GarmentDreamer: 3DGS Guided Garment Synthesis with Diverse Geometry and Texture Details

Boqian Li[1,2*] Xuan Li[1*] Ying Jiang[1,3*] Tianyi Xie[1] Feng Gao[4†]
Huamin Wang[5] Yin Yang[2] Chenfanfu Jiang[1]
[1]UCLA, [2]University of Utah, [3]HKU, [4] Amazon, [5]Style3D Research

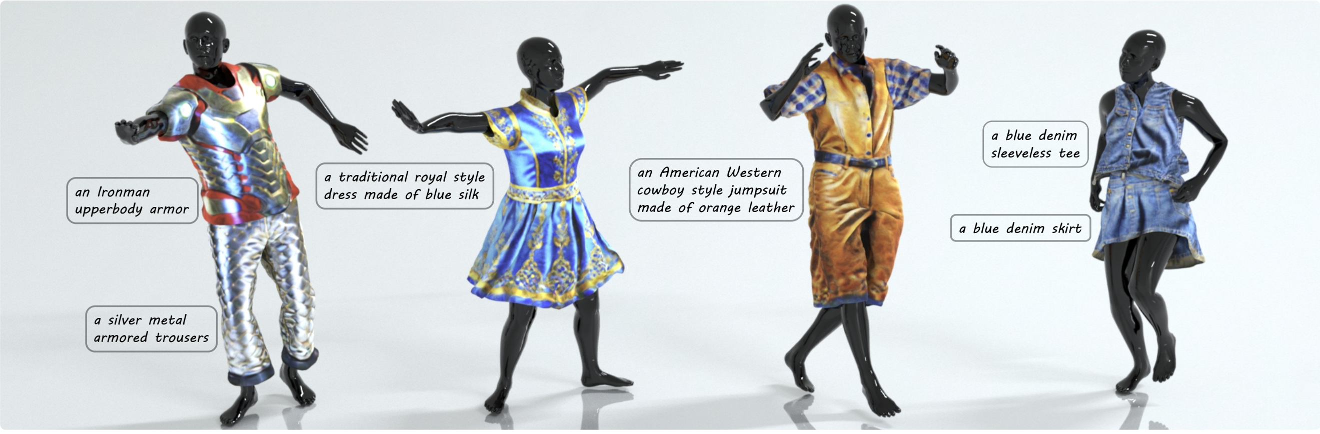
an Ironman upperbody armor
a silver metal armored trousers
a traditional royal style dress made of blue silk
an American Western cowboy style jumpsuit made of orange leather
a blue denim sleeveless tee
a blue denim skirt

Figure 1. GarmentDreamer is a garment synthesis framework for customizing simulation-ready high-quality textured garment meshes from text prompts.

Abstract

Traditional 3D garment creation is labor-intensive, involving sketching, modeling, UV mapping, and texturing, which are time-consuming and costly. Recent advances in diffusion-based generative models have enabled new possibilities for 3D garment generation from text prompts, images, and videos. However, existing methods either suffer from inconsistencies among multi-view images or require additional processes to separate cloth from the underlying human model. In this paper, we propose GarmentDreamer, a novel method that leverages 3D Gaussian Splatting (GS) as guidance to generate wearable, simulation-ready 3D garment meshes from text prompts. In contrast to using multi-view images directly predicted by generative models as guidance, our 3DGS guidance ensures consistent optimization in both garment deformation and texture synthesis. Our method introduces a novel garment augmentation module, guided by normal and RGBA information, and employs implicit Neural Texture Fields (NeTF) combined with Variational Score Distillation (VSD) to generate diverse geometric and texture details. We validate the effectiveness of our approach through comprehensive qualitative and quantitative experiments, showcasing the superior performance of GarmentDreamer over state-of-the-art alternatives[1].

1. Introduction

The creation of 3D digital garments is crucial in graphics and vision, driven by their extensive applications in fashion design, virtual try-on, gaming, animation, virtual reality, and robotics. Nevertheless, the conventional pipeline for 3D garment creation which encompasses sketching and modeling, followed by UV mapping, texturing, shading and simulation using commercial software [11, 38, 53] demands substantial manual effort. This process results in significant time and labor costs.

With the advancement of diffusion-based generative models [36, 41], 3D garment generation from text and images has flourished. Two primary methods have emerged. The first method, as explored in prior works [16, 35], starts by reconstructing 2D sewing patterns and subsequently

*Equal contributions.

†This work is not related to F. Gao's position at Amazon.

[1] Demos and codes are available at https://xuan-li.github.io/GarmentDreamerDemo/

generating 3D garments from these patterns. The second method involves generative models that directly predict the distribution of 3D target shapes based on image and text inputs[46, 52, 55, 65]. However, the former approach necessitates a vast amount of paired training data between sewing patterns and corresponding text or images [35]. The latter approach, while simpler, encounters issues such as multi-view inconsistency [46] and lacks high-fidelity details, which requires additional post-processing for downstream simulation tasks [26, 31, 32, 64]. Thus, generating simulation-ready, textured garments with high-fidelity details remains challenging.

Recognizing the advantages and limitations of both traditional pipeline and modern generative models, our goal is to create high-fidelity, simulation-ready textured garments with appropriately placed openings for the head, arms, and legs. We aim to achieve details comparable to that of the traditional pipeline. In this paper, we focus on *full-piece garment generation* without relying on 2D sewing patterns, which is sufficient for many graphics applications.

To achieve our goals, we leverage image/text-conditioned diffusion models. Several challenges revealed by prior methods must be addressed: (1) Many avatar generators [20, 33, 62] directly produce fused cloth-human models with watertight meshes. They require separation from humans and complex modifications to introduce openings for the head, arms, and legs for downstream tasks. (2) While some high-quality non-watertight garments are generated by deforming template meshes guided by multi-view images [46], predicting unsigned distance fields (UDF) via diffusion models [65], or optimizing meshes through differentiable simulators [32], these approaches often lack detailed and realistic geometrical features or complex textures. This limitation stems from the inherent challenges in 3D shape diffusion. (3) Deforming garment geometry solely based on multi-view images predicted by diffusion models can lead to inconsistency [8]. Additionally, refining textures in UV space can result in over-saturated, blocky artifacts [5, 55].

To address these challenges, we introduce GarmentDreamer, a 3DGS [22] guided garment synthesis method for simulation-ready, wearable garments featuring diverse geometry and intricate textures. We first leverage diffusion models and physical simulations to obtain a smooth garment template and generate corresponding Gaussian kernels via Score Distillation Sampling (SDS) loss [55]. Subsequently, we exploit the estimated normal map and RGBA information from 3DGS as guidance in our garment augmentation module to deform meshes using a coarse-to-fine optimization approach. In the coarse stage, we refine garment contour together with neck, arm, waist, and leg openings, and then in the fine stage, multi-scale details are created under the proposed guidance. Compared with guidance from generated multi-view images, multi-view consistent guidance extracted by Gaussian kernels creates more high-quality geometry and texture details. An implicit Neural Texture Field (NeTF) is then reconstructed and subsequently augmented by Variational Score Distillation (VSD) loss to offer high-quality garment textures. Compared with baking Gaussian kernels into a UV map directly, our texture extraction strategy offers more consistent results. Our contributions include:

- A novel 3D garment synthesis framework that integrates diffusion models with 3D Gaussian Splatting (3DGS) to generate wearable garments from text prompts.
- A new garment mesh deformation module using normal-based and RGBA-based guidance provided by 3DGS in course-to-fine stages to generate diverse garments with geometrical details.
- An effective texture reconstruction and fine-tuning strategy utilizing implicit Neural Texture Fields (NeTF) to generate high-quality garment textures.
- Comprehensive qualitative and quantitative experiments to evaluate the superior performance of GarmentDreamer as compared to prior methods.

