

OmniFabric: Coherent UV Space Texture Synthesis for 3D Garment Reconstruction

DING-JIUN HUANG, Carnegie Mellon University, United States of America

YUANHAO WANG, University of Washington, United States of America

CHENG ZHANG, Texas A&M University, United States of America

HUGO BERTICHE, Google, United States of America

ALEXANDRU-EUGEN ICHIM, Google, Switzerland

THABO BEELER, Google, Switzerland

FERNANDO DE LA TORRE, Carnegie Mellon University, United States of America



Fig. 1. Image-based 3D garment texture synthesis. Given a single in-the-wild clothing image, OmniFabric synthesizes high-quality, coherent fabric textures directly on garment sewing patterns. OmniFabric supports a wide range of textures and patterns, preserving fine details while maintaining global alignment with the input image. The resulting textured sewing patterns can be seamlessly simulated into 3D garments.

Automated generation of production ready 3D garment assets from a single image is a central challenge in digital content creation. While recent generative models have significantly advanced 3D geometry reconstruction, synthesizing high-quality textures remains a bottleneck. Existing methods often bake environmental illumination and shadows directly into the texture map, or they fail to maintain global structural coherence, making the resulting assets unusable for physical simulation and relighting. In this work, we introduce OmniFabric, a novel approach that synthesizes globally coherent texture maps directly within the 2D sewing pattern space. Given a single

reference image, our pipeline utilizes an estimated 3D mesh and generative priors of powerful Vision-Language models (VLM) to establish a complete but coarse texture initialization across the unwrapped sewing patterns. We then leverage a specialized diffusion transformer, trained via an automated synthetic data engine and conditioned on 3D positional features, to refine this initialization directly in the canonical UV domain. This effectively removes distortion and baked-in artifacts to extract a clean and normalized texture map that preserves the original garment design. Extensive experiments demonstrate that OmniFabric significantly outperforms state-of-the-art baselines, yielding photorealistic 3D garments with high-quality textures. Project page: <https://humansensinglab.github.io/OmniFabric>

CCS Concepts: • **Computing methodologies** → **Animation; Computer vision.**

Additional Key Words and Phrases: Texture Synthesis, 3D Garment Reconstruction, Garment Sewing Patterns

ACM Reference Format:

Ding-Jiun Huang, Yuanhao Wang, Cheng Zhang, Hugo Bertiche, Alexandru-Eugen Ichim, Thabo Beeler, and Fernando De la Torre. 2026. OmniFabric: Coherent UV Space Texture Synthesis for 3D Garment Reconstruction. In *SIGGRAPH Asia 2026 Conference Papers (SA Conference Papers '26)*, December 01–04, 2026, Kuala Lumpur, Malaysia. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3829340.3842275>



This work is licensed under a Creative Commons Attribution 4.0 International License. SA Conference Papers '26, Kuala Lumpur, Malaysia
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2842-6/2026/12
<https://doi.org/10.1145/3829340.3842275>

1 Introduction

Creating high-fidelity, simulation-ready 3D garments is essential for modern digital production, with applications spanning gaming [Habermann et al. 2021], e-commerce [Hwangbo et al. 2020; Kim and LaBat 2013], and cinematic visual effects [Hughes et al. 2007]. While traditional asset creation relies on labor-intensive manual modeling, there is growing interest in automating this process to synthesize assets directly from images. Despite recent strides in reconstructing accurate 3D garment geometry [Bian et al. 2025; He et al. 2024; Li et al. 2025b; Nakayama et al. 2025; Rong et al. 2025; Sarafianos et al. 2025; Wang et al. 2025], generating production-ready textures from a single observation remains an open problem.

Garment texture synthesis is uniquely difficult due to the complex dynamics of fabric, which frequently produce deep folds, self-occlusions, and extreme pose variations (see Figure 1). To be truly usable in downstream applications, a texture must not only look realistic in a static view but also be free from baked-in lighting and geometric distortion, capturing the garment’s intrinsic albedo. However, current state-of-the-art 3D texturing frameworks [Bensadoun et al. 2024; Chen et al. 2023b; Richardson et al. 2023; Zeng et al. 2024; Zhao et al. 2025] typically rely on multi-view diffusion priors to iteratively “paint” assets via direct projection onto draped 3D geometry. Consequently, unwrapping these folded surfaces inevitably causes large geometric distortion and occlusion. Furthermore, these models fail to disentangle intrinsic material properties from environmental illumination, baking transient lighting artifacts, such as shadows and geometry-induced wrinkles, directly into the texture map.

To circumvent these issues, recent work such as FabricDiffusion [Zhang et al. 2024] has explored texture synthesis directly within the regularized, flat 2D sewing pattern domain, successfully yielding simulation-ready texture maps. However, FabricDiffusion focuses exclusively on local texture synthesis by extracting tileable material patches from the source image, ignoring the global texture layout. Consequently, it struggles at reconstructing asymmetric designs, large-scale prints, or irregular spatial patterns. We argue that synthesizing a globally coherent texture map that faithfully reflects the visual identity of the input image requires leveraging the complete context of the reference observation, rather than relying on isolated patches. This presents a fundamental challenge:

How do we establish pixel-level correspondences between a reference image and 2D sewing panels while transferring textures that are free from baked-in artifacts and geometric distortions?

In this paper, we propose **OmniFabric**, to bridge this gap. We first employ a powerful image generation model [Google 2025] to repose the input clothing into a canonical A-pose, aligning its silhouette with the frontal render of the garment mesh simulated from the predicted sewing patterns. We further harness the temporal and spatial consistency of large-scale video generation models [Google DeepMind 2025] to hallucinate coherent, multi-view observations of this A-pose garment. Projecting these globally consistent multi-view images onto the unwrapped sewing patterns establishes a complete, albeit coarse, texture initialization that preserves the garment’s global design. Because this initial projection inevitably bakes visual artifacts such as geometric distortions and physics-induced wrinkles into the texture map, we introduce a subsequent refinement stage

operating entirely within the sewing pattern space. We design an automatic data engine that curates an extensive and highly diverse synthetic dataset of textured sewing patterns, paired with simulated texture initializations containing baked-in artifacts and distortions. With the curated dataset, we train a specialized Diffusion Transformer (DiT) that acts as a texture normalizer in the 2D sewing pattern space. This model rectifies distortions, and strips away transient illumination to produce a clean and normalized texture map. Finally, we conduct comprehensive experiments to demonstrate the superiority of OmniFabric over existing 3D texturing baselines. In summary, the key contributions of our work are:

- OmniFabric, a novel framework for synthesizing high-fidelity, globally coherent textures from a single in-the-wild image for 3D garment reconstruction.
- An automated data engine for creating realistic, diverse textured sewing patterns at scale, providing training data for learning complex garment textures.
- A strategy of using canonical 3D mesh as spatial anchor to map image pixels to sewing pattern UV space, enabling globally aligned texture synthesis.
- A coarse-to-fine texture generation pipeline that initializes globally aligned textures, and further rectifies distortion and baked-in artifacts with a DiT-based UV space normalization network.

