GAvatar: Animatable 3D Gaussian Avatars with Implicit Mesh Learning

Ye Yuan* Xueting Li* Yangyi Huang Shalini De Mello Koki Nagano Jan Kautz Umar Iqbal
NVIDIA

https://nvlabs.github.io/GAvatar

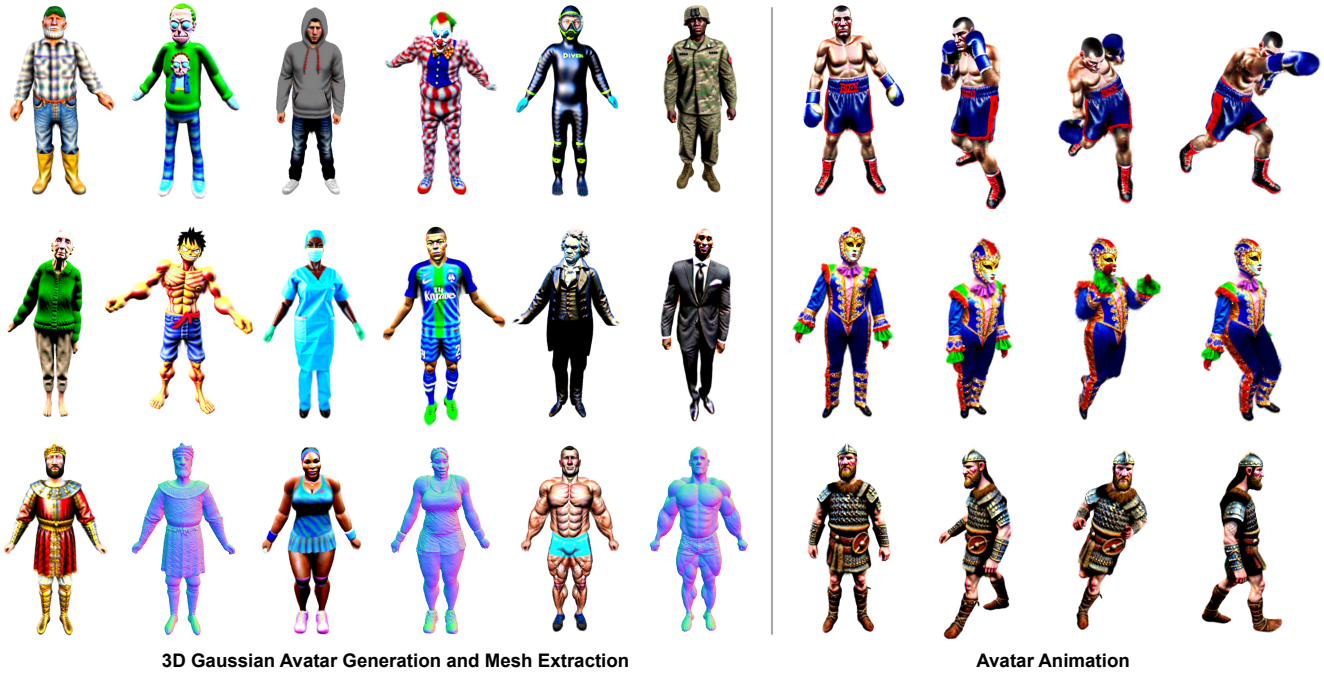

Figure 1. GAvatar synthesizes high-fidelity 3D animatable avatars from text prompts. Our novel primitive-based implicit Gaussian representation enables efficient avatar animation (100 fps, 1K resolution) and also extracts a highly detailed mesh from learned 3D Gaussians.

Abstract

Gaussian splatting has emerged as a powerful 3D representation that harnesses the advantages of both explicit (mesh) and implicit (NeRF) 3D representations. In this paper, we seek to leverage Gaussian splatting to generate realistic animatable avatars from textual descriptions, addressing the limitations (e.g., flexibility and efficiency) imposed by mesh or NeRF-based representations. However, a naive application of Gaussian splatting cannot generate high-quality animatable avatars and suffers from learning instability; it also cannot capture fine avatar geometries and often leads to degenerate body parts. To tackle these problems, we first propose a primitive-based 3D Gaussian representation where Gaussians are defined inside pose-driven primitives to facilitate animation. Second, to stabilize and amortize the learning of millions of Gaussians, we propose to use neural implicit fields to predict the Gaussian attributes (e.g., colors). Finally, to capture fine avatar geometries and extract detailed meshes, we propose a novel SDF-based implicit mesh learning approach for 3D Gaussians that regularizes the underlying geometries and extracts highly detailed textured meshes. Our proposed method, GAvatar, enables the large-scale generation of diverse animatable avatars using only text prompts. GAvatar significantly surpasses existing methods in terms of both appearance and geometry quality, and achieves extremely fast rendering (100 fps) at 1K resolution.

1. Introduction

Digital avatars play an essential role in numerous applications, from augmented and virtual reality to gaming, movie production, and synthetic data generation [8, 20, 21, 35, 44, 51–53]. However, highly realistic and animatable avatars are extremely difficult to create at scale due to the complexity and diversity of character geometries and appearances. Traditional approaches rely on manual modeling and rig-

*Equal contribution.

ging of digital avatars, which are labor-intensive and time-consuming. Recent advances in text-to-image generative models trained on large-scale data show impressive results in generating highly diverse and realistic human images from text [6, 33, 34, 48]. In light of this, several methods are proposed to generate 3D avatars from textual descriptions by distilling the 2D prior of these generative models into 3D avatar representations [9, 11, 18]. While their results are promising, the quality of the generated avatars is limited by the 3D representations they use, which are typically based on mesh or neural radiance field (NeRF) [27]. Mesh-based representations allow efficient rendering through rasterization, but the expressiveness to capture diverse geometry and fine details is limited due to the underlying topology. NeRF-based representations are expressive in modeling complex 3D scenes, but they are computationally expensive due to the large number of samples required by volume rendering to produce high-resolution images. As a result, existing avatar generation methods often fail to both generate fine-grained, out-of-shape geometric details, such as loose clothing, and efficiently render high-resolution avatars, which are critical for interactive and dynamic applications.

