Garment3DGen: 3D Garment Stylization and Texture Generation Supplementary Material

Nikolaos Sarafianos, Tuur Stuyck, Xiaoyu Xiang, Yilei Li, Jovan Popovic, Rakesh Ranjan
Meta Reality Labs
nsarafianos.github.io/garment3dgen

We refer the interested reader to the supplemental video where we provide a wide variety of results ranging from image/text to 3D textured garments as well as applications of our method in downstream tasks such as physics-based cloth simulation, hand-garment interaction in VR using a headset and sketch to 3D garment reconstruction. Below we provide some additional details regarding the implementation of our key components as well as some additional ablation studies to showcase the impact of our design decisions.

Garment3DGen General Details

We believe that our approach provides three key insights that will be valuable to the community:

1. Mesh-based deformations provide the right properties to generate (or stylize) new garments that we can utilize for downstream tasks other than rendering.
2. A text-prompt or a single image alone cannot provide enough guidance to generate the desired garment exactly the way a user might want it. This is evident from the results of WordRobe [6] which despite its mesh-quality the generated garments do not follow the provided text prompt.
3. 3D supervisions, if done right, can provide strong enough supervision signal in order to generate the desired garments with the proper topology and structure.

Our approach builds upon these insights and introduces a novel yet simple solution to generate high-quality, physically plausible garments. As input to the method, we require only a single garment image (or alternatively, a text prompt that can generate this image using a text to image model) and a base garment template mesh. The input image needs to contain a single piece of clothing, captured from a semi-frontal viewpoint with its pose being as occlusion free as possible. A person can be wearing this garment or there might be more than one piece of clothing in the image in which case we perform semantic segmentation (using SAM) to obtain the garment. The template mesh is not required to be similar to the image guidance. For example, we demonstrate results where our method can go from a shirt to a puffer jacket, from a tank-top to a dress or even a T-shirt to a fantastical sea armor. Note that the closer the base mesh is to the target geometry, the easier the task is. For example, starting from a dress mesh to go to a shirt is a difficult task while starting from something closer to the target simplifies this problem.

Algorithm 1: Automatic View Selection

Input: an input mesh M_{def} with UV texture T with front and back views painted, a binary mask T^B marking the painted pixels of T, and N uniformly distributed candidate views $\{C_i\}_{i=1}^{N}$;
for *number of iterations* **do**
  Calculate the binary mask T_i^B for each view i from T^B: $\{T_i^B\}_{i=1}^{N}$;
  Select the least painted view C_j: $j \leftarrow \arg\min_{i=1}^{N} \sum T_i^B$;
  Generate the appearance image I_i and update T^B;
end

Automatic View Selection: The goal of this algorithm is to automatically select the least-painted view and paint it. In this way, we can solve the 3D texture generation problem in a coarse-to-fine manner, and ensure the overall consistency. Alg. 1 provides a detailed description of the automatic view selection algorithm: given the input UV texture T with painted front and back views, there could be N candidate views. We maintain a binary mask T^B that marks the painted pixel as 1, and unpainted pixel as 0. We can select the view with the most unfilled pixels as the next view to generate the appearance, and update the binary mask T^B. This process is repeated iteratively until most of the pixels are painted, or reaching a certain iteration number.

Mesh Deformer Details

- **Alignment**: Using the nvdiffmodelling library the base mesh is aligned to the target mesh using the unit-size function that moves/rescales the input to match bounding boxes.

Table 1. Comparisons of different texture estimation methods. The runtime is measured on a single NVIDIA H100 GPU.

Method	Pros	Cons	Runtime
Mesh2Tex [1]	Infinite resolution & Global consistency	One model per class & No fine details	8mins
TEXTure [5]	Shape-aware & Local consistency	Bad Global consistency & texture artifacts & Janus	2mins
Text2Tex [2]	Shape-aware & Local consistency	Bad Global consistency & color/pattern shifts & Janus	5mins
Garment3DGen	Shape-aware & Local/Global consistency & Very fast	Disharmonious patterns & Janus	~4.5secs

