Get3DHuman: Lifting StyleGAN-Human into a 3D Generative Model using Pixel-aligned Reconstruction Priors

Zhangyang Xiong[1,2] Di Kang[3] Derong Jin[2] Weikai Chen[4] Linchao Bao[3]
Shuguang Cui[2,1] Xiaoguang Han[2,1*]
[1]FNii, CUHKSZ [2]SSE, CUHKSZ [3]Tencent AI Lab [4]Tencent America

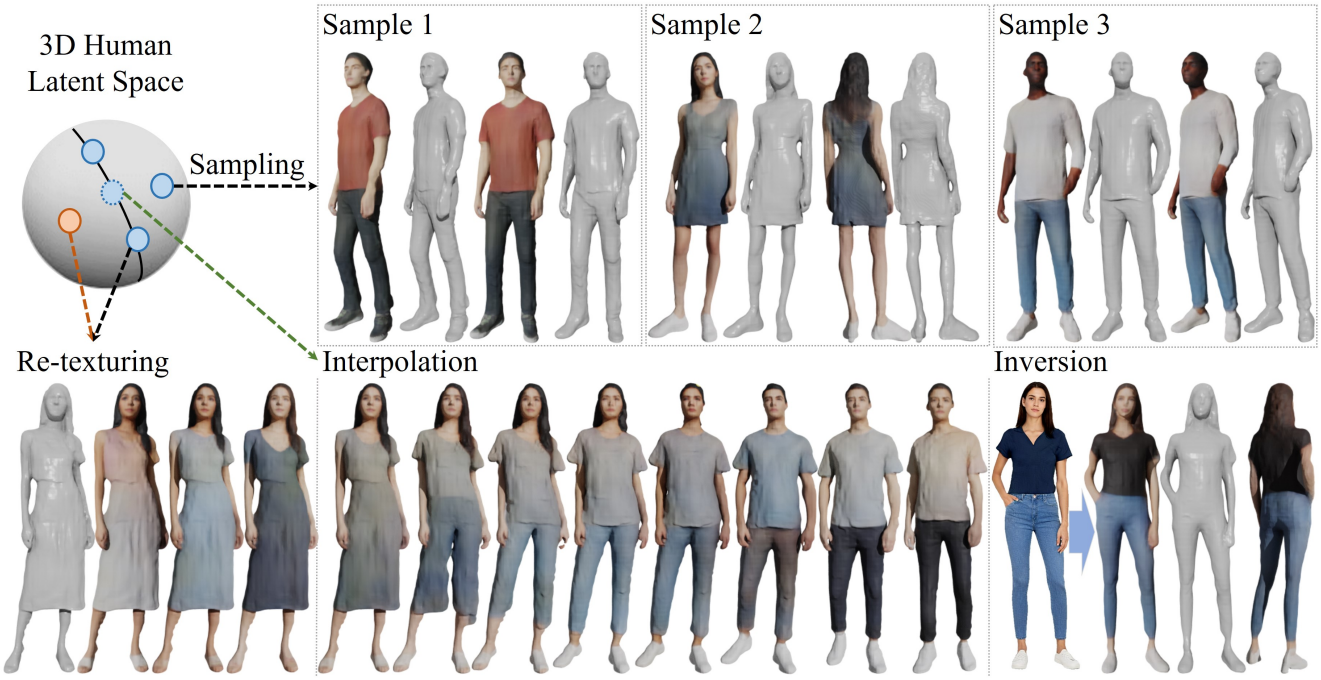

Figure 1: **Generations from Get3DHuman.** We export generated shapes and visualize them in Blender. Besides generating 3D textured human models from random codes, our method also supports re-texturing a given shape (bottom left), shape and texture interpolation (bottom middle), and inversion from a given reference image (bottom right). See Supp. for more results.

Abstract

Fast generation of high-quality 3D digital humans is important to a vast number of applications ranging from entertainment to professional concerns. Recent advances in differentiable rendering have enabled the training of 3D generative models without requiring 3D ground truths. However, the quality of the generated 3D humans still has much room to improve in terms of both fidelity and diversity. In this paper, we present Get3DHuman*, a novel 3D human framework that can significantly boost the realism and diversity of the generated outcomes by only using a limited budget of 3D ground-truth data. Our key observation is that the 3D generator can profit from human-related priors learned through 2D human generators and 3D reconstructors. Specifically, we bridge the latent space of Get3DHuman with that of StyleGAN-Human [13] via a specially-designed prior network, where the input latent code is mapped to the shape and texture feature volumes spanned by the pixel-aligned 3D reconstructor [50]. The outcomes of the prior network are then leveraged as the supervisory signals for the main generator network. To ensure effective training, we further propose three tailored losses applied to the generated feature volumes and the intermediate feature maps. Extensive experiments demonstrate that*

*Corresponding author: hanxiaoguang@cuhk.edu.cn

Get3DHuman greatly outperforms the other state-of-the-art approaches and can support a wide range of applications including shape interpolation, shape re-texturing, and single-view reconstruction through latent inversion.

1. Introduction

Generating diverse and high-quality virtual humans plays an important role in numerous applications, including VR/AR, visual effects, game production, etc. The advances in generative models have brought impressive advances to the state-of-the-art of generating 2D virtual avatars, such as images or videos. However, synthesizing 3D humans with high fidelity and large variations remains much under-explored due to the scarcity of 3D human data.

The conventional solution [9, 8, 55, 11] for acquiring a 3D avatar from a real person is typically a time-consuming and cumbersome process, requiring a specialized capture system, substantial manual efforts, and extensive computation. To circumvent the requirement of collecting a large corpus of 3D ground-truth data, recent works utilize the differentiable rendering technique to train a 3D generative model in a 3D-unsupervised manner [6, 14]. Specifically, instead of using direct 3D supervision, adversarial losses are applied to the images rendered from the synthesized 3D content. However, due to the lack of dense multi-view images for each model, these methods can only encourage geometry-to-image consistency in the selected views while failing to produce plausible reconstruction in the unseen regions. In addition, the differentiable rendering process is computationally heavy, making the network training highly inefficient.