We aim to address these issues by adopting a new 3D representation, 3D Gaussian Splatting [17], which represents a scene using a set of 3D Gaussians with color, opacity, scales, and rotations and produces rendering by differentiably splatting the Gaussians onto an image. Gaussian splatting combines the advantages of both mesh and NeRF-based representations and it is both efficient and flexible to capture fine details. However, naive applications of Gaussian splatting to avatar generation fail for several reasons due to the unconstrained nature of individual Gaussians. First, the Gaussian splatting representation is not animatable, as the Gaussians are defined in the world coordinate and cannot be easily transformed with the avatar's pose in a coherent manner. Second, a large number (millions) of Gaussians are required to model a highly detailed avatar, and the immense optimization space of individual Gaussian attributes (*e.g.*, color, opacity, scale, rotation) leads to unstable optimization, especially when using high-variance objectives such as SDS [29]. Third, the 3D Gaussians lack explicit knowledge of surfaces, and cannot easily incorporate surface normal supervision, which is crucial for extracting highly detailed 3D meshes [4, 10]. Without geometry supervision, missing or degenerate body parts can appear when using weak 3D supervision (*i.e.*, SDS), which we will show in the experiments.

To tackle these problems, we propose GAvatar, a novel approach that leverages Gaussian Splatting to generate realistic animatable avatars from textual descriptions. First, we introduce a new primitive-based 3D Gaussian representation that defines 3D Gaussians inside pose-driven primitives. This representation naturally supports animation and enables flexible modeling of fine avatar geometry and appearance by deforming both the Gaussians and the primitives. Second, we propose to use implicit Gaussian attribute fields to predict the Gaussian attributes, which stabilizes and amortizes the learning of a large number of Gaussians, and allows us to generate high-quality avatars using high-variance optimization objectives such as SDS. Additionally, after avatar optimization, since we can obtain the Gaussian attributes directly and skip querying the attribute fields, our approach achieves extremely fast (100 fps) rendering of neural avatars at a resolution of 1024×1024. This is significantly faster than existing NeRF-based avatar models [3, 18] that query neural field for each novel camera view and avatar pose. Finally, we also propose a novel signed distance function (SDF)-based implicit mesh learning approach that connects SDF with Gaussian opacities. Importantly, it enables GAvatar to regularize the underlying geometry of the Gaussian avatar and extract high-quality textured meshes. Our contributions are summarized as follows:

- We introduce a new primitive-based implicit Gaussian representation for animatable avatars, enabling more stable and high-quality 3D avatar generation. It also allows extremely fast rendering (100 fps) at 1K resolution.
- We propose a novel SDF-based method that effectively regularizes the underlying geometry of 3D Gaussians and also enables the extraction of high-quality textured meshes from the learned Gaussians avatar.
- Our approach generates 3D avatars with fine geometry and appearance details. We experimentally demonstrate that GAvatar consistently outperforms existing methods in terms of avatar quality.

2. Related Work

3D Representations for 3D Content Generation. Various 3D representations have been employed for 3D content generation, each with its own set of strengths and limitations. Triangulated meshes are a common choice due to their simplicity and compatibility with existing graphics pipelines [14]. However, their inflexible topology can pose challenges in accurately representing intricate geometries. Alternatively, volumetric representations, such as voxel grids [39], offer flexibility in modeling complex shapes. Nevertheless, their computational and memory costs grow cubically with resolution, impeding the faithful reconstruction of fine geometry details and smooth surfaces. Recently, NeRFs [27] have gained prominence for modeling 3D shapes, especially in text-to-3D applications, thanks to their ability to capture arbitrary topologies with minimal memory usage. Yet, their rendering cost increases significantly at higher resolutions. Some approaches adopt hybrid representations to harness the benefits of different techniques. The Mixture of Volumetric Primitives (MVP) rep-

resentation [25], for instance, introduces volumetric primitives onto a template mesh, achieving rapid rendering by leveraging a convolutional network to compute volumetric primitives. It generates images through ray-marching, accumulating colors and opacities from the primitives. Gaussian Splatting [17] has emerged as a promising 3D representation for efficiently rendering high-resolution images. It models objects using colored 3D Gaussians, which are rendered onto an image using splatting-based rasterization. However, a notable limitation is its difficulty in extracting meshes from learned Gaussians, as it predominantly captures appearance details through 3D Gaussians without modeling the underlying object surfaces.

In this work. we introduce a novel primitive-based 3D Gaussian representation with implicit mesh learning. It enables modeling dynamic and articulated objects like humans using Gaussian Splatting while also facilitating textured mesh extraction. In comparison to MVP, our Gaussian-based representation is more flexible and expressive, since each primitive comprises a variable number of 3D Gaussians with varying non-uniform locations that can go beyond the primitive boundaries. This allows it to capture finer details compared to the cubic primitives used in MVP. Moreover, our representation employs splatting-based rasterization, enabling efficient rendering of high-resolution images compared to traditional ray-marching techniques.

