# Gaussian Shell Maps for Efficient 3D Human Generation

Rameen Abdal[*1] Wang Yifan[*1] Zifan Shi[*†1,2] Yinghao Xu[1] Ryan Po[1] Zhengfei Kuang[1]

Qifeng Chen[2] Dit-Yan Yeung[2] Gordon Wetzstein[1]

[1]Stanford University [2]HKUST


Figure 1. **Gaussian Shell Maps.** Gaussian Shell Maps is an efficient framework for 3D human generation connecting 3D Gaussians with CNN-based generators. 3D Gaussians are anchored to "shells" derived from the SMPL template [36] (only two shells are visualized for clarity), and the appearance is modeled in texture space. Trained only on 2D images, we show that our method can generate diverse articulable humans in real-time with state-of-the-art quality directly in high resolution without the need for upsampling and hence avoiding aliasing artifacts.

## Abstract

*Efficient generation of 3D digital humans is important in several industries, including virtual reality, social media, and cinematic production. 3D generative adversarial networks (GANs) have demonstrated state-of-the-art (SOTA) quality and diversity for generated assets. Current 3D GAN architectures, however, typically rely on volume representations, which are slow to render, thereby hampering the GAN training and requiring multi-view-inconsistent 2D upsamplers. Here, we introduce Gaussian Shell Maps (GSMs) as a framework that connects SOTA generator network architectures with emerging 3D Gaussian rendering primitives using an articulable multi shell–based scaffold. In this setting, a CNN generates a 3D texture stack with features that are mapped to the shells. The latter represent inflated and deflated versions of a template surface of a digital human in a canonical body pose. Instead of rasterizing the shells directly, we sample 3D Gaussians on the shells whose attributes are encoded in the texture features. These Gaussians are efficiently and differentiably rendered. The ability to articulate the shells is important during GAN training and, at inference time, to deform a body into arbitrary user-defined poses. Our efficient rendering scheme bypasses the need for view-inconsistent upsamplers and achieves high-quality multi-view consistent renderings at a native resolution of* $512 \times 512$ *pixels. We demonstrate that GSMs successfully generate 3D humans when trained on single-view datasets, including SHHQ and DeepFashion.*
*Project Page: rameenabdal.github.io/GaussianShellMaps*

## 1. Introduction

The ability to generate articulable three-dimensional digital humans augments traditional asset creation and animation workflows, which are laborious and costly. Such generative artificial intelligence–fueled workflows are crucial in several applications, including communication, cinematic production, and interactive gaming, among others.

3D Generative Adversarial Networks (GANs) have emerged as the state-of-the-art (SOTA) platform in this domain, enabling the generation of diverse 3D assets at interactive framerates [6, 8, 15, 23, 41, 71, 76, 76]. Most existing 3D GANs build on variants of volumetric scene representations combined with neural volume rendering [65]. However, volume rendering is relatively slow and compromises the training of a GAN, which requires tens of millions of forward rendering passes to converge [11]. Mesh–based representations building on fast differentiable rasterization have been proposed to overcome this limitation [19, 71], but these are not expressive enough to realistically model features like hair, clothing, or accessories, which deviate significantly from the template mesh. These limitations,

[*] Equal Contribution
[†] Work done as a visiting student researcher at Stanford University

which are largely imposed by a tradeoff between efficient or expressive scene representations, have been constraining the quality and resolution of existing 3D GANs. While partially compensated for by using 2D convolutional neural network (CNN)–based upsamplers [11, 22], upsampling leads to multi-view inconsistency in the form of aliasing.

Very recently, 3D Gaussians have been introduced as a promising neural scene representation offering fast rendering speed and high expressivity [28]. While 3D Gaussians have been explored in the context of single-scene overfitting, their full potential in generative settings has yet to be unlocked. This is challenging because it is not obvious how to combine SOTA CNN-based generators [26, 27] with 3D Gaussian primitives that inherently do not exist on a regular Cartesian grid and that may vary in numbers.

We introduce Gaussian Shell Maps (GSMs), a 3D GAN framework that intuitively connects CNN generators with 3D Gaussians used as efficient rendering primitives. Inspired by the traditional computer graphics work on shell maps [47], GSMs use the CNN generator to produce texture maps for a set of "shell" meshes appropriately inflated and deflated from the popular SMPL mesh template for human bodies [36]. The textures on the individual shells directly encode the properties of 3D Gaussians, which are sampled on the shell surfaces at fixed locations. The generated images are rendered using highly efficient Gaussian splatting, and articulation of these Gaussians can be naturally enabled through deforming the scaffolding shells with the SMPL model. Since 3D Gaussians have spatial extent, they represent details on, in between, and outside the discrete shells. GSMs are trained exclusively on datasets containing single-view images of human bodies, such as SHHQ [17].

Our experiments demonstrate that GSM can generate highly diverse appearances, including loose clothing and accessories, at high resolution, without an upsampler, at a state-of-the-art rendering speed of 125 FPS (or 35FPS including generation). Among various architecture design choices, multiple shells with fixed relative locations of the 3D Gaussian achieve the best results in our experiments.

Specifically, our contributions include

1. We propose a novel 3D GAN framework combining a CNN-based generator and 3D Gaussian rendering primitives using shell maps.
2. We demonstrate the fastest 3D GAN architecture to date, achieving real-time rendering of $512^2$ px without convolutional upsamplers, with image quality and diversity matching the state of the art on challenging single-view datasets of human bodies.

## 2. Related Work

**3D-Aware Generative Models.** GANs and diffusion models are two very powerful generative models emerging as a result of efficient architectures and high-quality datasets [25, 48]. With the quality of image generation reaching photo-realism, the community started leveraging these SOTA generative models for 3D generation by combining neural rendering methods [64, 65] to produce high-quality multi-view consistent 3D objects from image collections. Several 3D GAN architectures have explored implicit or explicit neural volume representations for modeling 3D objects, including [2, 10, 11, 14, 22, 39, 40, 42, 57, 59, 62, 69, 70]. 3D diffusion models, on the other hand, typically use the priors encoded by pre-trained text-to-image 2D diffusion models, e.g. [4, 7–9, 12, 29, 33, 34, 44, 46, 51, 53, 66] (see this survey for more details [45]). Due to the lack of high-quality, large-scale multi-view datasets curated for specific categories like humans, the choice of a suitable generative model becomes critical. Using diffusion models to generate high-resolution multi-view consistent 3D objects is still an unsolved problem [45]. 3D GANs, on the other hand, exhibit better quality and multi-view consistency at higher resolutions that do not assume multi-view data [11, 15, 62, 71]. This motivates our choice of building an efficient representation using a 3D GAN framework.