To resolve the above issues, in this work, we present *Get3DHuman*, a novel 3D generator that can faithfully synthesize high-fidelity clothed 3D humans with a diversity of shapes and textures. Our key observation is that 3D human generators can benefit from the inductive bias from a 2D human synthesizer and the prior knowledge learned through relevant 3D modeling tasks. In particular, to bypass the limited availability of 3D ground truths, we propose to leverage the generative power of 2D human synthesizers which have shown more promising and stable quality than their counterpart 3D generators. We further lift the rich prior from the 2D generator, i.e. StyleGAN-Human [13], to 3D by using the pixel-aligned reconstruction priors, i.e. the pre-trained PIFu network [49], through single-view human reconstruction. Thanks to the strong generalization ability of pixel-aligned implicit reconstructor, by feeding it with a myriad of photo-realistic human images generated by StyleGAN-Human, we are able to obtain a vast number of 3D human models with highly diversified body shapes, apparel, poses, and textures. To further ensure the high quality of the generated shapes, we filter out inferior results via manual inspections.

We further materialize the above idea via a novel prior induction mechanism. Specifically, we first train a prior network to encode the 2D generator prior and the 3D reconstruction prior into three supervisory signals. That is, given a random latent code, the prior network would generate normal maps, depth maps, and shape and texture feature volumes of the 3D human corresponding to the input code (see Fig. 2). The input code is sampled from the latent space of StyleGAN-Human while the shape and texture feature volumes are consistent with the PIFu latent, and, hence, can be converted to the predicted human shape via the pre-trained PIFu decoder. We then supervise the training of the proposed 3D generators via three specially-tailored losses. First, a latent prior loss is introduced to provide direct supervision of the generated feature volumes for the shape and texture generation branches. Second, an adversarial loss is applied to the 3D feature volumes instead of the output signed distance field (SDF). This helps reduce the training complexity while ensuring the realism of the generated 3D humans. Lastly, normal maps and depth maps are used for supervising the generation of intermediate feature maps. Specifically, instead of directly transforming the input code into a 3D feature volume, we first map the code to 2D feature maps and then lift them into 3D feature volume. This additional intermediate supervision helps cast finer-grained geometry details as shown in our experiments (see Fig. 7). We further utilize a refinement module to improve the quality of our textured mesh as the texture prior is not always satisfactory.

Our method can support a wide range of applications, including shape generation, interpolation, shape re-texturing, and latent inversion from a single image. We evaluate Get3DHuman via extensive experiments and demonstrate that it strongly outperforms the state-of-the-art methods, both qualitatively and quantitatively.

To summarize, our main contributions include:

- We propose a novel 3D human generation framework that explicitly incorporates priors from top-tier 2D human generators and 3D reconstruction schemes to achieve high-quality and diverse 3D clothed human generation.
- We present specially-tailored prior induction losses for effective and efficient prior-based supervision.
- We set the new state of the art in the task of shape generation while supporting many applications including shape interpolation, re-texturing, and latent inversion.

2. Related Work

Constructing generative models for images, videos or 3D models is a fundamental problem in the field of computer graphics and computer vision. Here, we only review and summarize the work related to 2D or 3D human generation that is relevant to our work.

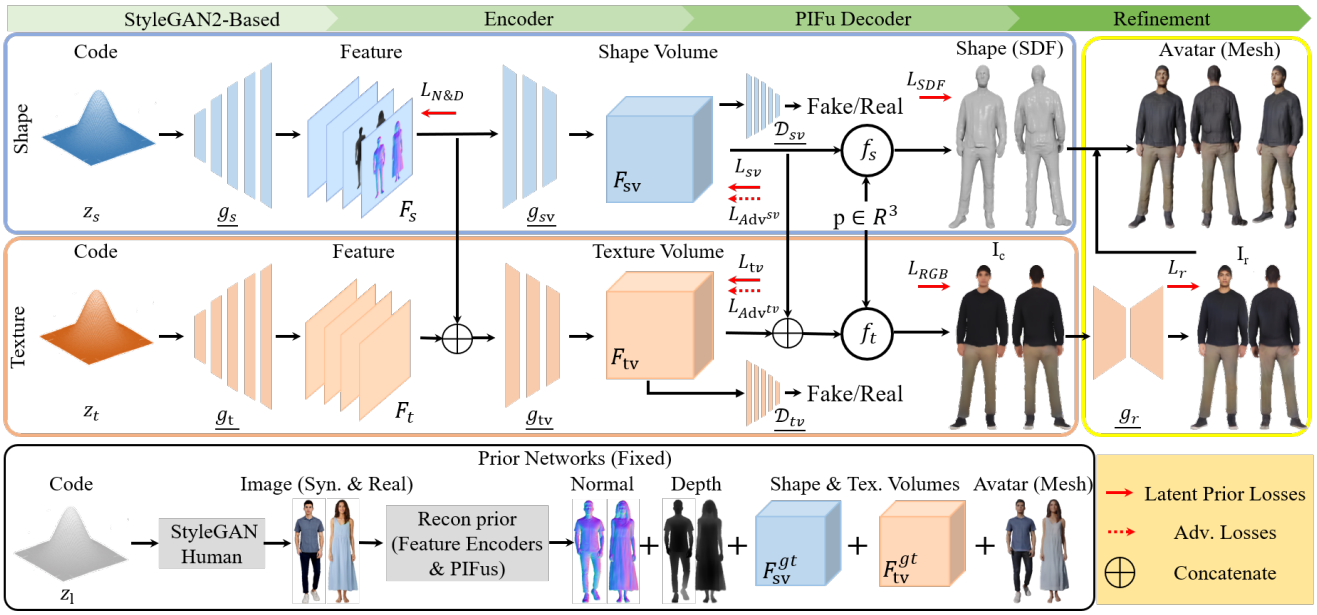

Figure 2: **Overview of our framework.** Our Get3DHuman consists of a shape generator (blue box) and a texture generator (orange box) with a refinement module (yellow box) that enables nonexistent 3D human creation. Shape generator responds for generating a high-quality full-body geometry from a shape code and sends shape features to the texture generator. Texture generator predicts RGB colors of all points in the 3D space from a texture code and intermediate shape features. Trainable modules are underlined, including g_s, g_{sv}, $\mathcal{D}_{sv}$, g_t, g_{tv}, $\mathcal{D}_{tv}$, and g_r. These seven modules are all trained from scratch. The prior network (black box) only produces supervisory signals for the training of form.

