the rendering quality on our test set using standard metrics: PSNR, SSIM, LPIPS, MSE, and L1. As shown in Table I, our method consistently outperforms the baseline approaches. Specifically, our pipeline achieves higher PSNR scores for photorealistic rendering, alongside lower LPIPS, MSE, and L1 errors, which demonstrates that our method recovers higher-quality and more accurate 4D avatars than existing techniques.

Qualitative Comparison. Figure 5 visually compares our approach with the baselines. Limited by its deformation capacity, FLARE produces blurry artifacts, especially in the mouth and teeth regions. INSTA generates overly smooth results, failing to preserve fine local details. Conversely, our method achieves higher visual quality. Compared to existing Gaussian-based avatars (FlashAvatar, SplattingAvatar, GaussianBlendShapes and HRAvatar), our approach recovers the finest high-frequency details, including hair, teeth, and precise lip shapes. The zoomed-in patches further validate that our model synthesizes a more detailed and photorealistic appearance.

D. Relighting Comparison

To demonstrate the effectiveness on relighting applications, we conduct comparing experiments qualitatively with SOTA relightable avatars including FLARE [26], RGAvatar [28] and HRAvatar [29] using different environmental lighting. Since

TABLE I
QUANTITATIVE COMPARISON ON THE TEST SET OF OUR COLLECTED DATASET FROM DIFFERENT COMPARISON APPROACHES, INCLUDING INSTA [40], FLASHAVATAR [20], SPLATTINGAVATAR [22] (ABBREVIATED AS SPLAVATAR FOR SIMPLICITY), FLARE [26] AND GBS(GAUSSIANBLENDSHAPES) [82] AND OURS, IN TERMS OF PSNR, SSIM, LPIPS, MSE AND L1 ACCURACY METRICS.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	L1 ↓
FLARE	25.6670	0.9122	0.0592	0.05921	0.0123
INSTA	26.6980	0.9268	0.0912	0.0509	0.0151
SPLAvatar	25.6255	0.9223	0.0979	0.0533	0.0139
FlashAvatar	28.7674	0.9415	0.0563	0.0406	0.0110
GBS	27.9699	0.9390	0.1051	0.0436	0.0126
HRAvatar	29.3708	0.9407	0.0776	0.0414	0.0121
Ours	**29.9664**	**0.9431**	**0.0557**	**0.0350**	**0.0101**

Fig. 6. Visual comparison results between FLARE [26], RGAvatar [28], HRAvatar [29] and Ours, under different environmental lights for different subjects.

HRAvatar lacks native support for body modeling and relighting, we only evaluate its relighting results of heads region. As shown in Fig. 6, Our approach predicts a more faithful and detailed albedo and achieves better relighting results than FLARE, RGAvatar, and HRAvatar. Furthermore, benefiting from our dynamic ray-tracing-based Gaussian splatting, our approach can render more advanced light transport, specular highlights, and shadow effects. Notably, compared to these

methods, ours is the only method that supports global illumination. Please refer to our supplementary video for dynamic relighting comparison between different approaches.

E. Ablation Study

Compact SDF Field Regularization. To study the impact of the SDF field regularization on the compact D-RayAvatar reconstruction, we implemented an additional version of our system that removes the SDF field regularization (termed as 'w/o SDF', relying on the normal consistency loss of vanilla 2DGS for geometry optimization). We experimentally compared this baseline with our full system ('Ours') on the test set of our collected dataset. Fig. 7 presents a visual comparison between the two variants. As observed, the version equipped with SDF priors not only yields more accurate rendering results but also optimizes a more accurate and smooth geometry. Besides, Table II shows the quantitative comparison between such two systems by evaluating the image rendering accuracy on the test set of our collected dataset. We can see that without using the Constraints from SDF field, the accuracy metrics including PSNR, SSIM, LPIPS, MSE and L1 consistently decrease compared with our full system. This shows that the extra SDF field takes effect for better 4D Gaussian reconstruction, thus achieving better image rendering results. This confirms that the extra SDF field ensures better Gaussian reconstruction and, consequently, higher-quality image rendering.

Normal-conditioned SH. During the training stage and ray-tracing embedded PBR rendering, we adopt to use 2D Gaussian primitives' normal vector n but not the view direction to query the SH color, which we call normal-conditioned SH. To study how normal-conditioned SH influence the final appearance rendering quality, we implemented another variants using the view direction to query the SH color (termed as 'view direction-conditioned SH'), and compare with our normal-conditioned SH (termed as 'normal-conditioned SH' or 'ours'). Table II shows the quantitative comparison on subjects with large pose variations between such two systems. We can see that without using the normal-conditioned SH, the accuracy metrics including PSNR, SSIM, LPIPS, MSE and L1 consistently also decrease compared with normal-conditioned SH system, which show the validation of our normal-conditioned SH in the ray-tracing embedded PBR rendering. Fig. 7 (left) show some visual comparison of such two systems, it can be observed that the 'view direction-conditioned SH' exhibits shading artifacts in the eye and face regions, whereas the 'normal-conditioned SH' avoids these issues. Furthermore, the normal-conditioned SH also suppresses geometric artifacts more effectively than the baseline.

Dynamic Ray-tracing. To evaluate the contribution of the ray-tracing module within our PBR rendering pipeline, we conducted an ablation study by training and relighting the model with and without indirect light tracing. As shown in Fig. 8,eliminating ray-tracing during training results in inaccurate albedo estimation; specifically, we observe unnatural brightness in occluded regions, such as the nose wings. This degradation stems from incomplete lighting modeling. By

TABLE II
QUANTITATIVE COMPARISON BETWEEN TWO VARIANT SYSTEMS WITH ('OURS') OR WITHOUT ('W/O SDF') SDF FIELD REGULARIZATION, AND NORMAL-CONDITIONED SH CALCULATION.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓
w/o SDF	27.2861	0.8927	0.1089	0.0462
w/o Normal-SH	27.8534	0.8991	0.0963	0.0432
Ours	**28.1932**	**0.9035**	**0.0924**	**0.0413**

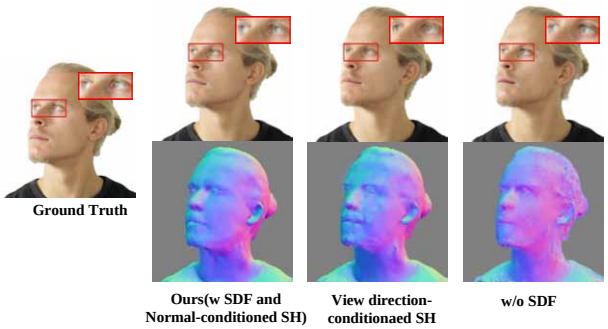

Fig. 7. The visual comparison results for Ablation Study of normal-conditioned SH (left) and SDF(right).

examining the optimized environment map in the w/o Ray-tracing setting, we observe that skin tone-liked radiance which physically originates from indirect lighting is erroneously baked into the direct lighting (environment map). In contrast, incorporating dynamic ray-tracing during training properly models these interactions, leading to a more robust disentanglement of lighting and material, and consequently, a more accurate albedo inference.