2. Related Work

Diffusion-based 3D Generation For 3D generation, many work distill 2D pre-trained diffusion models via SDS loss [41, 69] and Variational Score Distillation (VSD) [59, 67], or exploit 3D diffusion models to directly generate 3D representations such as point cloud [37], Neural Radiance Fields (NeRF) [18, 50], mesh [36, 42], SDF [10, 49], Unsigned Distance Field (UDF) [65], DMTets [47], and 3D Gaussian Splatting (GS) [55, 63]. To capture rich surface details and high-fidelity geometry of generated 3D shapes, normal maps [20, 30, 34, 36], depth maps [43], pose priors [66], skinned shape priors [23] have been adopted as guidance modules for 3D digital avatar [21, 32, 62], garment [16, 46], and scene synthesis tasks [27, 54, 56].

3D Garment Synthesis Traditional 3D garment creation usually begins with 2D sewing patterns in commercial fashion design software, necessitating significant labor and time costs [46]. To automate 3D garment generation, learning-based methods have been employed to infer garment shapes from text prompts, images, and videos [4, 6, 12, 16, 17, 24, 32, 35, 46, 48, 52, 61, 65, 68]. However, many methods focused on clothed human synthesis [6, 19, 57, 61, 68] typically generate garments fused together with digital human models, which restricts them to basic skinning-based animations and requires nontrivial work to separate the garments from the human body. In contrast, our work focuses on separately wearable geometry. Other closely related works include [32] which also generates high-quality simulation-ready clothes at the expense of creating clothing templates by artists and precise point clouds by scanners.

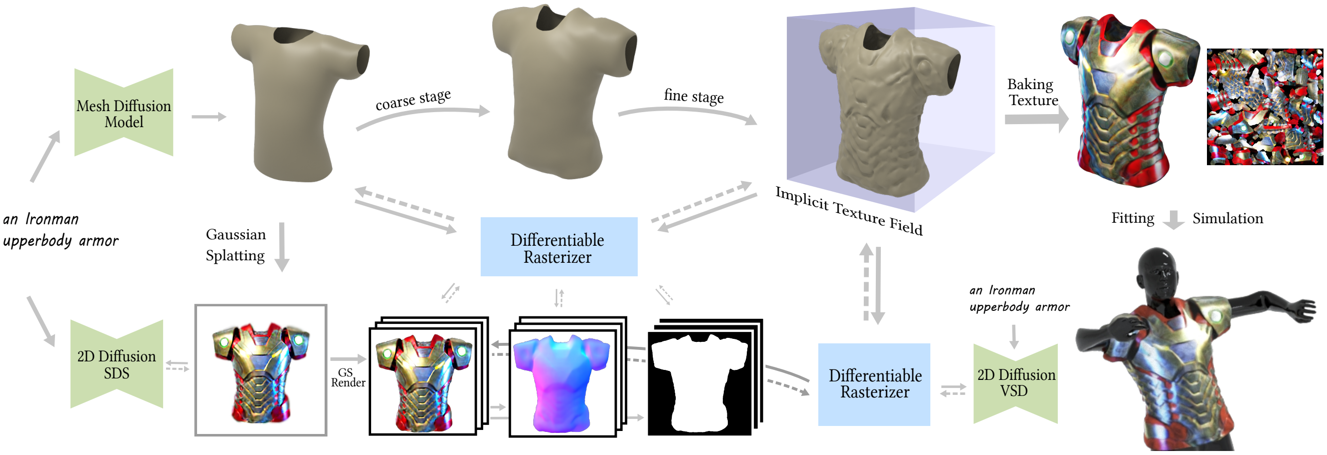

Figure 2. Starting with text prompts, we first generate a garment template mesh $\mathcal{F}_t$ using a diffusion model. We then optimize a 3DGS using the template $\mathcal{F}_t$ and text $\mathcal{T}$ as guidance. The template mesh $\mathcal{F}_t$ is refined in a two-stage process guided by 3DGS, utilizing RGBA, normal map, and mask losses to achieve the final shape $\mathcal{F}_g$ with enriched geometric details. Finally, we reconstruct and optimize an implicit texture field Φ via VSD, producing high-quality textured garment meshes suitable for downstream simulation/animation tasks.

Some recent methods [16, 35] generate non-watertight garments with sewing patterns, while ours generates a one-piece cloth similar to Sarafianos et al. [46]. Although one-piece 3D clothing is not suitable for manufacturing, it is still sufficient for many graphics downstream tasks. Note that [46] develops methods to improve multi-view consistency in image generation similarly to [36], while our method mitigates this challenge through a novel 3DGS guidance.

Garment Refinement Garment refinement involves diversification [23], draping [12], wrinkle generation [7, 40], stylization [46] and other techniques to enhance diversity and realism. Traditional methods optimize energy or geometric constraints to directly edit meshes [3, 51]. Recent learning-based approaches modify latent geometry and textures, potentially with conditioned diffusion [12, 16, 23, 52], and decode latent features to generate refined meshes. To modify garment meshes from text prompts, Textdeformer [14] represents mesh deformation through Jacobians and exploits global CLIP features as guidance. Garment3DGen [46] deforms garment geometry using 2D image guidance. However, these RGB-based or CLIP-based methods have a tendency to prefer modifying 2D textures over 3D geometric structures. On the other hand, Surf-D [65] edits garment geometry with sketch conditions but lacks texture synthesis. Kim et al. [23] utilize normal maps as references to carve clothed humans. Worchel et al. [60] exploit neural deferred shading results to modify mesh surfaces. Inspired by these works [23, 60], our approach explores both normal maps and RGBA image features with neural deferred shading to guide mesh deformation, creating realistic garment geometry and textures that can be directly applied to downstream simulation/animation tasks.

3. Method

Fig. 2 overviews our method. Given text $\mathcal{T}$, GarmentDreamer generates wearable textured garment meshes $\mathcal{F}_g$ with neck, arm, waist, and leg openings. Starting with $\mathcal{T}$, we generate garment template $\mathcal{F}_t$ based on predicted UDFs in § 3.1. In § 3.2, we optimize a 3DGS representation based on $\mathcal{T}$ and $\mathcal{F}_t$ and then design a 3DGS-guided two-stage training to refine $\mathcal{F}_t$ into the final garment shape $\mathcal{F}_g$, increasing the mesh diversity and introducing geometric details. Finally, we generate high-quality textures by optimizing an implicit Neural Texture Field (NeTF) Φ augmented with variational SDS loss in § 3.3.