2D Human Generation. In recent years, generative adversarial networks (GANs) [15] has been successfully applied to the task of image synthesis. Many methods have built photorealism generative models of 2D human faces [25, 26, 24]. Based on the success of these efforts, faces can also be efficiently edited [48, 53, 19]. On another line, some researchers attempt to generate 2D images of dressed humans from sketch, pose, or text conditions [32, 27, 2, 51, 20]. The recent works of [13, 12] have built large-scale datasets and resorted to StyleGAN to achieve impressive 2D human generation results. The success in 2D image generation and editing has inspired research in 3D generation.

3D Human Modeling. As a common representation, parametric human model, like [4, 31, 46, 5, 23, 22, 42, 44], controls a template through a series of low-dimensional parameters to obtain a 3D naked human body, with which the modeling can be performed by regression approaches. These works are further extended to model dressed human [3, 34], by introducing vertices' movements on the template meshes. Some other representations are also proposed for dealing with more complex clothed bodies, like point clouds [33, 35, 58], radiance fields [29, 41, 45, 54] and implicit fields [7, 37, 43, 57]. In particular, the implicit field representation has become a powerful tool for modeling 3D (clothed) human shapes, as it can capture arbitrary resolutions and fine-grained details.

PIFu [49] first proposes a pixel-aligned implicit function to extract local features from images to complete human body digitization. They are further expanded to PIFuHD [50], which uses predicted normal maps and a multi-level architecture to generate higher resolution and richer details consistent with the input image.

3D Human Generation. Early approaches aimed to extend GANs to voxel [56, 40], point cloud [1, 61, 38], implicit [37, 7]. However, these works focus mainly on generating geometry, not appearance.

EG3D [6] proposes to generate multi-view images of a 3D face via introducing an efficient tri-plane representation and a neural rendering design with the help of a pretrained StyleGAN.

Following EG3D, there are also some methods aiming to full-body human generation [59, 16]. Since their adversarial losses are only applied to the rendered images and there is no explicit supervision on the geometry, making the produced shape is far from satisfactory.

Similar to our work, GET3D [14] also targets to generate textured 3D meshes. Specifically, it generates an SDF field and a texture field by two latent codes and performs simultaneous optimization of the two fields by adversarial loss on 2D rendered images. In our work, we focus on 3D textured human model and aims to generate 3D textured meshes with

diverse human poses and appearances.

Recently, a concurrent work HumanGen [18] is posted on the Internet. Although it also involves the prior of StyleGAN-Human and PIFu for generative model construction, we have a key difference: HumanGen only builds a generative model to produce a texture field from a code where the geometry is directly borrowing from the prior, while ours aims to use two codes to generate both shape field and texture field. In this fashion, different from HumanGen, we truly built a latent space for 3D textured humans.

3. Method

Fig. 2 shows the overall network of the proposed 3D generator network. Get3DHuman contains parallel shape and texture branches, which is facilitated by a prior network branch during training. Take the shape branch as an example, given a shape latent code, Get3DHuman generates a shape feature F_s, a shape feature volume F_{sv}, and produces a high-quality human shape represented as an SDF (Sec. 3.1). Sec. 3.2 details our prior-facilitated learning of this 3D generator. Sec. 3.3 describes the texture network and its training. Finally, training details including data preparation, and training strategy are described in Sec. 3.5.

3.1. The Proposed 3D Generator

As shown in Fig. 2 top, it contains three parts, including a StyleGAN-like Mapping & Synthesis stage (g_s), a feature encoder stage (g_{sv}), and a PIFu-style geometry decoder (f_s). Given a shape latent code z_s drawn from a Gaussian distribution, we first use a StyleGAN2 network [26] g_s to generate a shape feature F_s, which is fed into a hourglass-style [39] fully convolutional feature encoder g_{sv} to extract a shape feature volume F_{sv}. *i.e.*:

$$F_s = g_s(z_s), \tag{1}$$

$$F_{sv} = g_{sv}(F_s), \tag{2}$$

where z_s is the random shape latent code, F_s and F_{sv} are the intermediate pixel-aligned high-dimensional **s**hape feature image and **s**hape feature volume, respectively.

With the shape feature volume, a PIFu [49] shape decoder f_s is used to evaluate the signed distance $s \in \mathbb{R}$ given a query point $p \in \mathbb{R}^3$ and its corresponding feature. *i.e.*:

$$f_s(F_{sv}(x), d(p)) = s : s \in \mathbb{R}, \tag{3}$$

where $p \in \mathbb{R}^3$ is the 3D query point, $F_{sv}(x)$ is the image feature at p's projected location $x = \pi(p)$ on the image plane, $d(p)$ is the depth value of p in the camera frame, and f_s is the PIFu shape decoder (MLPs) that is pretrained like [50] and fixed.

3.2. Prior-Facilitated Learning

Besides adversarial loss, we propose to fully utilize the prior information from well-trained networks to facilitate the training of the 3D GAN due to limited training data and the complexity of this task. Specifically, we first utilize image-latent pairs from StyleGAN-Human to learn the latent space better since it already has a reasonable structure after training. Then, we extract intermediate features using the PriorNet branch supervision to guide the training, which can also be seen as a kind of deep supervision. In the following, we first describe how to extract helpful prior information and then describe the training losses.

Prior extraction. The prior network is the concatenation of a StyleGAN-Human [26] generator and a PIFuHD-like [49, 50] 3D reconstructor (see Fig. 2). So, given a latent code, a full body human image is first synthesized by the StyleGAN-Human generator. Then the following 3D reconstructor takes as input the synthesized image and extracts a normal map N, a depth map D, a shape feature volume F_{sv}, and a texture feature volume F_{tv}, which are used as supervisory signals since they contain helpful human prior information.