**Generative Articulated 3D Digital Humans.** 3D-GAN frameworks have been proposed to generate the appearance, geometry, and identity-preserving novel views of digital humans [6, 13, 15, 23, 71, 72, 74, 76]. Most of these GANs are trained on single-view image collections [16, 32, 35]. A popular approach is to use a neural radiance representation [6, 23, 24, 41, 74] where a canonically posed human can be articulated via deformation. Other approaches are based on meshes, where a template can be fixed or learned during the training [3, 18, 19, 63, 72]. Related to this line of work, a concurrent paper, LSV-GAN [71], offsets the SMPL template meshes into layered surfaces and composites the per-layer rasterization result to form the final rendering. While it provides a faster alternative to the volume rendering-based approaches, it can only accommodate a small offset from the SMPL mesh, which hampers diversity. Our GSM method differs from LSV-GAN as we employ 3D Gaussians [28] as primitives on the layered surfaces, which, by having a learnable spatial span, allows for larger deviation from the template mesh and can thus generate more intricate details.

**Point-Based Rendering.** Earlier point-based methods efficiently render point clouds and rasterize them by fixing the size [20, 21, 54]. While efficient in terms of speed and parallel rasterization on graphics processing units [31, 56], they are not differentiable [28, 49]. To combine these methods with the neural networks and perform view synthesis, recent works have developed differentiable point-based rendering techniques [1, 52, 67, 68, 73, 75, 77]. More recently, 3D Gaussian-based point splatting gained traction

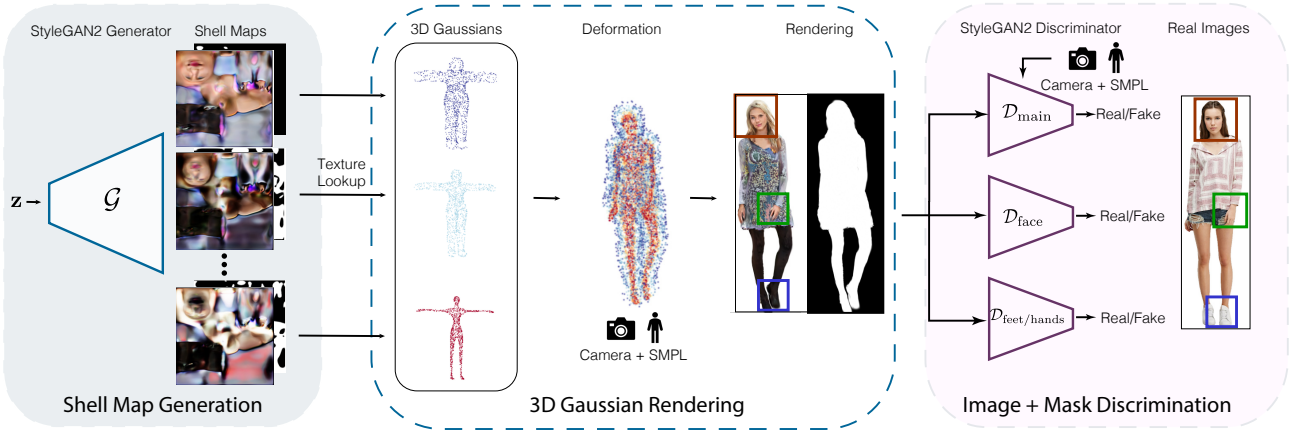


Figure 2. **Method Overview.** We propose an expressive yet highly efficient representation, Gaussian Shell Map (GSM), for 3D human generation. Combining the idea of 3D Gaussians and Shell Maps (Sec. 3), we sample 3D Gaussians on "shells", which are mesh layers offsetted from the SMPL template, forming a shell volume to model complex and diverse geometry and appearance; the Gaussian parameters are learned in the texture space, allowing us to leverage existing CNN-based generative architecture (Sec. 4.1). Articulation is straightforward by interpolating the deformation of the shell (Sec. 4.2). The generation is supervised by single-view 2D images using several discriminator critics, including part-specific face, hands, and feet discriminators (Sec. 4.3).

due to the flexibility of anisotropic covariance and density control with efficient depth sorting [28]. This allows 3D Gaussian splats to handle complex scenes composed of high and low-frequency features. Relevant to human bodies, 3D Gaussians have also been used in pose estimation and tracking [38, 50, 61]. While the Gaussian primitives have been used for efficient scene reconstruction and novel view synthesis, it is not trivial to deploy Gaussians in a generative setup. To the best of our knowledge, our method is the first to propose a combination of 3D Gaussians and 3D GANs.

## 3. Background

**3D Gaussians.** These point-based primitives can be differentiably and efficiently rendered using EWA (elliptical weighted average) volume splatting [78]. 3D Gaussians have recently demonstrated outstanding expressivity for 3D scene reconstruction [28], in which the Gaussian parameters, position $\mathbf{x} \in \mathbb{R}^3$, opacity $o \in \mathbb{R}$, color $\mathbf{c} \in \mathbb{R}^{sh}$ ($sh$ representing the spherical harmonic coefficients), scaling $\mathbf{s} \in \mathbb{R}^3$ , and rotation $\mathbf{q} \in \mathbb{R}^4$ parameterized as quaternions are jointly optimized to minimize the photometric errors of the rendered images in a set of known camera views. The optimization is accompanied by adaptive control of the density, where the points are added or removed based on the density, size, and gradient of the Gaussians.

Specifically, each Gaussian is defined as

$$G(\mathbf{x}'; \mathbf{x}, \Sigma) = \exp^{-\frac{1}{2}(\mathbf{x}'-\mathbf{x})^{\intercal}\Sigma^{-1}(\mathbf{x}'-\mathbf{x})}, \tag{1}$$

where $\Sigma = RSS^TR^T$ is the covariance matrix parameterized by the rotation and scaling matrices $R$ and $S$ given by the quaternions $\mathbf{q}$ and scaling $\mathbf{s}$.