Text-to-3D Generation. The field of text-to-3D generation has recently been revolutionized [4, 23, 29, 32, 32, 40, 43] with the availability of large text-to-image models [6, 33, 34, 48]. The earlier methods optimize the 3D objects by encouraging the 2D rendering to be consistent with the input text in the CLIP [30] embeddings space [5, 13, 14, 36, 41, 45]. While they demonstrated the usefulness of text-to-image models for 3D content generation, the resulting 3D models often lacked realism and fine geometry details. The seminal work DreamFusion [29] replaces the CLIP model with a text-to-image diffusion model and proposed Score Distillation Sampling (SDS) to optimize a NeRF-based representation of the 3D object. Since then multiple variants of this method have been proposed. Magic3D [23] enhances runtime efficiency with a two-staged framework and adopts a more efficient DMTet [7] representation. ProlificDreamer [43] addresses over-saturation/smoothing issues through a variational SDS objective. MVDream [38] fine-tunes text-to-image models to generate 3D-consistent multi-view images, enabling efficient 3D generations. Fantasia3D [4] disentangles geometry and appearance modeling, optimizing surface normals separately using the SDS loss. More recently, DreamGaussian [40] replaced the NeRF-based representation with Gaussian Splatting to significantly reduce runtime. However, this leads to 3D models with limited geometry and appearance quality, despite attempts to refine texture details through mesh-based fine-tuning. It is important to note that all these methods are limited to rigid objects only and cannot be animated easily.

Text-to-3D Avatar Generation Building upon the success achieved in generating static 3D objects, numerous methods have been proposed to model dynamic objects, particularly human or human-like avatars [3, 10–12, 14, 15, 18, 22, 47, 50]. ClipMatrix [14] is one of the first methods that showcased the creation of animatable avatars based on textual descriptions. It achieves this by optimizing a mesh-based representation using a CLIP-embedding loss. AvatarClip [9] follows a similar pipeline but employs a NeRF-based representation [42]. DreamAvatar [3] and AvatarCraft [15] utilize SDS loss instead of CLIP, and learn the NeRF representation in canonical space through the integration of human body priors from SMPL [26]. DreamHumans [18] introduces a deformable and pose-conditioned NeRF model by incorporating the imGHUM [2] model. DreamWaltz [11] and AvatarVerse [47] leverage pose-conditioned ControlNets [48], showcasing improved avatar quality with conditional SDS. However, a common limitation among these methods is their reliance on NeRF to generate images, resulting in the computation of SDS loss based on low-resolution images. For instance, DreamHumans [18] generates 64×64 images during optimization, leading to a compromise in avatar quality. In contrast, our approach can efficiently generate images with a resolution of 1024×1024, resulting in higher-quality avatars, as demonstrated in our experiments. There are several contemporary works that demonstrate impressive avatar quality [10, 22, 46]. TADA [22] shows that a mesh-based approach with adaptive mesh subdivision can be used to generate high-quality avatars. HumanNorm [10] finetunes text-to-image models to directly generate normal and depth maps from the input text. The adapted models are then utilized to optimize the avatar's geometry through the SDS loss, with texture optimization achieved using a normal-conditioned ControlNet [48]. Similarly, AvatarBooth [46] fine-tunes region-specific diffusion models, highlighting that employing dedicated models for distinct body regions enhances avatar quality. These improved optimization objectives are complementary to our method since they are compatible with our Gaussian-based 3D representation. Since our model can efficiently render high-resolution images and normals, we anticipate synergies between our approach and [10, 46, 47] to yield further enhancements.

3. Preliminaries

Primitive-based 3D Representation. Primitive-based methods represent a 3D scene by a set of primitives such as cubes [25, 31], points [1] or nerflets [49]. In this work, we adopt the primitive formulation used in [25, 31]: a set

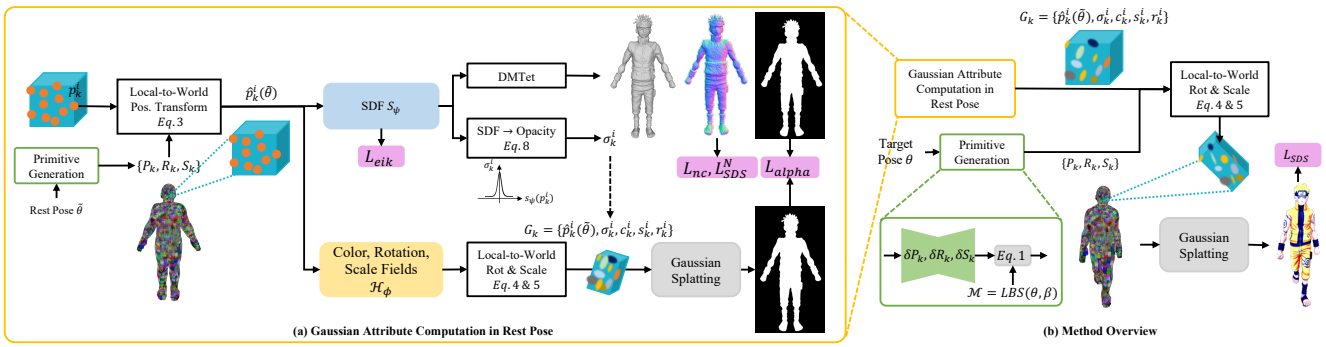

Figure 2. **Overview of GAvatar.** We first generate the primitives $V_k{=}(P_k, R_k, S_k)$ in the rest pose $\tilde{\theta}$. Each primitive consists of N_k 3D Gaussians with their position p_k^i, rotation r_k^i and scaling s_k^i defined in the primitive's local coordinate system. Next, we obtain the canonical positions, $\hat{p}_k^i(\tilde{\theta})$, of the Gaussians in the world coordinates by applying the global transforms of the primitives using Eq. 3. These positions are then used to query the color c_k^i, rotation r_k^i and scaling s_k^i of each Gaussian from a neural attribute field $\mathcal{H}_\phi$. Each Gaussian's SDF value is queried from a neural SDF $\mathcal{S}_\psi$ and is converted into the opacity σ_k^i through a kernel function $\mathcal{K}$. The 3D Gaussians with the predicted attributes are then rasterized onto the camera view using Gaussian splatting to produce the RGB image I and alpha image I_α. We use DMTet [37] to differentiably extract the mesh from the Gaussian SDF values and generate its normal map and silhouette for geometry regularization. For animating the avatar using any target pose θ, we generate the primitives using the target pose and use them to transform the 3D Gaussians, before rasterizing the image. A method walkthrough is also provided in the supplementary video.