Shape losses. Given a latent code, the shape generator produces intermediate shape feature F_s and shape feature volume F_{sv}, and the final SDF s (see Fig. 2). Our training losses include a latent adversarial loss $\mathcal{L}_{Adv^{sv}}$ applied on the shape feature volume F_{sv}, a SDF loss $\mathcal{L}_{SDF}$, a latent prior loss $\mathcal{L}_{sv}$ applied on the shape feature volume, a depth loss $\mathcal{L}_D$ and a normal loss $\mathcal{L}_N$ applied on the first four channels of the intermediate shape feature F_s. The loss terms are detailed in the following.

SDF loss $\mathcal{L}_{SDF}$. We sample M points per iteration, including near surface points (obtained from depth map) and random points, and apply L1 loss on them. The SDF value is predicted by a pretrained and fixed PIFu MLP decoder [49] $s = f_s(F_{sv}(x), d(p_i))$, where $x = \pi(p)$ is the 2D projection of query point p and $d(p)$ is the depth value of p in the camera coordinate space.

$$\mathcal{L}_{SDF} = \frac{1}{M}\sum_{i}^{M} ||\hat{s} - s||_1 \tag{4}$$

Latent prior loss $\mathcal{L}_{sv}$. To take full use of the prior knowledge, we also apply constraints on the intermediate features as follows. Another potential benefit is the sampled 3D points in the previous SDF loss are too sparse while this feature field loss can provide supervision on the whole feature maps.

$$\mathcal{L}_{sv} = ||F_{sv} - F_{sv}^{gt}||_1 \tag{5}$$

Geometry loss. We force several channels of the intermediate feature maps to predict helpful 2.5D information (i.e. normal) since these feature maps are pixel-aligned (i.e. the spatial information is retained). This operation is similar to

deep supervision [28] and helps the learning processing.

$$\begin{aligned}\mathcal{L}_{N\&D} =&\lambda_N||\hat{F}_{s(c:1,2,3)} - N||_1+\\ &\lambda_D||\hat{F}_{s(c:4)} - D||_1\end{aligned} \tag{6}$$

where $\hat{F}_{s(c:1,2,3)}$ and $\hat{F}_{s(c:4)}$ are the first three and fourth channels of the predicted shape feature F_s, D and N is the pseudo groundtruth normal map and depth map.

Latent adversarial loss $\mathcal{L}_{Adv^{sv}}$. $\mathcal{D}_{sv}$ is a discriminator taking the shape volume as input. We use the non-saturating GAN loss proposed in StyleGAN2 [26] and R1 regularization [15, 36].

$$\mathcal{L}_{Adv^{sv}} = \mathcal{L}_{GAN^{sv}} + \lambda_{Reg}\mathcal{L}_{Reg^{sv}} \tag{7}$$

In summary, the total shape loss $\mathcal{L}_{Shape}$ is

$$\begin{aligned}\mathcal{L}_{Shape} =&\lambda_{SDF}\mathcal{L}_{SDF} + \lambda_{sv}\mathcal{L}_{sv} + \lambda_N\mathcal{L}_N+\\ &\lambda_D\mathcal{L}_D + \lambda_{Adv^{sv}}\mathcal{L}_{Adv^{sv}}\end{aligned} \tag{8}$$

where $\lambda_{(\cdot)}$s are the corresponding loss weights. After training, the shape generator is fixed during the later texture branch learning.

3.3. Texture Generator & Training Losses

The texture branch is almost a mirror of the shape branch except several small differences.

Texture generator. The texture branch consists of a StyleGAN-style generator g_t, a feature encoder g_{tv}, a PIFu-style texture decoder f_t, and a texture volume discriminator $\mathcal{D}_{tv}$. The texture branch is similar to the shape branch except it is additionally conditioned on intermediate features from the shape branch (see Fig. 2 and Eqs. 1-3). *i.e.*:

$$F_t = g_t(z_t), \tag{9}$$

$$F_{tv} = g_{tv}(F_t \oplus F_s), \tag{10}$$

$$f_t((F_{tv}(x) \oplus F_{sv}(x)), d(p)) = c \in [0,1]^3, \tag{11}$$

where z_t is texture latent code, $\oplus$ represents channel-wise concatenation to introduce intermediate features (i.e. F_s, $F_{sv}(x)$) from the shape branch. Similarly, f_t is the PIFu texture decoder (MLPs) that is pretrained like [49] and fixed. The texture discriminator is also similar to the shape discriminator but with 512 input channels.

Texture losses. Similar to the shape branch, given a texture latent code, the texture branch first generates a texture feature (F_t) and a texture feature volume (F_{tv}). Different from the shape branch, it also takes as input intermediate shape features (e.g. F_s, F_{sv}) that are generated using the same latent code z (i.e. $z_s = z_t$). At the same time, the texture branch use the same latent code to predict a texture feature (F_t). Similarly, the individual texture losses are as follows:

$$\mathcal{L}_{RGB} = \frac{1}{M}\sum_i^M ||\hat{c} - c||_1, \tag{12}$$

$$\mathcal{L}_{tv} = ||F_{tv} - F_{tv}^{gt}||_1, \tag{13}$$

$$\mathcal{L}_{Adv^{tv}} = \mathcal{L}_{GAN^{tv}} + \lambda_{Reg}\mathcal{L}_{Reg^{tv}}, \tag{14}$$

The overall texture loss $\mathcal{L}_{Texture}$ is:

$$\mathcal{L}_{Texture} = \lambda_{RGB}\mathcal{L}_{RGB} + \lambda_{tv}\mathcal{L}_{tv} + \lambda_{Adv^{tv}}\mathcal{L}_{Adv^{tv}} \tag{15}$$

where $\lambda_{(\cdot)}$s are the corresponding loss weights. Note that we fix the shape branch and train the texture branch.