Remark. Generating 3D garment textures is not a new topic. While many prior works mainly focus on directly texturing 3D garment meshes, they often do not scale well to large collections of in-the-wild images and remain weakly connected to real-world garment production pipelines. We argue that fabric texture synthesis should be grounded in UV space of sewing patterns. This enables the model to maintain strong global coherence while remaining closely aligned with how garments are constructed in practice.

2 Related Work

2.1 3D Garment Reconstruction

Early approaches for garment reconstruction primarily relied on implicit functions [Saito et al. 2019, 2020; Xiu et al. 2022] to approximate surfaces, yet they often struggle with complex topologies and loose-fitting clothing. More recent works leverage advanced representations like 3DGS [Rong et al. 2025] and image-to-3D diffusion priors [Luo et al. 2024; Wang et al. 2025] to achieve impressive visual fidelity. However, these methods typically produce rigid meshes or unstructured point clouds that are inherently unsuitable for physical cloth simulation. While methods like Garment3DGen [Sarafianos et al. 2025] bridge this gap by utilizing template deformation to produce simulation-ready assets, they rely on a slow, computationally intensive per-asset optimization process. To obtain simulation-ready assets, a significant line of work attempts to infer 2D sewing patterns directly from 3D point clouds or images. NeuralTailor [Korosteleva and Lee 2022] focused on reconstructing pattern structures from 3D geometry, and subsequent works [Chen et al. 2024; Liu et al. 2023] predict these patterns directly from single-view images using discriminative transformer architectures. To facilitate more structured generation, GarmentCode [Korosteleva and Sorkine-Hornung 2023] introduced a programmatic domain-specific

language (DSL) for sewing patterns, enabling parametric control. Building on this representation, subsequent research has increasingly focused on advanced generative modeling. DressCode [He et al. 2024] synthesizes novel patterns from text prompts using a GPT-based framework, while Alpparel [Nakayama et al. 2025] scales into a multi-modal foundation model that handles complex pattern generation natively from mixed inputs. To further improve physical realism, Dress-1-to-3 [Li et al. 2025b] incorporates differentiable physics simulators to optimize geometric alignment of the inferred patterns. Concurrently, other state-of-the-art frameworks directly predict these programmatic structures from images leveraging large generative models [Bian et al. 2025; Li et al. 2025a; Zhou et al. 2025]. While these methods have established sewing patterns as a robust domain for 3D animation, the synthesis of normalized, artifact-free textures for these panels remains an open challenge.

2.2 Generative 3D Texturing

The rise of large-scale vision language models [Radford et al. 2021; Rombach et al. 2022; Saharia et al. 2022] has revolutionized 3D texture synthesis. Early optimization-based approaches either leverage CLIP [Radford et al. 2021] for optimizing texture map of 3D models [Chen et al. 2022; Hong et al. 2022; Michel et al. 2022; Mohammad Khalid et al. 2022], or employ Score Distillation Sampling (SDS) to optimize texture maps by distilling gradients from a frozen 2D diffusion model [Chen et al. 2023a; Metzger et al. 2023; Poole et al. 2022]. While capable of generating coherent global structures, these methods tend to hallucinate generic textures without high-frequency details. To improve fidelity, recent research has shifted toward projection-based inpainting, by utilizing depth-conditioned Stable Diffusion to progressively paint 3D meshes from multiple viewpoints [Cao et al. 2023; Chen et al. 2023b; Richardson et al. 2023]. These frameworks iteratively project the current texture into screen space, inpaint missing regions using 2D priors, and project the result back. More recent systems like Paint3D [Zeng et al. 2024], MVPaint [Cheng et al. 2025] and Meta 3D TextureGen [Bensadoun et al. 2024] further refine this process by synchronizing multi-view generation or employing coarse-to-fine UV refinement to minimize seams. Despite their popularity, a fundamental limitation persists across these general-purpose methods: they rely on priors trained on natural photography. Consequently, they inherently entangle illumination with surface appearance, inevitably “baking in” transient lighting effects—such as cast shadows and specular highlights—directly into the UV map. In addition, the generated textures are subject to distortion and blurring due to the complex surface topology, leading to suboptimal results.

In the specific area of garment texturing, FabricDiffusion [Zhang et al. 2024] addresses some of these pitfalls by treating texture synthesis as a material extraction task. It utilizes a diffusion model to extract tileable, distortion-free material patches from a single image without lighting artifacts. However, because it relies on local texture tiling, FabricDiffusion fails to capture global structural information, such as specific graphic placements, or handling non-uniform texture patterns across different UV islands. OmniFabric bridges this gap by shifting from local material extraction to global texture synthesis. We operate in the canonical UV space to disentangle

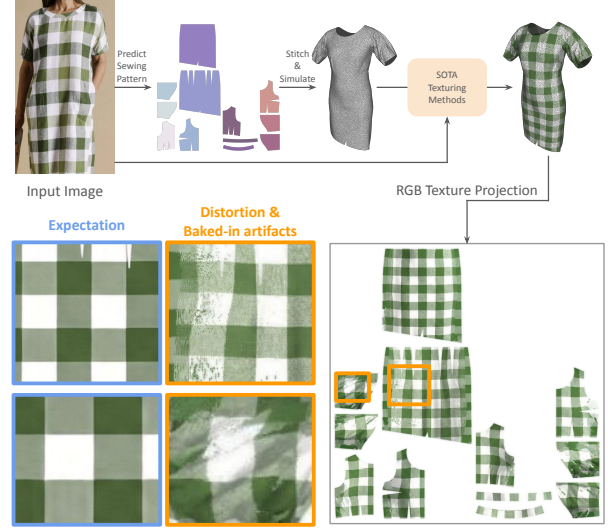


Fig. 2. **Limits of baseline texturing methods.** While existing methods can synthesize textures for 3D assets with decent visual quality, they fail to create garment textures free from distortions and baked-in illumination artifacts, rendering the asset unsuitable for downstream applications.

base-color from illumination, synthesizing a holistic texture map that respects the global texture design in the reference image. While previous methods [Bensadoun et al. 2024; Zeng et al. 2024] also deploy a 2D refinement stage on the UV map, they only perform inpainting on occluded regions and fail to rectify distortion or remove baked-in artifacts, which are the domain specific challenges for garment texturing, as shown in Figure 2.

3 Method

Figure 3 presents an overview of our approach. We aim to reconstruct 3D garments with coherent textures from real world images. We define the task and objective in Section 3.1. Next, we introduce a two stage pipeline for generating holistic textures directly onto sewing patterns. This includes a coarse texture map initialization (Section 3.2) and a UV space texture normalization (Section 3.3). To train the model, we develop an automated pipeline to synthesize a large scale dataset of textured sewing patterns (Section 4).