Regarding relighting capabilities, we evaluated the avatar's appearance under novel illumination with and without the ray-tracing module. First, we illuminated the avatar using a single point light. As observed in Fig. 8, the method enabled with ray-tracing produces significantly more accurate and fine-grained shadows, particularly in self-occluded regions. Furthermore, when relighting with a HDR environment map with complex lighting, the baseline without ray-tracing suffers from noticeable visual distortions and artifacts. In contrast, the ray-tracing-enabled version handles the intricate lighting interactions effectively, yielding a much more natural and photorealistic appearance.

Gaussian-wise Ray-tracing v.s. Pixel-wise Ray-tracing. To validate the ray-tracing efficiency of our design, we implemented a pixel-wise ray-tracing variant to compare against our proposed Gaussian-wise strategy. In terms of training and relighting quality, as shown in Fig. 9, both methods successfully recover accurate material attributes, achieve consistent ray-tracing results in terms of visibility and indirect light, and produce photorealistic relighting results with approximate visual differences. However, a substantial divergence (4.79 FPS vs. 0.92 FPS) in computational efficiency is evident, as detailed in Table III. This performance advantage stems from two key factors. First, the number of Gaussian primitives (28,000) is an order of magnitude smaller than the pixel count

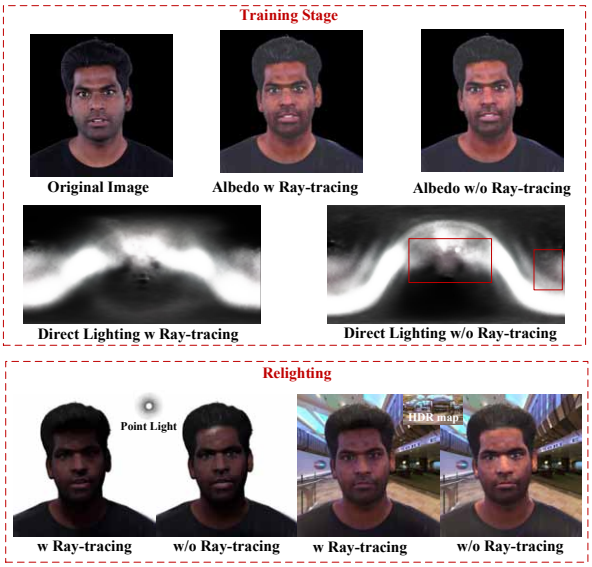

Fig. 8. Different visual results with/without dynamic ray-tracing.

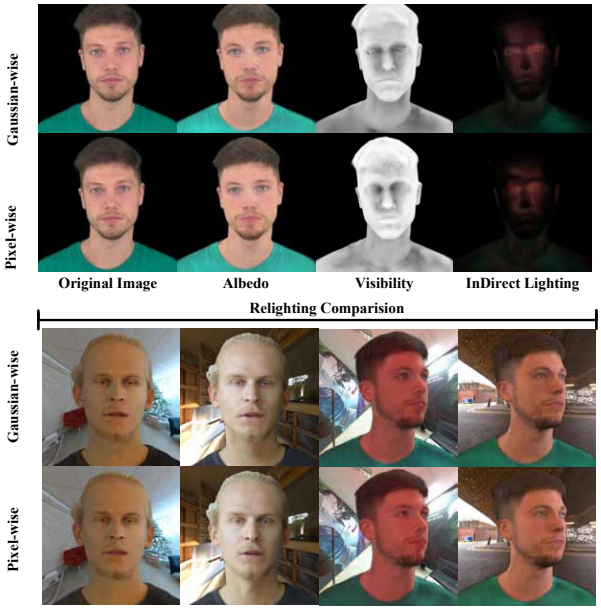

Fig. 9. The visual results of Gaussian-wise v.s. pixel-wise ray-tracing.

(262,144), which drastically reduces the workload for ray-tracing intersections. Second, the pixel-wise baseline follows a deferred shading pipeline, incurring the overhead of pre-splatting intermediate G-buffer maps (i.e., BRDF, normals, and depth). Conversely, our Gaussian-wise strategy circumvents this costly step by performing the rendering equation directly on the Gaussians, then splatting only the PBR color of Gaussians to the image space directly. This architectural design represents an optimal trade-off, achieving visual quality approaching the pixel-wise version while significantly reducing the computational burden, making it highly suitable for dynamic deformable avatars scenarios.

Albedo Regulation. To evaluate the contribution of our albedo regularization strategy, we conducted an ablation study by comparing our full method against two baselines: (1) training without albedo regularization, (2) training with the albedo prior finetuned from [79] without our illumination-guided refinement. Implementation-wise, we employed a semantic mask to enforce the constraint solely on skin pixels.

As shown in Fig. 10, removing the albedo regularization (w/o Albedo Regulation) results in significant baked-in lighting artifacts. Since the decomposition between albedo and shading is inherently ill-posed, the model tends to explain high-frequency shading details(e.g., highlights) as surface textures.Incorporating the finetuned prior improves the disentanglement of lighting and albedo. However, as observed in the second column, the referential albedo suffers from inconsistent skin tones. In contrast, after applying our proposed correction, the corrected albedo prior aligns significantly better with the actual skin color, as shown in the right column. Consequently, this accurate guidance enables our model to reconstruct a more faithful albedo map.

F. Time Efficiency Analysis

We conducted a runtime efficiency analysis by comparing our method against previous approaches, including FLARE, RGAvatar, and HRAvatar, on the same RTX 3090 GPU. Table III shows the relighting frame rates for the different methods. Due to the high computational overhead of ray-tracing, our full method (Ours, Gaussian-wise tracing) currently operates at a lower frame rate compared to FLARE (5.4 FPS), RGAvatar (16.0 FPS), and HRAvatar (40 FPS). To evaluate the inherent efficiency of our core representation, we also analyzed the variant without ray-tracing (i.e., omitting visibility and indirect illumination). In this setting, our method becomes significantly faster than FLARE. It also outperforms RGAvatar achieved by discarding the heavy SDF module—which is employed solely as auxiliary supervision for training—during the relighting stage.

TABLE III
RENDERING EFFICIENCY COMPARISION WITH OTHER RELIGHTING METHODS. THE RENDERING TIME HERE REFERS TO THE AVERAGE TOTAL RUNNING TIME REQUIRED TO GENERATE EACH RELIGHTED IMAGE.