3.1. Garment Template Mesh Generation

Human clothing generation can utilize strong inductive biases, as garments within the same category tend to have similar topology and overall orientation. Thus, it is reasonable to warm-start our optimization using a template and modify it. The starting point is to generate a template mesh $\mathcal{F}_t$ from text $\mathcal{T}$. Using meshes from Cloth3D [1] and Sew-Factory [35], we train a mesh diffusion model on a dataset containing 15 different categories of garments, each with over 100 meshes, which is sufficient for generating common template garment meshes. Leveraging the efficiency and effectiveness of generation in latent space [45], we represent the geometry information of the garment mesh in a compact vector form by extracting its latent code.

Simulation-based Preprocessing A straightforward approach to obtaining the garment latent space is to train an autoencoder with high-quality garment meshes [65]. However, through experiments, we found the autoencoder struggles to capture high-frequency geometric details, such as overly dense wrinkles. Additionally, some meshes in the dataset exhibit self-intersections due to these noisy details, resulting in unsatisfactory reconstruction. We believe these high-frequency details are not only challenging for the autoencoder to learn but also redundant for the purpose of template generation. Recognizing that we only need a warm-start template and will obtain geometric details in later phases, we propose a simulation-based data preprocessing step that involves smoothing the garment meshes using a physics-based cloth simulator. Specifically, we set the rest bending angle between every two adjacent triangles to zero. By minimizing the bending energy in quasi-static Finite Element Method (FEM) simulation steps [28, 29] together with a Neo-Hookean stretching energy, each garment mesh transforms to a rest state with minimal high-frequency wrinkles, without altering the overall characteristic shape.

Garment Latent Space To encode the 3D garment template meshes $\mathcal{F}_t$ into latent space, we utilize the Dynamic Graph CNN (DGCNN) [58], producing a 64-dimensional garment latent code γ. To reconstruct the garment geometry from the latent code, we employ a Multilayer Perceptron (MLP) with Conditional Batch Normalization [13] as the decoder. This decoder processes the latent code alongside a set of query points, which are sampled from the input meshes and their surroundings during training, and randomly sampled in a canonicalized space during testing. The decoder then predicts the UDF values for these points. We use both distance loss and gradient loss as suggested by De Luigi et al. [12] in training the autoencoder.

Latent Diffusion Building upon the garment latent space, we train a latent diffusion model to predict the latent code γ conditioned on the garment category specified in the text prompt $\mathcal{T}$, such as a skirt or T-shirt. Using the garment decoder alongside a set of query points, we obtain the garment UDF field. Finally, the desired garment template mesh $\mathcal{F}_t$ is extracted from the predicted UDF using MeshUDF [15].

3.2. Garment Geometry Deformer

While the template mesh from the previous step provides a basic structure, it lacks detailed geometry and is constrained by the dataset's limited diversity. Our next step is to introduce greater diversity and enrich the mesh with more geometric details. To achieve this, we first generate appropriate guidance models to enhance the mesh. While prior work has utilized multi-view image generators, they suffer from risks of inconsistency [8]. To mitigate these risks, we turn to 3DGS representations. We utilize a two-stage training process to optimize the geometry of a garment based on the multi-view guidance of 3DGS. In the first, coarse stage, masks are used to refine the garment's contour. The second, fine stage uses RGB renderings and normal maps to add local details. This 3DGS-guided process applies various displacements to the garment surface without compromising wearability, significantly enhancing the diversity and introducing fine geometric details for better visual quality.

3.2.1 3D Gaussian Generation

We adopt 3DGS [22] to guide the deformation of garment geometry, which exploits 3D anisotropic Gaussian kernels to reconstruct 3D scenes with learnable mean μ, opacity σ, covariance Σ, and spherical harmonic coefficients $\mathcal{S}$. We utilize the garment template mesh $\mathcal{F}_t$ with the same text prompt $\mathcal{T}$ offering color, material, and pattern descriptions to generate 3D Gaussian kernels for further guiding geometry refinement and texture generation. Similar to the query points used in the garment latent code decoder, the initial 3D Gaussians are randomly sampled at the surroundings of the template mesh surface. We optimize Gaussian kernels using SDS loss [41] with a frozen 2D diffusion model ϕ conditioning on text prompts $\mathcal{T}$. The rendered image is produced by a differentiable renderer g with the parameters of 3DGS θ, notated as $\mathbf{x} = g(\theta)$. The formula for computing the gradient to guide the updating direction of θ is: $\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x} = g(\theta)) \triangleq \mathbb{E}_{t,\epsilon}\left[w(t)\left(\hat{\epsilon}_\phi\left(\mathbf{z}_t; \mathcal{T}, t\right) - \epsilon\right)\frac{\partial \mathbf{x}}{\partial \theta}\right]$, where $\hat{\epsilon}_\phi\left(\mathbf{z}_t; \mathcal{T}, t\right)$ is the score estimation function, predicting the sampled noise $\hat{\epsilon}$ given the noisy image $\mathbf{z}_t$, text prompt $\mathcal{T}$, and noise level t; $w(t)$ is a weighting function. We render RGB images and masks of Gaussian kernels from 24 views for garment geometry deformation. The masks are generated using a step function with an empirical threshold ϑ applied to the opacity $\sigma \in [0, 1]$ of each Gaussian kernel.

3.2.2 Coarse Stage

The objective of the coarse stage is to deform the mesh to align with the overall shape of the 3DGS. To achieve this, given the template mesh $\mathcal{F}_t$ and a camera view C_i, we use a differentiable rasterizer to generate the mask and use the following mask loss to guide the contour optimization:

$$\mathcal{L}_{\text{M}}(\boldsymbol{v}) = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \text{MSE}(R_M(\mathcal{F}_t, C_i), M_i), \tag{1}$$

where the optimization variable $\boldsymbol{v}$ is the concatenation of all vertex positions, $\mathcal{I}$ is the set of camera views, R_M is the differentiable contour rasterizer empowered by Nvdiffrast [25], and $\{M_i\}$ is the ground truth mask guidance.

Optimization under merely $\mathcal{L}_{\text{M}}$ is stochastic and unstable. One reason is that the vertices inside the masks can move freely without changing the output mask. Considering that the template mesh is smooth, we can maintain the smooth surface and stabilize the deformation process during the coarse stage by imposing constraints on the surface curvature. This is achieved through a normal-consistency loss $\mathcal{L}_{\text{NC}}$ and a Laplacian loss $\mathcal{L}_{\text{L}}$:

$$\mathcal{L}_{\text{NC}}(\boldsymbol{v}) = \frac{1}{N}\sum_{j\sim k}(1 - \boldsymbol{n}_j \cdot \boldsymbol{n}_k)^2, \;\; \mathcal{L}_{\text{L}}(\boldsymbol{v}) = \frac{1}{M}\sum_{j\sim k} w_{jk}\|\boldsymbol{v}_j - \boldsymbol{v}_k\|^2, \tag{2}$$

where $\boldsymbol{n}_i, \boldsymbol{n}_j$ are adjacent face normals, N is the number of adjacent face pairs, $\boldsymbol{v}_i, \boldsymbol{v}_j$ are adjacent vertex positions, M

is the number of adjacent vertex pairs, and $\{w_{jk}\}$ are the Laplacian edge weights.

