Supplementary for WordRobe (Text-Guided Generation of Textured 3D Garments)

Astitva Srivastava[1], Pranav Manu[1],
Amit Raj[2], Varun Jampani[3], and Avinash Sharma[1,4]

[1]IIIT Hyderabad [2]Google Research [3]Stability AI [4]IIT Jodhpur
{astitva.srivastava, pranav.m}@research.iiit.ac.in
amitrajs@google.com, varunjampani@gmail.com, avinashsharma@iitj.ac.in

Link to Project Page.

1 Training & Implementation Details

Our garment generation framework uses DGCNN [1] as the garment encoder ξ which takes $20,000$ points sampled on the garment surface as input. The decoders D_{coarse} & D_{fine} are implemented as MLPs, both having 5 hidden layers of 512 neurons each. We also use conditional Batch Normalization [2] for conditioning on the garment latent vector while decoding. We train ξ and D_{coarse} together in the coarse training stage for 20 epochs, and D_{fine} separately in the fine stage for 10 epochs. The value of hyperparameters λ_{dist}, λ_{grad} & λ_{latent} involved in training objectives is 1.0, 0.3 & 0.2, respectively. The mapping network MLP_{map} is modelled using a 10 layer MLP with a skip-connection from the input layer to 4^{th} hidden layer. All the networks are trained using AdamW [3] optimizer on an NVIDIA RTX 4090 GPU.

We use OptCuts [4] for mesh parametrization as it eliminates the possibility of overlap in UV space. For projecting textures, we take a texel of UV map and identify on which face (triangle) of the mesh it lies. Once we have the triangle, we take 3D vertex positions of the triangle and calculate the 3D position of the texel using Barycentric interpolation. This 3D point is then projected on π_{rgb} and bilinearly interpolated to get the color information which is then stored into the corresponding texel of the UV map. Doing this process for every texel gives us the final texture UV map and, eventually, texture 3D garment mesh.

2 Text-driven Garment Editing

In this section, we describe how to enable text-driven editing of garments. Given a garment latent code ϕ, methods like [5] employ a pretrained latent code classifier to identify the garment type and to identify which dimension to interpolate in order to induce category-specific changes, e.g. manipulate sleeve lengths. Though this approach works well, it requires explicit manual annotations to enable control over interpolation. In our case, we make use of CLIP embeddings to automatically identify the dimensions to control certain aspects of the 3D garment. Given a CLIP embedding ψ corresponding to a text-prompt (say, *"skirt"*),

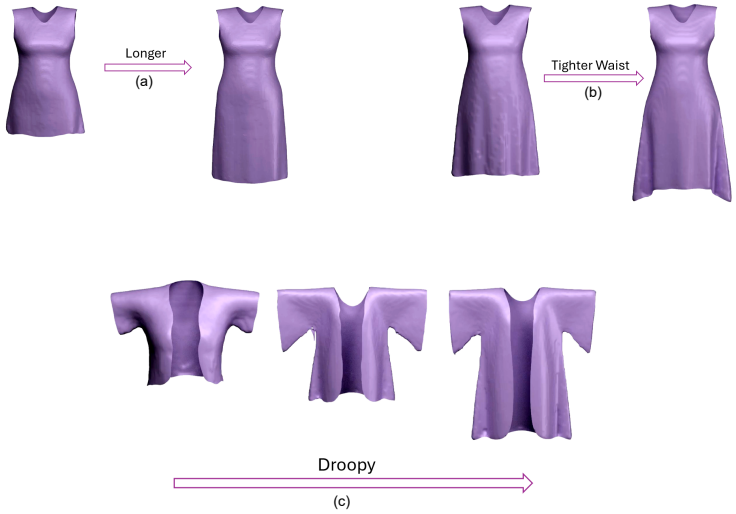

Fig. 1: Text-driven manipulation of the latent code.