Method	time per image↓	FPS↑
FlARE	0.1849s	5.41
RGAvatar	0.0622s	16.06
HRAvatar	0.0249s	40.15
Ours(w/o ray-tracing)	0.0351s	28.43
Ours(Pixel-Wise ray-tracing)	1.0869s	0.92
Ours(Gaussian-Wise ray-tracing)	**0.2088s**	**4.79**

V. LIMITATION AND DISCUSSION

One main limitation of our D-RayAvatar is that our current implementation sometimes has a failure to adequately disentangle high-frequency illumination from the training videos. This is due to our simplified modeling of environment map and our adoption of the normal-conditioned SH function to calculate the indirect light during training. This issue could be

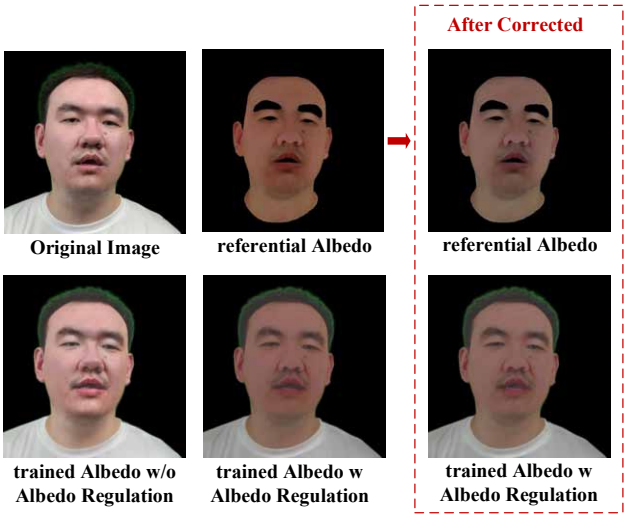

Fig. 10. The visual results with/without albedo regulation.

addressed by augmenting our current illumination model with an additional high-frequency component, such as Spherical Gaussians (SGs) or explicit point lights. A further limitation is that our relighting results occasionally show artifacts under extreme lighting conditions. This occurs because ray-tracing is highly sensitive to the underlying geometry accuracy; even minor geometric flaws can be drastically amplified, degrading the final rendering quality. Potential solutions include employing and optimizing advanced Gaussian representations to achieve more accurate underlying geometry.

Ethic. The goal of D-RayAvatar is to enable a high fidelity, subject-specific relightable Gaussian avatar creation, which can be used to synthesize virtual portraits given novel poses/expressions. However, this would provide ways for new malicious content by training a subject from their monocular video on the internet and generating new content without their consent. To mitigate these risks, future work should develop protective measures, such as adding digital watermarks to track the generated videos, and training specific detectors to identify deepfakes created by our method.

VI. Conclusions

This paper proposes a new dynamic ray-tracing-based Gaussian splatting (D-RayGS), which efficiently performs ray-tracing embedded Physical Based Rendering directly on the 2D Gaussian primitives. To preserve the high quality inverse rendering, we also provide a compact D-RayGS learning strategy which regularizes the compactness of Gaussian primitives using an auxiliary SDF field. Based on our D-RayGS and compact learning, we build an D-RayAvatar reconstruction from monocular videos, which achieves better Gaussian avatar reconstruction and enables better relightable rendering. We hope this work can inspire subsequent works for better realistic and efficient relightable Gaussian-based avatar reconstruction in the community, especially to simulate more advanced light effects in the dynamic scenarios.

References

[1] B. Egger, W. A. Smith, A. Tewari, S. Wuhrer, M. Zollhoefer, T. Beeler, F. Bernard, T. Bolkart, A. Kortylewski, S. Romdhani *et al.*, "3d morphable face models—past, present, and future," *ACM Trans. Graph.*, vol. 39, no. 5, pp. 1–38, 2020.

[2] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt, "State of the art on monocular 3d face reconstruction, tracking, and applications," in *Computer Graphics Forum*, vol. 37, no. 2, 2018, pp. 523–550.

[3] T. Liao, Y. Zheng, A. Karmanov, L. Hu, L. Jin, Y. Xiu, and H. Li, "Soap: Style-omniscient animatable portraits," in *ACM SIGGRAPH Conference*, 2025.

[4] G. Gafni, J. Thies, M. Zollhofer, and M. Nießner, "Dynamic neural radiance fields for monocular 4d facial avatar reconstruction," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8649–8658.

[5] P.-W. Grassal, M. Prinzler, T. Leistner, C. Rother, M. Nießner, and J. Thies, "Neural head avatars from monocular rgb videos," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18 653–18 664.

[6] Y. Hong, B. Peng, H. Xiao, L. Liu, and J. Zhang, "Headnerf: A real-time nerf-based parametric head model," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 20 374–20 384.

[7] Y. Zheng, V. F. Abrevaya, M. C. Bühler, X. Chen, M. J. Black, and O. Hilliges, "Im avatar: Implicit morphable head avatars from videos," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13 545–13 555.

[8] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[9] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10 318–10 327.

[10] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5865–5874.

[11] X. Gao, C. Zhong, J. Xiang, Y. Hong, Y. Guo, and J. Zhang, "Reconstructing personalized semantic facial nerf models from monocular video," *ACM Trans. Graph.*, vol. 41, no. 6, pp. 1–12, 2022.

[12] Y. Xu, L. Wang, X. Zhao, H. Zhang, and Y. Liu, "Avatarmav: Fast 3d head avatar reconstruction using motion-aware neural voxels," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023.

[13] X. Zhao, L. Wang, J. Sun, H. Zhang, J. Suo, and Y. Liu, "Havatar: High-fidelity head avatar via facial model conditioned neural radiance field," *ACM Trans. Graph.*, 2023.

[14] L. Wang, X. Zhao, J. Sun, Y. Zhang, H. Zhang, T. Yu, and Y. Liu, "Styleavatar: Real-time photo-realistic portrait avatar from a single video," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023.

[15] Z. Zheng, X. Zhao, H. Zhang, B. Liu, and Y. Liu, "Avatarrex: Real-time expressive full-body avatars," *ACM Trans. Graph.*, vol. 42, no. 4, 2023.

[16] X. Deng, Z. Zheng, Y. Zhang, J. Sun, C. Xu, X. Yang, L. Wang, and Y. Liu, "Ram-avatar: Real-time photo-realistic avatar from monocular videos with full-body control," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.

[17] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–14, 2023.

[18] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, and M. Nießner, "Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.

[19] L. Hu, H. Zhang, Y. Zhang, B. Zhou, B. Liu, S. Zhang, and L. Nie, "Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.

[20] J. Xiang, X. Gao, Y. Guo, and J. Zhang, "Flashavatar: High-fidelity head avatar with efficient gaussian embedding," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.

[21] Y. Xu, B. Chen, Z. Li, H. Zhang, L. Wang, Z. Zheng, and Y. Liu, "Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.

[22] Z. Shao, Z. Wang, Z. Li, D. Wang, X. Lin, Y. Zhang, M. Fan, and Z. Wang, "SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2024, pp. 1606–1616.