With these proposed loss terms, the coarse stage can deform the mesh surface to align with 3DGS while preserving the hole region. Furthermore, the boundaries of the hole regions coincide with the boundaries of the garment rendered by Gaussian rendering, which serves as good intermediate results. By utilizing the deformed hole regions, we can continue to preserve the openings and the garment's wearability in the next deformation stage.

3.2.3 Fine Stage

While contour mask guidance optimizes the garment mesh to match the overall shape of the generated 3DGS, it does not encourage the generation of local geometric details characterized by local displacement variations, as such information is not available in masks. To generate realistic local details of garment geometry, we exploit rich visual information from the RGB renderings of 3DGS to deform the garment mesh after the coarse stage.

Inspired by Worchel et al. [60], we propose using a neural shader module to utilize the RGB information to enrich geometry details, which is an implicit shading field $S(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{d})$ that maps a query position along with its normal and view direction to an RGB color. The module is combined with the same differentiable rasterizer R to render RGB images. Given a camera view C_i with the camera center $\boldsymbol{c}_i$, we jointly optimize the shader's parameters θ and garment vertices $\boldsymbol{v}$ using the following RGB loss:

$$\mathcal{L}_{\mathrm{RGB}}(\boldsymbol{v}, \theta) = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \mathrm{L}_1(S(\tilde{\boldsymbol{v}}_{jk}, \tilde{\boldsymbol{n}}_{jk}, \tilde{\boldsymbol{v}}_{jk} - \boldsymbol{c}_i), (I_i)_{jk}), \tag{3}$$

where $\{I_i\}$ is the ground truth RGB images rendered from the generated 3DGS, $\tilde{\boldsymbol{n}} = R(\{\boldsymbol{n}_j\}, \mathcal{F}_t, C_i)$ and $\tilde{\boldsymbol{v}} = R(\{\boldsymbol{v}_j\}, \mathcal{F}_t, C_i)$ are rasterized vertex normals and positions, respectively.

Simultaneously, we observed that the neural shader sometimes brings noise and unnecessary patterns from texture into geometry, like the light variations on the armor and logos on clothes. To address this, we need to maintain the necessary geometry details like wrinkles while removing the noise. We propose using normal estimation models to obtain estimated normal maps used for additional guidance, which can capture wrinkles and the overall normal information of the garment. The ground truth normal maps $\{N_i\}$ are inferred from the rendered RGB images from the generated 3DGS by a pre-trained normal estimator. We then use the following normal loss to guide the geometry optimization:

$$\mathcal{L}_{\mathrm{N}}(\boldsymbol{v}) = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \mathrm{L}_1(R(\{\boldsymbol{n}_j\}, \mathcal{F}_t, C_i), N_i), \tag{4}$$

where $\{\boldsymbol{n}_j\}$ is the set of vertex normals.

Furthermore, the normals at the garment hole regions are not reliable since the renderings from the 3DGS at these regions are blurry and directly applying the normal map at the hole regions to the mesh deform process can result in the closure of the original openings. We propose using the coarse-stage mesh as the hole guidance to maintain the openings. We detect the hole region by checking the dot product between the garment surface normal and the camera direction. Specifically, the hole region mask within a rendered image is defined as

$$(\tilde{M}_i^H)_{jk} = \tilde{\boldsymbol{n}}_{jk} \cdot (\tilde{\boldsymbol{v}}_{jk} - \boldsymbol{c}_i) > 0. \tag{5}$$

We use the following hole loss to maintain the holes initially present in the template mesh after the coarse stage:

$$\mathcal{L}_H(\boldsymbol{v}) = \frac{1}{|\mathcal{I}|}\sum_{i\in I} \mathrm{MSE}(\tilde{M}_i^H, M_i^H), \tag{6}$$

where M_i^H is the hole mask at the beginning of the fine stage. However, the mask values are boolean, which are not differentiable. We manually skip the gradient of the binarization by letting

$$\frac{\partial \mathcal{L}^H}{\partial\{\tilde{\boldsymbol{n}}_{jk} \cdot (\tilde{\boldsymbol{v}}_{jk} - c_i)\}} = \frac{\partial \mathcal{L}^H}{\partial(\tilde{M}_i^H)_{i,j}}, \tag{7}$$

which can provide correct gradient directions.

In summary, the loss function for the fine stage is the weighted sum of $\mathcal{L}_{\mathrm{M}}, \mathcal{L}_{\mathrm{NC}}, \mathcal{L}_{\mathrm{L}}, \mathcal{L}_{\mathrm{N}}, \mathcal{L}_{\mathrm{H}}, \mathcal{L}_{\mathrm{RGB}}$. These losses not only align the geometry with the 3DGS appearance but also preserve the garment openings, ensuring wearability and simulation-ready features.

3.3. Texture Synthesis

The final step of GarmentDreamer is to generate high-quality, detailed textures for garment meshes. Directly using vertex colors from the neural shader is insufficient due to the limited vertex count. Instead, we propose using a Neural Texture Field (NeTF) $\Phi(\boldsymbol{x})$, which exploits a hash grid encoder to map the xyz coordinates of any position on the mesh to RGB, enabling high-resolution texturing. Furthermore, we use UV unwrapping to map the 3D coordinates to 2D UV space, forming a connection pathway of 2D coordinates - 3D coordinates - RGB. Rather than relying on vertex colors, we use the multiview images rendered by 3DGS as guidance to fit the NeTF, ensuring high-quality texture preservation even with a limited number of vertices. We optimize NeTF using the following loss function:

$$\mathcal{L}_{\mathrm{T}}(\omega) = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \mathrm{L}_1(\Phi(\tilde{\boldsymbol{p}}_{jk}), (I_i)_{jk}), \tag{8}$$

where ω is the parameters of the NeTF Φ and $\tilde{\boldsymbol{p}} = R(\{\boldsymbol{p}_j\}, \mathcal{F}_t, C_i)$ represents the positions on the mesh that have been rasterized.