3.1 Problem Definition

Given a single reference image I and an estimated 3D garment mesh $\mathcal{M}_R$ parameterized by sewing patterns $\mathcal{P}$ from an off-the-shelf sewing pattern prediction model, our goal is to generate a normalized texture map $\mathcal{T}$ directly within the sewing pattern space. Unlike previous methods [Zhang et al. 2024] that focus on extracting local and repeatable material patches, we formulate our objective as a global texture synthesis task. We define $\mathcal{T} \in \mathbb{R}^{H \times W \times 3}$ as a representation of normalized texture map of the garment, which is completely free of distortion, view dependent illumination, cast shadows, and geometry induced wrinkles. Following the paradigm of diffusion models, we formulate the synthesis of this holistic texture map as a conditional distribution mapping problem. We seek

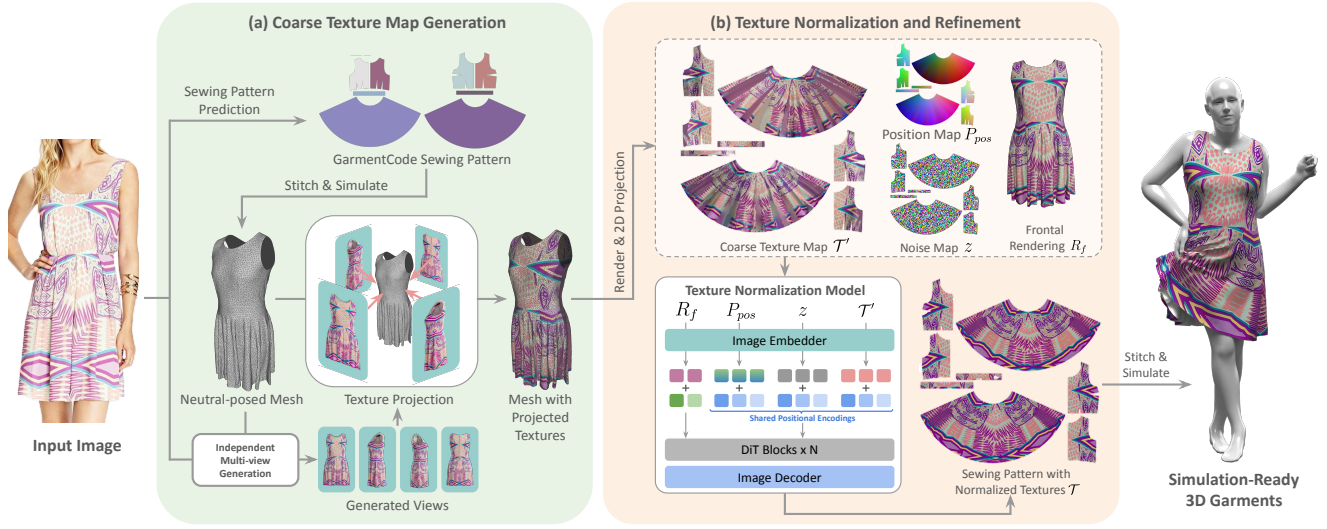


Fig. 3. **Overview of OmniFabric.** Given a reference clothing image, OmniFabric synthesizes high-quality garment textures in sewing pattern UV space via a two-stage pipeline. **Stage 1:** We generate a coarse UV texture map by first predicting garment sewing patterns and reconstructing a canonical 3D garment. The input image texture is transferred onto the garment surface and expanded into multi-view using an off-the-shelf video generative model. These views are projected onto the canonical garment and reflected in UV space to produce a globally aligned coarse texture map. **Stage 2:** We refine the coarse texture with a DiT-based texture normalization network operating on UV patches. By conditioning on positional maps and spatially aligned tokens, the model removes baked-in artifacts and projection inconsistencies, producing clean and coherent sewing pattern textures that can be directly simulated into 3D garments.

to learn a mapping function $\mathcal{G}$ such that:

$$\mathcal{T} \sim \mathcal{G}(\mathcal{I}, \mathcal{P}, \epsilon), \epsilon \sim \mathcal{N}(0, \mathbf{I}) \quad (1)$$

We note that although $\mathcal{T}$ is not intrinsic albedo, it’s a close approximation and can be directly imported into cloth simulation engines, e.g., CLO [CLO Virtual Fashion Inc. 2024], with externally specified material assumption for remaining properties including roughness, metallic and normal map, enabling the synthesis of simulation-ready 3D garments.

3.2 Coarse Texture Map Initialization

To guide texture synthesis in the sewing pattern space, we first aim to obtain a coarse texture initialization that fully leverages the visual information from the reference image. This requires establishing pixel wise correspondences between the reference image and the sewing pattern. While this mapping is challenging, we recast the task via a VLM driven and pose aware texture alignment step. We further leverage the multiple view generative priors of large video models to provide a complete and globally coherent texture initialization.

3.2.1 Pose aware texture alignment. Using Nano Banana Pro [Google 2025], we rebase the reference image I into I' to perfectly align with the frontal silhouette of the rest-pose garment mesh $\mathcal{M}_R$. This alignment enables direct pixel projection of I' onto $\mathcal{M}_R$ and its corresponding sewing patterns $\mathcal{P}$, producing the textured frontal render R_f . We observe that Nano Banana Pro successfully manipulates the input garment into a desired pose while preserving the original texture with high accuracy and fidelity.

3.2.2 Consistent multiple view synthesis. After the initial pose alignment, a significant portion of the UV space occluded from the frontal view remains untextured. To hallucinate these missing regions while

encouraging strong spatial continuity, we leverage a large-scale video generation model for multi-view generation. By conditioning the model on the textured frontal render R_f , we synthesize a 360-degree spinning video of the garment, naturally exploiting the model’s inherent temporal priors for structural consistency. Following recent projection-based texturing paradigms, we sample multi-view V , four orthogonal views (front, back, left, and right) from the generated video. These observations are then projected onto $\mathcal{M}_R$ and mapped to their corresponding UV coordinates on $\mathcal{P}$ to form the coarse texture map $\mathcal{T}'$.

3.3 UV Space Texture Refinement

While $\mathcal{T}'$ preserves the coherent global texture of the garment, it contains baked-in artifacts like shadows, physical wrinkles and unpainted areas. Furthermore, spatial distortions in $\mathcal{T}'$ leave it far short of production-ready fabric quality, as shown in Figure 2. To address these issues, we introduce a texture normalization model designed to rectify $\mathcal{T}'$ into a normalized texture map $\mathcal{T}$.