of K cubic primitives $\{V_1, \ldots, V_K\}$ are attached to the surface of a SMPL-X [28] mesh $\mathcal{M} = \text{LBS}(\theta, \beta)$, where θ and β are the SMPL-X pose and shape parameters, and LBS is the linear blend skinning function. Each primitive $V_k = \{P_k, R_k, S_k\}$ is defined by its location $P_k \in \mathbb{R}^3$, per-axis scale $S_k \in \mathbb{R}^3$ and orientation $R_k \in \text{SO}(3)$. The primitive parameters are generated by:

$$\begin{aligned} P_k(\theta) &= \hat{P}_k(\mathcal{M}) + \delta P_\omega(\theta)_k, \\ R_k(\theta) &= \delta R_\omega(\theta)_k \cdot \hat{R}_k(\mathcal{M}), \\ S_k(\theta) &= \hat{S}_k(\mathcal{M}) + \delta S_\omega(\theta)_k, \end{aligned} \tag{1}$$

where we first compute a mesh-based primitive initialization $\hat{P}_k(\mathcal{M}), \hat{R}_k(\mathcal{M}), \hat{S}_k(\mathcal{M})$, and then apply pose-dependent correctives $\delta P_\omega(\theta), \delta R_\omega(\theta), \delta S_\omega(\theta)$, which are represented by neural networks with parameters ω. The mesh-based initialization is computed by placing the primitives on a 2D grid in the mesh's uv-texture space and generating the primitives at the 3D locations on the mesh surface points corresponding to the uv-coordinates. The overall deformation process is illustrated in Fig. 2 (green box) and more details can be found in [31].

Score Distillation Sampling. First proposed in DreamFusion [29], score distillation sampling (SDS) can be used to optimize the parameters η of a 3D model g using a pre-trained text-to-image diffusion model. Given a text prompt y and the noise prediction $\hat{\epsilon}(I_t; y, t)$ of the diffusion model, SDS optimizes model parameters η by minimizing the difference between the noise ϵ added to the rendered image $I = g(\eta)$ and the predicted noise $\hat{\epsilon}$ by the diffusion model:

$$\nabla_\eta \mathcal{L}_{\text{SDS}} = E_{t,\epsilon}\left[w(t)(\hat{\epsilon}(I_t; y, t) - \epsilon)\frac{\partial I}{\partial \eta}\right], \tag{2}$$

where $g(\eta)$ denotes the differentiable rendering process of the 3D model, t is the noise level, I_t is the noised image, and $w(t)$ is a weighting function.

4. Approach

Our approach, GAvatar, generates a 3D Gaussian-based animatable avatar given a text prompt. Our key ideas are two-fold: (1) we introduce a new primitive-based implicit 3D Gaussian representation (Sec. 4.1) that not only enables avatar animation but also stabilizes and amortizes the learning of a large number of Gaussians using the high-variance SDS loss; (2) we represent the underlying geometry of 3D Gaussians with an SDF that enables extracting high-quality textured meshes and regularizing the avatar's geometry (Sec. 4.2). The training process of our approach is described in Sec. 4.3 and an overview of our method is provided in Fig. 2.

4.1. Primitive-based Implicit Gaussian Avatar

Recently, Gaussian Splatting [17] has emerged as a powerful representation for 3D scene reconstruction and generation thanks to its efficiency and flexibility. However, naive application of Gaussian Splatting to human avatar generation poses animation and training stability challenges. Specifically, two essential questions arise: (1) how to transform the Gaussians defined in the world coordinate system along with the deformable avatar and (2) how to learn Gaussians with consistent attributes (i.e., color, rotation, scaling, etc.) within a local neighborhood. In the following, we answer both questions by proposing a primitive-based implicit Gaussian representation.

Primitive-based 3D Gaussian Avatar. To generate an

animatable human avatar, we start with the primitive formulation discussed in Sec. 3, where the human body is represented by a set of primitives attached to its surface. Since the primitives are naturally deformed according to the human pose and shape, we propose to attach a set of 3D Gaussians $\{G_k^1, \dots, G_k^{N_k}\}$ to the local coordinate system of each primitive $V_k{=}\{P_k, R_k, S_k\}$ and deform them along with the primitive. Specifically, each Gaussian $G_k^i{=}\{p_k^i, r_k^i, s_k^i, c_k^i, \sigma_k^i\}$ is defined by its position p_k^i, rotation r_k^i, and scaling s_k^i in the primitive's local coordinates, as well as its color features c_k^i and opacity σ_k^i. Given a target pose θ, we first obtain the location P_k, scale S_k, and orientation R_k of each deformed primitive using Eq. 1. Then the global location $\hat{p}_k^i$, scale $\hat{s}_k^i$, and orientation $\hat{r}_k^i$ of each Gaussian G_k^i associated with the primitive are computed as:

$$\hat{p}_k^i(\theta) = R_k(\theta) \cdot (S_k(\theta) \odot p_k^i) + P_k(\theta) \tag{3}$$

$$\hat{s}_k^i(\theta) = S_k(\theta) \cdot s_k^i \tag{4}$$

$$\hat{r}_k^i(\theta) = R_k(\theta) \cdot r_k^i \tag{5}$$

This primitive-based Gaussian representation naturally balances constraint and flexibility. It is more flexible compared to the native primitive representation in [25, 31] since it allows a primitive to deform beyond a cube by equipping it with Gaussians. Meanwhile, the Gaussians within each primitive share the motion of the primitive and are more constrained during animation.