MLP_Map predicts corresponding garment latent code ϕ which is decoded to obtain the initial garment. For a text prompt *"longer"*, we first compute the CLIP embedding $\psi_{feature}$. Then, we perform the following operation:

$$\psi_{edit} = w\psi + (1 - w)\psi_{feature} \tag{1}$$

In other words, we perturb the ψ in the direction of $\psi_{feature}$ ($w = 0.5$), which results in the modified CLIP embedding ψ_{edit}. MLP_{map} takes this ψ_{edit} and predicts ϕ_{mod}, which if decoded results in the 3D garment which resembles *"long skirt"*. However, predicting the entire garment's latent code from scratch disturbs other characteristics of the original garment as well. To retain the other characteristics as much as possible and change only the *length*, we first compute the difference ϕ_{delta} between original latent code ϕ and modified latent code ϕ, i.e. $\phi_{delta} = \phi - \phi_{mod}$. We then identify top-$k$ values in ϕ_{delta}, which denote the corresponding dimensions to change in order to achieve the desired modification. Thus, perturbing the garment latent code ϕ along only these k dimensions results in intended modification while maintaining other characteristics of the garment (choosing the value of k is flexible and is driven by the user preference, we use $k = 7$). In Figure 5 of the main paper, we perform text-driven editing of ***"skirt"*** according to the keyword *"long"*, and that of a ***"cardigan"*** by introducing *"sleeves"* and then *"hood"*. Here, in Figure 1, we demonstrate similar text-based latent editing on other garment samples. The aforementioned approach does not require any manual intervention except for the amount of perturbation the end-user intends to introduce.

3 Extended Results

In this section, we present additional qualitative results and demonstrate potential applications of our framework.

3.1 Results on Complex Text Prompts

Though we aim to generate 3D garments with simplistic, user-friendly prompts, our framework supports detailed and complex prompts as well, as shown in Figure 2 & Figure 3. In Figure 2, we comprehensively describe the geometry and appearance of the garment to be modelled. Our framework is able to understand detailed prompts and decode them correctly to generate expected geometry and appearance. Similarly, in Figure 3, detailed prompts for separate garments can be given to generate the final combined clothing (by merging the UDF of each garment using [6]).

3.2 Sketch-guided 3D Garment Generation & Editing

Our weakly supervised strategy to train the mapping network on CLIP-embeddings allows us to enable various other interesting applications apart from text-driven generation and editing. Since CLIP space is joint image & text embedding space, we can pass any image which represents a garment and generate the corresponding 3D garment geometry. Figure 4 demonstrate the way to generate and edit the 3D garments by sketching or scribbling the garments. We pass the garment sketch image to the CLIP and get the corresponding CLIP embedding vector, which is then further passed to the Mapping Network to predict the associated garment latent code. This latent code is then decoded by the coarse and fine decoders to generate the 3D garment geometry. On modifying a part of the sketch, only the corresponding 3D garment part undergoes significant change, while we observe small insignificant changes on the remaining parts of the garments. For textures, we follow the proposed Texture Synthesis module on the 3D geometry obtained via sketches, as shown in Figure 5. As can be seen from the figure, the geometry of the output 3D garments accurately adheres to the input sketch (at least at a coarser level) in terms of shape and semantics. This feature enables a more controllable and descriptive way to generate the 3D garments compared to text prompts.

3.3 3D Garments Extraction from Images

We extend the idea from the previous subsection to demonstrate the 3D garment recovery from in-the-wild random garment images. Since we have trained the Mapping Network on data generated by ControlNet, it has seen a wide variety of garment textures and lighting conditions encoded within the CLIP embedding vector. This allows us to pass any internet image to CLIP's image encoder, obtain the corresponding CLIP embedding, and feed it to the Mapping Network

Fig. 2: Generating high-quality textured 3D garments with detailed text-prompts.

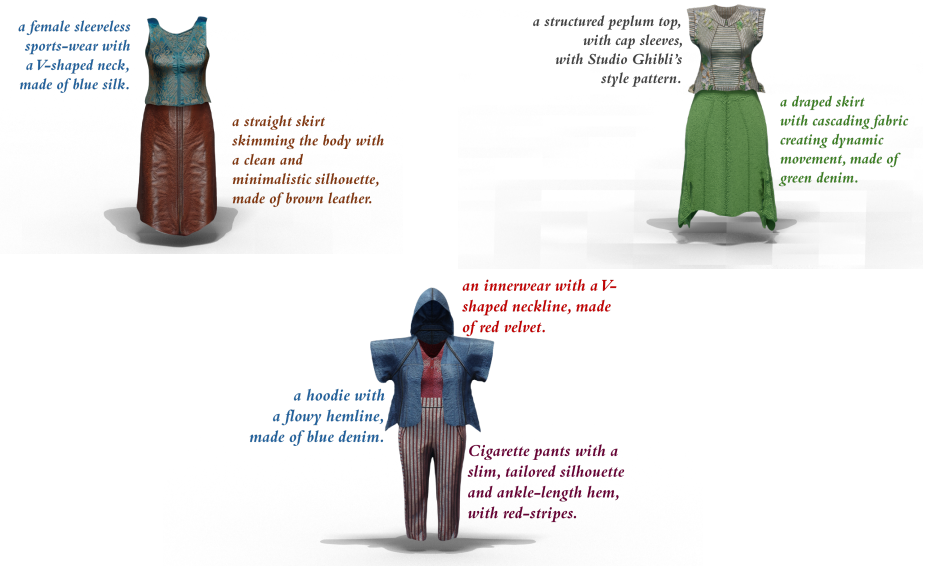

Fig. 3: Generating composition of garments using detailed text-prompts.