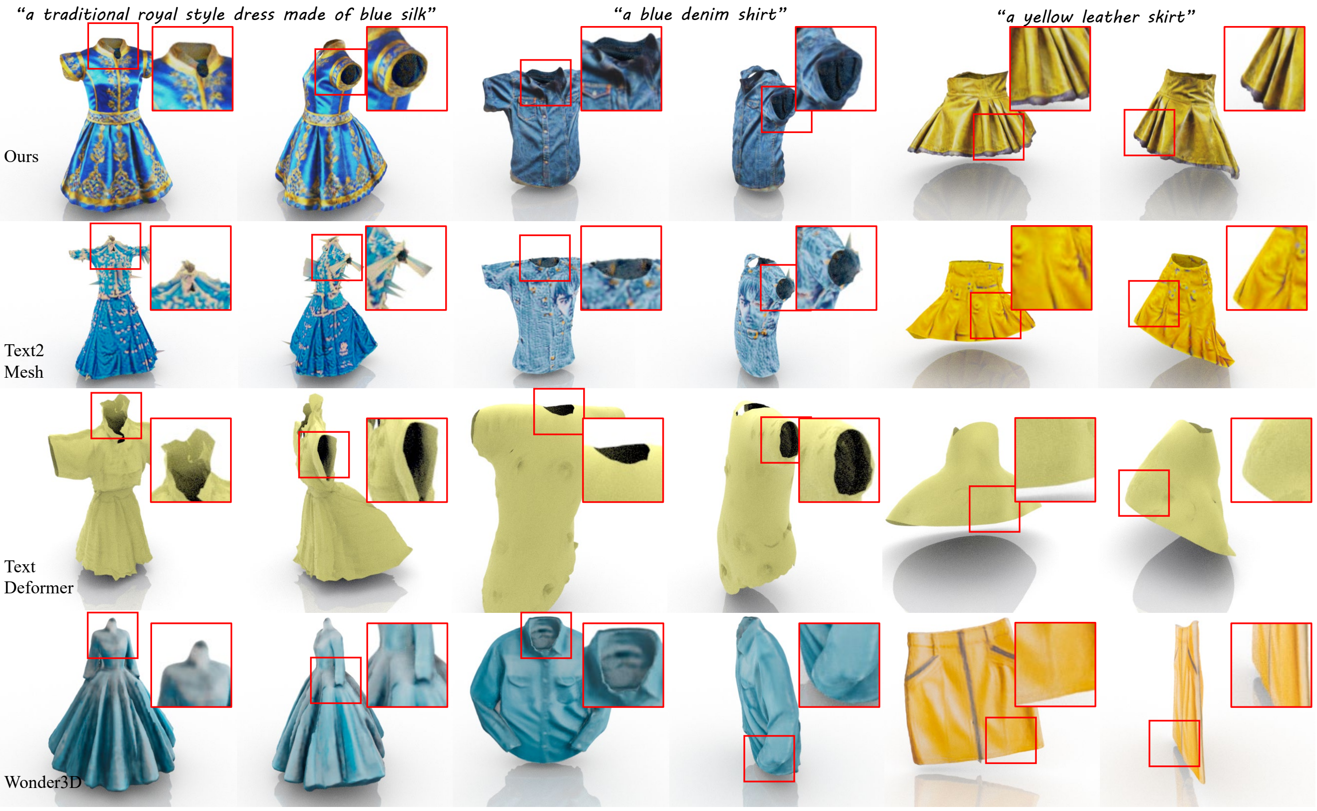

Figure 3. **Qualitative Comparisons**. While baseline methods either produce unrealistic geometric artifacts, e.g. spikes and excessive smoothness, or non-garment textures, GarmentDreamer excels in generating high-quality, simulation-ready non-watertight garments with detailed textures and fine wrinkle details.

For better visual quality and enhanced texture details, we utilize Variational Score Distillation (VSD) [59] to fine-tune the implicit texture fields. This process involves forwarding the NeTF to project a 2D image from a random view and feeding this projected image into the VSD framework to compute a perceptual loss. By backpropagating this loss, we optimize the parameters of NeTF implicitly and improve the overall texture quality. After reconstruction and fine-tuning, we can easily query the color of any mesh point $\boldsymbol{p}$, which is baked onto a texture map.

4. Experiments

In this section, we conduct a thorough evaluation of GarmentDreamer across various garment categories, providing both quantitative and qualitative comparisons with other state-of-the-art 3D generation methods. We also present ablation studies to highlight the effectiveness of the key components in our pipeline. About more showcases, please see details in supplementary material.

4.1. Comparison

We compare GarmentDreamer against several state-of-the-art 3D generation methods: Text2Mesh [39], TextDeformer [14], and Wonder3D [36]. Text2Mesh and TextDeformer aim to optimize and deform an initial mesh to the desired shape indicated in the text prompt. We use the same template mesh for GarmentDreamer and these deformation-based methods to ensure fairness. To compare our method with Wonder3D, which reconstructs 3D meshes from single-view images, we use DALLE-3 [2] to generate the images from text prompts as its input. To comprehensively compare GarmentDreamer with these three methods, we generate 21 distinct types of garments with each method, including shirts, dresses, skirts, and pants.

Table 1. **Quantitative Comparisons.** Our approach outperforms deformation-based and generative methods on both FashionCLIP similarity score (FCSS) which is pretrained on fashion datasets and vanilla OpenAI CLIP similarly score (CSS), supports generating openings on clothing as well as textures, and runs faster than prior deformation-based methods.

Methods	FCSS	CSS	Wearable	Texture	Runtime
Text2Mesh	0.3396	0.2655	✓	✓	∼ 20 mins
TextDeformer	0.2367	0.1657	✓		∼ 35 mins
Wonder3D	0.3402	0.2509		✓	∼ 4 mins
Ours	0.3413	0.2731	✓	✓	∼ 15mins

4.1.1 Quantitative Comparison

In the absence of a standardized metric for 3D generation quality, we focus on measuring the consistency between the

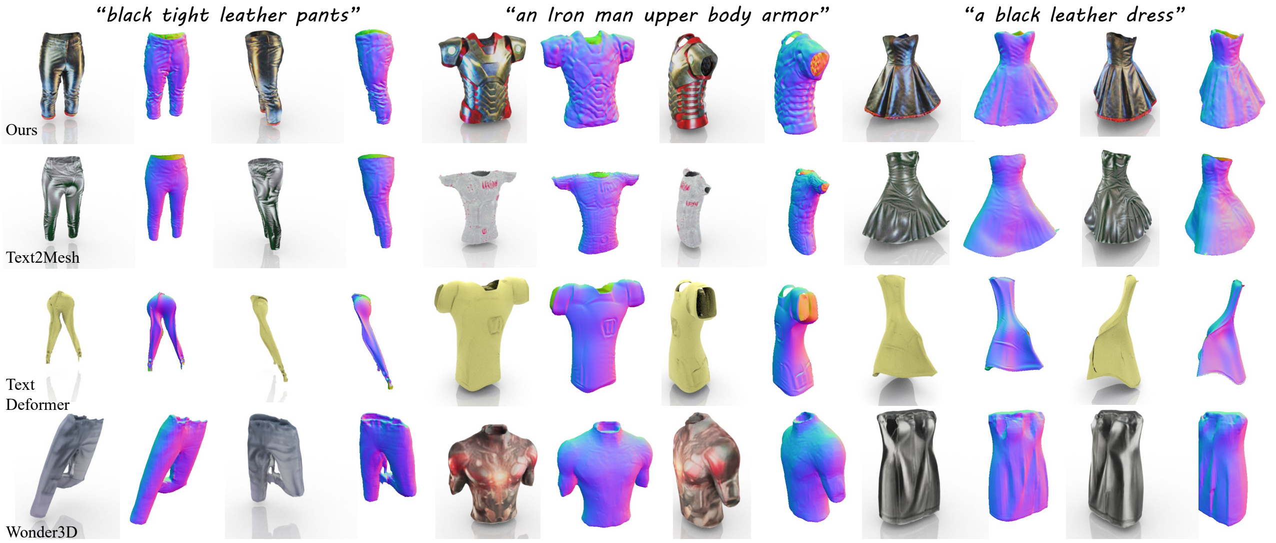

Figure 4. **Normal Comparisons**. We visualize the normal maps for a better comparison of garment geometry between GarmentDreamer and other methods. Our proposed method generates visually plausible garment meshes, featuring finer geometric details such as natural wrinkles and smooth boundaries.

