Supplementary Material for A Repulsive Force Unit for Garment Collision Handling in Neural Networks

Qingyang Tan[1], Yi Zhou[2], Tuanfeng Wang[2], Duygu Ceylan[2], Xin Sun[2], and Dinesh Manocha[1]

[1] Department of Computer Science, University of Maryland at College Park
{qytan,dmanocha}@umd.edu
[2] Adobe Research {yizho,yangtwan,ceylan,xinsun}@adobe.com

1 SDF Network

In this section, we will give details about the math formulation of the SDF network, the sampling scheme, and the loss functions we use for training the human body SDF network.

To train a generalized SDF network that can predict the implicit function of human bodies with different shapes and poses in real-time, we design the network to predict SDF conditioned on the SMPL [3] parameters.

$$f(\boldsymbol{x}, \vec{\beta}, \vec{\theta}) \approx \mathrm{SDF}^{M(\vec{\beta},\vec{\theta})}(x). \tag{1}$$

SMPL is a PCA model computed from a large human shape data. $\vec{\beta}$ and $\vec{\theta}$ are its shape and pose parameters. $M(\vec{\beta}, \vec{\theta})$ is the human shape reconstructed from $\vec{\beta}$ and $\vec{\theta}$.

To train the SDF network, we combine both the regression loss on sampled points in the space and the geometric regularization loss on the gradient as proposed by Park et al. and Gropp et al. [4,2]. For each garment-body pair in the TailorNet dataset, we collect three categories of SDF value samples:

1. Randomly sampled points from the body surface, with or without Gaussian disturbance. For samples right on the body surface, we also collect their normals. Note that, we can only get correct SDF gradients for the surface points which are their normals. For other points, we can estimate their gradients through analytic methods.
2. Randomly sampled points from the garment surface, with or without Gaussian disturbance.
3. Randomly sampled points inside the bounding box of the body. We use a general bounding box for all the samples with size $4m \times 4m \times 4m$, centering at $[0, 0, 0]$.

For points from the body surface without disturbance, we denote them as $\{\boldsymbol{x}_i\}_{i \epsilon I_S}$, their normals as $\{\boldsymbol{n}_i\}_{i \epsilon I_S}$. For other points, we denote them as $\{\boldsymbol{x}_j\}_{j \epsilon I_E}$.

The ground truth SDF values for all the points are $\{s_i\}_{i\epsilon I_S \cup I_E}$. We compute the loss for training SDF as:

$$\mathcal{L}_{SDF} = \lambda_a \mathcal{L}_v + \lambda_b \mathcal{L}_{sg} + \lambda_c \mathcal{L}_{se} \tag{2}$$

$$\mathcal{L}_v = \mathbb{E}_{i\epsilon I_S \cup I_E}(|f(\boldsymbol{x}_i) - s_i|) \tag{3}$$

$$\mathcal{L}_{sg} = \mathbb{E}_{i\epsilon I_S}(\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_i) - \boldsymbol{n}_i\|) \tag{4}$$

$$\mathcal{L}_{se} = \mathbb{E}_{i\epsilon I_E}(\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_i)\| - 1)^2, \tag{5}$$

where $\mathcal{L}_v$ is a regression loss for the values [4], $\mathcal{L}_{sg}$ and $\mathcal{L}_{se}$ are losses for the gradients [2]. Specifically, $\mathcal{L}_{se}$ is based on the Eikonal equation[1]. We set the weights to balance each term as $\lambda_a = 2, \lambda_b = 1, \lambda_c = 0.1$.

We include the performance for the approximated SDF on the datasets we used in Table 1. We use two metrics:

Mean Absolute Error defined in Eq. 3;

Mean Relative Error defined as

$$\mathbb{E}_{i\epsilon I_S \cup I_E}\left(\left|\frac{f(\boldsymbol{x}_i) - s_i|}{s_i}\right| \cdot 100\%\right). \tag{6}$$

Using those loss functions, we can have supervision on the absolute values for the SDF samples, but no supervision on the norm of the gradient for vertices that are not on the body surfaces. Consequently, the mean relative error is much worse than the mean absolute error. Thus, in the main paper, we use the predicted offset scale to help ReFU improve its collision handling ability using the approximated SDF.

Table 1: Mean absolute error and mean relative error of the SDF network.

Dataset	Mean Absolute Error	Mean Relative Error
Shirt Male	2.38mm	28.22%
T-shirt Male	2.37mm	25.85%
Short-pant Male	2.46mm	31.66%
Skirt Female	3.10mm	32.64%

2 Penetration Energy Histogram

Although our method cannot eliminate all the collisions when using the neural network approximated SDF due to the inaccuracies of SDF, it brings a significant decrease in the overall penetration energy as shown in the distribution histogram in Fig. 1 and Fig. 2. Fig. 1 shows all the results and Fig. 2 shows the zoomed-in results with collision energy less than 2.5×10^{-3}. We compute the penetration energy as the way described in [7].