Fig. 4: Sketch-guided editing of the 3D garments.

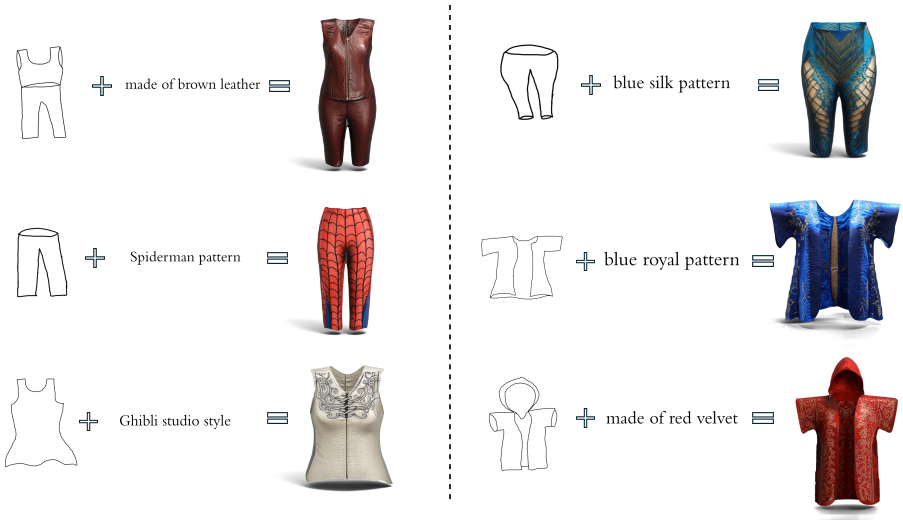

Fig. 5: Generating high-quality 3D garments by combining sketch (for geometry) and text (for texture).

to obtain the garment latent code for 3D garment generation. Figure 6 shows the results of the aforementioned approach, where we generate 3D garments from arbitrary garment images whether they contain humans or not.

4 Extended Evaluation

In this section, we demonstrate generalization of our method on a different dataset and explain the details regarding qualitative user study.

4.1 Generalization on CLOTH3D

We train our encoder-decoder architecture for learning the garment latent space on *unposed* 3D garments from [7] dataset. However, CLOTH3D [8] dataset is a widely popular choice for existing state-of-the-art methods like [5]. Therefore, we demonstrate generalization pn both **topwear** and **bottomwear** classes of CLOTH3D dataset after training only on [7] dataset and computing the standard evaluation metrics, namely Point-to-Surface distance (P2S) and Chamfer Distance (CD) on the test set of CLOTH3D. We define both the metrics below:

- **P2S:** Point to surface distance is the shortest distance between a point and a surface. [9]
- **CD:** It is defined as the sum of squared distances of nearest neighbor correspondences of the two point clouds. [10]

Fig. 6: Generating 3D garments using a reference image

We report both the metrics in Table 1, while comparing with DrapeNet [5], which has been trained specifically on CLOTH3D. As can be observed from the table, we perform at par, if not better, than DrapeNet, even without training our method on CLOTH3D. This justifies that **(a)** [7] has a more diverse and better training distribution than widely popular [8] dataset, in terms of garment geometry learning; and **(b)** our encoder-decoder is not overfitted to the samples in [7] dataset and can generalize to unseen garment types. However, it is important to note that both the datasets, CLOTH3D and [7] are synthetically generated, as it is very challenging to capture 3D real-world garments in canonical pose.

Table 1: Quantitative evaluation of garment encoding-decoding framework on both **topwear** and **bottomwear** garments from CLOTH3D dataset. Please note that for this experiment, we train our method on [7] dataset and evaluate on CLOTH3D, while we train and evaluate DrapeNet on CLOTH3D.