generated garments and their text prompt inputs. We render each garment from 36 different views and compute the average CLIP similarity score between these rendered images and the corresponding text prompts. Given that TextDeformer does not handle textures, we render its results using a default color. Typically, text-to-2D/3D works utilize the vanilla CLIP model [44] for evaluating text-to-image alignment. However, this model is optimized for general subjects and lacks specificity for garment evaluation. To address this, we employ FashionCLIP [9], a model similar to CLIP but fine-tuned on a fashion dataset, making it more appropriate for our purposes.

We report the comparison results in Table. 1, including both the FashionCLIP Similarity Score (FCSS) and the CLIP Similarity Score (CSS). Our method outperforms all baselines in terms of text-garment alignment. Furthermore, our approach demonstrates faster performance compared to the two deformation-based baselines.

4.1.2 Qualitative Evaluation

GarmentDreamer ensures that the garment meshes maintain their geometric integrity and exhibit rich, detailed textures, making them suitable for high-quality visual applications. We visualize the results of GarmentDreamer and other baselines in Fig. 3 and Fig. 4. The meshes produced by Text2Mesh appear distorted with spiky artifacts due to the direct optimization of all vertex coordinates. TextDeformer alleviates this issue by parameterizing deformation as Jacobians, but it fails to capture high-frequency details, causing overly smooth geometry. Wonder3D relies heavily on input images and generates garments with closed sleeves or necklines due to limited garment-specific knowledge. In contrast, our method produces wearable, simulation-ready garments with realistic textures, enabling seamless downstream tasks like animation and virtual try-on.

Additionally, by observing Fig. 4, it can be seen that our method achieves the best geometric-texture alignment: for the wrinkles of the dress and pants, as well as the armor engraved designs, our results are not only displayed in the texture but also reflected at the geometric level thanks to the capability of the garment geometry deformer module. This makes the rendered results more realistic, as can be clearly observed from the side edges of the clothing. We also compare with Garment3DGen [46] and WordRobe [52] using the same text prompts; see Fig. 7. Cargo shorts generated by [52] have fewer details in grain lines, hemline, seam allowance, and side seams than ours. Armor created by Garment3DGen [46] aligns textures with image guidance, while ours generates detailed high-quality geometric structures.

4.2. Ablation

In Fig. 6 and Fig. 5, we ablate key components in GarmentDreamer, using the same 21 generated garments in § 4.1.

Hole Loss We first examine the effectiveness of the proposed hole loss $\mathcal{L}_H$, a crucial component of our proposed 3DGS-guided mesh deformer, which ensures clean edges of the openings and, accordingly, wearability. Without this, mesh deformation tends to enclose the arm and head holes, resulting in unwearable garment meshes.

Normal Loss Normal loss $\mathcal{L}_N$ ensures noise-free meshes with necessary geometry details. As shown in Fig. 6, for the denim shirt and jeans, normal loss ensures clean geometry. For the Spiderman shirt example, the garment surface without normal loss guidance incorrectly represents Spider-

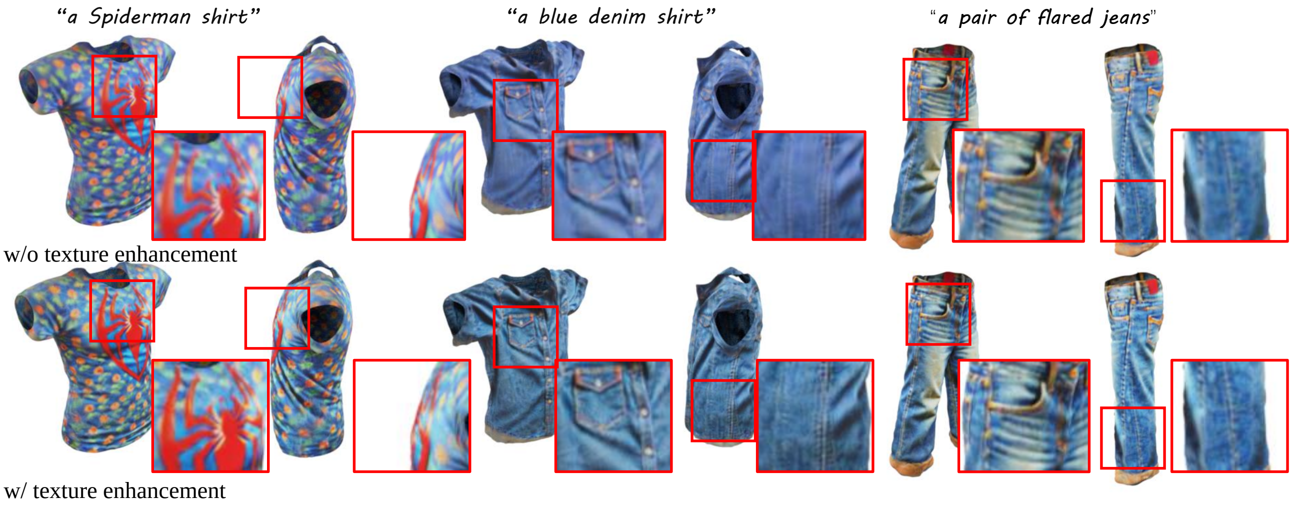

Figure 5. **Ablation of NeTF Enhancement**. From top to bottom are the results of texture without refinement, and final textured mesh respectively. Texture enhancement provides high-quality details in the hemline, seam allowance, grain line and wrinkles.

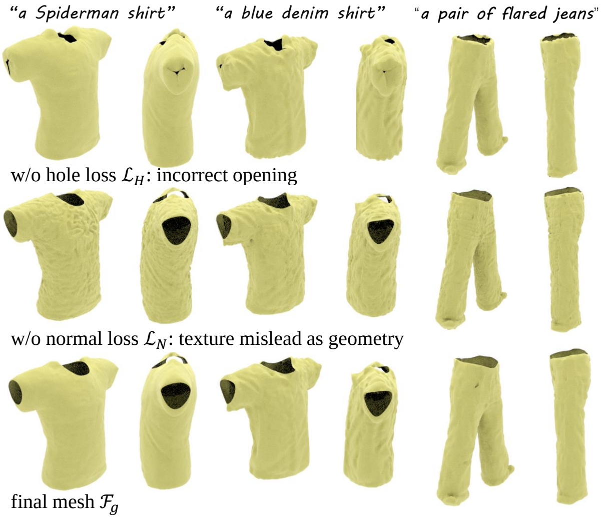
"a Spiderman shirt"
"a blue denim shirt"
"a pair of flared jeans"
w/o hole loss $\mathcal{L}_H$: incorrect opening
w/o normal loss $\mathcal{L}_N$: texture mislead as geometry
final mesh $\mathcal{F}_g$

Figure 6. **Ablation of Hole and Normal Loss**. From top to bottom are the results of GarmentDreamer without hole loss $\mathcal{L}_H$, normal loss $\mathcal{L}_N$, final mesh without texture respectively. Hole loss $\mathcal{L}_H$, normal loss $\mathcal{L}_N$ offers clear openings, noise-free surface of the generated garment mesh.

man texture as geometry structures, which is against fabric design in reality.