- **Deformation**: We use the same formulation with the Neural Jacobian Fields(NJF) as described in Sec.3.1 of their paper and Sec.3 of TextDeformer(TD). Once the deformation map is obtained using Eq.(1) we obtain the updated vertices of the input mesh. While NJF could deform a garment to have a different pose it's not possible to change its style (t-shirt-¿sea-creature-armor) because of the supervision signals. Our goal was not to do mesh-registration but instead stylize input base garment templates via deformation. Hence NJF was chosen for its versatility across heterogeneous mesh collections, because it's triangulation-agnostic and it provided a flexible and easy-to-use framework to accomplish our goal in a fast plug-n-play manner.
- **Losses**: Our goal is to deform the base mesh enough to match the pseudo-ground-truth mesh extracted from the image but not fully, since if that was the case we'd end up with a watertight mesh unable to be fit to parametric bodies and simulated. Hence we opted for point-to-point meshes for the 3D supervision as well as embedding and image-based losses in the 2D space.

Texture Details

This approach prioritized filling in the large areas first before moving on to smaller and more occluded regions.

Texture Comparisons: In Table 1 we provide a comparison between the pros and cons of recent texture estimation approaches. Our approach is significantly faster compared to past works due to key optimizations described in the main paper while maintaining good local and global consistency. Similar to past works one can notice the Janus problem appearing in texture maps, which can be handled by training a multiview generation diffusion model with more explicit camera pose injection in the future works.

Quantitative Comparisons - Details: In terms of garment base meshes we utilized the publicly available dataset provided by DiffAvatar which comprises 6 template geometries. For each one of the 6 garments we provide 4 different image inputs (2 real and 2 AI-generated) and to quantitatively evaluate the different approaches we render the untextured outputs of all methods from 36 views. No prompts are used during the quantitative evaluation procedure.

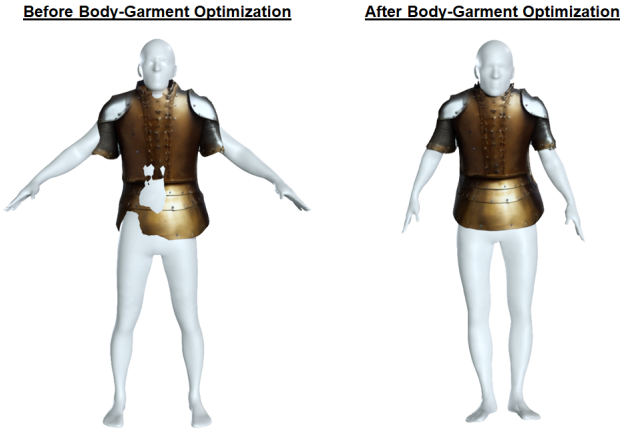

Figure 1. **Fitting a parametric body to a generated garment**: We start with the generated textured 3D garment (in this case a medieval armor) and a parametric body in its canonical pose (left). After the body-garment optimization process the body pose and shape parameters are optimized such that the generated garment can fit in the body accurately without penetrations.

Garment Fitting to Parametric Bodies

While the aforementioned supervisions and regularizations aim to ensure that the quality of the generated garments will be satisfactory for simulation, the produced garment will still need to be scaled, positioned and oriented to fit the parametric body [4] to be draped on and simulated. To accomplish this task, we run an optimization procedure during which the generated garment remains fixed in the generated pose and the pose and shape of the parametric body are transformed such that the garment can accurately fit the body. This optimization process shown in in Fig. 1, starts with a rigid transformation and scaling of the body and continues with an optimization of the body pose and shape using the Chamfer distance loss.

Rigid Transformation and Scaling: The optimization is initialized by applying a rigid transformation (rotation, translation and scaling) of the parametric body model. This step roughly aligns the body with the garment.

Pose and Shape Optimization: The body pose and shape parameters are optimized to minimize the Chamfer distance between the body mesh and the garment mesh.

Collision Handling: After body model optimization, an ad-

Figure 2. **Impact of the pre-trained CLIP on garment data**: We disable all other supervisions and explore the impact of a pre-trained CLIP model on fashion data versus using the regular model to enforce embedding supervisions. We observe that regular CLIP embeddings result in distorted and unusable geometries regardless of whether the input is a text prompt or an image.