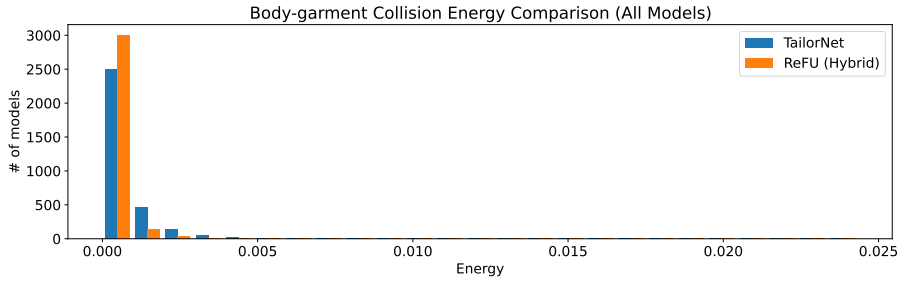

Fig. 1: We show the histogram of the collision energy for TailorNet without ReFU (blue) and TailorNet with ReFU trained in the "Hybrid" SDF mode (yellow). It shows with ReFU, the network can produce much more garments with low collision energy.

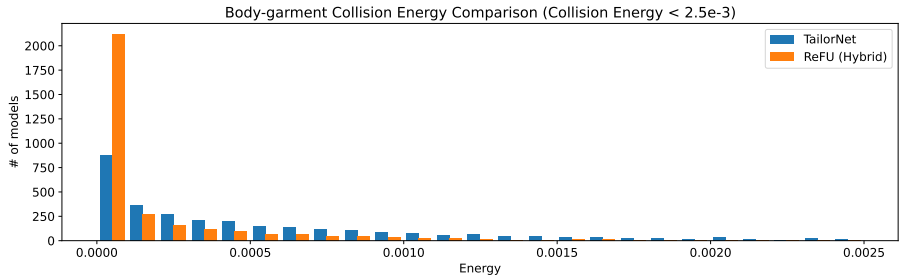

Fig. 2: Here is a zoomed-version of Fig. 1 for output garments with collision energy less than 2.5×10^{-3}.

3 Results with GCNN

In this section, we show how ReFU works when applied in a backbone network based on a graph convolutional neural network (GCNN) [8] for general 3D models. This GCNN does not have a frequency division process as in TailorNet, thus we can directly plug in the ReFU and train from scratch. We mentioned in Sec. 3.2. of the main paper, that SDF for collision handling is only useful when the points are near the surface, so we may better train with ReFU in the refining stage when the predicted cloth already satisfies this condition. If we train from scratch, the initial network prediction may not satisfy the condition. Nevertheless, we still find the network can learn to cope with these, and have good results when the network becomes steady as shown in Table 2. The results demonstrate significant improvements in reducing collisions by training ReFU with GCNN from scratch. This experiment illustrates that ReFU can work with different kinds of backbone networks.

Table 2: Garment prediction results of the GCNN network trained with or without the ReFU layer.

Metric	Method			
	GCNN[8]	w/ ReFU		
		Approx. SDF	Acc. SDF	Hybrid
MPVE	14.05	13.05	12.82	12.93
VFCP	2.26%	0.81%	0.00%	0.72%
CFMP	8.70%	19.92%	40.98%	36.78%

4 Results on VTO Dataset [6]

VTO dataset from [6] is a new public garment-body dataset with more complex human poses. The dataset contains 17 different shapes and several different motion sequences, including walking, running, jumping, dancing, etc. We evaluate our results on test sets that include four unseen sequences similar to the original paper [6]. We choose two options for the number of shapes: one with five shapes resulting in a similar amount of samples to the TairlorNet dataset and another with all 17 shapes. We use a 5-layer Multilayer Perception (MLP) as our baseline. We show the results in Table 3. Our method works well in this new dataset.

Table 3: Results on the new VTO dataset [6]

Shape Num.	Train Num.	Test Num.	Metric	Method	
				Baseline	ReFU (Approx. SDF)
5	33525	2060	MPVE	15.36	13.15
			VFCP	1.78%	0.56%
			CFMP	18.69	52.53%
17	113985	7004	MPVE	18.23	16.12
			VFCP	3.18%	0.86%
			CFMP	13.30%	38.39%

5 Ablation Study for Networks Computing α_i

We compare alternative options for computing the scale α_i. In the final ReFU structure, we use the following networks to compute:

$$\alpha_i = g(k(\boldsymbol{z})_i, f(\boldsymbol{x}_i)), \boldsymbol{z} \in \mathbb{R}^M, \tag{7}$$

with $k : \mathbb{R}^M \rightarrow \mathbb{R}^{N \times D}$ as a topology-dependent MLP network that infers the latent vector for every vertex from the global feature $\boldsymbol{z}$.