The image formation is governed by classic point-based $\alpha$-blending of overlapping Gaussians ordered from closest to farthest [30]:

$$\mathbf{C} = \sum_{i \in \mathcal{N}} \mathbf{c}'_i o'_i \prod_{j=1}^{i-1} (1 - \alpha_\mathbf{j}), \tag{2}$$

where color $\mathbf{c}'_i$ and opacity $o'_i$ is computed by

$$\mathbf{c}'_i = G(\mathbf{x}'; \mathbf{x}_i)\mathbf{c}_i \quad \text{and} \quad o'_i = G(\mathbf{x}'; \mathbf{x}_i)o_i. \tag{3}$$

Thanks to their ability to fit complex geometry and appearance, 3D Gaussians are gaining popularity for 3D scene reconstruction. However, deploying 3D Gaussians for generative tasks remains an unexplored topic and is challenging for several reasons. First, Gaussians are of "Lagrangian" nature – their sparse number and learnable positions are challenging to combine with SOTA "Eulerian" (i.e., grid-based) CNN generators. Second, the parameters of Gaussians are highly correlated. The same radiance field can be equally well explained by many different configurations of Gaussians, varying their locations, sizes, scales, rotations, colors, and opacities. This ambiguity makes it challenging to generalize over a distribution of objects or scenes, which generative methods do.

**Shell Maps.** Our representation is related to Shell Maps [14, 47, 58], a technique in computer graphics designed to model near-surface details. Shell maps use 3D texture maps to store the fine-scale features in a shell-like volume close to a given base surface, typically represented as a triangle mesh. In essence, they extend UV


Figure 3. **Novel Views and Animation.** We can render a generated identity in novel views and articulate it for animations.

Figure 4. **Latent Code Interpolation.** Latent code interpolation of our model trained on DeepFashion Dataset.

maps such that every point in the shell volume can be bijectively mapped to the 3D texture map for efficient modeling and rendering in texture space. The shell volume is constructed by offsetting the base mesh along the normal direction while maintaining the topology and avoiding self-intersection. The volume is discretized into tetrahedra that connect the vertices of the base and offset meshes. A unique mapping between the shell space and the texture space can be established by identifying the tetrahedron and subsequently querying the barycentric coordinates inside it. Formally, the mapping from $\mathbf{x}$ in the shell volume to its position in the 3D texture map is defined as

$$\mathbf{x}_t = \phi\left(\mathbf{T}_t, B\left(\mathbf{x}, \mathbf{T}\right)\right), \tag{4}$$

where $\mathbf{T}, \mathbf{T}_t$ refer to corresponding tetrahedra in the shell and texture space, $B\left(\mathbf{x}, \mathbf{T}\right)$ is the barycentric coordinates of $\mathbf{x}$ in $\mathbf{T}$, and $\phi$ denotes barycentric interpolation.

# 4. Gaussian Shell Maps (GSM)

GSM is a framework that connects 3D Gaussians with SOTA CNN-based generator networks [11] in a GAN setting. The key idea is to anchor the 3D Gaussians at fixed locations on a set of "shells" derived from the SMPL [36] human body template mesh. These shells span a volume to model surface details that deviate from the unclothed template mesh. We learn the relevant Gaussian parameters in the texture space, allowing us to leverage established CNN-based generative backbones while seamlessly utilizing the parametric deformation model for articulation. The overall pipeline is shown in Figure 2.

## 4.1. Representation

Our method utilizes the concept of shell maps to leverage the inherent planar structure of texture space for seamless integration with CNN-based generative architectures and, at the same time, encapsulate diverse and complex surface details without directly modifying the template mesh. Specifically, the shell volume is defined by the boundary mesh layers, created by inflating and deflating the T-pose SMPL mesh using Laplacian Smoothing [60] with the smoothing factor set to negative for inflation and positive for deflation. We then represent this shell volume using $N$ mesh layers, i.e. "shells", by linearly spacing the vertices between the aforementioned boundary shells. In parallel, we apply a similar discretization strategy to the 3D texture space, creating $N$ 2D texture maps storing neural features that can be referenced for each shell using the UV mapping inherited from the SMPL template.

As shown in Figure 2, we use the shell maps to generate all Gaussian parameters except the positions, as those are sampled and anchored w.r.t. to the shells at every iteration (explained below in detail). This results in a feature volume of $\mathsf{T}^{N\times H\times W\times 11}$, comprised of 3D color $\mathsf{T}^{\mathbf{c}}$, 1D opacity $\mathsf{T}^{o}$, 3D scaling $\mathsf{T}^{\mathbf{s}}$, and 4D rotation $\mathsf{T}^{\mathbf{q}}$ features.

We create Gaussians in our shell volume. This is done by sampling a fixed number of Gaussians quasi-uniformly on the shells based on the triangle areas. Once sampled, the Gaussians are anchored on the shells using barycentric coordinates so that every Gaussian center $\mathbf{x}$ can be mapped to a point $\mathbf{x}_t$ on the corresponding shell map using Eq. (4), except that the tetrahedra are replaced with the triangle in which the Gaussian resides. This anchoring is a crucial design choice, as it enables straightforward feature retrieval from the shell maps. More importantly, it simplifies the learning by fixing the Gaussian positions and offloading geometry modeling to opacity $o$ and sigma $\Sigma$.

The spatial span of the Gaussians plays a significant role in reconstructing the features defined on the discrete shells into a continuous signal within the 3D space. It enables every point—whether on the shells, between them, or outside them—to receive a valid opacity and color value by evaluat-


Figure 5. **Qualitative Comparison.** We compare our results with GNARF and EVA3D baselines on DeepFashion and SHHQ datasets. In each case, we show the deformed body poses of the identities generated by the methods. The competing methods exhibit artifacts marked in red. Notice that our approach generates high-quality textures, like facial details and more realistic deformations.

ing Eq. (3). This process allows us to model diverse appearances and body shapes, excelling mesh-based representation while at the same time maintaining efficient rendering, which is critical for GAN training.

Formally, Gaussian opacity, color, scale, and rotation can be interpolated from the shell maps $\mathsf{T}$

$$f = \mathsf{T}^f\left(\mathbf{x}_t\right), \text{ where } f = \{o, \mathbf{c}, \mathbf{s}, \mathbf{q}\}. \tag{5}$$

### 4.2. Deformation

The deformation step updates the Gaussians' locations and orientations based on the SMPL template mesh $\mathbf{M}$, pose $\boldsymbol{\theta}$, and shape $\boldsymbol{\beta}$. Note that, different from SMPL, our template mesh could be any of the shells. Since each Gaussian is anchored on the shells using barycentric coordinates, we can query its new location and orientation simply from the associated vertices. In particular, given the barycentric coordinates $B\left(\mathbf{x}, \mathbf{T}\right)$ of a Gaussian inside triangle $\mathbf{T}$, and the deformed position $\mathbf{T}_{\text{new}}$ and the rotation quaternions $\mathbf{P}$ on the vertices, we can obtain the new location and orientation of the Gaussian as

$$\mathbf{x}_{\text{new}} = \phi\left(\mathbf{T}_{\text{new}}, B\left(\mathbf{x}, \mathbf{T}\right)\right), \tag{6}$$

$$\mathbf{q}_{\text{new}} = \frac{\hat{\mathbf{p}}}{\|\hat{\mathbf{p}}\|}\mathbf{q}, \text{where } \hat{\mathbf{p}} = \phi\left(\mathbf{P}, B\left(\mathbf{x}, \mathbf{T}\right)\right). \tag{7}$$