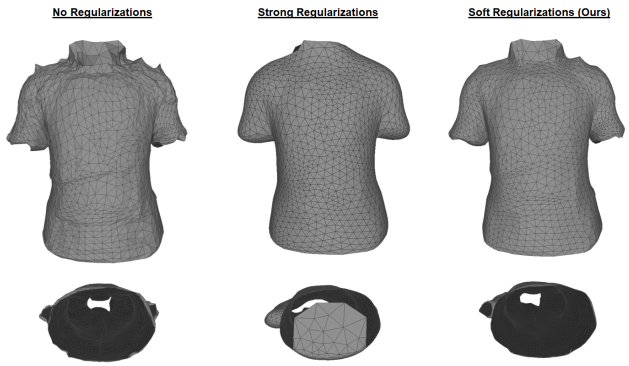

Figure 3. **Impact of regularizations on the final armor geometry**: Enforcing no regularizations (Laplacian smoothing, penalization of small triangles etc.) on the output mesh results in a crisp output armor mesh with arm/body holes but its quality is not at the level required to perform physics-based simulation. On the other hand, enforcing strong regularizations results in overly smoothed meshes with closed holes. Our output strikes a good balance between capturing those fine-level details that make an armor geometry look like one yet making it suitable for downstream tasks.

ditional step is performed to resolve potential body-cloth collisions. This is achieved by minimizing an interpenetration loss that penalizes any intersections between the body mesh and the garment mesh.By combining these steps, the garment mesh can be accurately fit to the parametric body model, enabling realistic draping and physics-based cloth simulations for downstream applications.

Additional Ablation Studies

Supervisions: When it comes to supervisions we observed that: a) utilizing regular CLIP embeddings provides minimal supervision guidance when it comes to garments and results in poorly deformed meshes which is why we opted for a garment fine-tuned model as shown in Fig. 2, b) explicitly enforcing multi-view consistency losses is not necessary as 3D supervisions can provide better guidance, and iii) there is a trade-off between allowing for heavy garment stylizations/deformations and maintaining a good mesh quality that can be used later on as shown in Fig. 3. Thus we propose to use a combination of 3D supervisions to guide the deformation process to obtain an accurate 3D shape along with 2D and embedding supervisions to obtain the fine-level details of the garment that the 3D pseudo ground-truth might fail to capture. We train for ∼1000 iterations with the weights of each loss described in Eq. (6) as follows: $w_{CD} = 20, w_{Lap} = 1, w_{triag} = 1, w_{2D} = 2, w_E = 4$ with the weight of W_{CD} gradually decreasing after the first 500 iterations once we have obtained a fairly accurate pose and shape of the garment to allow for the remaining of the supervisions to distill the fine-level garment details. Note that if we were to enforce strong 3D supervisions we would end up with deformed garments that would have no holes for the body, arms and head.

Figure 4. **Impact of Texture Module**: given the left image as a condition, the texture enhancement module enriches the details and enhances the overall image quality by effectively utilizing the powerful 2D priors.

Adding Components one at a time: As described in the main paper we conducted an ablation study depicted in Fig. 6. We start with the off-the-shelf TextDeformer which takes a text prompt and a base-mesh and deforms this to match the target text. Text prompts are not ideal to capture the fine-level details of a garment as there can be many "medieval armors". In addition, a pretrained CLIP model is not capable of capturing the subtle differences between a "jacket" and a "puffer jacket". To overcome this limitation we adapt TextDeformer to take image inputs as guidance (ImageDeformer) and observe that the deformed geometries are improved. Nonetheless, they still fail to capture the details of the image. By swapping the original CLIP model and introducing a model that is fine-tuned on fashion data we observe that details are better preserved across garments. Noting that image-based reconstruction methods can accu-

Figure 5. **3D Garment Generation**: Given an image (1^{st} row) or a text prompt (2^{nd} row) as guidance and a base geometry mesh (bottom left inset) that can be far from the target we generate high-quality textured 3D geometries of both real as well as fantastical garments.

rately capture geometry but produce coarse and watertight meshes that are unsuitable for subsequent tasks, we utilize these meshes as pseudo ground-truth for our proposed approach. Our Garment3DGen results in garments that faithfully follow the image guidance while containing wrinkles and fine details. However, the quality of the output geometry is not always ideal for physics-based downstream tasks because they produce poorly conditioned triangles which result in instabilities when simulated in addition to poorly tessellated geometry which will result in unnatural fabric behavior. Because of this, we introduced additional 3D supervisions that preserve a better mesh quality.