3.3.1 Training objective of multi-condition diffusion. We formulate texture normalization as a holistic UV synthesis problem by developing a conditional distribution mapping network. Unlike FabricDiffusion [Zhang et al. 2024], which relies on a single condition, i.e., a local textile patch, for normalization task, our framework incorporates a multi-modal conditioning set y to guide global texture normalization. We extend Eq. 1 and minimize the following:

$$\mathcal{L} = \mathbb{E}_{\mathcal{E}(\mathcal{T}), y, \epsilon \sim \mathcal{N}(0, \mathbf{I}), t} [\|\epsilon - \epsilon_\theta(z_t, t, y)\|^2], y = \{\mathcal{T}', R_f, P_{pos}\} \quad (2)$$

where z_t represents the noisy latent of the ground-truth texture at timestep t . The multi-modal conditioning set y includes the coarsely-textured map $\mathcal{T}'$, textured frontal rendering R_f and a 3D position map P_{pos} , created by projecting the vertices’ 3D positions of $\mathcal{M}_R$

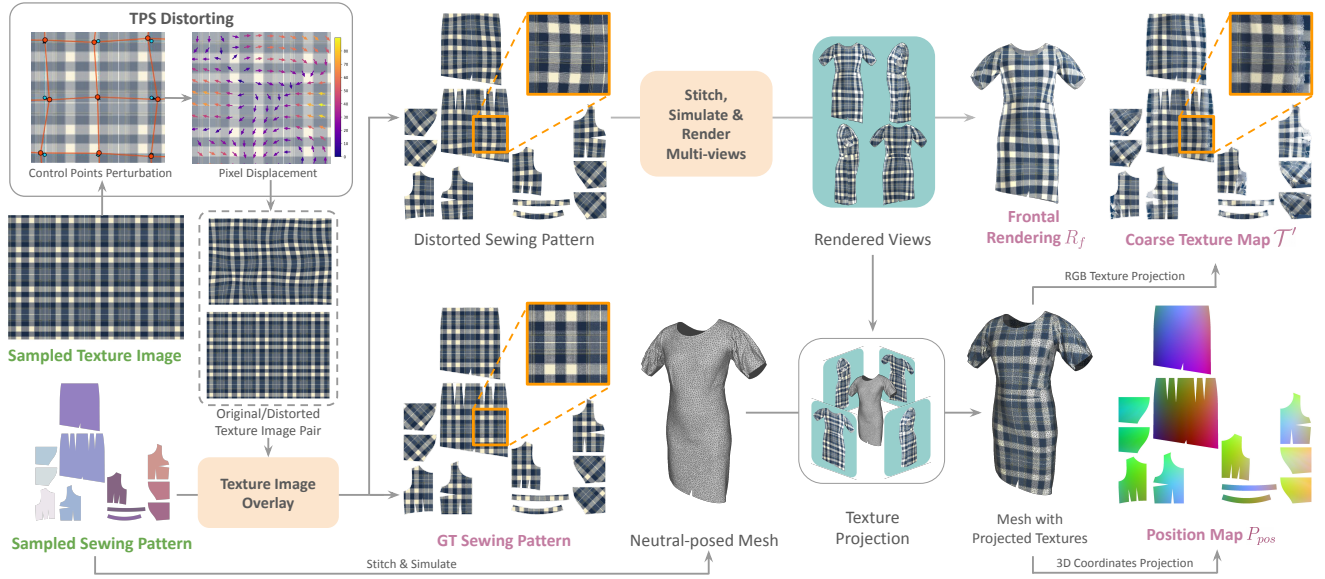


Fig. 4. **Framework of automated data creation.** We first create distorted texture image through TPS distorting from the sampled normalized counterpart, which will both be printed onto sampled sewing pattern. The textured sewing pattern can then be simulated into a training data point of rest-posed mesh M_R , position map P_{pos} , coarse texture map $\mathcal{T}'$ and frontal view rendering R_f .

onto the 2D layout of $\mathcal{P}$. P_{pos} helps the model learn spatial connectivity between separate UV panels, ensuring seamless textures across fragmented UV islands.

3.3.2 Model architecture and training. To process multi-modal conditioning set y , we adopt a Diffusion Transformer (DiT) architecture following previous works [Tan et al. 2025a,b], and leverage a unified tokens processing strategy that treats all conditions as a unified sequence of tokens. We utilize a joint self-attention mechanism where the condition tokens (from $\mathcal{T}'$, R_f , and P_{pos}) and the noisy latent tokens from z_t interact within the same transformer blocks. Because $\mathcal{T}'$, P_{pos} , and the target $\mathcal{T}$ all share the same 2D panel layout, we apply a dynamic positional encoding strategy: image patches at the same spatial coordinates across these maps are assigned identical positional encodings, while only R_f retains its own coordinate system through distinct encodings. This minimal yet universal design allows the model to attend to the structural cues of P_{pos} while simultaneously hallucinating missing textures in $\mathcal{T}'$. We use a pre-trained weight of DiT, and fine-tune the model on our synthetic textured sewing pattern data with LoRA [Hu et al. 2022].

4 Synthetic Training Data Creation

To train our texture normalization model, we develop a data engine (Figure 4) to curate a large-scale dataset of textured sewing patterns. The core objective is to synthesize training pairs that exhibit the complex appearance of real-world garments while providing ground-truth normalized texture maps for supervised learning.

4.1 Textured Sewing Pattern Creation

A key bottleneck in garment dataset construction is the lack of complex and high-resolution textures; most available texture sources

provide only simple, repetitive patterns. To overcome this, we introduce a pipeline to produce diverse, high-fidelity texture images by using fashion images from real-world datasets, e.g., DeepFashion [Liu et al. 2016], as references. For each sample, we apply a Large Language Model (LLM) [Bai et al. 2023] to generate a detailed description encompassing texture layout, semantic elements, and color palettes. These descriptions, combined with the reference image, are fed into a vision language model [Google 2025] to synthesize textures with diverse designs. We then overlay the generated texture image to sewing patterns sampled from GarmentCodeData. Note that the goal here is not to reconstruct the exact texture from the example image, but to generate plausible and realistic textures that provide sufficient diversity for training the texture synthesis model.

4.2 Training Pairs Construction

For each textured sewing pattern, we generate a training tuple $(\mathcal{T}', R_f, P_{pos}, \mathcal{T})$ consisting of a coarse texture map, a frontal rendering, a positional map, and the ground-truth texture. To explicitly train the network to rectify geometric distortions, we apply a random Thin Plate Spline (TPS) [Bookstein 2002] distortion to the initial texture image. The original, undistorted texture forms the ground-truth map $\mathcal{T}$, while the TPS-distorted texture is overlaid onto the sewing pattern for physical simulation. Specifically, we perturb each control point p_i of a regular grid fit to the texture’s aspect ratio, giving $\hat{p}_i = p_i + \delta_i$ with

$$\delta_i = \varepsilon_i \cdot \lambda \cdot \frac{D}{800}, \varepsilon_i \sim \mathcal{U}(-0.5, 0.5) \quad (3)$$

where $\lambda = 25$ by default controls the overall distortion magnitude and D is the resolution of a texture image. A TPS is then fit between the regular grid $\{p_i\}$ and its perturbed counterpart $\{\hat{p}_i\}$ and applied densely to warp the full-resolution texture. We stitch and

draped the sewing pattern onto an A-pose SMPL body to yield a 3D garment mesh M_R , from which we capture the frontal rendering R_f . The positional map P_{pos} is then derived by projecting the xyz coordinates of the mesh vertices onto the 2D panel layout $\mathcal{P}$. Finally, we synthesize the coarse texture map $\mathcal{T}'$ by projecting R_f and a randomly sampled view R_a onto a neutral-posed mesh and unwrapping them back into the 2D pattern space. This explicitly bakes physics-induced artifacts and deformations into the initialization, compelling the model to learn undistortion and normalized base-color recovery.