Method	**Evaluation on CLOTH3D**	
	CD(topwear)↓	**P2S(topwear)↓**
DrapeNet (trained on CLOTH3D)	1.522	**0.631**
Ours (trained on [7])	**1.491**	0.635
	CD(bottomwear)↓	**P2S(bottomwear)↓**
DrapeNet (trained on CLOTH3D)	1.585	0.739
Ours (trained on [7])	**1.568**	**0.703**

4.2 User Study

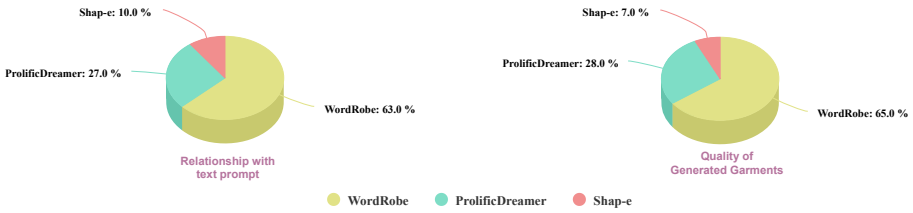

Fig. 7: Distribution of preferences in the qualitative user study.

For the subjective evaluation of our method, we perform a qualitative user study among 67 participants. First, the participants were presented with the results of our method and were asked two questions :

– **How would you rate the relationship between the input text prompt and the generated result on a scale of 1 to 3? [1-not related, 2-**

somewhat related, 3-highly related]

– **How would you rate the quality of the results in general, on a scale of 1 to 5? [1-very bad, 2-bad, 3-acceptable, 4-good, 5-very good]**

Our method achieves an average rating of **2.57** on a scale of $[1-3]$ in terms of the relationship between the result and the input text prompt and an average rating of **4.01** on a scale of $[1-5]$ in terms of quality of the generated 3D garment.

All the participants were also asked to select one of the methods among [11], [12] and WordRobe on two basis − relationship between the result & text-prompt, and overall quality of the generated garment. About 63% of the participants prefer WordRobe, 27% prefer [12], and 10% prefer [11] in terms of the relationship between the result and text prompt. In terms of quality, 65% of the participants prefer WordRobe, 28% prefer [12], and 7% prefer [11].

Finally, we also asked participants to choose between Text2Tex [13] and WordRobe for the text-driven synthesis of textures over existing meshes. About 54% participants opted for WordRobe when it comes to the relationship between the result and input text prompt, and about 76% opted for WordRobe when it comes to the quality of the generated 3D garment.

5 Discussion

5.1 Why *unposed*?

Unposed simply means garments in canonical T-pose, which is a standard rest pose defined for human characters. We aim towards generating *unposed* 3D garments because it has several advantages. First, the garments are free from any pose-specific deformations, which is undesirable while defining garment characteristics, as symmetries are important in the garment designing process, which gets disturbed when garments undergo pose-specific deformations. Second, standard animation or dynamic character simulation pipelines keep their characters in the canonical pose for rigging and skinning purposes; therefore, it makes sense to have garments also defined in the canonical pose. Additionally, recent learning-based cloth simulation methods [14–16] also require garments to be in T-pose/canonical pose, thereby making the 3D garments generated by our framework directly usable in all such scenarios.

5.2 Front-back vs Multiview Projection for Texture Synthesis

As stated in the main paper, we use front-back as a natural choice for partitioning a 3D garment to reduce the number of visible seams on the mesh. Figure 8 shows a result with 4 views (instead of just front and back), where prominent seams can be seen (circled in red) on salient regions of the garment mesh. Additionally, we demonstrate better global consistency of the proposed front-back projection via superior CLIP scores (Table.2 main paper), evaluated across *multiple random*

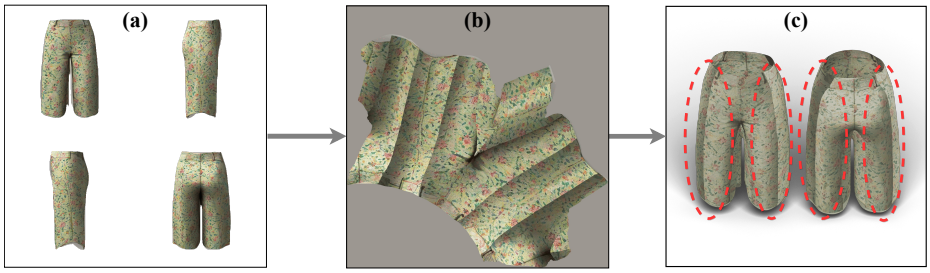

Fig. 8: **(a)** View-composited image (with 4 views) generated using depth-conditioned ControlNet [17]. **(b)** Corresponding UV texture map with multiple seams. **(c)** Noticeable seams on the salient parts of the garments.

views, surpassing SOTA method Text2Tex [13] (which uses multiview projection), while also being optimization-free **(13 *times faster*)**. This improvement is also justified through qualitative comparison (Figure.8 main paper) and the user study.