Texture Module: The impact of the texture enhancement module is shown in Fig. 4. The textures directly synthesized by 3D generation models are low-resolution, smooth and over-simplified, which is due to the scarcity of high quality 3D training data. Thus, the texture enhancement module aims to effectively utilize the 2D priors learned from the large high-quality image dataset. After our image-conditioned image enhancement, we bring back vivid details to the texture, improving the perceptual quality.

Limitations

Garment3DGen handles a variety of garment types both realistic and fantastical. Due to the requirement of a template mesh, there is a limitation on what garments can be generated whilst still providing distortion-free meshes. This can be mitigated by providing a more diverse template library. Our estimated textures, while faithful to the image, sometimes do not fully preserve fine-level details. We plan to address this by tuning the texture enhancement module to be conditioned on the reference image across all views while maintaining its multi-view color consistency properties. Finally, it is worth noting that the closer the input template mesh is to the target garment, the easier the deformation task becomes. We currently select the closest template manually but performing automatic retrieval based on the image could be tackled in future work.

Additional Results

Comparisons with SewFormer: In Fig. 7 we provide a qualitative comparisons against SewFormer [3] which predicts garment patterns from a single RGB image.To facilitate a meaningful comparison, we use a physics-based cloth simulator to sew the generated panels together, creating an assembled 3D garment. In the top row T-shirt example, SewFormer incorrectly generates pattern pieces for a dress instead of the requested T-shirt. After assembly, it's clear that the predicted garment resembles a dress rather than the intended T-shirt. The second example shows that while SewFormer works reasonably well for certain garments, it fails to fully match the guidance, missing sleeves in this case. Both examples demonstrate that SewFormer produces results that don't accurately match the image guidance, and its method for obtaining 3D assembled garments is more complex, requiring specialized software to sew individual pattern pieces together. Additionally, SewFormer is limited to producing V-neck designs, whereas our method correctly follows the visual guidance.

Additional Qualitative Comparisons: In Fig. 5 we provide multi-view renders of our 3D textured garments that we generate from text prompts and an image guidance. From these results we gain the following insights: a) Garment3DGen works just as well with fantastical garments (armors or dresses) that are outside the regular garment distribution, b) our texture estimation module results into high-quality textures that closely match the input text prompt and c) our output geometry does not have to be similar to the input base mesh. Finally in Fig. 8 we showcase the plethora of applications that Garment3DGen has ranging from text/image/sketch to simulation-ready 3D garments to hand-garment interaction in a VR environment using the on-device hand-tracking.