5 Experiments

In this section, we present a comprehensive evaluation of OmniFabric through quantitative and qualitative analyses. We first detail experimental setup, including curated synthetic dataset and metrics used for evaluation. We then demonstrate the effectiveness of our framework by comparing against SOTA methods for 3D texturing. Finally, we conduct ablation studies to validate our core architectural designs. Additional implementation details, extended qualitative results, our method’s adaptability to other sewing pattern prediction methods besides ChatGarment [Bian et al. 2025], examples of our curated garment data, and discussion on potential future directions are provided in the supplementary material.

5.1 Setup

5.1.1 Datasets. To train and evaluate our model, we construct a large-scale synthetic dataset of textured sewing patterns, as described in Section 4. We sample a total of 3K unique and diverse garment data from GarmentCodeData [Korosteleva et al. 2024]. We then use the pipeline in Figure 4 to generate textures images, and print them onto sampled sewing patterns, finally constructing a dataset of 30K textured sewing patterns with diverse appearance and structural design (10 texture variations per garment style on average). We simulate garments using NVIDIA Warp [Macklin 2022] with a per-sample random seed, so repeated simulations of the same sewing pattern produce distinct M_R meshes and, consequently, different baked-in wrinkles in $\mathcal{T}'$. For lighting, we sample random subsets of white point lights to introduce varied illumination and baked-in shading across the dataset. We train our model with this synthetic dataset, and keep a held-out set for quantitative comparison with other baseline models with a test/train ratio of 0.1. Besides synthetic data, we evaluate all methods on in-the-wild images, which is the major focus of this work. These in-the-wild images are obtained from DeepFashion [Liu et al. 2016] dataset and also latest images of clothing design. Due to the lack of ground-truths of in-the-wild images, we evaluate mainly through qualitatives.

5.1.2 Metrics. We evaluate fidelity and structural coherence of the generated textures with a suite of standard metrics. We use LPIPS [Zhang et al. 2018] and DISTs [Ding et al. 2020] to measure visual similarity and structural consistency. We also compare with SSIM [Wang et al. 2004] and MS-SSIM [Wang et al. 2003], metrics employed to assess pixel-level structural accuracy. CLIP-score (CLIP-s) [Gal et al. 2022] measures the semantic alignment between the final textured garment and the reference images.

5.1.3 Baseline methods. We compare against several SOTA methods in 3D texturing and garment-specific texture transfer. Among them, Hunyuan3D-2.0 [Zhao et al. 2025] and Paint3D [Zeng et al. 2024] use multi-view generation to provide additional prior other than the given single-view observation. FabricDiffusion [Zhang et al. 2024] is capable of creating fabric textures with no baked-in artifacts by normalizing local textile pattern with a diffusion model.

5.1.4 Implementation details. Our texture normalization model is based on the Diffusion Transformer (DiT) architecture, and we follow the multi-conditioning strategy of OminiControl [Tan et al. 2025a] by treating tokens from all input conditions as a unified token sequence. The model is fine-tuned using LoRA on a single NVIDIA A6000 GPU with a batch size of 8, keeping the VAE and primary DiT backbone fixed to preserve generative stability. We resize and pad sewing pattern maps into 1024×1024 resolution in a batch for training efficiency. During inference, we reverse the padding and resizing process, and deploy an image super-resolution model [Wang et al. 2021] to upscale the output. Since we are fine-tuning the model with LoRA, a higher training resolution can be easily achieved without GPU memory issue, making our framework scalable for training data of higher resolution. We fine-tune the model until convergence on our synthetic dataset. We utilize Veo 3 [Google DeepMind 2025] as the multi-view video generation prior and Nano Banana Pro [Google 2025] for the initial texture transfer to the frontal rendering R_f . We used ChatGarment [Bian et al. 2025] as the model for sewing pattern prediction throughout all experiments in the main paper. The average runtime for a single inference is roughly 5 minutes, depending on the current load on the Gemini model servers. The failure rate of Gemini models, e.g., generating unrelated textures on R_f or multi-views, is lower than 2%, estimated from randomly sampled generations. No additional filtering is necessary thanks to its stable performance. Additional details including prompts used for generation are provided in supplementary material.

5.2 Qualitative Comparisons

5.2.1 Texture synthesis by OmniFabric from in-the-wild images. We first show our results in Figure 5. Given a reference image, OmniFabric first generates a coarse texture map, serving as roughly textured sewing pattern. Then, the texture normalization model removes the distortions and baked-in artifacts. From the front views of simulated 3D garments provided in Figure 5, we show that OmniFabric is capable of creating seamless textures across the 3D surface of garments. Note that while OmniFabric excels in textured garment synthesis from real-world images, we fine-tune our model only with synthetic data, which poses great scalability for model improvement.

5.2.2 Simulation results. We present simulations of synthesized 3D garments in Figure 9 via CLO. The sewing patterns generated by OmniFabric provide the normalized RGB base-color as albedo. Other parameters not predicted by our model—roughness, metallic, reflection intensity, and auto-generated normal map—are left at CLO’s default Fabric_Matte preset values. The Environment/HDRI maps for relighting examples are also chosen from CLO’s lighting presets. The simulated assets exhibit a convincing and photorealistic



Fig. 5. **Qualitative results of OmniFabric.** Our method synthesizes 3D garments with high appearance consistency by predicting sewing patterns of normalized textures. In particular, OmniFabric handles intrinsic geometric distortions and reconstructs diverse texture types, including logos and printed graphics (1st and 2nd rows), and complex high-frequency patterns with dense visual details. Please zoom in to check the details.

dynamic appearance, maintaining structural integrity and texture coherence even under extreme dynamics and varying illumination.

5.2.3 Comparison with state-of-the-art methods. We compare with other methods in Figure 6 using in-the-wild images as well. For input with complex textures and elements, both Hunyuan3D and



Fig. 6. **Qualitative comparisons to state-of-the-art methods.** OmniFabric is capable of generating accurate asymmetric textures and logos, while existing methods fail to preserve the high-frequency details on input cases with complex patterns.

Table 1. **Quantitative comparisons.** We compare with baselines on our synthetic dataset. The results show that OmniFabric greatly surpasses all other methods in terms of both pixel and perceptual quality.

	LPIS ↓	SSIM ↑	MS-SSIM ↑	DISTS ↓	CLIP-s ↑
FabricDiffusion [Zhang et al. 2024]	0.273	0.645	0.707	0.259	0.906
Paint3D [Zeng et al. 2024]	0.311	0.657	0.704	0.266	0.890
Hunyuan3D-2.0 [Zhao et al. 2025]	0.223	0.712	0.717	0.220	0.924
OmniFabric (ours)	0.092	0.868	0.905	0.121	0.963

Paint3D fail to preserve the high-frequency details. Even though FabricDiffusion can create normalized textile pattern and texture the given mesh through tiling, it is limited to local textile patch and fails to generate coherent global appearance. Conversely, OmniFabric creates normalized textures while preserving coherent details.