3.4. Refinement Module

A textured mesh can be obtained from the two implicit fields similar to PIFu [49]. However, due to limited training data, the reconstruction prior learned in the texture field is not always satisfactory. For example, the rendered images could be blurry or sometimes erroneous, which means the corresponding textured meshes extracted from the implicit fields could also be problematic.

To obtain high-quality mesh colors, we propose an extra refinement module, which consists of an image refinement step and a (mesh) texture update step.

Image refinement. The image refinement is realized as an image-to-image (I2I) translation task using a UNet-style [17] network g_r.

$$I_r = g_r(I_c) \tag{16}$$

where g_r is the network, I_c is the rendered image from shape and texture volumes, and I_r is the refined image for later texture extraction. We impose L1 loss and perceptual loss [21] between the refined result and the ground truth image I^{gt} during training. The loss $\mathcal{L}_r$ is defined as:

$$\begin{aligned}\mathcal{L}_{\text{r}} =&\lambda_r||I_r - I^{gt}||_1+\\ &\lambda_P\sum_l \left\|\Phi^l(I_r) - \Phi^l(I^{gt})\right\|_2^2,\end{aligned} \tag{17}$$

where Φ^l denotes the multi-level feature extraction operation using a pretrained VGG network, $\lambda_{(\cdot)}$s are the corresponding loss weights.

After image refinement, we can get an person image in the same pose but with sharper and more correct face/cloth details (see Fig. 6).

Vertex-color refinement. With the improved multi-view images, we can paint the extracted mesh $\mathcal{M}$ with new colors accordingly. Specifically, we utilize surface tracking [30] to render multi-view depth maps paired with those images. With paired depth maps and their corresponding refined images, we can obtain a high-quality colored point cloud $\mathcal{P}$ via 2D-to-3D projection. Finally, for every mesh vertex v, we paint it with the color of its nearest neighbor point in the colored point cloud $\mathcal{P}$.

3.5. Data Preparation & Training

Collecting & filtering data. Three different types of data are used in this work, including 396 high-quality 3D human models, real-world Internet images, and StyleGAN-Human [13] synthesized images. More concretely, the 3D human models are purchased from RenderPeople [47], which include both 3D mesh models and high-resolution texture maps that could be rendered into photorealistic images. The real images are crawled from the Internet and 14,097 are left after a manual selection to exclude extreme poses and oddly shaped costumes that can not be handled by reconstruction prior. The synthesized images are generated using StyleGAN-Human [13] and 69,099 are left after a manual selection to exclude images with obvious artifacts (e.g. distorted body parts).

Extracting pseudo-GT. We extract pseudo-GT for a given image using two prior networks, a human image synthesis network [60] and a single-view reconstruction network [49], which are trained using our RenderPeople data. For a real image, the corresponding pseudo-GT includes a real image, a depth map, a normal map, the shape volume $\mathcal{F}_{sv}$, and the texture volume $\mathcal{F}_{tv}$. For a synthesized image, we also stored its corresponding latent code (z_l). A pretrained PIFu decoder can easily extract 3D or texture information from specific feature volumes. After the aforementioned pre-processing, we randomly choose 500 (real) and 1500 (synthesis) pseudo-GT for evaluation and use the remaining 81,196 samples during training.

Network training. We train the shape branch, the texture branch, and the refinement network in three separate stages.

In the $1st$ stage, we only train the shape branch (with the PIFu shape decoder f_s fixed). Specifically, we train the shape branch using the shape reconstruction losses in Eq. 8. The $\lambda_{(.)}$s are set to $\{20, 40, 20, 20, 1\}$, empirically. We found the weight of the adversarial loss has an obvious influence on the synthesized 3D models. For example, using too large adversarial loss usually produces unrealistic high-frequency noise, and using too small adversarial loss usually results in overly-smoothed 3D models, showing similar trends to Fig. 4.

In the $2nd$ stage, we train the texture branch with the shape branch and the PIFu texture decoder f_t fixed. The prior losses $\mathcal{L}_{RGB}$, $\mathcal{L}_{tv}$ and the adversarial loss $\mathcal{L}_{adv^{tv}}$ in Eq. 15 are applied simultaneously. The $\lambda_{(.)}$s are set to $\{20, 40, 1\}$, empirically. Note that examples generated with both paired shape-texture latent codes (i.e. $z_s = z_t$) and unpaired latent codes (i.e. $z_s \neq z_t$) will be used for adversarial training so that we can obtain different and reasonable textures for the same shape latent code, which enables the re-texturing application.

In the $3rd$ stage, we only train the I2I refinement network g_r with all the other parts fixed. The L1 loss $\mathcal{L}_r$ and the perceptual loss $\mathcal{L}_P$ in Eq. 17 are applied simultaneously. The $\lambda_{(.)}$s are set to $\{1, 1\}$. Note that 360° images rendered from RenderPeople data are additionally used for training in this stage.

Table 1: Quantitative comparisons with SOTA methods.

	COV↑ (%)	MMD↓	FPD↓	FID↓	FID_{3D} ↓
EG3D	15.33	2.54	2.21	76.55	221.73
SDF-StyleGAN	23.35	1.12	1.02	-	-
GET3D	35.93	0.77	0.87	61.69	88.15
Ours	**39.22**	**0.58**	**0.85**	**54.39**	**69.70**

Figure 3: Visual comparisons with state-of-the-art 3D human generators. SDF-StyleGAN can only generate geometry. For the other methods, we visualize the geometry and appearance rendered with Blender, and the images (rightmost of each set of results) generated directly by the network. EG3D uses volume rendering to generate images, while our method and GET3D query the RGB values on the surfaces. Compared to the others, our results contain sharper and more plausible details in both geometry and texture, achieving the best scores on all the metrics in Tab. 1.

4. Experiments

In Sec. 4, we first introduce the quantitative evaluation used in our experiments. In Sec. 4.1, we compare the proposed method with other state-of-the-art methods, showing that our method generates more diverse and higher-quality textured meshes. We conduct ablation studies of our method to verify the effectiveness of the adversarial loss, the reconstruction prior, and the refinement module in Sec. 4.2. Finally, we demonstrate three downstream applications of

our method in Sec. 4.3, including re-texture the generated meshes, interpolation between two latent codes, and inversion from the real-world image.