Implicit Gaussian Attribute Field. To fully exploit the expressiveness of 3D Gaussians, we allow each Gaussian to have individual attributes, *i.e.*, color features, scaling, rotation, and opacity. However, this potentially results in unstable training where Gaussians within a local neighborhood possess different attributes, leading to noisy geometry and rendering. This is especially true when the gradient of the optimization objective has high variance, such as the SDS objective in Eq. 2. To stabilize and amortize the training process, instead of directly optimizing the attributes of the Gaussians, we propose to predict these attributes using neural implicit fields. As shown in the yellow block in Fig. 2, for each Gaussian G_k^i, we first compute its canonical position $\hat{p}_k^i(\tilde{\theta})$ in the world coordinate system (Eq. 3), where $\tilde{\theta}$ represents the rest pose. We can then query the color c_k^i, rotation r_k^i, scaling s_k^i and opacity σ_k^i of each Gaussian using the canonical position $\hat{p}_k^i(\tilde{\theta})$ from two neural implicit fields $\mathcal{H}_\phi$ and $\mathcal{O}_\psi$, which are represented by neural networks with parameters ϕ and ψ:

$$(c_k^i, r_k^i, s_k^i) = \mathcal{H}_\phi(\hat{p}_k^i(\tilde{\theta})) \tag{6}$$

$$\sigma_k^i = \mathcal{O}_\psi(\hat{p}_k^i(\tilde{\theta})) \tag{7}$$

where we use a separate neural field $\mathcal{O}_\psi$ to output the opacities of the Gaussians, while other attributes are predicted by $\mathcal{H}_\phi$. This design is because the opacities of the Gaussians are closely related to the underlying geometry of the avatar and require special treatment, which will be discussed in Sec. 4.2. Note that by querying the neural field with a canonical rest pose $\tilde{\theta}$, we canonicalize the Gaussian attributes, which can then be shared across different poses and animations. Our use of neural implicit fields constrains nearby Gaussians to have consistent attributes, which greatly stabilizes and amortizes the training process and enables high-quality avatar synthesis using high-variance losses.

Rendering and Objectives. After obtaining the positions and attributes of 3D Gaussians, we adopt the efficient Gaussian splatting technique described in [17] to render an RGB image I and also an alpha image I_α. The RGB image I is then used for the SDS loss defined in Eq. 2 as one of the main training objectives. To prevent the Gaussians from straying far away from the primitives, we also utilize a local position regularization loss $\mathcal{L}_{\text{pos}}{=}\sum_{k,i} \|p_k^i\|^2$, which constrains the Gaussians to be close to the origin of the associated primitives.

4.2. SDF-based Mesh Learning for 3D Gaussians

A crucial aspect yet to be addressed in our primitive-based 3D Gaussian representation is how to properly represent the underlying geometry of the 3D Gaussians. This is important for two reasons: (1) 3D Gaussians are transparent "point clouds" that do not have well-defined surfaces, which can lead to degenerate body parts or holes in the generated avatars (see Fig. 5); (2) Currently, there is no efficient and effective way to extract textured meshes from a large number of 3D Gaussians, which are often important for applications in traditional graphics pipelines.

SDF-based Gaussian Opacity Field. To address this problem, we propose to represent the underlying geometry of 3D Gaussians through a signed distance field (SDF) function $\mathcal{S}_\psi$ with parameters ψ. Specifically, we parametrize the opacity σ_k^i of each 3D Gaussian based on their signed distance to the surface using a kernel function $\mathcal{K}$ inspired by NeuS [42]:

$$\sigma_k^i = \mathcal{K}(\mathcal{S}_\psi(p_k^i))\,, \tag{8}$$

where $\mathcal{K}(x) = \gamma e^{-\lambda x}/(1+e^{-\lambda x})^2$ is a bell-shaped kernel function with learnable parameters $\{\gamma, \lambda\}$ that maps the signed distance to an opacity value. Intuitively, this opacity parametrization builds in the prior that Gaussians should stay close to the surface in order to obtain high opacity. The parameter λ controls the tightness of the high-opacity neighborhood of the surface and α controls the overall scale of the opacity. The SDF-based Gaussian opacity parametrization naturally fits our primitive-based implicit Gaussian representation, since now we can define the aforementioned opacity field $\mathcal{O}_\psi$ as the product of the SDF

and the kernel function: $\mathcal{O}_\psi = \mathcal{K} \circ \mathcal{S}_\psi$, and we can directly use a neural network to represent the SDF $\mathcal{S}_\psi$.

Mesh Extraction and Geometry Regularization. An important advantage of using an SDF $\mathcal{S}_\psi$ to represent the underlying geometry of 3D Gaussians is that it allows us to extract a mesh $\widetilde{\mathcal{M}}$ from the SDF through differentiable marching tetrahedra (DMTet [37]):

$$\widetilde{\mathcal{M}} = \text{DMTET}(\mathcal{S}_\psi)\,. \tag{9}$$

Both the SDF and extracted mesh allow us to utilize various losses to regularize the geometry of the 3D Gaussian avatar. Specifically, we first employ an Eikonal regularizer to maintain a proper SDF, which is defined as:

$$\mathcal{L}_{\text{eik}} = (\|\nabla_p \mathcal{S}_\psi(p)\| - 1)^2\,, \tag{10}$$

where $p \in \mathcal{P}$ contains both the center points of all Gaussians in the world coordinates as well as points sampled around the Gaussians using a normal distribution. Next, we also employ an alpha loss to match the mask I_M rendered using the extracted mesh to the alpha image I_α from the Gaussian splatting:

$$\mathcal{L}_{\text{alpha}} = \|I_M - I_\alpha\|^2\,. \tag{11}$$

Inspired by Fantasia3D [4], we also use a normal SDS loss to supervise the normal rendering I_N of the extracted mesh using differentiable rasterization [19]. The SDS gradient can be computed as:

$$\nabla_\theta \mathcal{L}^N_{\text{SDS}} = E_{t,\epsilon}\left[w(t)(\hat{\epsilon}(I_{N,t}; y, t) - \epsilon)\frac{\partial I_N}{\partial \theta}\right]\,, \tag{12}$$

where $I_{N,t}$ is the noised normal image. We further use a normal consistency loss $\mathcal{L}_{\text{nc}}$ which regularizes the difference between the adjacent vertex normals of mesh $\widetilde{\mathcal{M}}$.