The deformed mesh is given by the SMPL deformation model $\mathrm{SMPL}$, and the quaternions $\mathbf{P}$ are a result of the linear blend skinning (LBS) from regressed joints and skinning weights, $\{w_j\}_1^{N_J}$, i.e., $\mathbf{M}_{\text{new}} = \mathrm{SMPL}\left(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{M}\right)$ and $\mathbf{p} = \text{rot2quat}\left(\sum_{j=1}^{N_J} w_j R_j\left(\boldsymbol{\theta}, J\left(\boldsymbol{\beta}\right)\right)\right)$, where $R_j$ is the rotation matrix of the $j$-th joint, and $J$ is the regressor function that maps the shape parameters to the joint locations.

### 4.3. GAN training

**Generator.** Similar to prior work [11], we adopt StyleGAN2 without camera and pose conditioning [6, 71] for the generator. We use separate MLPs with different activations for $\mathbf{c}, o, \mathbf{q}$, and $\mathbf{s}$: for $\mathbf{c}$, we use shifted sigmoid following the practice proposed in Mip-NeRF [5]; for $o$ we use sigmoid to constrain the range to $(0, 1)$; for $\mathbf{q}$ we normalize the raw MLP output to ensure they are quaternions (see Eq. (7)); for $\mathbf{s}$ we use clamped exponential activation to limit the size of the Gaussians, which is critical for convergence based on our empirical study.

**Discriminator.** Our discriminator closely follows that of [6]. Since it does not have an upsampler, the input to the discriminator is the RGB image concatenated with the alpha channel (foreground mask), which is rendered using Gaussian rasterization with the Gaussian color set to 1 for fake samples and precomputed using off-the-shelf segmentation network [55] for the real samples. We refer to the discriminator using foreground mask "Mask Discriminator". Including the alpha channel helps prevent the white background from bleeding into the appearance, which causes artifacts during articulation.

**Face, Hand, and Feet Discriminator.** As the human body and clothes are diverse, the discriminator may choose to focus on these features and provide a weak signal to the facial, hand, and foot areas, which are crucial for visually

Table 1. **Quantitative Evaluation.** We compare our method with 3D-GAN baselines using DeepFashion and SHHQ datasets. We compute the FID score to evaluate the quality and diversity of the generated samples. Notice that our scores are comparable to state-of-the-art methods. To evaluate deformation consistency, we compute the PCK metric, where our approach consistently outperforms the baselines. INF. represents the inference speed measured in ms/img on an A6000 GPU at $512^2$ resolution. Note our method is the fastest across all competing methods. $*$ numbers are adopted from EVA3D [23]; NA represents Not Available; — represents Not Applicable, and $+$ numbers are provided by the authors.

| Model | Deep Fashion | | SHHQ | | Comp. |
|---|---|---|---|---|---|
| | FID ↓ | PCK ↑ | FID ↓ | PCK ↑ | INF. ↓ |
| EG3D* | 26.38 | — | 32.96 | — | 38 |
| StyleSDF* | 92.40 | — | 14.12 | — | 32 |
| ENARF* | 77.03 | 43.74 | 80.54 | 40.17 | 104 |
| GNARF | 33.85 | 97.83 | 14.84 | <u>98.96</u> | 72 |
| EVA3D* | 15.91 | 87.50 | **11.99** | 88.95 | 200 |
| StylePeople | 17.72 | <u>98.31</u> | 14.67 | 98.58 | **28** |
| GetAvatar | 19.00+ | NA | NA | NA | 44 |
| AG3D | **10.93** | NA | NA | NA | 105 |
| Ours | <u>15.78</u> | **99.48** | <u>13.30</u> | **99.27** | **28** |

appealing results. To avoid this problem, we adopt a dedicated Face Discriminator $\mathcal{D}_{\text{face}}$, Feet-and-Hands Discriminator $\mathcal{D}_{\text{feet/hands}}$ in addition to the main Discriminator $\mathcal{D}_{\text{main}}$. All of these part-focused discriminators share the same base architecture as the main discriminator, except that we no longer use any conditioning since these cropped images do not contain distinct pose information. The inputs are crops of the corresponding parts, whose spatial span is determined from the SMPL pose and camera parameters.

**Scaling Regularization.** In our empirical study, we observe when unconstrained, the network tends to learn overly large or extremely small Gaussians early on during the training and rapidly leads to divergence or model collapse. We experimented with multiple regularization strategies and different activation functions and found the following scaling regularization most effective:

$$\mathcal{L}_{\text{scale}} = \mathsf{M} \circ \|\mathsf{T}^{\mathbf{s}} - s_{\text{ref}}\|^2 \text{ with } s_{\text{ref}} = \frac{1}{P}\sum_{i=1}^{P} \ln(\delta_i), \quad (8)$$

where $\mathsf{M}$ is the binary mask indicating UV-mapped region on the shell maps, and $s_{\text{ref}}$ is a reference scale determined by the closest-neighbor distance $\delta$ averaged among all the Gaussians.

# 5. Experiment Settings

## 5.1. Datasets

We evaluate our method using the two most common human datasets, DeepFashion [35] and SHHQ [17]. DeepFashion and SHHQ do not provide SMPL parameters, so we use SMPLify-X [43] to obtain the SMPL parameters and camera poses.

Table 2. **Ablation: Number of Shells.** Ablation on the number of shells performed on $128^2$ resolution trained for $1600k$ images.

| # of Shells | 1 | 2 | 8 | 10 | 15 |
|---|---|---|---|---|---|
| FID ↓ | 40.3 | 25.31 | 18.70 | 21.57 | 21.10 |

## 5.2. Training Details

By default, we generate the shell maps at $512 \times 512$ resolution for $8$ shells with a total of $100k$ Gaussians equally distributed across the shells. The offset between adjacent shells is $0.08$. The last fully-connected layer of the scaling prediction is adjusted to ensure the initial scaling is within a reasonable range, which is determined similar to $s_{\text{ref}}$ in Eq. (8). The loss weights for all discriminators are set to 1, whereas the scaling regularization is set to $0.1$. $R1$ regularization is used for all the discriminators with a weight of $10$. We apply progressive training starting at $256 \times 256$ rendering resolution for $6000k$ training images and progressively grow the resolution to $512 \times 512$ for $1000k$ images, then continue training at the fixed $512 \times 512$ resolution for $3000k$ images. We train our method on 8 A100 GPUs with a batch size of 32. The learning rates are initialized at $0.002$ for both the generator and discriminator.