Geometry evaluation. Similar to any 3D GAN, we adopt Fréchet point cloud distance (FPD) [10] to evaluate the diversity and quality of the generated shapes. We use Chamfer Distance based (d_{CD}) Coverage (COV) and Minimum Matching (MMD) following [1, 14] to evaluate the similarity between a set of generated meshes and the reference set of pseudo-GT meshes. See our Supp. for detailed settings.

Texture evaluation. To evaluate the quality of the generated textures, we used the common Fréchet Inception Distance (FID) metric on 2D images directly generated from the model and FID_{3D} on 2D images rendered using the textured 3D mesh model. we randomly choose 10k real images and 40k synthetic images as references to calculate the FID.

4.1. Comparisons with State-of-the-art Methods

Baselines. We compare our Get3DHuman with three state-of-the-art methods, including EG3D [6], SDF-StyleGAN [60], and GET3D [14].

EG3D is a 3D-aware image generation method focusing more on the rendered image rather than the geometry.

SDF-StyleGAN, which only generates geometry, is a StyleGAN2-based network plus local and global shape discriminators that take as input SDF values and gradients.

GET3D, similar to ours, also uses shape and texture branches. Its shape branch utilizes a differentiable surface extraction method (i.e. DMTet [52]) and its texture branch is based on EG3D. Different to ours, its supervisions are purely applied on 2D renderings (i.e. images, silhouettes). The original GET3D [14] paper is trained in a T-posed RenderPeople dataset. We have attempted to train GET3D with our purchased non-T-posed RenderPeople data. However, the results are poor in terms of geometry and textures (see Supp. for the results), possibly due to the large pose space and limited training data. Its results could possibly be improved with more training data, but high-quality 3D data are expensive and difficult to obtain.

For a fair comparison, all these three approaches are trained with our pseudo-GT data. We utilize shape and texture fields to render images to train EG3D and GET3D and get the SDF fields to train SDF-StyleGAN.

Results. Tab. 1 shows quantitative comparisons and Fig. 3 visualizes the results. Our results beat the others by a substantial margin, especially on COV, MMD, and FID. We can easily observe the differences from the visual comparisons.

SDF-StyleGAN, which is only able to generate shapes, generates a little better-looking shapes than EG3D. But it often generates disconnected regions so that artifacts are frequently observed in the elbows and ankles.

EG3D focuses more on rendering high-quality images and cannot produce a reasonable geometry sometimes. The textured mesh does not look good because the geometries are problematic.

GET3D results and our results look better since the shapes are clear and contain more details while having fewer artifacts. Compared to ours, GET3D generates over-smoothed shapes with more artifacts and its textures are not are sharp as ours. This is also reflected by the FID (54.39 vs 61.69) and FID_{3D} scores (69.70 vs 88.15) in Tab 1. Since the geometry generated by EG3D is not accurate enough, the FID_{3D} is much higher (221.73) than other methods.

Visualization. We follow Get3D and render textured meshes in Blender with lighting (Fig. 5 right). Renderings via PyTorch without lighting, which are shown in Fig. 5 left, are the original results.

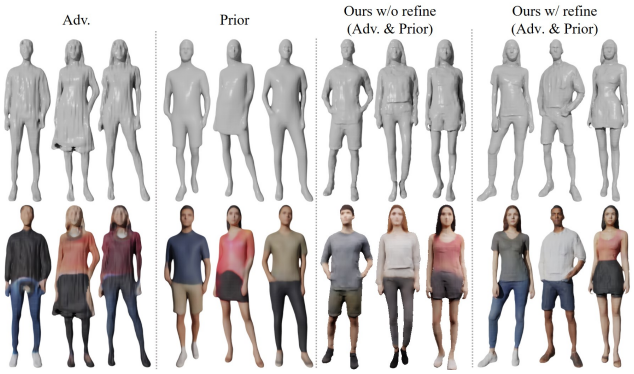

Figure 4: Ablation study on adversarial losses and prior losses. We compare the human models generated from "Adv. only", "Prior only", and "Ours Adv. + Prior". Ours (Adv. & Prior) produces the best results with plausible details on both the shape and the appearance. The refinement module further improves the appearance.

Table 2: Quantitative evaluation on adversarial and prior.

	COV↑ (%)	MMD↓	FPD↓	FID↓	FID_{3D} ↓
Adv. only	35.62	0.72	0.63	89.39	105.63
Prior only	31.51	0.80	0.88	69.81	89.32
Ours (w/o refine) (Adv. & Prior)	39.22	0.58	0.85	59.95	74.38
Ours (w/ refine) (Adv. & Prior)	**39.22**	**0.58**	**0.85**	**54.39**	**69.70**

Figure 5: Visualize the results of two rendering methods. Blender renders textured meshes with lighting(right). PyTorch renderings look sharper and brighter(left).

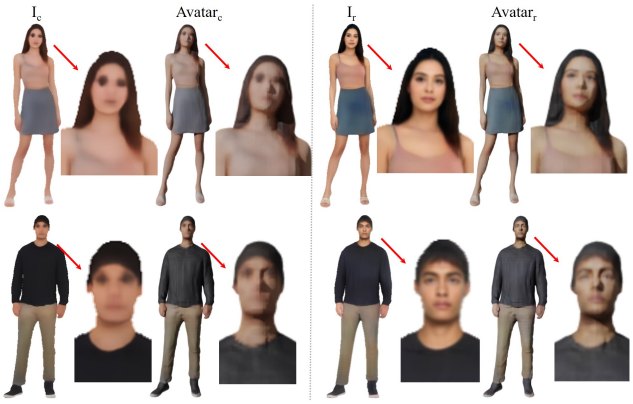

Figure 6: Ablation on the refine module. Using the refinement module brings realistic texture details (right). I_r and I_c are images directly generated from the network.Avatars are rendered from texture models by using Blender.