[23] L. Li, Y. Li, Y. Weng, Y. Zheng, and K. Zhou, "Rgbavatar: Reduced gaussian blendshapes for online modeling of head avatars," *arXiv preprint arXiv:2503.12886*, 2025.

[24] L. Schoneveld, Z. Chen, D. Davoli, J. Tang, S. Terazawa, K. Nishino, and M. Nießner, "Sheap: Self-supervised head geometry predictor learned via 2d gaussians," *arXiv preprint arXiv:2504.12292*, 2025.

[25] Y. Zheng, W. Yifan, G. Wetzstein, M. J. Black, and O. Hilliges, "Pointavatar: Deformable point-based head avatars from videos," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023.

[26] S. Bharadwaj, Y. Zheng, O. Hilliges, M. J. Black, and V. Fernandez-Abrevaya, "Flare: Fast learning of animatable and relightable mesh avatars," *ACM Trans. Graph.*, vol. 42, no. 6, 2023.

[27] S. Saito, G. Schwartz, T. Simon, J. Li, and G. Nam, "Relightable gaussian codec avatars," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.

[28] Z. Fan, S.-S. Huang, Y. Zhang, D. Shang, J. Zhang, Y. Guo, and H. Huang, "Rgavatar: Relightable 4d gaussian avatar from monocular videos," *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[29] D. Zhang, Y. Liu, L. Lin, Y. Zhu, K. Chen, M. Qin, Y. Li, and H. Wang, "Hravatar: High-quality and relightable gaussian head avatar," in *IEEE CVPR*, 2025.

[30] N. Moenne-Loccoz, A. Mirzaei, O. Perel, R. de Lutio, J. Martinez Esturo, G. State, S. Fidler, N. Sharp, and Z. Gojcic, "3d gaussian ray tracing: Fast tracing of particle scenes," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–19, 2024.

[31] H. Chen, Z. Lin, and J. Zhang, "Gi-gs: Global illumination decomposition on gaussian splatting for inverse rendering," 2025.

[32] S. Govindarajan, D. Rebain, K. M. Yi, and A. Tagliasacchi, "Radiant foam: Real-time differentiable ray tracing," *arXiv preprint arXiv:2502.01157*, 2025.

[33] R. Tobiasz, G. Wilczyński, M. Mazur, S. Tadeja, and P. Spurek, "Meshsplats: Mesh-based rendering with gaussian splatting initialization," *arXiv preprint arXiv:2502.07754*, 2025.

[34] K. Byrski, M. Mazur, J. Tabor, T. Dziarmaga, M. Kądziołka, D. Baran, and P. Spurek, "Raysplats: Ray tracing based gaussian splatting," *arXiv preprint arXiv:2501.19196*, 2025.

[35] C. Gu, X. Wei, Z. Zeng, Y. Yao, and L. Zhang, "Irgs: Inter-reflective gaussian splatting with 2d gaussian ray tracing," in *IEEE CVPR*, 2025.

[36] T. Xie, X. Chen, Z. Xu, Y. Xie, Y. Jin, Y. Shen, S. Peng, H. Bao, and X. Zhou, "Envgs: Modeling view-dependent appearance with environment gaussian," in *IEEE CVPR*, 2025.

[37] J. Gao, C. Gu, Y. Lin *et al.*, "Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing," in *Eur. Conf. Comput. Vis.* Springer Nature Switzerland, 2024, pp. 73–89.

[38] Y. Jiang, J. Tu, Y. Liu *et al.*, "Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 5322–5332.

[39] Y. Hong, Y. Wu, Z. Shen, C. Guo, Y. Jiang, Y. Zhang, J. Yu, and L. Xu, "Beam: Bridging physically-based rendering and gaussian modeling for relightable volumetric video," *arXiv preprint arXiv:2502.08297*, 2025.

[40] W. Zielonka, T. Bolkart, and J. Thies, "Instant volumetric head avatars," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4574–4584.

[41] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *CGI*, 1999, pp. 187–194.

[42] S. Ploumpis, H. Wang, N. Pears, W. A. Smith, and S. Zafeiriou, "Combining 3d morphable models: A large scale face-and-head model," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 934–10 943.

[43] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4d scans." *ACM Trans. Graph.*, vol. 36, no. 6, pp. 194–1, 2017.

[44] R. Daněček, M. J. Black, and T. Bolkart, "Emoca: Emotion driven monocular face capture and animation," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 20 311–20 322.

[45] Z. Qiu, Y. Li, D. He, Q. Zhang, L. Zhang, Y. Zhang, J. Wang, L. Xu, X. Wang, Y. Zhang *et al.*, "Sculptor: Skeleton-consistent face creation using a learned parametric generator," *ACM Trans. Graph.*, 2022.

[46] A. Lattas, S. Moschoglou, B. Gecer, S. Ploumpis, V. Triantafyllou, A. Ghosh, and S. Zafeiriou, "Avatarme: Realistically renderable 3d facial reconstruction" in-the-wild"," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 760–769.

[47] J. Shang, T. Shen, S. Li, L. Zhou, M. Zhen, T. Fang, and L. Quan, "Self-supervised monocular 3d face reconstruction by occlusion-aware multi-view geometry consistency," in *Eur. Conf. Comput. Vis.*, 2020, pp. 53–70.

[48] A. Tewari, F. Bernard, P. Garrido, G. Bharaj, M. Elgharib, H.-P. Seidel, P. Pérez, M. Zollhofer, and C. Theobalt, "Fml: Face model learning from videos," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 812–10 822.

[49] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Niessner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt, "Deep video portraits," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, 2018.

[50] X. Ji, H. Zhou, K. Wang, Q. Wu, W. Wu, F. Xu, and X. Cao, "Eamm: One-shot emotional talking face via audio-based emotion-aware motion model," in *SIGGRAPH Asia*, 2022.

[51] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2387–2395.

[52] J. Thies, M. Elgharib, A. Tewari, C. Theobalt, and M. Nießner, "Neural voice puppetry: Audio-driven facial reenactment," in *Eur. Conf. Comput. Vis.*, 2020, pp. 716–731.

[53] Y. Xu, H. Zhang, L. Wang, X. Zhao, H. Han, Q. Guojun, and Y. Liu, "Latentavatar: Learning latent expression code for expressive neural head avatar," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023.

[54] J. Sun, X. Wang, L. Wang, X. Li, Y. Zhang, H. Zhang, and Y. Liu, "Next3d: Generative neural texture rasterization for 3d-aware head avatars," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023.

[55] Y. Wu, S. Xu, J. Xiang, F. Wei, Q. Chen, J. Yang, and X. Tong, "Aniportraitgan: Animatable 3d portrait generation from 2d image collections," in *SIGGRAPH Asia 2023 Conference Proceedings*, 2023, pp. 1–9.

[56] C. Liu, T. Jing, C. Ma, X. Zhou, Z. Lian, Q. Jin, H. Yuan, and S.-S. Huang, "Emodifftalk: Emotion-aware diffusion for editable 3d gaussian talking head," in *IEEE CVPR*, 2026.