There are two additional possible choices. The first one ("Alt. 1"):

$$\alpha_i = g(k'(\boldsymbol{z}), f(\boldsymbol{x}_i)), \boldsymbol{z} \in \mathbb{R}^M, \tag{8}$$

with $k : \mathbb{R}^M \rightarrow \mathbb{R}^{D'}$ as another MLP inferring one shared latent vector from z. Here we let $D' \propto N \times D$ to maintain the parameter size of the whole network and ensure a fair comparison.

The second one ("Alt. 2"):

$$\alpha_i = g'(f(\boldsymbol{x}_i)), \boldsymbol{z} \in \mathbb{R}^M, \tag{9}$$

where g' is a network directly predicting the scale from each vertex's SDF value.

We show the comparison results on "Shirt Male" dataset in Table 4. For all the experiments, we use approximate SDF. The results show that our final choice in Eq. 7 achieves better results since it considers each vertex's information to compute the final scale.

Table 4: Results trained with different α_i computing networks.

Metric	Method			
	Baseline	Alt. 1	Alt. 2	ReFU
MPVE	11.27	10.65	10.66	10.59
VFCP	1.18%	0.76%	0.79%	0.62%
CFMP	11.92%	21.36%	20.81%	26.9%

Table 5: Per-frame running time, including approximated SDF query, accurate SDF query computed using spatial data structures, ReFU layer inference, and the backbone network based on TailorNet inference [5].

Dataset	Component			
	Approx. SDF	Acc. SDF	ReFU	Backbone
Shirt Male	1.97ms	121.96ms	0.29ms	22.14ms
T-Shirt Male	1.77ms	99.27ms	0.28ms	21.51ms
Short-pant Male	1.58ms	89.69ms	0.21ms	18.82ms
Skirt Female	1.67ms	107.38ms	0.23ms	19.76ms
All Garments	1.75ms	105.50ms	0.25ms	20.57ms

6 Running Time

We include the running time for SDF, ReFU layer, and the backbone network TailorNet in Table 5.

7 Moving Offset Analysis

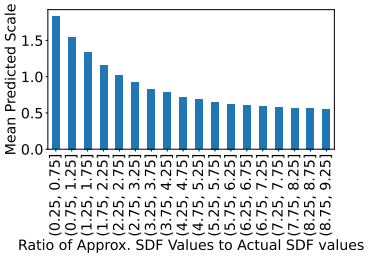

Fig. 3: We show the bar plot of the predicted scale grouped by the ratio of approximated to actual SDF values. A larger scale can push collided vertices further to compensate for smaller than actual SDF value prediction, and vice versa. It shows that our moving offset based on the predicted scale can compensate for the error from the SDF approximation.

We analyze the ratio of the approximated SDF values to the actual ones with the predicted moving offset scale α. We show the bar plot for all collision-resolved vertices in Fig. 3. When the ratio is smaller than one, the approximated SDF value is smaller than the actual one. A larger moving offset scale lets ReFU push the vertices even further and compensate for the error from SDF prediction. Similarly, when the ratio is larger than one, the approximated SDF value is larger than the actual one; a smaller scale can avoid putting the vertex too far away. Among the collision-resolved vertices, the minimum ratio is 0.39 with a scale of 2.68; the maximum ratio is 9.13 with a scale 0.25. Notice that the mean predicted scale equals 1 for the group $(2.25, 2.75]$, which shows that our layer learns to push the vertices even further than the distance to the surface to resolve the EE cases.

8 Local Geometric Comparison with Optimization Post-Processing

We use local Laplacian error on the collision resolving regions to show that our method can better preserve small-scale geometric details, as compared to other post-processing methods. For each initial collided vertex, we compute its 1-ring neighborhood Laplacian error. We summarize the mean error on the 'shirt male'

Table 6: Local Laplacian error for collision resolving region on 'shirt male' dataset. Our method can preseve local geometry details than previous methods.

w/ Opt. Post-Process		w/ ReFU		
Approx. SDF	Acc. SDF	Approx. SDF	Acc. SDF	Hybrid
7.02	5.85	5.29	4.24	4.36

dataset in Table 6. Our method with different settings results in lower errors compared to the post-processing counterparts.