NeTF Enhancement Our proposed NeTF enhancement via VSD facilitates better capture of fabrication attributes. As shown in Fig. 5, it could be observed that without NeTF enhancement, the pocket of the shirt and the waistband are blurry. In contrast, texture enhancement leads to high-quality fabrication details, such as fold lines, grain lines, the seam allowance of the shirt, and clear fly piece, hemline, seam allowance, and side seam of the jumpsuit, which are crucial in traditional garment design.

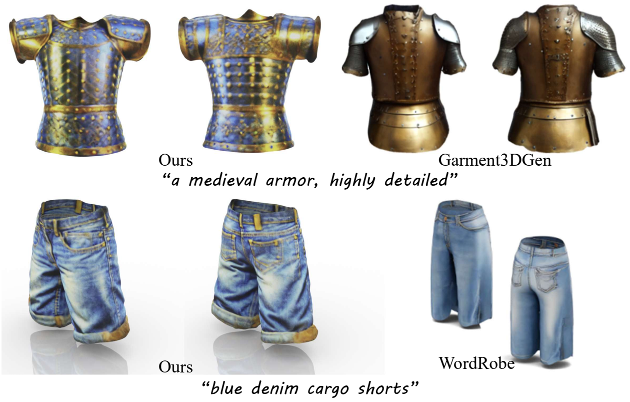
Ours
Garment3DGen
"a medieval armor, highly detailed"
Ours
WordRobe
"blue denim cargo shorts"

Figure 7. We compare with images from Garment3DGen [46] and WordRobe [52]. Ours show high-quality details.

5. Limitations and Future Work

There are several avenues for future research. Firstly, our method takes minutes rather than seconds for each generation process. Improving its efficiency and scalability is essential for large-scale garment collections. Second, integrating differentiable simulators could enhance the realism of generated garments. Furthermore, parameterizing the geometry with 2D sewing patterns could offer multiple benefits: they facilitate a seamless connection to the manufacturing process, ensure a closer match between real and simulated clothing, and provide a more intuitive design workflow for traditional fashion designers. Lastly, like other SDS-based 3D generation approaches, our method bakes lighting effects, such as specular highlights and shadows, into the texture, making them non-relightable and inconsistent with physical laws. Learning Physically-Based Rendering (PBR) materials could help separate these effects, enhancing the quality and realism of synthesized garments

References

[1] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020. 3

[2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 6

[3] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE transactions on visualization and computer graphics*, 14(1):213–230, 2007. 3

[4] Andrés Casado-Elvira, Marc Comino Trinidad, and Dan Casas. Pergamo: Personalized 3d garments from monocular video. In *Computer Graphics Forum*, pages 293–304. Wiley Online Library, 2022. 2

[5] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18558–18568, 2023. 2

[6] Jianchuan Chen, Wentao Yi, Liqian Ma, Xu Jia, and Huchuan Lu. Gm-nerf: Learning generalizable model-based neural radiance fields from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20648–20658, 2023. 2

[7] Lan Chen, Lin Gao, Jie Yang, Shibiao Xu, Juntao Ye, Xiaopeng Zhang, and Yu-Kun Lai. Deep deformation detail synthesis for thin shell models. In *Computer Graphics Forum*, page e14903. Wiley Online Library, 2023. 3

[8] Zilong Chen, Yikai Wang, Feng Wang, Zhengyi Wang, and Huaping Liu. V3d: Video diffusion models are effective 3d generators. *arXiv preprint arXiv:2403.06738*, 2024. 2, 4

[9] Patrick John Chia, Giuseppe Attanasio, Federico Bianchi, Silvia Terragni, Ana Rita Magalhães, Diogo Goncalves, Ciro Greco, and Jacopo Tagliabue. Contrastive language and vision learning of general fashion concepts. *Scientific Reports*, 12(1):18958, 2022. 7

[10] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2262–2272, 2023. 2

[11] CLO3D. Clo3d garment design software, 2024. 1

[12] Luca De Luigi, Ren Li, Benoît Guillard, Mathieu Salzmann, and Pascal Fua. Drapenet: Garment generation and self-supervised draping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1451–1460, 2023. 2, 3, 4

[13] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. *Advances in neural information processing systems*, 30, 2017. 4

[14] William Gao, Noam Aigerman, Thibault Groueix, Vova Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 3, 6

[15] Benoit Guillard, Federico Stella, and Pascal Fua. Meshudf: Fast and differentiable meshing of unsigned distance field networks. In *European conference on computer vision*, pages 576–592. Springer, 2022. 4

[16] Kai He, Kaixin Yao, Qixuan Zhang, Jingyi Yu, Lingjie Liu, and Lan Xu. Dresscode: Autoregressively sewing and generating garments from text guidance. *arXiv preprint arXiv:2401.16465*, 2024. 1, 2, 3

[17] Yi He, Haoran Xie, and Kazunori Miyata. Sketch2cloth: Sketch-based 3d garment generation with unsigned distance fields. In *2023 Nicograph International (NicoInt)*, pages 38–45. IEEE, 2023. 2

[18] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 2

[19] Shuo Huang, Zongxin Yang, Liangting Li, Yi Yang, and Jia Jia. Avatarfusion: Zero-shot generation of clothing-decoupled 3d avatars using 2d diffusion. page 5734–5745, 2023. 2

[20] Xin Huang, Ruizhi Shao, Qi Zhang, Hongwen Zhang, Ying Feng, Yebin Liu, and Qing Wang. Humannorm: Learning normal diffusion model for high-quality and realistic 3d human generation. *arXiv preprint arXiv:2310.01406*, 2023. 2

[21] Suyi Jiang, Haimin Luo, Haoran Jiang, Ziyu Wang, Jingyi Yu, and Lan Xu. Mvhuman: Tailoring 2d diffusion with multi-view sampling for realistic 3d human generation. *arXiv preprint arXiv:2312.10120*, 2023. 2

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4):1–14, 2023. 2, 4

[23] Byungjun Kim, Patrick Kwon, Kwangho Lee, Myunggi Lee, Sookwan Han, Daesik Kim, and Hanbyul Joo. Chupa: Carving 3d clothed humans from skinned shape priors using 2d diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15965–15976, 2023. 2, 3

[24] Taeksoo Kim, Byungjun Kim, Shunsuke Saito, and Hanbyul Joo. Gala: Generating animatable layered assets from a single scan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1535–1545, 2024. 2