5.3 Quantitative Comparisons

5.3.1 Synthetic dataset. We show quantitative results where all baselines are evaluated with our curated synthetic dataset for fair comparison. For each test case, we provide every baseline with the identical frontal rendering R_f and rest-posed mesh M_R . Each method then performs its respective 3D texturing task on M_R . To compute image-based metrics, the resulting 3D textures are projected back into the sewing pattern UV space to generate a comparable texture map T for each baseline. Table 1 shows OmniFabric excels in all metrics, particularly in preserving global structural coherence and eliminating projection artifacts.

Table 2. **User study for real-world data.** We compare with baselines on real-world input images in 4 aspects: overall quality, fidelity, back-view correctness and global coherence.

	Overall rank ↓	Fidelity ↑	Back-view ↑	Global Coherence ↑
FabricDiffusion [Zhang et al. 2024]	3.40	1.75	2.69	3.15
Paint3D [Zeng et al. 2024]	3.41	1.86	2.81	3.03
Hunyuan3D-2.0 [Zhao et al. 2025]	2.10	3.43	3.41	3.50
OmniFabric (ours)	1.09	4.44	4.35	4.41

5.3.2 Real-world data. Besides comparing with other methods in Figure 6, we conduct user study to evaluate the performances of the methods due to the absence of real-world dataset of textured sewing patterns. As detailed in Table 2, the study involves 13 participants evaluating results from 10 real-world image inputs based on four criteria: overall quality ranking (1 to 4, lower is better), fidelity to the input appearance, back-view correctness, and global texture coherence (1 to 5, higher is better). OmniFabric achieves the best results across all criteria, demonstrating that it is preferred on real images and more faithfully preserves coherent garment textures.

5.4 Reproducibility via Open-Source Models

While using Gemini models for pose transferring and multi-view generation (Section 3.2) as default implementation of OmniFabric, we further construct a fully open-source alternative to facilitate reproducibility. To replace the closed-source models, we use FLUX2 [Labs 2025] as image generation backbone and incorporate Mistral-3 [Liu et al. 2026] for prompt up-sampling. We feed textured

Table 3. We construct a fully open-source alternative and it achieves similar performance to the default settings using closed-source models, showing the great reproducibility of our framework.

	LPIPS ↓	SSIM ↑	DISTS ↓	CLIP-s ↑
OmniFabric w/ default settings	0.092	0.868	0.121	0.963
OmniFabric w/ open-source models	0.114	0.847	0.149	0.955

Table 4. We analyze error accumulation across certain modules by replacing intermediate outputs in our model pipeline with ground-truth (GT) counterparts. “Pred.” and “GT” mean the intermediate output is provided by Gemini models and GT respectively, and “SP Norm.” refers to sewing pattern normalization. Results show that our texture normalization model effectively removes the accumulated error.

Replacement Setting	LPIPS ↓	SSIM ↑	DISTS ↓	CLIP-s ↑
(a) GT I' + GT V	0.105	0.849	0.159	0.944
(b) Pred. I' + GT V	0.121	0.837	0.176	0.935
(c) Pred. I' + Pred. V	0.144	0.804	0.166	0.938
(d) Pred. I' + Pred. V + SP Norm.	0.092	0.868	0.121	0.963

Table 5. **Ablation study** shows that our designs are essential for the texture normalization model to generate position-aligned details with high fidelity.

Setting	LPIPS ↓	SSIM ↑	DISTS ↓	CLIP-s ↑
OmniFabric (ours)	0.092	0.868	0.121	0.963
(a) w/o coarse texture map $\mathcal{T}'$	0.407	0.601	0.306	0.887
(b) w/o frontal view rendering R_f	0.097	0.850	0.129	0.962
(c) w/o position map P_{pos}	0.098	0.848	0.130	0.961

frontal view R_f and silhouettes of four orthogonal views as conditions to generate the multi-view with this alternative. Table 3 shows open-source alternative can reproduce similar performance with the same experiment settings in Section 5.3.

5.5 Error-Accumulation Analysis

We analyze error accumulation across modules in our model architecture in this section by progressively replacing intermediate outputs, specifically the reposed reference image I' and generated multi-view V , with ground-truth counterparts. As shown in Table 4, settings (a) to (c) accumulate errors across stages including texture projection, pose alignment and multi-view generation, and (d) shows that our normalization model effectively fixes these aggregated artifacts on sewing patterns.

5.6 Ablation Studies and Analyses

5.6.1 Importance of the canonical 3D and coarse texture map. As shown in Table 5-(a), without the coarse texture map, which serves as the distillation of generative priors, the performance drops significantly. This shows that the projected textures do provide essential guidance for the texture generation. Figure 8 shows that the model is randomly generating textures that “look similar” instead of pixel-aligned content without coarse texture map $\mathcal{T}'$.

5.6.2 Effect of the frontal view rendering condition. We observe that removing R_f during training leads to inferior results (Table 5-(b)), suggesting that the frontal-view rendering provides important cues for the model. In particular, R_f offers a holistic visual reference of



Fig. 7. **Limitations.** OmniFabric may synthesize mismatched textures from inaccurately generated multi-views (1st row), or generate garments with incorrect geometry even though appearance is well-transferred (2nd row).

the garment appearance, helping the model better understand global layout and consistency of the texture across sewing pattern pieces.

5.6.3 Effect of position map. As shown in Table 5-(c), without P_{pos} , model not only shows sub-optimal results, but generates textures and elements in wrong positions. As illustrated in Figure 8, a lemon is mistakenly generated in a blank region of the sewing pattern.

5.6.4 Importance of rectifying geometric distortion in synthetic dataset. The right of Figure 8 shows the necessity of creating geometric distortion in the synthetic training data. Through training to remove geometric distortion, the model can not only generate normalized structured pattern on the sleeves, e.g., the checker pattern, but generate clean appearances on all other elements, e.g., texts and logos.

5.7 Limitations

As shown in Figure 7, correctness of synthesized textures can be affected by inaccurately generated multi-views (1st row). OmniFabric is also limited to refine the mismatched geometry $\mathcal{M}_R$ predicted by ChatGarment [Bian et al. 2025] (2nd row); however, it does not change the role of our refinement model: given a sewing pattern, it normalizes the textures by reducing distortions, wrinkles, shadows, and projection artifacts, regardless of specific garment geometry. We put more discussion of limitations in the supplementary material.

6 Conclusion and Discussion

We introduce OmniFabric, a novel framework for synthesizing 3D garments with normalized and globally coherent textures. By leveraging large-scale generative priors and training a texture normalization model, we effectively disentangle intrinsic albedo from view-dependent artifacts including baked-in shadows and wrinkles. Our method bridges the gap between single-view observations and simulation-ready 3D assets, maintaining structural integrity across complex sewing patterns. For this study, we focus on the synthesis of sewing patterns with normalized textures within the GarmentCode data framework. An immediate and promising extension will be considering the joint prediction of complete PBR material maps, including roughness, metallic, and normal maps, alongside the albedo. We believe this direction will further enhance the photorealism of the reconstructed garments under diverse environmental lighting, providing even greater utility for immersive digital content creation.