9 Comparison with Collision Loss

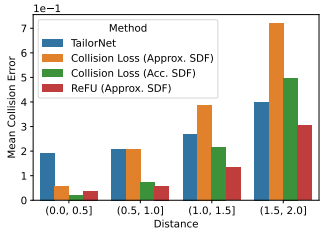

Fig. 4: We highlight the benefits of our approach on samples distinct from the training set over methods based on collision loss. Collision loss can only help reduce collisions for samples close to the training set.

As we mentioned in the main paper, adding collision loss introduces even more collisions for testing samples that are farther away from the training set. For each sample in the test set, we compute the minimal Euclidean distance to the training set samples in the parameter space (pose, shape, and style). In Fig. 4, we show the mean collision error for "Shirt Male" grouped by the distance. The soft constraint can only reduce collisions for samples near the training set and even introduces more errors for samples far away. In contrast, ReFU can still resolve some collisions for samples with great differences from the seen training ones.

10 Visualized Comparison Results

We include more visulizations for the results generated with or without our ReFU layer, in Fig. 5, Fig. 6, Fig. 7 and Fig. 8.

Fig. 5: Additional examples from Shirt Male dataset, showing collisions resolved by applying our ReFU in TailorNet.

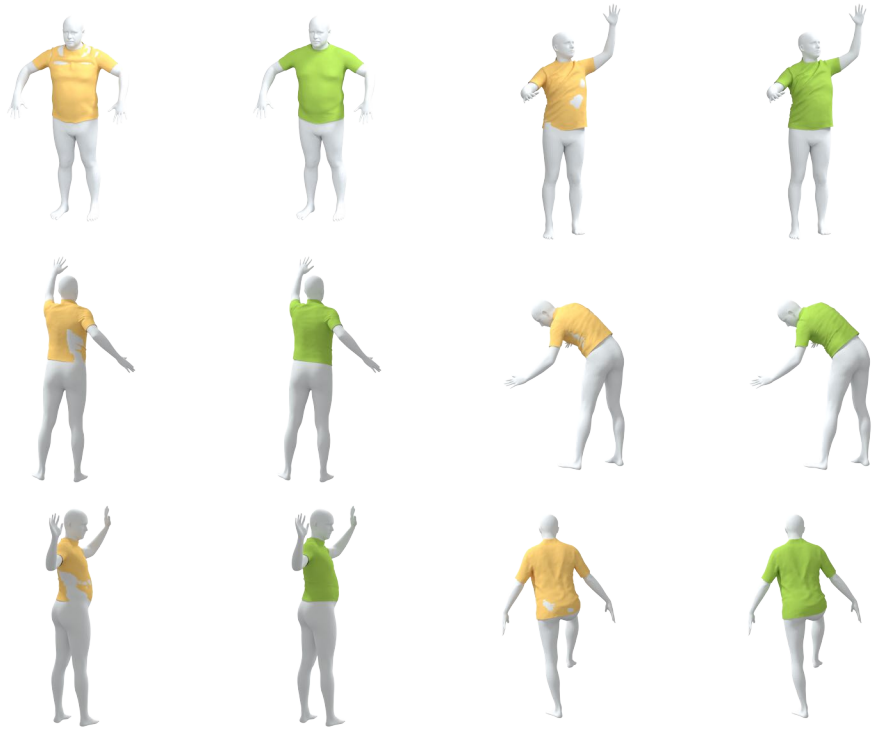

Fig. 6: Additional examples from T-Shirt Male dataset, showing collisions resolved by applying our ReFU in TailorNet.

Fig. 7: Additional examples from Short-pant Male dataset, showing collisions resolved by applying our ReFU in TailorNet.

Fig. 8: Additional examples from Skirt Female dataset, showing collisions resolved by applying our ReFU in TailorNet.

References

1. Crandall, M.G., Lions, P.L.: Viscosity solutions of hamilton-jacobi equations. Transactions of the American mathematical society **277**(1), 1–42 (1983)
2. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. arXiv preprint arXiv:2002.10099 (2020)
3. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. ACM transactions on graphics (TOG) **34**(6), 1–16 (2015)
4. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)
5. Patel, C., Liao, Z., Pons-Moll, G.: Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7365–7375 (2020)
6. Santesteban, I., Thuerey, N., Otaduy, M.A., Casas, D.: Self-supervised collision handling via generative 3d garment models for virtual try-on. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
7. Tan, Q., Pan, Z., Manocha, D.: Lcollision: Fast generation of collision-free human poses using learned non-penetration constraints. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI (2021)
8. Zhou, Y., Wu, C., Li, Z., Cao, C., Ye, Y., Saragih, J., Li, H., Sheikh, Y.: Fully convolutional mesh autoencoder using efficient spatially varying kernels. arXiv preprint arXiv:2006.04325 (2020)