## 5.3. Evaluation Details

**Baselines.** We compare with the following volume rendering and rasterization-based methods: EG3D [11], StyleSDF [42], GNARF [6], ENARF [41], StylePeople [19], EVA3D [23], GetAvatar [76], and AG3D [15]. Among these methods, EG3D and StyleSDF do not support articulation; except StylePeople, all methods use neural field and volumetric rendering. Regardless of representation, all these baselines use 2D convolution for post-rendering upsampling and/or postprocessing, causing flickering artifacts (see *Supplementary Materials*).

**Metrics** Following prior work, We use two metrics to evaluate the quality, diversity, and quality of animations: Fréchet Inception Distance (FID) and Percentage of Correct Keypoint (PCK). The former measures the visual quality and diversity of the generated samples (using $50k$ samples and the full dataset). The latter measures how close the generated image aligns with the pose control. We compute PCK on $5k$ samples using the implementation provided by GNARF [6].

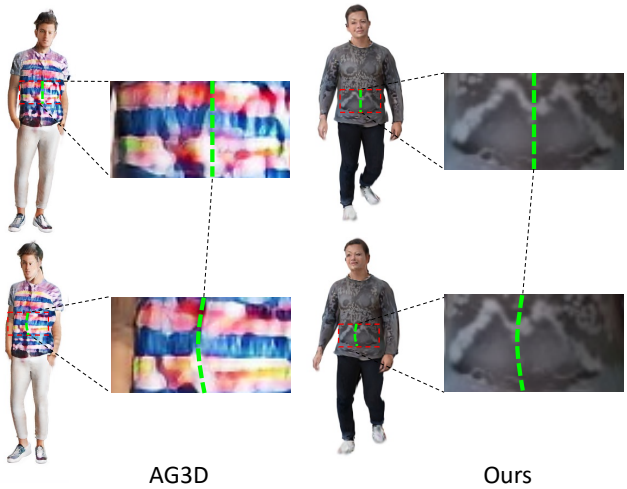


Figure 6. **Multi-view Consistency.** The figure shows a closeup of the garment in two different views. The green dashed line follows the body shape and indicates the same 3D position. In AG3D [15], the pattern significantly changes, while ours, utilizing texture maps, has built-in view consistency.

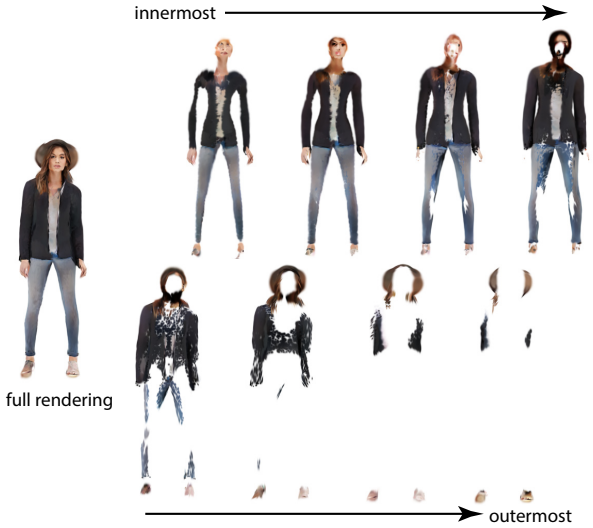


Figure 7. **Per-shell Visualization**. We visualize the per-shell contributions. The shells decompose the human body into different parts, where the inner shells capture the torso, and the outer shells capture the clothing and hair.

## 6. Experiment Results

### 6.1. Qualitative

Figure 1 showcases some of our generated results. Notice that our model is able to generate diverse attributes, including loose clothing and accessories like hats. Further, in Figure 3, we visualize the generated results in varying camera views and body poses. To demonstrate that the latent space of the proposed model is smooth, we show visual results of latent code interpolation in Figure 4 computed on the DeepFashion trained model.

Figure 5 shows the visual comparison of our method with the baselines on DeepFashion and SHHQ datasets, with more examples to be found in *Supplementary Materials*. Volumetric methods, such as GNARF, operate in a canonical pose and rely on accurate correspondence matching between the observed space and the canonical space, which is ill-defined; thus, these methods tend to produce artifacts for limbs, which undergo large deformation and occlusion. Our method is able to model the complex geometry and appearance of the human body, generating characters with loose clothing and accessories (see Figure 1) and producing convincing results under various articulations.

### 6.2. Quantitative

In Table 1, we show the results of our method compared to the competing methods. Many of the methods have not released training or data processing code; we have collected these numbers with our best effort from the authors directly.

Our method performs on par with the SOTA methods in terms of visual quality and diversity measured by FID while ranking top in terms of inference speed. By utilizing the texture space, our model can control the articulation in a straightforward manner, contributing to consistent alignment with the pose control input, as reflected by the PCK score. Note that GetAvatar is trained on multiview data, and AG3D uses normal maps for supervision.

All other baseline methods adopt post-rendering convolution layers to upsample and add texture details, leading to severe aliasing artifacts not reflected in the quantitative evaluation. We demonstrate this in Figure 6, where we render a generated identity in two different views using our method and AG3D, which achieves the lowest FID on the DeepFashion dataset. Notice that with view change, the features produced by AG3D change significantly, leading to unnatural flickering in animation even though the FID is low.

### 6.3. Ablation

We conduct ablation studies on different configurations and components of our method to understand their effect on the final model's performance. To identify the trends, we test these configurations at $128^2$ resolution on the SHHQ dataset, trained for $1600k$ images without progressive training. $200k$ Gaussians are sampled for all the experiments.