5.3 Contrast with Recent 3D Generative Methods

Recent advancements in 3D generative deep learning have given rise to zero-shot text-to-3D or image-to-3D generative models. However, the geometries obtained from such methods are plausible but nowhere near production-ready, especially when it comes to modelling complex geometries with openings, e.g. garments. Since most of these methods model 3D surfaces as SDFs, they fail to handle open garment surfaces. As shown in Figure 9, we highlight the output quality of a recent state-of-the-art 3D generative method [18], where noisy and poor garment geometries can be seen. Contrast that with the output of our method in Figure 6 on the same garment images, which generate plausible and ready-to-use 3D garments. This performance difference is due to the obvious fact that our method is specialized for 3D garments as compared to generic text-to-3D or image-to-3D methods, which can model arbitrary objects but with poor geometric quality.

5.4 Text Prompts for Evaluation

Due to the lack of any 3D garment dataset with text annotations, in order to come up with several diverse text prompts describing the geometry and appearance of the garments, we leverage large language models with powerful language generation capabilities. We asked ChatGPT-3.5 to generate 300 random text prompts describing different garment styles while just focusing on the geometry and & 300 text prompts describing textures only. More specifically, after several trials and errors, we came up with the following two prompts:

– *"Write* 300 *descriptive text prompts to describe various clothing styles. Don't describe color or texture information, just the valid geometrical details, such*

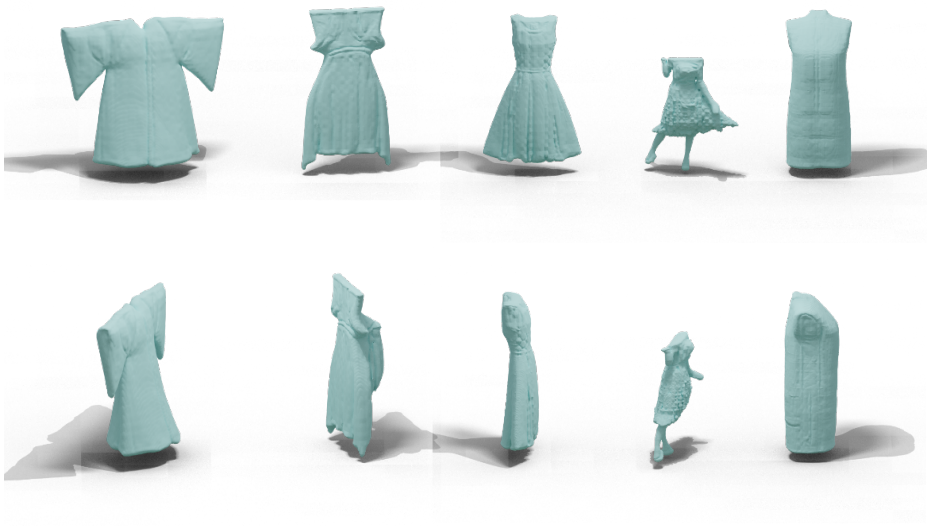

Fig. 9: Output from state-of-the-art 3D generative method [18], after giving just garment images as the input. The geometry from the input view looks plausible (first row) but is poor and unusable when observed from the side (second row).

as size, shape, curvature and so on (do not include knit-type or material type). Remove bullet points and put every point in a new line. Make sure prompts are highly diverse and distinct from each other."

– *"Write* 300 *descriptive text prompts to describe various textures and patterns that can be put onto clothing. Be creative and make sure to take inspiration from famous fashion designers. Remove bullet points and put every point in a new line. Make sure prompts are highly diverse and distinct from each other."*

We manually verified the correctness, quality and diversity of the text prompts. We report the CLIP score and conduct the user study on the prompts generated using the aforementioned approach. We will release the exact prompts along with the source code in the public domain.