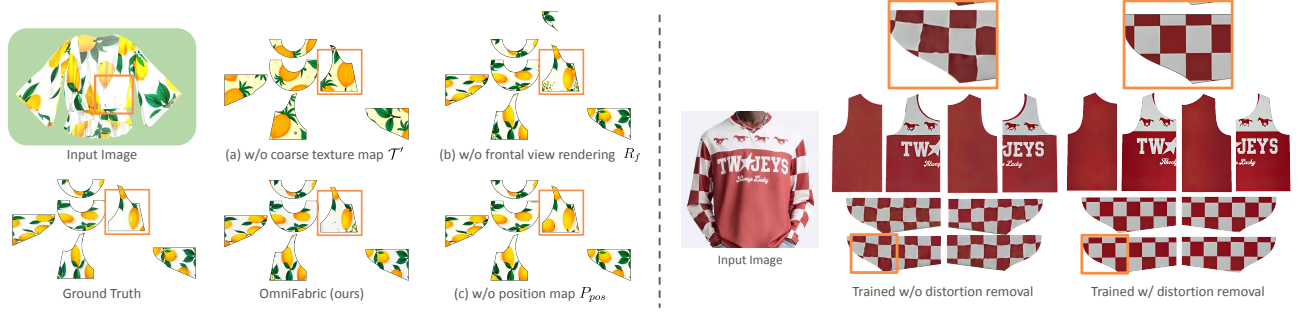


Fig. 8. **Qualitative analysis of key design mechanisms.** We prove the effectiveness of the core designs of our texture normalization model. Without $\mathcal{T}'$, the model fails to generate consistent appearance but arbitrary textures with similar colors and elements. We also observe that R_f and P_{pos} are both essential conditions for the model to generate pixel-aligned content observed from the input image (left figure). We also show that our method can effectively rectify distortion when trained with distortion-augmented dataset (right figure).

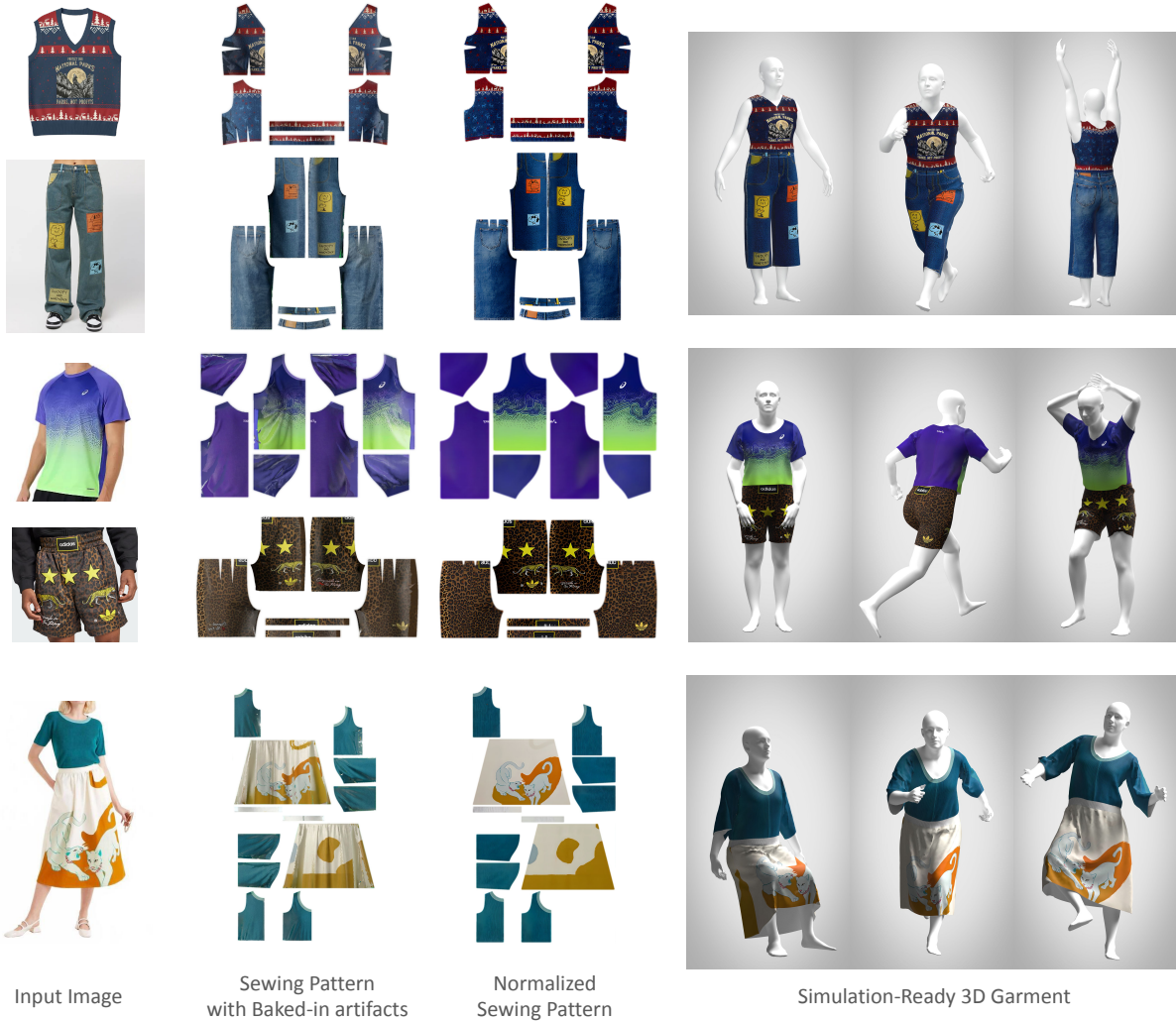


Fig. 9. **Simulation results of our generated 3D garments.** 3D garments generated by our method can be directly simulated under varying human pose and lighting conditions and show convincing results.