**Number of Shells.** The number of shells determines the granularity of variation we can model inside the shell volume, thus impacting the expressivity of our representation. We investigate this effect in Table 2. With only one shell, although the Gaussians can vary in size and cover regions off

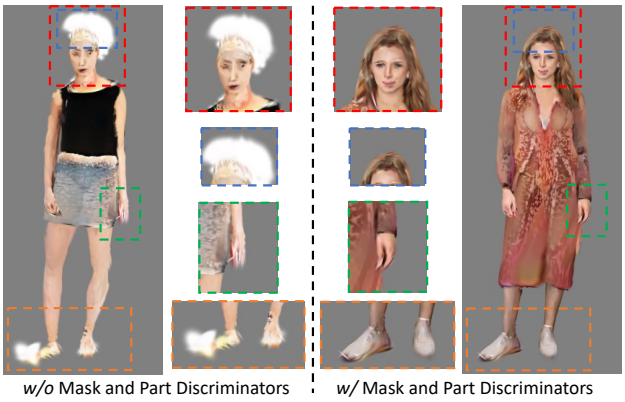


Figure 8. **Mask + Part Discriminators.** The blue box highlights the improved silhouette after concatenating the alpha channels in the input to the discriminator. The other boxes showcase better details using the face, feet, and hands discriminator.

the shell surface, they still lack the capacity to model diverse human body shapes and appearances. Using two shells essentially defines the boundary for the shell volume. This variation significantly outperforms the single-shell variation, demonstrating the importance of shell volume. Adding 8 shells (our default setting) further improves the visual quality, as the finer discretization enhances the capability to model more complex geometry and appearance. The performance gain stagnates with even more shells, likely because the model struggles to converge when the feature map becomes too large. Figure 7 shows an example identity of what each Gaussian shell has learned: progressing from the innermost to the outermost shell, every shell captures some texture details, and the outer shell models details that deviate from the base template, such as hat and hair.

**Mask and Part Discriminator** The mask discriminator involves the inclusion of the alpha channel as part of the input to the RGB images and is very beneficial for disambiguating background by learning plausible human silhouettes. As demonstrated in Figure 8, the model trained without mask discriminator generates disturbing white artifacts outside the human shape. Figure 8 also shows the improvement in small features in the facial area and better structure for hands and feet introduced by facial and feet/hands discriminator. We also quantify the joint improvement of the mask and part discriminator and observed an FID boost from 20.63 to 18.7 at $128^2$ resolution.

**Types of Gaussian Anchors.** In our representation, we opt to sample the Gaussians on shell meshes. There are other alternatives to sample the Gaussians, including constructing tetrahedra akin to the original Shell Map proposed by Porumbescu et al. [47]; or completely discarding the template by sampling uniformly inside a bounding box. We compared with these alternatives and found the shell representation achieves the best visual quality and convergence speed. A more detailed discussion can be found in *Supplementary Materials*.

## 7. Discussion

**Limitations.** Our method is not without limitations. Like almost all existing 3D human generation approaches, our method also relies on a parametric deformation model to enable articulation, which falls short in handling the dynamics of hair and loose clothing. Due to the irregularity and sparsity of Gaussians, it is not obvious how to extract the accurate geometry and normals. Surface splatting, with a stronger surface prior may be a potential solution, which we plan to investigate in the future. Lastly, while ranking among the best methods, our current results still cannot achieve photorealism. The challenge stems from the complexity of human appearance. A promising direction is to combine a small amount of multi-view studio captures with in-the-wild datasets to leverage priors discovered in controlled and richly annotated datasets.

**Ethical Concerns.** GANs, like the one we developed, carry the risk of being utilized for creating altered images of real individuals. This inappropriate application of image generation methods represents a danger to society, and we firmly oppose the use of our research for disseminating false information or damaging reputations. Additionally, we acknowledge that there might be a deficit in diversity in our outcomes, which could be a consequence of inherent biases in the datasets we utilize.

**Conclusions.** In this work, we present Gaussian Shell Maps (GSMs), a novel framework that effectively combines CNN-based generators with 3D Gaussian rendering primitives for the generation of digital humans. Our experiments demonstrate the potential of GSMs in generating diverse and detailed human figures, including complex features like clothing and accessories. The framework shows promise in enhancing the efficiency of rendering processes, a crucial factor for real-time applications in industries like virtual reality and cinematic production. The potential of GSMs in practical scenarios invites further exploration, and continued research could enhance digital human modeling.

**Acknowledgements.** We thank Thabo Beeler and Guandao Yang for fruitful discussions and Alexander Bergman for help with baseline comparisons. We also thank the authors of GetAvartar for providing additional evaluation results. This work was in part supported by Google, Samsung, Stanford HAI, and the Swiss Postdoc Mobility Fund.

# Supplementary Materials


Figure 9. **Appearance Editing.** 3D Gaussians offer an explicit representation, thereby facilitating convenient post-generation editing. In this example, we demonstrate swapping the clothing of two generated identities. Please refer to Section 9.3 for further details.

## 8. Further Analysis

### 8.1. Types of Gaussian Anchors.

GSM anchors the 3D Gaussians on shell meshes. We evaluated multiple alternative ways to anchor the Gaussians, testing the following three methods at $128^2$ resolution after training with $1.6k$ Kimgs: (i) **in bounding box**: The Gaussians are uniformly sampled within the bounding box of the 3D human mesh, with Gaussian features interpolated from axis-aligned triplane features. This variant differs from our proposed GSM as it does not utilize the shell map to learn features in texture space. Instead, 3D Gaussian features are learned in world space, requiring the generator to also model the distribution of diverse human body poses. With this variant, we demonstrate the importance of using the shell map. (ii) **on a single shell with learned offset**: This variant samples only on the base mesh, the SMPL mesh, but allows deviations from the mesh template by applying a learned offset per Gaussian, predicted by the generator as part of the feature textures. This approach emulates the typical pipeline of existing 3D human GANs, where clothing and hair are captured by offsetting the template unclothed mesh. (iii) **in tets**: The Gaussians are sampled not only on the shell meshes but also in between them in tetrahedra, constructed by connecting mesh vertices. This variant is more akin to the original Shell Map proposed by Porumbescu et al. As shown in Table 3, the bounding box variant underperforms, as the generator struggles to handle deformation jointly with appearance. Learning offsets to model surface details different from the template mesh yields subpar quality. This suggests that varying the Gaussians' positions complicates the already non-convex optimization problem, as the positions are highly correlated with the rest of the Gaussian properties. Finally, sampling in tets shows slow convergence and does not improve the FID. Additionally, this model exhibits a slower rendering speed (speed for deformation and rasterization for a generated identity) of 20 ms/img versus 9 ms/img.

### 8.2. Sampling Densities

We evaluate the effect of the number of Gaussians on the generation quality. For this study, we train on $512^2$ reso-

Figure 10. **Visualization.** 3D humans rendered in different poses using our GSM method.