Texture Extraction. Our proposed implicit Gaussian attribute field $\mathcal{H}_\phi$ naturally facilitates texturing the extracted mesh $\widetilde{\mathcal{M}}$, since we can use the Gaussian color field as the 3D texture field used by the differentiable rasterization. Once the Gaussian-based avatar is fully optimized, directly using the Gaussian color field already provides a good initial texture for the mesh, but we can further improve the texture quality by finetuning the color field using an SDS loss $\mathcal{L}^{\widetilde{\mathcal{M}}}_{\text{sds}}$ on the RGB rendering $I_{\widetilde{\mathcal{M}}}$ of the textured mesh. We observe that only a small number of finetuning iterations is required for convergence.

4.3. Optimization

The overall objective of our method can be summarized as:

$$\mathcal{L} = \mathcal{L}_{\text{SDS}} + \mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{eik}} + \mathcal{L}_{\text{alpha}} + \mathcal{L}^N_{\text{SDS}} + \mathcal{L}_{\text{nc}}\,, \tag{13}$$

where we omit the weighting terms for brevity. Using this objective, we optimize the Gaussian local positions $\{p^i_k\}$, Gaussian attribute field $\mathcal{H}_\phi$ and SDF $\mathcal{S}_\psi$, opacity kernel parameters $\{\gamma, \lambda\}$, primitive motion corrective networks $\delta P_\omega, \delta R_\omega, \delta S_\omega$, as well as the SMPL-X shape parameters β.

Initialization. We divide the uv-map of SMPL-X into a 64×64 grid, which gives us 4096 primitives. We assign 64 Gaussians to each primitive V_k and initialize their local positions $\{p^i_k\}$ with a uniform grid of $4 \times 4 \times 4$.

Training. We perform Gaussian densification as described in [17] every 100 iterations, which leads to different numbers of Gaussians per primitive. We stop densification when the total number of Gaussians exceeds 2 million. To render the RGB image I for the SDS loss $\mathcal{L}_{\text{sds}}$, we take the target pose θ from two sources: (1) a natural pose θ_N optimized together with the aforementioned variables; (2) a random pose θ_A sampled from an animation database to ensure realistic animation.

5. Experiments

In Fig. 3, we showcase example avatars generated by our method and their geometry and textured meshes. Notice the intricate geometry details captured by our method, thanks to our SDF-based implicit mesh learning for 3D Gaussians. Due to its primitive-based design, our approach readily supports avatar animation. We showcase various animations in Fig. 1 and on the project website.

Rendering Efficiency. Since GAvatar no longer needs to query the Gaussian attributes from the implicit fields after optimization, it achieves extremely fast rendering speed due to the use of 3D Gaussians. Specifically, a generated avatar with 2.5 million Gaussians can be rendered with 1024×1024 resolution at 100 fps, which is tremendously faster than most NeRF-based approaches. Moreover, the Gaussian rendering only takes about 3ms (300+ fps), so further speedup is possible by optimizing the speed of non-rendering operations such as LBS and primitive transforms.

5.1. Qualitative Evaluation

Fig. 4 compares our method, GAvatar, with the state-of-the-art approaches: DreamGaussian [40], AvatarCLIP [9], AvatarCraft [15] and Fantasia3D [4]. For completeness, we also compare with contemporary works, DreamHumans [18] and TADA [22]. For DreamHumans [18] we use the avatar renderings provided on the project page, while for other methods we use the publicly available source codes. Our method clearly produces higher-quality avatars both in terms of geometry and appearance. DreamGaussian [40], AvatarCLIP [9], AvatarCraft [15] and Fantasia3D [4] fail catastrophically to model complex avatars. DreamHumans [18] creates low-resolution avatars since it is trained with a resolution of 64×64 only. TADA [22] can

Figure 3. **Generated avatars by our method and their mesh normals and texture meshes.**

Figure 4. **Comparison with the state-of-the-art methods.** From top to bottom, the prompts used in each row are "a person dressed at the venice carnival", "a professional boxer" and "a bedouin dressed in white". Our method consistently produces the best quality avatars.

render high-resolution images due to a mesh-based rendering but can produce degenerate solutions with implausible shapes. It also provides smoother texture and less geometry details as compared to our method. GAvatar generates significantly better avatars as compared to all methods as we will also show in our user study next.

5.2. Quantitative Evaluation

To quantitatively evaluate the proposed method, we follow previous works [9, 22, 40] and carry out an extensive A/B user study. We adopt 24 prompts commonly used in the baselines to generate the avatars. In total, we collected 1512 responses from 42 participants. For each vote, we show a pair of randomly chosen 3D avatars synthesized by our method and one of the baseline methods. We ask the participant to choose the method that has better 1) geometry quality, 2) appearance quality, and 3) consistency with the given prompt. Table 1 summarizes the preference percentage of our method over the baseline methods. Notably, our method

Compared Method	Geometry Quality	Appearance Quality	Consistency with Prompt
AvatarCLIP [9]	98.81	97.62	97.62
AvatarCraft [15]	96.43	98.81	98.81
DreamGaussian [40]	100.0	98.81	98.81
Fantasia3D [4]	92.86	92.86	91.67
DreamHuman [18]*	73.81	73.81	65.48
TADA [22]*	61.90	69.05	67.86

Table 1. **User Study.** We show a *preference percentage* of our method over state-of-the-art methods (* denotes contemporary methods). GAvatar is preferred by the users over all baselines.