[57] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. pmlr, 2015, pp. 2256–2265.

[58] D. Gao, G. Chen, Y. Dong, P. Peers, K. Xu, and X. Tong, "Deferred neural lighting: free-viewpoint relighting from unstructured photographs," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–15, 2020.

[59] X. Zhang, S. Fanello, Y.-T. Tsai, T. Sun, T. Xue, R. Pandey, S. Orts-Escolano, P. Davidson, C. Rhemann, P. Debevec *et al.*, "Neural light transport for relighting and view synthesis," *ACM Trans. Graph.*, vol. 40, no. 1, pp. 1–17, 2021.

[60] K. Sarkar, M. C. Bühler, G. Li, D. Wang, D. Vicini, J. Riviere, Y. Zhang, S. Orts-Escolano, P. Gotardo, T. Beeler *et al.*, "Litnerf: Intrinsic radiance decomposition for high-quality view synthesis and relighting of faces," in *SIGGRAPH Asia 2023 Conference Proceedings*, 2023, pp. 1–11.

[61] Y. Xu, G. Zoss, P. Chandran, M. Gross, D. Bradley, and P. Gotardo, "Renerf: Relightable neural radiance fields with nearfield lighting," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 22 581–22 591.

[62] Z. Xu, K. Sunkavalli, S. Hadap, and R. Ramamoorthi, "Deep image-based relighting from optimal sparse samples," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–13, 2018.

[63] A. Meka, C. Haene, R. Pandey, M. Zollhöfer, S. Fanello, G. Fyffe, A. Kowdle, X. Yu, J. Busch, J. Dourgarian *et al.*, "Deep reflectance fields: high-quality facial reflectance field inference from color gradient illumination," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, 2019.

[64] A. Meka, R. Pandey, C. Häne, S. Orts-Escolano, P. Barnum, P. David-Son, D. Erickson, Y. Zhang, J. Taylor, S. Bouaziz, C. Legendre, W.-C. Ma, R. Overbeck, T. Beeler, P. Debevec, S. Izadi, C. Theobalt, C. Rhemann, and S. Fanello, "Deep relightable textures: volumetric performance capture with neural rendering," *ACM Trans. Graph.*, vol. 39, no. 6, nov 2020.

[65] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. Lensch, "Nerd: Neural reflectance decomposition from image collections," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12 684–12 694.

[66] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "Nerv: Neural reflectance and visibility fields for relighting and view synthesis," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7495–7504.

[67] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 1–18, 2021.

[68] Y. Zhang, J. Sun, X. He, H. Fu, R. Jia, and X. Zhou, "Modeling indirect illumination for inverse rendering," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18 643–18 652.

[69] J. Munkberg, J. Hasselgren, T. Shen, J. Gao, W. Chen, A. Evans, T. Müller, and S. Fidler, "Extracting triangular 3d models, materials, and lighting from images," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8280–8290.

[70] H. Kim, M. Jang, W. Yoon, J. Lee, D. Na, and S. Woo, "Switchlight: Co-design of physics-driven architecture and pre-training framework for human portrait relighting," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2024, pp. 25 096–25 106.

[71] L. Zhang, A. Rao, and M. Agrawala, "Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport," in *ICLR*, 2025.
[72] A. Dib, C. Thebault, J. Ahn, P.-H. Gosselin, C. Theobalt, and L. Chevallier, "Towards high fidelity monocular face reconstruction with rich reflectance using self-supervised learning and ray tracing," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12 819–12 829.
[73] H. Feng, T. Bolkart, J. Tesch, M. J. Black, and V. Abrevaya, "Towards racially unbiased skin tone estimation via scene disambiguation," in *Eur. Conf. Comput. Vis.* Springer, 2022, pp. 72–90.
[74] Z. Xu, S. Peng, C. Geng, L. Mou, Z. Yan, J. Sun, H. Bao, and X. Zhou, "Relightable and animatable neural avatar from sparse-view video," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.
[75] J. Li, C. Cao, G. Schwartz *et al.*, "Uravatar: Universal relightable gaussian codec avatars," in *SIGGRAPH Asia 2024 Conference Papers*, 2024, pp. 1–11.
[76] B. Huang, Z. Yu, A. Chen *et al.*, "2d gaussian splatting for geometrically accurate radiance fields," in *ACM SIGGRAPH 2024 Conference Proceedings*, 2024, pp. 1–11.
[77] J. T. Kajiya, "The rendering equation," in *ACM SIGGRAPH*, 1986, pp. 143–150.
[78] Y. Yao, J. Zhang, J. Liu, Y. Qu, T. Fang, D. McKinnon, Y. Tsin, and L. Quan, "Neilf: Neural incident light field for physically-based material estimation," in *Eur. Conf. Comput. Vis.* Springer, 2022, pp. 700–716.
[79] X. Chen, S. Peng, D. Yang *et al.*, "Intrinsicanything: Learning diffusion priors for inverse rendering under unknown illumination," in *European Conference on Computer Vision (ECCV)*. Cham: Springer Nature Switzerland, 2024, pp. 450–467.
[80] W. Zielonka, T. Bolkart, and J. Thies, "Towards metrical reconstruction of human faces," in *Eur. Conf. Comput. Vis.* Springer, 2022, pp. 250–269.
[81] Z. Zhang, L. Li, Y. Ding, and C. Fan, "Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3661–3670.
[82] S. Ma, Y. Weng, T. Shao, and K. Zhou, "3d gaussian blendshapes for head avatar animation," in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–10.

Zhe Fan is currently a Ph.D. candidate at the School of Computer Science & Technology, Beijing Institute of Technology. Before this, he received his Master's degree from Beihang University in 2019. His research interests include computer graphics and digital human.

Shi-Sheng Huang is currently an associate professor in the school of Artificial Intelligence, Beijing Normal University. Before this, he was a PostDoc researcher at Tsinghua University. He got his Ph.D. degree in computer science and technology from Tsinghua University, Beijing, in 2015. His primary research interests include fields of computer graphics, and has published relevant research works in ACM TOG/IEEE TVCG/IEEE CVPR etc.

ZhaoChen Li began his studies at Beijing Normal University in 2022 and is currently enrolled there. His current research interests include computer vision and video generation models.

Hua Huang (Senior Member, IEEE) received the BS and PhD degrees from Xi'an Jiaotong University, Xi'an, China, in 1996 and 2006, respectively. He is currently a professor in the School of Artificial Intelligence, Beijing Normal University. His main research interests include image and video processing, computational photography, and computer graphics. He received the Best Paper Award of ICML2020 / EURASIP2020 / PRCV2019 / ChinaMM2017.

APPENDIX A
IMPLEMENTATION DETAILS

A. Network Architecture

We present the architecture of the the geometry network, motion deformation network and the shape deformation network in Fig. 1.