Figure 11. **LSV comparison.** LSV-GAN [71] suffers from discontinuities and facial artifacts in DeepFashion and background bleeding into textures in SHHQ. In comparison, our results show high-quality facial details and consistency.

lution and evaluate the FID score after training with $10k$ KImgs. Since the sampling density will affect the Gaussian scale, we adjust the scaling regularization and initialization accordingly. As shown in Table 4, using 100K Gaussians yields empirically the best result in terms of FID for the SHHQ dataset. Using $50k$ Gaussian samples yields the highest FID, suggesting that Gaussians are likely too few to fully model the appearance complexity exhibited in the



Figure 12. **Random Samples.** Randomly generated samples of 3D humans under same pose using or GSM method trained on DeepFashion [35] and SHHQ [17] datasets.

dataset. On the other hand, using too many Gaussians, e.g. $200k$, can harm the FID. We observe that this drop

Table 3. **Anchoring types.** Ablation on the anchoring type performed on $128^2$ resolution trained for $1.6k$ KImgs on SHHQ.

| Anchoring | bbox | tets | learned offset | triangles (proposed) |
|---|---|---|---|---|
| FID ↓ | 63.90 | 24.66 | 29.30 | **20.63** |

under a high sampling density scenario is due to the tendency of Gaussians to learn small scales while modeling high-frequency details. This adds complexity to the already challenging task of optimizing opacity and scaling. As a result, we might notice unwanted dotted patterns, especially in cloth areas. The FID score easily detects such an unnatural appearance.

Table 4. **Ablation: Number of Gaussians.** Ablation on the number of Gaussians performed on $512^2$ resolution trained for 10K KImgs on SHHQ dataset.

| Number | $50k$ | $100k$ | $200k$ |
|---|---|---|---|
| FID ↓ | 23.83 | 13.30 | 19.96 |

### 8.3. Relation and Comparison with LSV-GAN

Concurrent work, LSV-GAN [71], also employs shell meshes and rasterization to efficiently model diverse human shapes and appearances. Our approach, however, distinguishes itself from LSV-GAN by populating the shell meshes with 3D Gaussians and employing differentiable Gaussian Splatting for rendering [28]. The spatial span of these Gaussians fills the space between shells with continuous functions. As illustrated in Figure 11, LSV-GAN often exhibits artifacts at silhouette boundaries due to its discontinuous representation of shell volume. In contrast, our method yields smoother and more natural boundaries. Moreover, the utilization of 3D Gaussians allows us to define RGB and alpha values beyond the boundary shell, effectively expanding our capability to model deviations from the template mesh. This leads to more varied geometry and appearances of the human body, including loose clothing and accessories, as seen in Figure 1 of the main manuscript and Figure 10 in this document.

## 9. Additional Qualitative Results

In this section, we present further qualitative results. All visual examples have been sampled using the truncation technique detailed in EG3D [11]. Additional animated results can be found on the webpage.

### 9.1. Random Samples

To showcase the quality and diversity of our GSM method, we display randomly sampled results under identical poses in Figure 12. For both the DeepFashion [35] and SHHQ [17] datasets, our method successfully generates a variety of body shapes, accessories like hats, loose clothing, and intricate details on clothes.

### 9.2. Articulation and Novel View Rendering

On the webpage, we present videos demonstrating articulation and novel view rendering results. The articulation sequences are provided by the AMASS [37] dataset. Notably, our method avoids the temporal flickering artifacts common in other models, as it directly renders at the target resolution. This efficiency is due to our use of rasterization instead of the more costly volumetric rendering approach.

### 9.3. Appearance Editing

A significant advantage of explicit representations like 3D Gaussians, especially when compared to implicit representations such as radiance fields, is their enhanced editability. In Figure 9, we illustrate this benefit through a redressing application, where we interchange the upper and lower body appearances between multiple generated instances. This editing process involves selecting Gaussians within a specific region (e.g., lower or upper body) and then swapping their properties with those from another instance. This method is feasible because the Gaussian positions are anchored on the shell meshes and remain consistent across instances, with the appearance being defined solely by their properties.

## References


[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In ECCV, 2020. 2

[2] Sizhe An, Hongyi Xu, Yichun Shi, Guoxian Song, Umit Y Ogras, and Linjie Luo. Panohead: Geometry-aware 3d full-head synthesis in 360deg. In CVPR, 2023. 2

[3] Shivangi Aneja, Justus Thies, Angela Dai, and Matthias Nießner. ClipFace: Text-guided Editing of Textured 3D Morphable Models. In ArXiv preprint arXiv:2212.01406, 2022. 2

[4] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. ArXiv, abs/2211.01324, 2022. 2

[5] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In ICCV, 2021. 5

[6] Alexander Bergman, Petr Kellnhofer, Wang Yifan, Eric Chan, David Lindell, and Gordon Wetzstein. Generative neural articulated radiance fields. NeurIPS, 2022. 1, 2, 5, 6

[7] Alexander W. Bergman, Wang Yifan, and Gordon Wetzstein. Articulated 3d head avatar generation using text-to-image diffusion models. 2023. 2

[8] 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, 2023. 1

[9] Eric Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. ArXiv, abs/2304.02602, 2023. 2

[10] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In CVPR, 2021. 2

[11] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In CVPR, 2022. 1, 2, 4, 5, 6, 13

[12] Rui Chen, Y. Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. ArXiv, abs/2303.13873, 2023. 2

[13] Xinya Chen, Jiaxin Huang, Yanrui Bin, Lu Yu, and Yiyi Liao. Veri3d: Generative vertex-based radiance fields for 3d controllable human image synthesis, 2023. 2

[14] Stavros Diolatzis, Jan Novak, Fabrice Rousselle, Jonathan Granskog, Miika Aittala, Ravi Ramamoorthi, and George Drettakis. Mesogan: Generative neural reflectance shells. Comput. Graph. Forum, 2023. 2, 3

[15] Zijian Dong, Xu Chen, Jinlong Yang, Michael J.Black, Otmar Hilliges, and Andreas Geiger. AG3D: Learning to generate 3D avatars from 2D image collections. In ICCV, 2023. 1, 2, 6, 7

[16] Jianglin Fu, Shikai Li, Yuming Jiang, Kwan-Yee Lin, Chen Qian, Chen Change Loy, Wayne Wu, and Ziwei Liu. Stylegan-human: A data-centric odyssey of human generation. In ECCV, 2022. 2

[17] Jianglin Fu, Shikai Li, Yuming Jiang, Kwan-Yee Lin, Chen Qian, Chen Change Loy, Wayne Wu, and Ziwei Liu. Stylegan-human: A data-centric odyssey of human generation. In ECCV, 2022. 2, 6, 12, 13

[18] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. NeurIPS, 2022. 2

[19] Artur Grigorev, Karim Iskakov, Anastasia Ianina, Renat Bashirov, Ilya Zakharkin, Alexander Vakhitov, and Victor Lempitsky. Stylepeople: A generative model of fullbody human avatars. In CVPR, 2021. 1, 2, 6