4.2. Ablation Studies

Ablation study on the adversarial and prior. The proposed method consists of two kinds of losses, i.e. adversarial losses and prior losses. Each type of loss is able to obtain a certain result. Ablative experiments, including "Adv. only", "Prior only" and "Adv. & Prior" with or without refinement, are conducted to demonstrate their effectiveness in Tab. 2. Fig. 4 shows some examples synthesized using random latent code. Only using adversarial losses in training (Adv. only) easily produces results that contain high-frequency noise, i.e. unrealistic details. Only using prior losses (Prior only) usually produces overly-smoothed results, which has been observed in 2D image synthesis too. Ours (Adv. & Prior) produces the best results with plausible details on both the geometry and the texture. When refinement module is included, the final textured mesh further gets a huge improvement in Fig. 6.

Ablation on normal guidance. The explicit use of normal map in 3D reconstruction networks has been demonstrated very helpful [50]. Thus, we introduce a normal map as guidance for the intermediate feature maps to guide the later geometry generation and conduct ablative experiments using only prior losses. Results are shown in Fig. 7. With the help of this normal supervision, more plausible details, especially the separation between upper-/lower-body clothes, emerge on the generated shapes, demonstrating the effectiveness of using the normal map as guidance.

Ablation on refinement module. Because the reconstruction prior might be inaccurate and sometimes leads to the unrealistic appearance, especially in the face area. To this end, we designed a refinement module. Tab. 2 and Fig. 6 illustrate the effectiveness of the refinement module in improving the appearance.

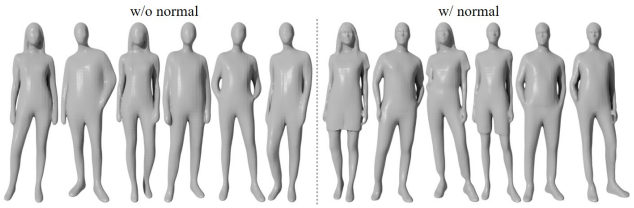

Figure 7: Ablation study on the intermediate normal supervision. Using normal supervision in training brings substantial geometry details (left). Note the abrupt depth changes between cloth and skin and the facial geometry.

Figure 8: Visualization of re-texturing the fixed geometry by different texture latent codes. More results are in Supp. We can see different textures are diverse, plausible, and suitable for the given shape since our texture branch is conditioned on shape branch features.

4.3. Applications

Re-texturing the generated meshes. The shape/texture latent codes could be different in our two-branch 3D generator. In fact, we intentionally include some unpaired shape-texture latent codes (i.e. different values for the shape/texture latent codes) during training and apply adversarial loss on their generations during training. Since the texture branch does not affect the shape branch, the generated shape is completely fixed when the shape latent code is given. Thus, we can randomly sample texture latent code with the shape latent code fixed, resulting in a useful re-

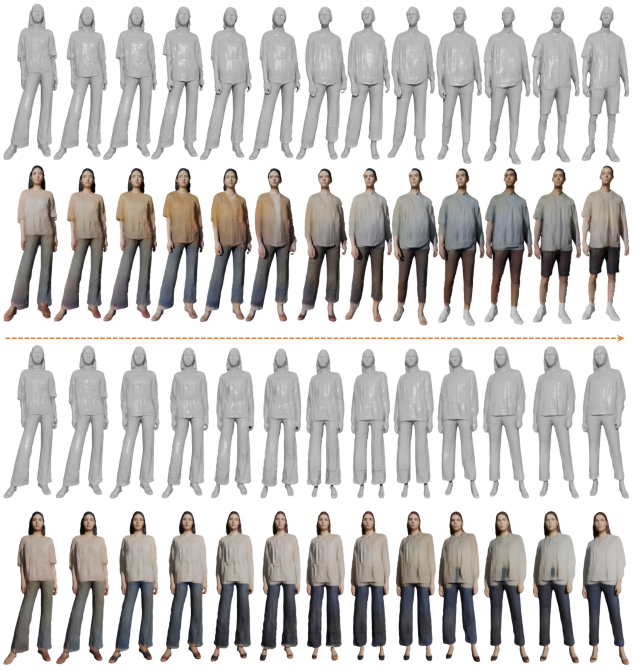

Figure 9: Interpolation examples. We randomly sample two sets of shape/texture latent codes to generate the right-/left-most examples, then interpolate both the shape and texture latent codes to generate the in-between examples. See Supp. for more results.

texturing application. Two examples are shown in Fig. 8. Our Get3DHuman successfully paint the 3D models with different and reasonable texture and takes the conditioned geometry into consideration.

Interpolation on shape and appearance. With our prior-facilitate learning, our method learns a better structured latent space z. Thus, we can produce reasonable interpolations given two reference latent codes. Specifically, we linearly interpolate two random codes and generate textured 3D models accordingly. Smooth and valid transition models with plausible details are produced as shown in Fig. 9.

Inversion results. In 2D GAN, inversion refers to find the corresponding *latent code* that generates a given *target* (image). Similarly, we are also searching suitable latent z_s, z_t for given feature volumes F_{sv}, F_{tv}. Get3DHuman is also able to perform image inversion similar to StyleGAN shown in Fig. 10. Given a human image, we first extract its shape/texture field features using the prior network, which are used as the optimization targets to search its corresponding shape/texture latent codes in the latent space. Note that we use shape/texture field features instead of the reference image when conducting this inversion. When the optimization is done, a textured human model resembling its reference image is produced. Compared to previous 3D reconstruction methods (e.g. PIFu), we can easily manipulate

Figure 10: Three inversion examples from given reference images. The reference images are shown on the left, inversely optimized human models are shown in different views in the middle, several re-textured models are shown on the right.

these results through latent space editing (e.g. re-texturing, interpolation).

5. Conclusion

In the paper, We introduced Get3DHuman, a novel 3D human generator that is able to synthesize diverse and high-quality clothed 3D humans. It utilizes the priors of the well-trained 2D human generator and 3D reconstructor. Numerous experiments have shown that our method greatly outperforms other methods and can support a wide range of applications, including shape interpolation, re-texturing, and single-view reconstruction via latent inversion.