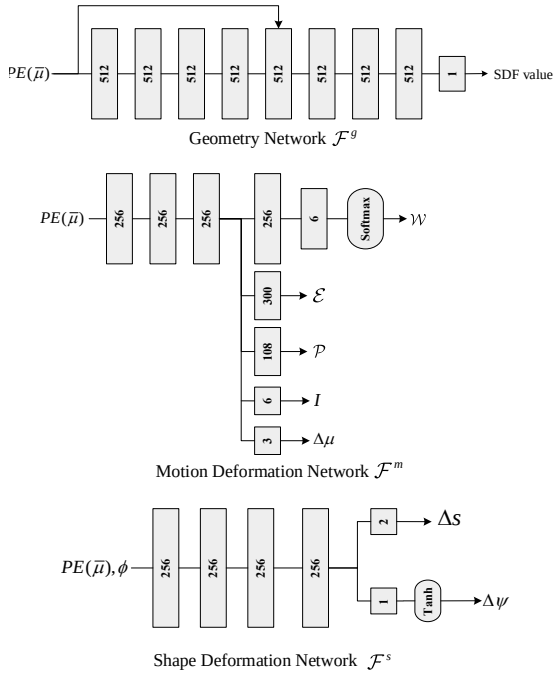

Fig. 1. Network architectures of the Geometry, Motion Deformation and Shape Deformation MLPs.

B. Training Details

We train the model for 50 epochs in the coarse stage and 30 epochs in the fine stage. For each epoch, We randomly sample a subset consisting half of the images for training. All experiments were conducted on a single NVIDIA RTX 3090 GPU device. During the coarse stage, we observed that directly optimizing the SDF and canonical 2D Gaussian positions jointly leads to poor convergence. Specifically, since the SDF is inaccurate at the beginning, applying the SDF constraint too early severely limits the movement of 2D Gaussians, preventing them from moving to correct geometric positions. This results in over-smoothed geometry that lacks high-frequency details. To address this, we adopt a scheduled strategy for the coarse stage. For the first 30 epochs, we disable the SDF constraint on the Gaussians, only allowing the SDF to fit the Gaussian positions while using Normal Regularization to guide their rotation. After obtaining a reasonable geometry initialization, we perform full joint optimization in the remaining 20 epochs. In this phase, the SDF and Gaussians are mutually optimized, ensuring the 2D Gaussians are tightly aligned with the surface represented by the SDF.

In the fine stage, we reduce the learning rates of all geometry-related parameters by a factor of 4 to 16, shifting the optimization focus primarily towards material and lighting parameters. Specifically, the learning rates for albedo, roughness, and Fresnel reflectivity are set to 0.005, 0.005, and 0.001, respectively, with the environment map set to 0.02. Regarding the rendering, we sample 64 rays per Gaussian to jointly evaluate direct illumination and perform ray-tracing. Furthermore, the Bounding Volume Hierarchy (BVH) is updated at every deformation to ensure accurate intersection. Our full coarse-to-fine training process requires approximately 9 hours, due to the computation required by optimizing the deep SDF network and perform dynamic ray-tracing. We intend to improve this efficiency in our future works.

C. Albedo Regulation Details

As mentioned in the main text, we fine-tuned the albedo inference checkpoint of Intrinsic Anything [1] and further proposed an albedo correction method. Fig. 2 illustrates a subset of frames from a video sequence alongside the corresponding albedo priors involved in albedo regulation.

Fine-tuning of the Prior model. The original Intrinsic Anything framework leverages diffusion models to address material inference for generic objects. However, due to the inherent complexity of facial materials and the lack of face-specific optimization, its inference on human subjects sometimes yields inaccurate results. As shown in the second row of Fig. 2, the albedo directly estimated by the vanilla model exhibits inconsistencies with the actual skin tone and fails to disentangle specular highlights from the diffuse component. To address this limitation, we constructed a specialized dataset for human head albedo inference, utilizing high-quality PBR digital assets from [2]. The dataset construction pipeline is designed to mimic the characteristics of monocular portrait videos. Specifically, for each head 3D model, we randomly sampled 30 dense viewpoints combined with 50 different HDR environment maps. We utilized the Blender Cycles rendering engine to generate paired data consisting of the rendered images and their corresponding ground-truth albedo maps. The dataset comprises 120 head models, covering a diverse range of skin tones and genders. Following the aforementioned protocol, a total of 184,500 image pairs were generated. For the fine-tuning process, we followed the training methodology from IID [3] but fine-tuning only the attention layers. The model was trained on two NVIDIA A800 GPUs for a duration of five days. Since the synthetic head models utilized in our dataset construction exclude hair geometry, the fine-tuned model is primarily specialized for the facial and neck regions. After this fine-tuning process, the prior model demonstrates a significantly improvement in the accuracy of facial albedo inference. The third row of Fig. 2 visualizes the results generated by our fine-tuned model on the input sequence. As observed, compared to the vanilla version, the baked-in specular highlights and shadows are effectively eliminated.

Albedo Correction. Although the fine-tuned model generally recovers albedo colors close to the intrinsic skin tone, failures still occur due to the inherent ambiguity of single-image albedo inference (as shown in Fig.10 of the main