[25] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. 4

[26] Lei Lan, Zixuan Lu, Jingyi Long, Chun Yuan, Xuan Li, Xiaowei He, Huamin Wang, Chenfanfu Jiang, and Yin Yang. Mil2: Efficient cloth simulation using non-distance barriers and subspace reuse. *arXiv preprint arXiv:2403.19272*, 2024. 2

[27] Jumin Lee, Sebin Lee, Changho Jo, Woobin Im, Juhyeong Seon, and Sung-Eui Yoon. Semcity: Semantic scene generation with triplane diffusion. *arXiv preprint arXiv:2403.07773*, 2024. 2

[28] Minchen Li, Danny M Kaufman, and Chenfanfu Jiang. Codimensional incremental potential contact. *ACM Transactions on Graphics (TOG)*, 40(4):1–24, 2021. 4

[29] Minchen Li, Chenfanfu Jiang, and Zhaofeng Luo. *Physics-Based Simulation*. 2024. 4

[30] Weiyu Li, Rui Chen, Xuelin Chen, and Ping Tan. Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. *arXiv preprint arXiv:2310.02596*, 2023. 2

[31] Xuan Li, Yu Fang, Lei Lan, Huamin Wang, Yin Yang, Minchen Li, and Chenfanfu Jiang. Subspace-preconditioned gpu projective dynamics with contact for cloth simulation. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–12, 2023. 2

[32] Yifei Li, Hsiao-yu Chen, Egor Larionov, Nikolaos Sarafianos, Wojciech Matusik, and Tuur Stuyck. Diffavatar: Simulation-ready garment optimization with differentiable simulation. *arXiv preprint arXiv:2311.12194*, 2023. 2

[33] Tingting Liao, Hongwei Yi, Yuliang Xiu, Jiaxaing Tang, Yangyi Huang, Justus Thies, and Michael J Black. Tada! text to animatable digital avatars. *arXiv preprint arXiv:2308.10899*, 2023. 2

[34] Fangfu Liu, Diankun Wu, Yi Wei, Yongming Rao, and Yueqi Duan. Sherpa3d: Boosting high-fidelity text-to-3d generation via coarse 3d prior. *arXiv preprint arXiv:2312.06655*, 2023. 2

[35] Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. Towards garment sewing pattern reconstruction from a single image. *ACM Transactions on Graphics (TOG)*, 42(6):1–15, 2023. 1, 2, 3

[36] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023. 1, 2, 3, 6

[37] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 2

[38] Maya. Autodesk maya, 2024. 1

[39] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022. 6

[40] Matthias Müller and Nuttapong Chentanez. Wrinkle meshes. In *Symposium on Computer Animation*. Madrid, Spain, 2010. 3

[41] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2, 4

[42] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. 2

[43] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. *arXiv preprint arXiv:2311.16918*, 2023. 2

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 7

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[46] Nikolaos Sarafianos, Tuur Stuyck, Xiaoyu Xiang, Yilei Li, Jovan Popovic, and Rakesh Ranjan. Garment3dgen: 3d garment stylization and texture generation. *arXiv preprint arXiv:2403.18816*, 2024. 2, 3, 7, 8

[47] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021. 2

[48] Yu Shen, Junbang Liang, and Ming C Lin. Gan-based garment generation using sewing pattern images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 225–247. Springer, 2020. 2

[49] Jaehyeok Shim, Changwoo Kang, and Kyungdon Joo. Diffusion-based signed distance fields for 3d shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20887–20897, 2023. 2

[50] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023. 2

[51] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004. 3

[52] Astitva Srivastava, Pranav Manu, Amit Raj, Varun Jampani, and Avinash Sharma. Wordrobe: Text-guided generation of textured 3d garments. *arXiv preprint arXiv:2403.17541*, 2024. 2, 3, 7, 8

[53] Style3D. Style3d digital fashion solution, 2024. 1

[54] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Scene graph denoising diffusion probabilistic model for generative indoor scene synthesis. *arXiv preprint arXiv:2303.14207*, 2023. 2

[55] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 2

[56] An Dinh Vuong, Minh Nhat Vu, Toan Nguyen, Baoru Huang, Dzung Nguyen, Thieu Vo, and Anh Nguyen. Language-driven scene synthesis using multi-conditional diffusion model. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[57] Jionghao Wang, Yuan Liu, Zhiyang Dou, Zhengming Yu, Yongqing Liang, Xin Li, Wenping Wang, Rong Xie, and Li Song. Disentangled clothed avatar generation from text descriptions. 2023. 2

[58] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 4

[59] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 6

[60] Markus Worchel, Rodrigo Diaz, Weiwen Hu, Oliver Schreer, Ingo Feldmann, and Peter Eisert. Multi-view mesh reconstruction with neural deferred shading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6187–6197, 2022. 3, 5

[61] Donglai Xiang, Fabian Prada, Zhe Cao, Kaiwen Guo, Chenglei Wu, Jessica Hodgins, and Timur Bagautdinov. Drivable avatar clothing: Faithful full-body telepresence with dynamic clothing driven by sparse rgb-d input. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023. 2

[62] Yuanyou Xu, Zongxin Yang, and Yi Yang. Seeavatar: Photorealistic text-to-3d avatar generation with constrained geometry and appearance. *arXiv preprint arXiv:2312.08889*, 2023. 2

[63] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. *arXiv preprint arXiv*, 2310, 2023. 2

[64] Chang Yu, Yi Xu, Ye Kuang, Yuanming Hu, and Tiantian Liu. Meshtaichi: A compiler for efficient mesh-based operations. *ACM Transactions on Graphics (TOG)*, 41(6):1–17, 2022. 2

[65] Zhengming Yu, Zhiyang Dou, Xiaoxiao Long, Cheng Lin, Zekun Li, Yuan Liu, Norman Müller, Taku Komura, Marc Habermann, Christian Theobalt, et al. Surf-d: High-quality surface generation for arbitrary topologies using diffusion models. *arXiv preprint arXiv:2311.17050*, 2023. 2, 3

[66] Huichao Zhang, Bowen Chen, Hao Yang, Liao Qu, Xu Wang, Li Chen, Chao Long, Feida Zhu, Daniel Du, and Min Zheng. Avatarverse: High-quality & stable 3d avatar creation from text and pose. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7124–7132, 2024. 2

[67] Qihang Zhang, Chaoyang Wang, Aliaksandr Siarohin, Peiye Zhuang, Yinghao Xu, Ceyuan Yang, Dahua Lin, Bolei Zhou, Sergey Tulyakov, and Hsin-Ying Lee. Scenewiz3d: Towards text-guided 3d scene composition. *arXiv preprint arXiv:2312.08885*, 2023. 2

[68] Xuanmeng Zhang, Jianfeng Zhang, Rohan Chacko, Hongyi Xu, Guoxian Song, Yi Yang, and Jiashi Feng. Getavatar: Generative textured meshes for animatable human avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2273–2282, 2023. 2

[69] Zhenglin Zhou, Fan Ma, Hehe Fan, and Yi Yang. Headstudio: Text to animatable head avatars with 3d gaussian splatting. *arXiv preprint arXiv:2402.06149*, 2024. 2