5.5 Limitations & Future Directions

WordRobe generates high-fidelity 3D garments with high-quality textures at an unprecedented speed and scale. However, there are certain limitations to our approach that we wish to overcome in future work. One of the drawbacks of front-back orthographic rendering is the loss of information around the tangential regions (see Figure 10). In order to fill in the missing details, we employ an off-the-shelf inpainting method, which occasionally leaves blurry seams along the

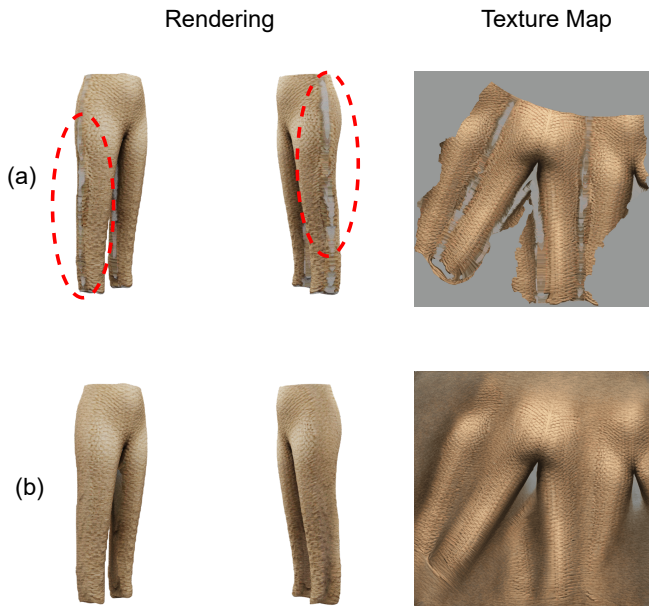

Fig. 10: (a) Missing tangential information due to orthographic projection. (b) Details inpainted within the texture map.

boundary of the front & back regions. Another limitation is the lack of fine-grain geometrical details on garment parts (e.g. pockets, buttons, etc.) which makes it challenging to model using implicit representations such as UDF.

In the context of text-driven texture synthesis, one major limitation that every existing method encounters is the hallucination of shadows, lights, and edges, which are purely textural and not a part of the surface of the garment. Though it may enhance the garment's appearance, but from the rendering point of view, this limits the applicability of the extracted textures to new lighting environments. As a part of future work, we wish to explore relighting to get rid of false shading, retaining the true albedo of the garment geometry. We would also like to enable support for layered clothing and material property of the garments. We hope our work paves a path towards high-fidelity production-ready garment generation from natural language prompts.

References

1. 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)*, 2019. 1
2. Harm de Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron Courville. Modulating early visual processing by language, 2017. 1
3. Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 1
4. Minchen Li, Danny M. Kaufman, Vladimir G. Kim, Justin Solomon, and Alla Sheffer. Optcuts: Joint optimization of surface cuts and parameterization. *ACM Transactions on Graphics*, 37(6), 2018. 1
5. Luca De Luigi, Ren Li, Benoit 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*, 2023. 1, 6, 8
6. Igor Santesteban, Miguel A. Otaduy, Nils Thuerey, and Dan Casas. ULNeF: Untangled layered neural fields for mix-and-match virtual try-on. In *Advances in Neural Information Processing Systems, (NeurIPS)*, 2022. 3
7. Maria Korosteleva and Sung-Hee Lee. Generating datasets of 3d garments with sewing patterns. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. 6, 8
8. Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020. 6, 8
9. Lingli Zhu, Antero Kukko, Juho-Pekka Virtanen, Juha Hyyppä, Harri Kaartinen, Hannu Hyyppä, and Tuomas Turppa. Multisource point clouds, point simplification and surface reconstruction. *Remote Sensing*, 11(22), 2019. 6
10. Ainesh Bakshi, Piotr Indyk, Rajesh Jayaram, Sandeep Silwal, and Erik Waingarten. A near-linear time algorithm for the chamfer distance, 2023. 6
11. Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions, 2023. 9
12. 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. *arXiv preprint arXiv:2305.16213*, 2023. 9
13. 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. 9, 10
14. Igor Santesteban, Miguel A. Otaduy, and Dan Casas. Snug: Self-supervised neural dynamic garments, 2022. 9
15. Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Neural cloth simulation. *ACM Transactions on Graphics*, 41(6):1–14, November 2022. 9
16. Artur Grigorev, Bernhard Thomaszewski, Michael J Black, and Otmar Hilliges. HOOD: Hierarchical graphs for generalized modelling of clothing dynamics. 2023. 9
17. Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 10
18. Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image, 2024. 10, 11