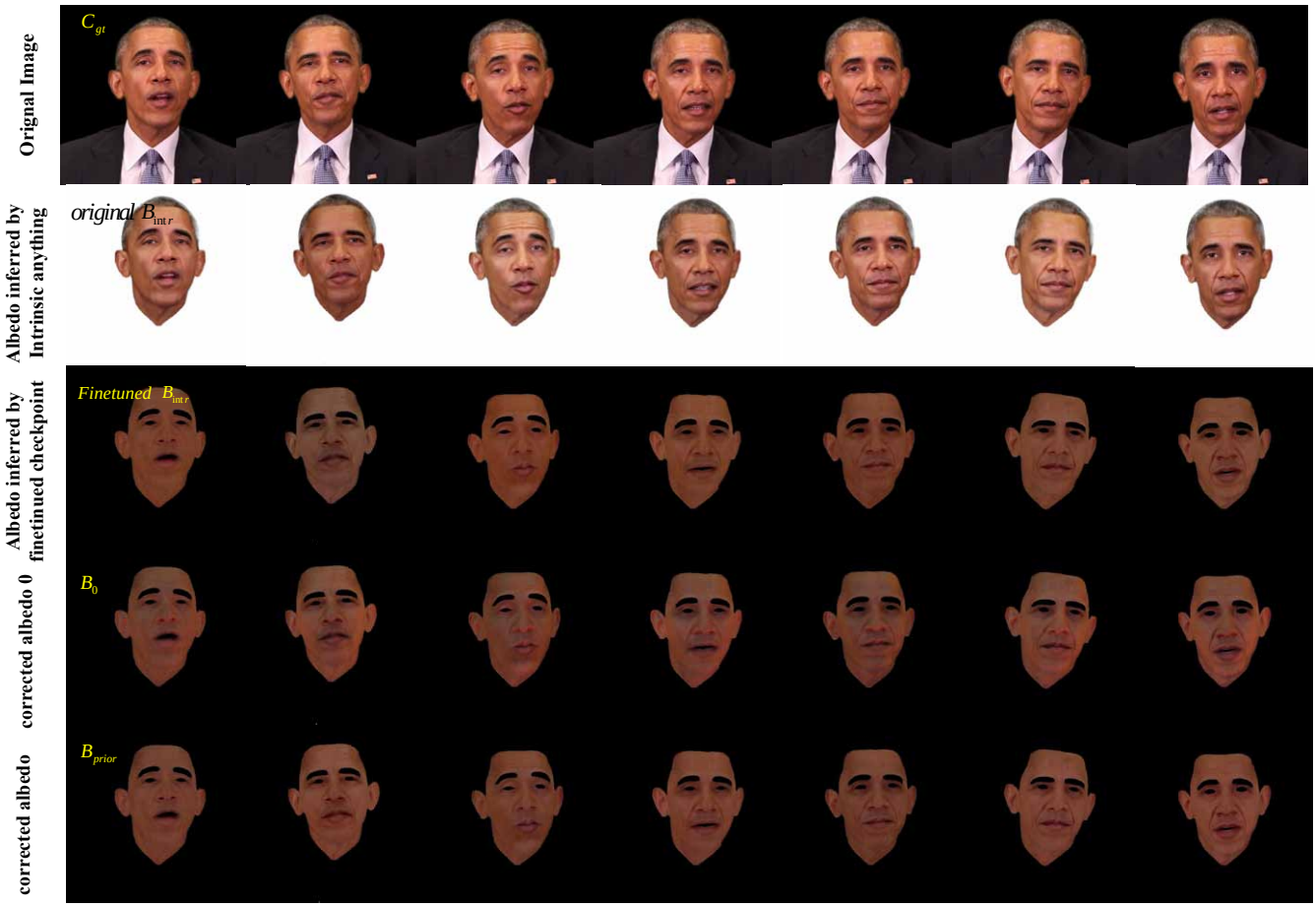

Fig. 2. Albedo Regulation Details.

text). Moreover, when processing image sequences, temporal inconsistency becomes another challenge. As shown in the third row of Fig. 2, the estimated albedo B_{intr} for the same subject exhibits noticeable fluctuations across different frames, shifting inconsistently between darker and lighter tones. To mitigate these issues, we proposed the albedo correction method mentioned in the main text.

we first convert the colored shading, which is calculated by the albedo B_{intr} and the training image C_{gt}, into a grayscale color space S_{gray}. The corrected albedo $B_{prior}$is then obtained as follows

$$S_{gray} = f_{lum}(\frac{C_{gt}}{B_{intr}}), B_0 = \frac{C_{gt}}{S_{gray}}$$
$$B_{prior} = B_{intr} + \frac{\sum_p M_p(B_0 - B_{intr})}{\sum_p M_p} \quad (1)$$

where f_{lum} is the weighted luminescence transformation, $\mathbf{M} \in \{0,1\}^{H\times W}$ is the binary mask and $M_{h,w} = 1$ indicates the face skin region.

After obtaining the final B_{prior}, we formulate the albedo regularization term as follows:

$$\mathcal{L}_{albedo} = |\mathbf{B} - f_{lin}(\mathbf{B}_{prior})| \quad (2)$$

where f_{lin} denotes the sRGB-to-linear transformation function (or inverse gamma correction). We apply f_{lin} to linearize the sRGB prior B_{prior}, ensuring alignment with our linear PBR rendering pipeline.

It is worth noting that we did not directly adopt the B_0 (calculated after shading rectification) as the final prior. This is because the conversion from colored shading to white (grayscale) shading inevitably introduces intensity errors. As observed in the fourth row of Fig. 2, B_0 suffers from color unevenness caused by these shading conversion inaccuracies; especially, some highlights that were previously removed re-emerge in this map. Consequently, we employed B_0 solely as a reference to perform global color correction on the initial estimate B_{intr}. We calculated the average color difference between B_{intr} and B_0 and applied it to shift the tone of B_{intr}, thereby preserving the spatial uniformity of B_{intr}'s color distribution while correcting its global chromaticity. As demonstrated in the fifth row of Fig. 2, our final corrected results B_{prior} effectively resolve the aforementioned issues, achieving precise albedo-shading disentanglement while maintaining both skin tone accuracy and temporal consistency.

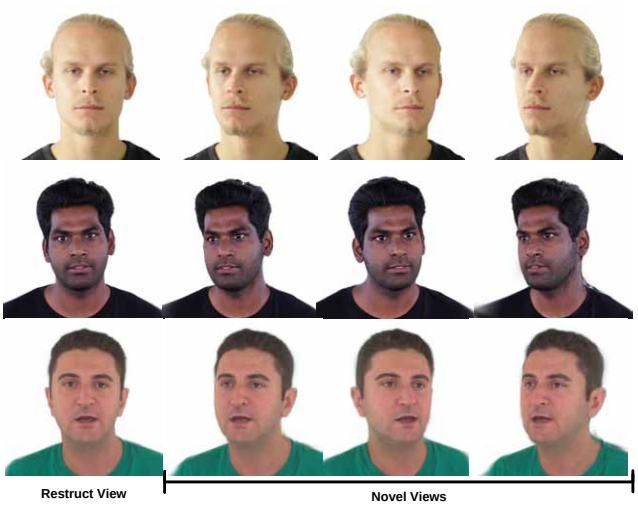

Fig. 3. visual results of novel Views Synthesis results of different subjects.

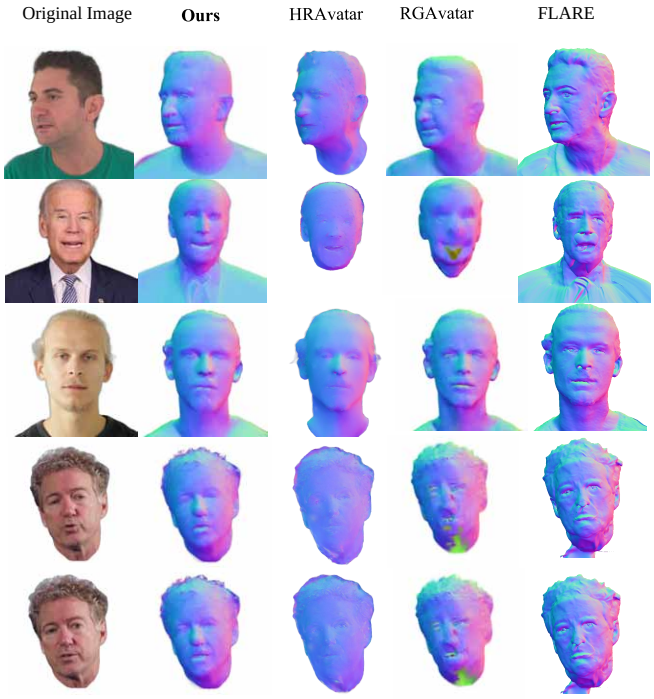

Fig. 4. geometry comparison results of different subjects between FLARE, HRAvatar, RGAvatar and our approach.