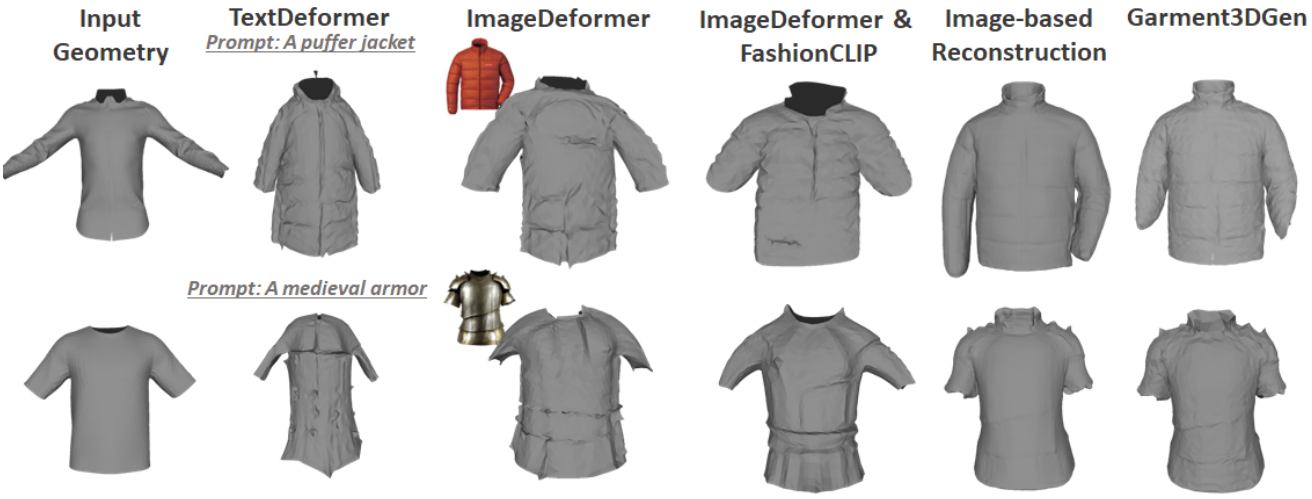

Figure 6. **Ablation Study**: Starting from a base input mesh we showcase that our key contributions result in deformed geometries that capture the input image guidance, comprise fine-level garment details and are suitable for our downstream tasks.

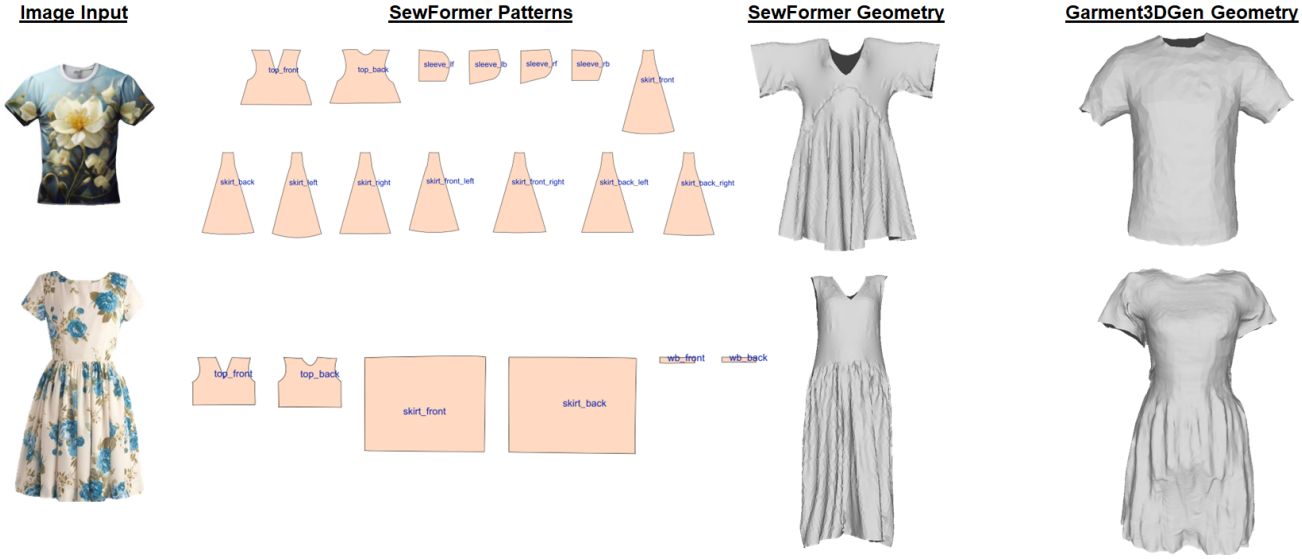

Figure 7. **Qualitative Comparisons**: SewFormer [3] produces 2D garment patterns from a single RGB image input. We further process these panels by virtually sewing them together at the seams to create assembled 3D garments, which enables us to perform a qualitative comparison. We showcase two garment examples and demonstrate that Garment3DGen produces results that are more faithful to the image input with a less complex pipeline.

Acknowledgments

We thank Michelle Guo, Will Gao and Astitva Srivastava for their help on running baseline comparisons.

References

[1] Alexey Bokhovkin, Shubham Tulsiani, and Angela Dai. Mesh2tex: Generating mesh textures from image queries. In *ICCV*, 2023. 2

[2] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023. 2

[3] 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. 4, 5

[4] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 2

[5] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3D shapes. *arXiv preprint arXiv:2302.01721*, 2023. 2

Figure 8. **Garment3DGen Applications**: We showcase various applications of our proposed approach.

[6] Astitva Srivastava, Pranav Manu, Amit Raj, Varun Jampani, and Avinash Sharma. Wordrobe: Text-guided generation of textured 3D garments. In *ECCV*, 2024. 1