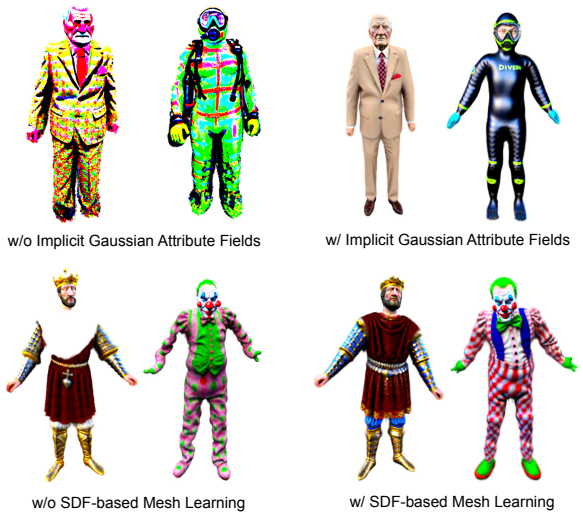

Figure 5. **Ablation Studies.**

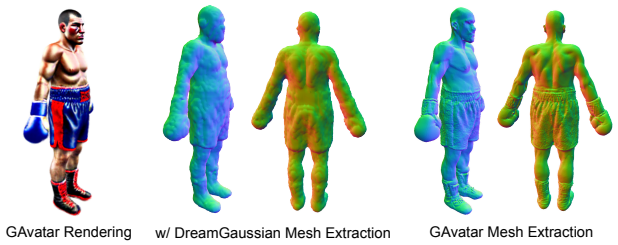

Figure 6. **Mesh Extraction Comparison.**

consistently outperforms existing and contemporary methods by a substantial margin.

5.3. Ablation Study

Effect of Implicit Gaussian Attribute Field. In Fig. 5 (Top), we design a variant of our method by disabling the implicit Gaussian attribute field and directly optimizing the Gaussian attributes. We observe that the generated avatars are significantly worse than our method, with pronounced noise and color oversaturation. This aligns with our intuition that directly optimizing millions of Gaussians individually with high-variance loss like SDS is quite challenging. In contrast, our implicit Gaussian attribute field allows a much more stable and robust optimization process.

Effect of SDF-based Mesh Learning. In Fig. 5 (Bottom), we design a variant of our approach by disabling the SDF-based mesh learning and instead letting the Gaussian attribute field additionally output the Gaussian opacities. As shown in Fig. 5, the generated avatars without mesh learning can have missing body parts and distorted body shapes. Our SDF-based mesh learning tackles these issues by regularizing the underlying geometry of the Gaussian avatar.

Mesh Extraction Comparison. An important benefit of our approach is that it allows us to extract a high-quality differentiable mesh representation of the Gaussian avatar. We compare our mesh extraction approach with the Gaussian density-based approach used in DreamGaussian [40], one of the few works that extract meshes from 3D Gaussians. In particular, we provide its mesh extraction pipeline with our optimized Gaussian attributes to obtain the final mesh. The results are shown in Fig. 6. We observe the mesh extracted by DreamGaussian is more noisy and lacks geometry details, while our approach obtains much smoother meshes with fine-grained geometry details.

6. Discussion and Limitations

We have presented a novel approach for generating diverse and animatable avatars with geometry learning and regularization. Our primitive-based 3D Gaussian representation allows us to flexibly model avatar geometry and appearance while enabling animation with extremely fast rendering. We demonstrated our neural implicit Gaussian attribute fields stabilize the learning of millions of 3D Gaussian under noisy objectives. We further propose a novel SDF-based mesh learning approach that regularizes the underlying geometry of the Gaussian avatar and extracts a high-quality textured mesh from 3D Gaussians. Our experiments and user study indicate that our approach surpasses state-of-the-art methods in terms of appearance and geometry quality.

While our approach has shown promising results, it still has several limitations to be addressed in future work. First, similar to other SDS-based approaches, our method sometimes also suffers from color oversaturation. We believe that exploring various techniques for improving SDS [16, 24, 43] can help mitigate this issue. Second, there can still be misalignment between the geometry and appearance of the generated avatars, where some geometry details in the rendering are embedded in the colors of the 3D Gaussians, similar to how texture can embed geometry details in mesh-based rendering. We believe that having consistent geometry and appearance supervisions such as those in HumanNorm [10] can help alleviate this issue. Disentangling lighting and appearance details within the 3D Gaussian-based representation is also an interesting future direction. Lastly, animating loose clothing with correct temporal deformations is still challenging, especially when no direct image or temporal supervision is provided. Leveraging temporal priors such as physics simulation or video diffusion models can be a promising future avenue to explore.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *ECCV*, 2020. 3

[2] Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. imghum: Implicit generative models of 3d human shape and articulated pose. In *International Conference on Computer Vision (ICCV)*, pages 5461–5470, 2021. 3

[3] Yukang Cao, Yan-Pei Cao, Kai Han, Ying Shan, and Kwan-Yee K Wong. DreamAvatar: Text-and-Shape Guided 3D Human Avatar Generation via Diffusion Models. *arXiv preprint:2304.00916*, 2023. 2, 3

[4] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation. In *International Conference on Computer Vision (ICCV)*, 2023. 2, 3, 6, 8

[5] Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. TANGO: Text-driven Photorealistic and Robust 3D Stylization via Lighting Decomposition. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 3

[6] dalle2. https://openai.com/dall-e-2, 2022. 2, 3

[7] Jun Gao, Wenzheng Chen, Tommy Xiang, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Learning deformable tetrahedral meshes for 3d reconstruction. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 9936–9947, 2020. 3