APPENDIX B
ADDITIONAL RESULTS

A. Novel Views Synthesis

In addition to deformation control via FLAME pose and expression parameters, our avatar supports explicit viewpoint manipulation.Fig. 3 presents the novel view synthesis results for 3 subjects. As observed, despite the inherent ambiguity of monocular video input, our approach maintains high-fidelity details and consistency for both the head and upper body across a range of viewing angles. This robustness is largely attributed to the effective modeling and optimization strategy of our proposed Compact D-RayGS.

B. Geometry Comparison

Fig. 4 presents a qualitative geometric comparison among HRAvatar, RGAvatar, FLARE, and ours. It can be observed that, FLARE is capable of capturing fine details but inevitably introduces noisy artifacts and holes, limited by its mesh representation. RGAvatar can reconstruct the overall geometric shape but fails to recover accurate geometry in certain instances, particularly for cases with a limited range of head rotation in training videos, it yields coarse shapes with noticeable artifacts. HRAvatar consistently recovers smooth and high-quality geometry. However, in some cases, it struggles to sufficiently disentangle and reconstruct continuous geometry from the source images. For instance, the normal map in the last row exhibits several white regions, which erroneously correspond to the specular highlight details of the face. By leveraging a Signed Distance Field (SDF) to focus on continuous and accurate surface reconstruction, our method facilitates precise per-Gaussian ray tracing while simultaneously achieving superior geometry quality.

C. Relighting Comparison

Fig. 5 presents a qualitative relighting comparison with two state-of-the-art monocular Gaussian-based avatar approaches, including HRAvatar and RGAvatar. We exclude FLARE(mesh-based) to focus on the more competitive Gaussian-based baselines. As observed, our method more faithfully preserves the subject's intrinsic skin tone while simultaneously exhibiting richer and more nuanced shading variations. Ours renders superior specular highlights and realistic shadow effects, which is attributed to our dynamic ray-tracing-based Gaussian Splatting. In contrast, the other two baselines rely on simplified illumination models, resulting in a flat appearance that fails to resolve complex lighting interactions.

REFERENCES

[1] X. Chen, S. Peng, D. Yang *et al.*, "Intrinsicanything: Learning diffusion priors for inverse rendering under unknown illumination," in *European Conference on Computer Vision (ECCV)*. Cham: Springer Nature Switzerland, 2024, pp. 450–467.
[2] Epic Games, "Metahuman," https://www.unrealengine.com/en-US/metahuman, 2021.
[3] P. Kocsis, V. Sitzmann, and M. Nießner, "Intrinsic image diffusion for indoor single-view material estimation," 2024.

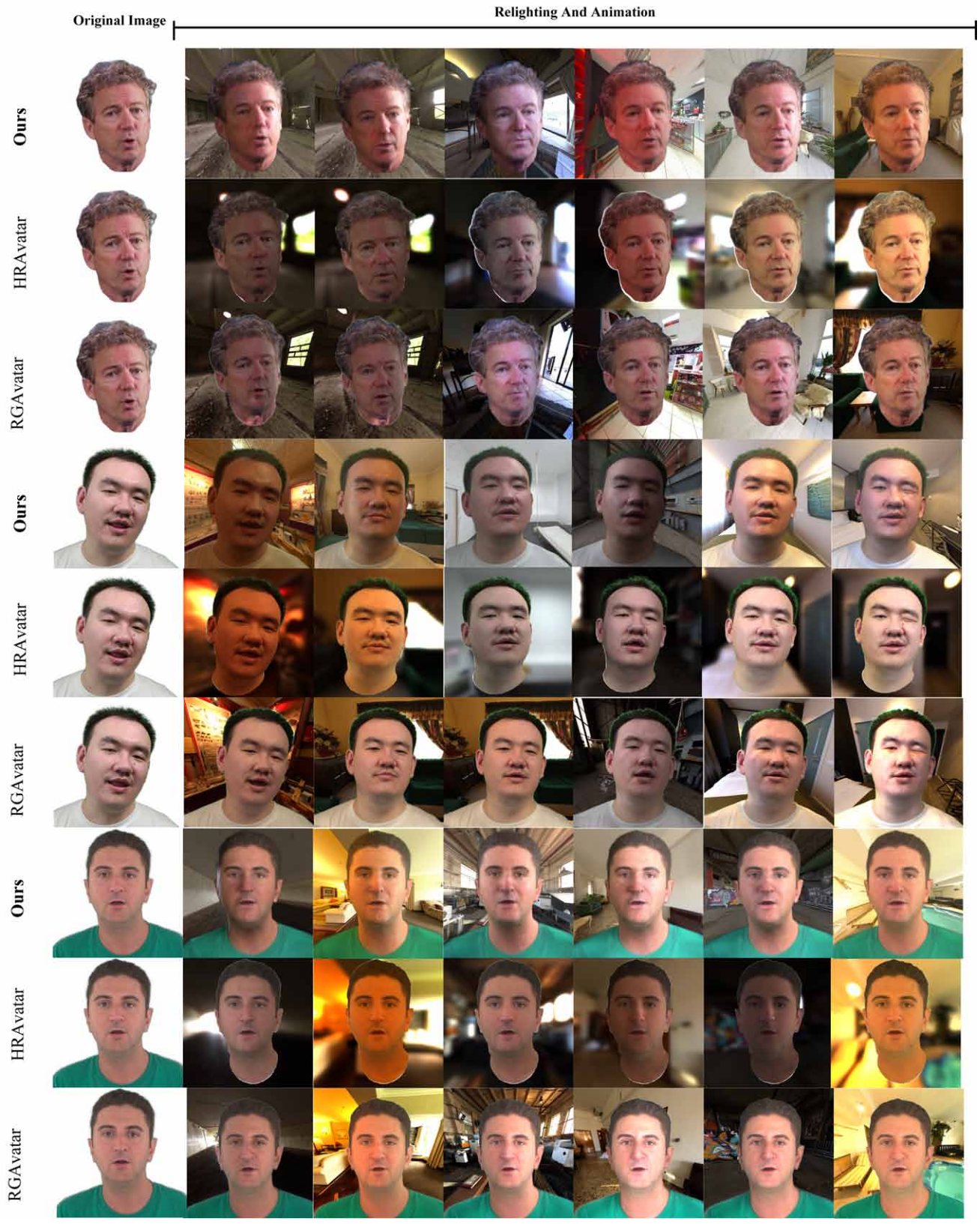

Fig. 5. More relighting comparison between HRAvatar, RGAvatar and our approach, by giving different environmental lights.