References

- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).
- Raphael Bensadoun, Yanir Kleiman, Idan Azuri, Omri Harosh, Andrea Vedaldi, Natalia Neverova, and Oran Gafni. 2024. Meta 3d texturegen: Fast and consistent texture generation for 3d objects. *arXiv preprint arXiv:2407.02430* (2024).
- Siyuan Bian, Chenghao Xu, Yuliang Xiu, Artur Grigorev, Zhen Liu, Cewu Lu, Michael J Black, and Yao Feng. 2025. Chatgarment: Garment estimation, generation and editing via large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2924–2934.
- Fred L. Bookstein. 2002. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence* 11, 6 (2002), 567–585.
- Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. 2023. Textfusion: Synthesizing 3d textures with text-guided image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4169–4181.
- Cheng-Hsiu Chen, Zheng-Wei Su, Min-Chun Hu, Chih-Yuan Yao, and Hung-Kuo Chu. 2024. Panelformer: Sewing pattern reconstruction from 2D garment images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 454–463.
- Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. 2023b. Text2tex: Text-driven texture synthesis via diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*. 18558–18568.
- Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023a. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 22246–22256.
- Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. 2022. Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. *Advances in neural information processing systems* 35 (2022), 30923–30936.
- Wei Cheng, Juncheng Mu, Xianfang Zeng, Xin Chen, Anqi Pang, Chi Zhang, Zhibin Wang, Bin Fu, Gang Yu, Ziwei Liu, et al. 2025. Mvpaint: Synchronized multi-view diffusion for painting anything 3d. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 585–594.
- CLO Virtual Fashion Inc. 2024. CLO. <https://www.clo3d.com>. Version 7.2.
- Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. 2020. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence* 44, 5 (2020), 2567–2581.
- Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618* (2022).
- Google. 2025. Gemini 3 Pro Image. <https://gemini.google.com/>.
- Google DeepMind. 2025. Veo: Google’s most capable video generation model. <https://deepmind.google/models/veo/>. Accessed: September 8, 2026.
- Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2021. Real-time deep dynamic characters. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–16.
- Kai He, Kaixin Yao, Qixuan Zhang, Jingyi Yu, Lingjie Liu, and Lan Xu. 2024. Dresscode: Autoregressively sewing and generating garments from text guidance. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–13.
- Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. 2022. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *arXiv preprint arXiv:2205.08535* (2022).
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.
- Christopher J Hughes, Radek Grzeszczuk, Eftychios Sifakis, Daehyun Kim, Sanjeev Kumar, Andrew P Selle, Jatin Chhugani, Matthew Holliman, and Yen-Kuang Chen. 2007. Physical simulation for animation and visual effects: parallelization and characterization for chip multiprocessors. *ACM SIGARCH Computer Architecture News* 35, 2 (2007), 220–231.
- Hyunwoo Hwangbo, Eun Hie Kim, So-Hyun Lee, and Young Jae Jang. 2020. Effects of 3D virtual “try-on” on online sales and customers’ purchasing experiences. *Ieee Access* 8 (2020), 189479–189489.
- Dong-Eun Kim and Karen LaBat. 2013. Consumer experience in using 3D virtual garment simulation technology. *Journal of the Textile Institute* 104, 8 (2013), 819–829.
- Maria Korosteleva, Timur Levent Kesdogan, Fabian Kemper, Stephan Wenninger, Jasmin Koller, Yuhang Zhang, Mario Botsch, and Olga Sorkine-Hornung. 2024. Garment-CodeData: A dataset of 3D made-to-measure garments with sewing patterns. In *European Conference on Computer Vision*. Springer, 110–127.
- Maria Korosteleva and Sung-Hee Lee. 2022. Neuraltailor: Reconstructing sewing pattern structures from 3d point clouds of garments. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–16.
- Maria Korosteleva and Olga Sorkine-Hornung. 2023. Garmentcode: Programming parametric sewing patterns. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–15.
- Black Forest Labs. 2025. FLUX.2: Frontier Visual Intelligence. <https://bfl.ai/blog/flux-2>.
- Xinyu Li, Qi Yao, and Yuanda Wang. 2025a. GarmentDiffusion: 3D garment sewing pattern generation with multimodal diffusion transformers. *arXiv preprint arXiv:2504.21476* (2025).
- Xuan Li, Chang Yu, Wenxin Du, Ying Jiang, Tianyi Xie, Yunuo Chen, Yin Yang, and Chenfanfu Jiang. 2025b. Dress-1-to-3: Single image to simulation-ready 3d outfit with diffusion prior and differentiable physics. *ACM Transactions on Graphics (TOG)* 44, 4 (2025), 1–16.
- Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. 2026. Ministral 3. *arXiv preprint arXiv:2601.08584* (2026).
- Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. 2023. Towards garment sewing pattern reconstruction from a single image. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–15.
- Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. 2016. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhongjin Luo, Haolin Liu, Chenghong Li, Wanghao Du, Zirong Jin, Wanhu Sun, Yinyu Nie, Weikai Chen, and Xiaoguang Han. 2024. Garverselod: High-fidelity 3d garment reconstruction from a single in-the-wild image using a dataset with levels of details. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–12.
- Miles Macklin. 2022. Warp: A High-performance Python Framework for GPU Simulation and Graphics. NVIDIA GPU Technology Conference (GTC). <https://github.com/NVIDIA/warp>
- Gal Metzger, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12663–12673.
- Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. 2022. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13492–13502.
- Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. 2022. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 conference papers*. 1–8.
- Kiyohiro Nakayama, Jan Ackermann, Timur Levent Kesdogan, Yang Zheng, Maria Korosteleva, Olga Sorkine-Hornung, Leonidas J Guibas, Guanda Yang, and Gordon Wetzstein. 2025. Aipare: A multimodal foundation model for digital garments. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 8138–8149.
- Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
- Elad Richardson, Gal Metzger, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. 2023. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH 2023 conference proceedings*. 1–11.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
- Boxiang Rong, Artur Grigorev, Wenbo Wang, Michael J Black, Bernhard Thomaszewski, Christina Tsalicoglou, and Otmar Hilliges. 2025. Gaussian garments: Reconstructing simulation-ready clothing with photorealistic appearance from multi-view video. In *2025 International Conference on 3D Vision (3DV)*. IEEE, 1054–1063.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* 35 (2022), 36479–36494.
- Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2304–2314.
- Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. 2020. PifuHD: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 84–93.
- Nikolaos Sarafianos, Tuur Stuyck, Xiaoyu Xiang, Yilei Li, Jovan Popovic, and Rakesh Ranjan. 2025. Garment3dgen: 3d garment stylization and texture generation. In *2025 International Conference on 3D Vision (3DV)*. IEEE, 1382–1393.
- Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. 2025a. Ominicontrol: Minimal and universal control for diffusion transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14940–14950.
- Zhenxiong Tan, Qiaochu Xue, Xingyi Yang, Songhua Liu, and Xinchao Wang. 2025b. Ominicontrol2: Efficient conditioning for diffusion transformers. *arXiv preprint arXiv:2503.08280* (2025).
- Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. 2021. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the*

- IEEE/CVF international conference on computer vision*. 1905–1914.
- Yuanhao Wang, Cheng Zhang, Gonalo Frazo, Jinlong Yang, Alexandru-Eugen Ichim, Thabo Beeler, and Fernando De la Torre. 2025. GarmentCrafter: Progressive Novel View Synthesis for Single-View 3D Garment Reconstruction and Editing. *arXiv preprint arXiv:2503.08678* (2025).
- Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.
- Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The thirty-seventh asilomar conference on signals, systems & computers, 2003*, Vol. 2. Ieee, 1398–1402.
- Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J Black. 2022. Icon: Implicit clothed humans obtained from normals. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 13286–13296.
- Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. 2024. Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4252–4262.
- Cheng Zhang, Yuanhao Wang, Francisco Vicente, Chenglei Wu, Jinlong Yang, Thabo Beeler, and Fernando De la Torre. 2024. FabricDiffusion: High-fidelity texture transfer for 3D garments generation from in-the-wild images. In *SIGGRAPH Asia 2024 Conference Papers*. 1–12.
- Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.
- Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. 2025. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202* (2025).
- Feng Zhou, Ruiyang Liu, Chen Liu, Gaofeng He, Yong-Lu Li, Xiaogang Jin, and Huamin Wang. 2025. Design2GarmentCode: Turning design concepts to tangible garments through program synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 23712–23722.