[8] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020. 1

[9] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. AvatarCLIP: Zero-Shot Text-Driven Generation and Animation of 3D Avatars. *Transactions on Graphics (TOG)*, 41(4):1–19, 2022. 2, 3, 6, 7, 8

[10] Xin Huang, Ruizhi Shao, Qi Zhang, Hongwen Zhang, and Ying Feng. Humannorm: Learning normal diffusion model for high-quality and realistic 3d human generation, 2023. 2, 3, 8

[11] Yukun Huang, Jianan Wang, Ailing Zeng, He Cao, Xianbiao Qi, Yukai Shi, Zheng-Jun Zha, and Lei Zhang. Dreamwaltz: Make a scene with complex 3d animatable avatars. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 2, 3

[12] Yangyi Huang, Hongwei Yi, Yuliang Xiu, Tingting Liao, Jiaxiang Tang, Deng Cai, and Justus Thies. TeCH: Text-guided Reconstruction of Lifelike Clothed Humans. In *International Conference on 3D Vision (3DV)*, 2024. 3

[13] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[14] Nikolay Jetchev. Clipmatrix: Text-controlled creation of 3d textured meshes. *arXiv preprint arXiv:2307.05663*, 2023. 2, 3

[15] Ruixiang Jiang, Can Wang, Jingbo Zhang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Avatarcraft: Transforming text into neural human avatars with parameterized shape and pose control. In *International Conference on Computer Vision (ICCV)*, 2023. 3, 6, 8

[16] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. *arXiv preprint arXiv:2310.17590*, 2023. 8

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 2, 3, 4, 5, 6

[18] Nikos Kolotouros, Thiemo Alldieck, Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Fieraru, and Cristian Sminchisescu. Dreamhuman: Animatable 3d avatars from text. *arXiv preprint:2306.09329*, 2023. 2, 3, 6, 8

[19] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *Transactions on Graphics (TOG)*, 39(6), 2020. 6

[20] Ruilong Li, Kyle Olszewski, Yuliang Xiu, Shunsuke Saito, Zeng Huang, and Hao Li. Volumetric human teleportation. In *ACM SIGGRAPH 2020 Real-Time Live*, 2020. 1

[21] Ruilong Li, Yuliang Xiu, Shunsuke Saito, Zeng Huang, Kyle Olszewski, and Hao Li. Monocular real-time volumetric performance capture. In *European Conference on Computer Vision (ECCV)*, pages 49–67. Springer, 2020. 1

[22] Tingting Liao, Hongwei Yi, Yuliang Xiu, Jiaxiang Tang, Yangyi Huang, Justus Thies, and Michael J Black. Tada! text to animatable digital avatars. In *International Conference on 3D Vision (3DV)*, 2024. 3, 6, 7, 8

[23] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-Resolution Text-to-3D Content Creation. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[24] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. *arXiv preprint arXiv:2305.08891*, 2023. 8

[25] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Trans. Graph.*, 2021. 3, 5

[26] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 3

[27] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[28] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 4

[29] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 3, 4

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 3

[31] Edoardo Remelli, Timur Bagautdinov, Shunsuke Saito, Chenglei Wu, Tomas Simon, Shih-En Wei, Kaiwen Guo, Zhe Cao, Fabian Prada, Jason Saragih, et al. Drivable volumetric avatars using texel-aligned features. In *ACM SIGGRAPH 2022 Conference Proceedings*, 2022. 3, 4, 5

[32] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint:2302.01721*, 2023. 3

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 2, 3

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 36479–36494, 2022. 2, 3

[35] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, 2020. 1

[36] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 18603–18613, 2022. 3

[37] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 6087–6101, 2021. 4, 6

[38] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 3

[39] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[40] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 3, 6, 7, 8

[41] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3835–3844, 2022. 3

[42] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021. 3, 5

[43] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 3, 8

[44] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J. Black. ECON: Explicit Clothed humans Optimized via Normal integration. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[45] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[46] Yifei Zeng, Yuanxun Lu, Xinya Ji, Yao Yao, Hao Zhu, and Xun Cao. Avatarbooth: High-quality and customizable 3d human avatar generation, 2023. 3

[47] Huichao Zhang, Bowen Chen, Hao Yang, Liao Qu, Xu Wang, Li Chen, Chao Long, Feida Zhu, Kang Du, and Min Zheng. Avatarverse: High-quality & stable 3d avatar creation from text and pose, 2023. 3

[48] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2, 3

[49] Xiaoshuai Zhang, Abhijit Kundu, Thomas Funkhouser, Leonidas Guibas, Hao Su, and Kyle Genova. Nerflets: Local radiance fields for efficient structure-aware 3d scene representation from 2d supervision. *CVPR*, 2023. 3

[50] Xuanmeng Zhang, Jianfeng Zhang, Chacko Rohan, Hongyi Xu, Guoxian Song, Yi Yang, and Jiashi Feng. Getavatar: Generative textured meshes for animatable human avatars. In *ICCV*, 2023. 3

[51] Yang Zheng, Ruizhi Shao, Yuxiang Zhang, Tao Yu, Zerong Zheng, Qionghai Dai, and Yebin Liu. Deepmulticap: Performance capture of multiple characters using sparse multiview cameras. In *International Conference on Computer Vision (ICCV)*, 2021. 1

[52] Zerong Zheng, Xiaochen Zhao, Hongwen Zhang, Boning Liu, and Yebin Liu. Avatarrex: Real-time expressive full-body avatars. *ACM Transactions on Graphics (TOG)*, 42(4), 2023.

[53] Luyang Zhu, Konstantinos Rematas, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman. Reconstructing nba players. In *European Conference on Computer Vision (ECCV)*, 2020. 1