Limitations. Our method is only able to synthesize simple standing-pose models, which are restricted by the image synthesis generator and reconstruction prior in our prior networks.

6. Acknowledgement

The work was supported in part by NSFC with Grant No. 62293482, the Basic Research Project No. HZQB-KCZYZ-2021067 of Hetao ShenzhenHK S&T Cooperation Zone, the National Key R&D Program of China with grant No. 2018YFB1800800, by Shenzhen Outstanding Talents Training Fund 202002, by Guangdong Research Projects No. 2017ZT07X152 and No. 2019CX01X104, by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), and by Shenzhen Key Laboratory of Big Data and Artificial Intelligence (Grant No. ZDSYS201707251409055). It was also partially supported by NSFC62172348, Outstanding Yound Fund of Guangdong Province with No. 2023B1515020055 and Shenzhen General Project with No. JCYJ20220530143604010.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *International conference on machine learning*, 2018.

[2] Badour AlBahar, Jingwan Lu, Jimei Yang, Zhixin Shu, Eli Shechtman, and Jia-Bin Huang. Pose with Style: Detail-preserving pose-guided image synthesis with conditional stylegan. *ACM TOG*, 2021.

[3] Thiemo Alldieck, Marcus A. Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. *CoRR*, 2018.

[4] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape completion and animation of people. *ACM TOG*, 2005.

[5] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016.

[6] 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.

[7] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019.

[8] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *SIGGRAPH*, 2015.

[9] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 145–156, 2000.

[10] Junseok Kwon Dong Wook Shu, Sung Woo Park. 3d point cloud generative adversarial network based on tree structured graph convolutions. *arXiv*, 2019.

[11] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4d: Real-time performance capture of challenging scenes. *SIGGRAPH*, 2016.

[12] Anna Frühstück, Krishna Kumar Singh, Eli Shechtman, Niloy J. Mitra, Peter Wonka, and Jingwan Lu. InsetGAN for full-body image generation. In *CVPR*, 2022.

[13] 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. *arXiv preprint*, 2022.

[14] 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. In *NeurIPS*, 2022.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *NeurIPS*, 2014.

[16] Fangzhou Hong, Zhaoxi Chen, Yushi Lan, Liang Pan, and Ziwei Liu. EVA3D: Compositional 3D human generation from 2d image collections. *arXiv*, 2022.

[17] Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *CVPR*, 2021.

[18] Suyi Jiang, Haoran Jiang, Ziyu Wang, Haimin Luo, Wenzheng Chen, and Lan Xu. HumanGen: Generating human radiance fields with explicit priors. *arXiv*, 2022.

[19] Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. Talk-to-edit: Fine-grained facial editing via dialog. In *ICCV*, 2021.

[20] Yuming Jiang, Shuai Yang, Haonan Qiu, Wayne Wu, Chen Change Loy, and Ziwei Liu. Text2Human: Text-driven controllable human image generation. *ACM TOG*, 2022.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV 2016*, pages 694–711. Springer, 2016.

[22] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018.

[23] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

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

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

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

[27] Christoph Lassner, Gerard Pons-Moll, and Peter V. Gehler. A generative model for people in clothing. In *ICCV*, 2017.

[28] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570. PMLR, 2015.

[29] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM TOG (SIGGRAPH Asia)*, 2021.

[30] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *CVPR*, 2020.

[31] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM TOG*, 2015.

[32] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NeurIPS*, 2017.

[33] Qianli Ma, Shunsuke Saito, Jinlong Yang, Siyu Tang, and Michael J. Black. SCALE: Modeling clothed humans with a surface codec of articulated local elements. In *CVPR*, 2021.

[34] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to Dress 3D People in Generative Clothing. In *CVPR*, 2020.

[35] Qianli Ma, Jinlong Yang, Siyu Tang, and Michael J. Black. The power of points for modeling humans in clothing. In *ICCV*, 2021.

[36] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[37] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, 2019.

[38] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. StructureNet: Hierarchical graph networks for 3D shape generation. *ACM TOG*, 2019.

[39] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[40] Thu H Nguyen-Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy Mitra. BlockGAN: Learning 3D object-aware scene representations from unlabelled images. In *NeurIPS*, 2020.

[41] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *ICCV*, 2021.

[42] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *3DV*, 2018.

[43] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.

[44] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, 2019.

[45] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, 2021.

[46] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. Dyna: A model of dynamic human shape in motion. *ACM TOG*, 2015.

[47] RenderPeople. https://renderpeople.com/.

[48] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *SIGGRAPH*, 2021.

[49] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. *arXiv preprint arXiv:1905.05172*, 2019.

[50] 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.

[51] Kripasindhu Sarkar, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Humangan: A generative model of humans images, 2021.

[52] 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 *NeurIPS*, 2021.

[53] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5541–5550, 2017.

[54] Shih-Yang Su, Frank Yu, Michael Zollhoefer, and Helge Rhodin. A-nerf: Surface-free human 3D pose refinement via neural rendering. *arXiv preprint arXiv:2102.06199*, 2021.

[55] Zhuo Su, Lan Xu, Zerong Zheng, Tao Yu, Yebin Liu, and Lu Fang. Robustfusion: Human volumetric capture with data-driven visual cues using a rgbd camera. In *European Conference on Computer Vision*, pages 246–264. Springer, 2020.

[56] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *NeurIPS*, 2016.

[57] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3D reconstruction. In *NeurIPS*, 2019.

[58] Ilya Zakharkin, Kirill Mazur, Artur Grigorev, and Victor Lempitsky. Point-based modeling of human clothing. In *ICCV*, 2021.

[59] 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. In *Arxiv:2208.00561*, 2022.

[60] Xin-Yang Zheng, Yang Liu, Peng-Shuai Wang, and Xin Tong. Sdf-stylegan: Implicit sdf-based stylegan for 3D shape generation. In *Computer Graphics Forum*, 2022.

[61] Linqi Zhou, Yilun Du, and Jiajun Wu. 3D shape generation and completion through point-voxel diffusion. In *ICCV*, 2021.