[20] Markus Gross and Hanspeter Pfister. Point-based graphics. Elsevier, 2011. 2

[21] Jeffrey P Grossman and William J Dally. Point sample rendering. In Proceedings of the Eurographics Workshop, 1998. 2

[22] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In ICLR, 2022. 2

[23] Fangzhou Hong, Zhaoxi Chen, Yushi LAN, Liang Pan, and Ziwei Liu. EVA3d: Compositional 3d human generation from 2d image collections. In ICLR, 2023. 1, 2, 6

[24] Suyi Jiang, Haoran Jiang, Ziyu Wang, Haimin Luo, Wenzheng Chen, and Lan Xu. Humangen: Generating human radiance fields with explicit priors. In CVPR, 2023. 2

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In CVPR, 2019. 2

[26] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In Proc. CVPR, 2020. 2

[27] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In NeurIPS, 2021. 2

[28] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. ACM TOG, 2023. 2, 3, 13

[29] Nikos Kolotouros, Thiemo Alldieck, Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Fieraru, and Cristian Sminchisescu. Dreamhuman: Animatable 3d avatars from text. ArXiv, abs/2306.09329, 2023. 2

[30] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In Comput. Graph. Forum, 2021. 3

[31] Samuli Laine and Tero Karras. High-performance software rasterization on gpus. In Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics, 2011. 2

[32] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation, 2021. 2

[33] 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. ArXiv, abs/2211.10440, 2022. 2

[34] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. ArXiv, abs/2303.11328, 2023. 2

[35] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In CVPR, 2016. 2, 6, 12, 13

[36] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. ACM TOG, 2015. 1, 2, 4

[37] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In International Conference on Computer Vision, pages 5442–5451, 2019. 13

[38] Youssef A Mejjati, Isa Milefchik, Aaron Gokaslan, Oliver Wang, Kwang In Kim, and James Tompkin. Gaussigan:

Controllable image synthesis with 3d gaussians from unposed silhouettes. arXiv preprint arXiv:2106.13215, 2021. 3

[39] Michael Niemeyer and Andreas Geiger. Campari: Camera-aware decomposed generative neural radiance fields. In 3DV, 2021. 2

[40] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In CVPR, 2021. 2

[41] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Unsupervised learning of efficient geometry-aware neural articulated representations. In ECCV, 2022. 1, 2, 6

[42] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. arXiv preprint arXiv:2112.11427, 2021. 2, 6

[43] 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 CVPR, 2019. 6

[44] Ryan Po and Gordon Wetzstein. Compositional 3d scene generation using locally conditioned diffusion. arXiv preprint arXiv:2303.12218, 2023. 2

[45] Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T. Barron, Amit H. Bermano, Eric Ryan Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, C. Karen Liu, Lingjie Liu, Ben Mildenhall, Matthias Nießner, Björn Ommer, Christian Theobalt, Peter Wonka, and Gordon Wetzstein. State of the art on diffusion models for visual computing, 2023. 2

[46] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988, 2022. 2

[47] Serban D. Porumbescu, Brian Budge, Louis Feng, and Kenneth I. Joy. Shell maps. In ACM SIGGRAPH, 2005. 2, 3, 8

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125, 2022. 2

[49] Liu Ren, Hanspeter Pfister, and Matthias Zwicker. Object space ewa surface splatting: A hardware accelerated approach to high quality point rendering. In Comput. Graph. Forum, 2002. 2

[50] Helge Rhodin, Nadia Robertini, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. A versatile scene model with differentiable visibility applied to generative pose estimation. In ICCV, 2015. 3

[51] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. CVPR, 2021. 2

[52] Darius Rückert, Linus Franke, and Marc Stamminger. Adop: Approximate differentiable one-pixel point rendering. ACM TOG, 2022. 2

[53] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. ArXiv, abs/2205.11487, 2022. 2

[54] Miguel Sainz and Renato Pajarola. Point-based rendering techniques. Computers & Graphics, 28(6):869–879, 2004. 2

[55] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In CVPR, 2018. 5

[56] Markus Schütz, Bernhard Kerbl, and Michael Wimmer. Software rasterization of 2 billion points in real time. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2022. 2

[57] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In NeurIPS, 2020. 2

[58] Zackary P. T. Sin, Peter H. F. Ng, and Hong Va Leong. Nerfahedron: A primitive for animatable neural rendering with interactive speed. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2023. 3

[59] Ivan Skorokhodov, Aliaksandr Siarohin, Yinghao Xu, Jian Ren, Hsin-Ying Lee, Peter Wonka, and Sergey Tulyakov. 3d generation on imagenet. In ICLR, 2023. 2

[60] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In Eurographics, 2004. 4

[61] Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In ICCV, 2011. 3

[62] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. ACM TOG, 2022. 2

[63] Jingxiang Sun, Xuan Wang, Lizhen Wang, Xiaoyu Li, Yong Zhang, Hongwen Zhang, and Yebin Liu. Next3d: Generative neural texture rasterization for 3d-aware head avatars. In CVPR, 2023. 2

[64] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In Comput. Graph. Forum, 2020. 2

[65] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In Comput. Graph. Forum, 2022. 1, 2

[66] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. ArXiv, abs/2304.12439, 2023. 2

[67] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In CVPR, 2020. 2

[68] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In CVPR, 2022. 2

[69] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. arXiv preprint arXiv:2112.10759, 2021. 2

[70] Yinghao Xu, Menglei Chai, Zifan Shi, Sida Peng, Ivan Skorokhodov, Aliaksandr Siarohin, Ceyuan Yang, Yujun Shen, Hsin-Ying Lee, Bolei Zhou, et al. Discoscene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis. In CVPR, 2023. 2

[71] Yinghao Xu, Wang Yifan, Alexander W. Bergman, Menglei Chai, Bolei Zhou, and Gordon Wetzstein. Efficient 3d articulated human generation with layered surface volumes. In 3DV, 2024. 1, 2, 5, 11, 13

[72] Zhuoqian Yang, Shikai Li, Wayne Wu, and Bo Dai. 3dhumangan: 3d-aware human image generation with 3d pose mapping. In ICCV, 2023. 2

[73] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. ACM TOG, 2019. 2

[74] Jianfeng Zhang, Zihang Jiang, Dingdong Yang, Hongyi Xu, Yichun Shi, Guoxian Song, Zhongcong Xu, Xinchao Wang, and Jiashi Feng. Avatargen: A 3d generative model for animatable human avatars. ArXiv, 2023. 2

[75] Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. Differentiable point-based radiance fields for efficient view synthesis. In SIGGRAPH Asia, 2022. 2

[76] 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. 1, 2, 6

[77] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In CVPR, 2023. 2

[78] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In Proceedings Visualization, 2001. 3