Supplementary Material of Towards Multi-Layered 3D Garments Animation

Yidi Shao[1] Chen Change Loy[1] Bo Dai[2]
[1]S-Lab for Advanced Intelligence, Nanyang Technological University
yidi001@e.ntu.edu.sg, ccloy@ntu.edu.sg
[2]Shanghai AI Laboratory
daibo@pjlab.org.cn

Table 1: Frequently used symbols in main text. We group the symbols into two parts: simulation-related symbols and Transformer[9]-related symbols,

Simulation-Related	
Symbols	**Descriptions**
M^t, P^t	Mesh and patch-based mesh.
$\boldsymbol{V}^t, \boldsymbol{V}_p^t$	Set of states for vertices and patches.
$\boldsymbol{x}_i^t, \dot{\boldsymbol{x}}_i^t, \ddot{\boldsymbol{x}}_i^t$	The position, velocity, and acceleration for vertex i.
$\boldsymbol{E}^M, \boldsymbol{E}^W$	Edge sets between vertices in mesh and world space.
$\boldsymbol{E}_p^M, \boldsymbol{E}_p^W$	Edge sets between patches in mesh and world space.
e_{ij}	Edge between vertex i and j.
$\boldsymbol{a}_g$	Attributes of garments.
$\boldsymbol{w}^t$	The wind descriptor.
$\boldsymbol{\eta}^t$	Quaternion rotation of the wind.
s^t	Strength of wind.
$\boldsymbol{\beta}_k, \boldsymbol{\alpha}_k$	Predicted velocity and acceleration.
Transformer-Related	
Symbols	**Descriptions**
W_*	Trainable parameters.
$\boldsymbol{v}_i, \boldsymbol{q}_i$	State and query tokens for vertex/patch i.
$\boldsymbol{r}_i, \boldsymbol{s}_i$	Receiver and sender tokens for vertex/patch i.
$\boldsymbol{\mu}_{a,b}, \sigma_{a,b}$	Mean and standard deviation between token a and b.
ω_{ij}	Attention score between vertex/patch i and j.
R	Rotation matrix in 3D space.
$\boldsymbol{f}_{i,j}, \boldsymbol{f}_{i,j}^R$	Interactions in shared and canonical hidden space.

A. LayersNet

A.1. Symbols

We list the symbols mentioned in our main text as shown in Table 1, which are grouped into simulation-related symbols and Transformer[9]-related symbols.

A.2. Rotation Equivalent Transformation

In the main text, we propose Rotation Equivalent Transformation with rotation invariant attention mechanism as follows:

$$\boldsymbol{q}_i = W_q\boldsymbol{v}_i, \qquad \boldsymbol{r}_i = W_r\boldsymbol{v}_i, \qquad \boldsymbol{s}_i = W_s\boldsymbol{v}_i, \tag{1}$$

$$\boldsymbol{f}_{i,j} = \frac{\boldsymbol{r}_i + \boldsymbol{s}_j}{\|\boldsymbol{r}_i - \boldsymbol{s}_j\|}. \tag{2}$$

$$\omega_{ij} = \mathrm{softmax}(\boldsymbol{q}_i^\top \boldsymbol{f}_{i,j}), \tag{3}$$

where $\boldsymbol{v}_i$ is state token, $\boldsymbol{q}_i$ is query token, $\boldsymbol{r}_i$ is receiver token and $\boldsymbol{s}_j$ is sender token, W_q, W_r, W_s are trainable parameters. Here we show the proof that our proposed attention mechanism in equation 3 is rotation invariant. Given query token $\boldsymbol{q}_i$, receiver token $\boldsymbol{r}_i$, and sender token $\boldsymbol{s}_j$ with d dimensions, we apply a high dimensional rotation $R \in \mathbb{R}^{d\times d}$ to rotate the tokens into corresponding canonical space as:

$$\boldsymbol{q}_i^R = R\boldsymbol{q}_i, \qquad \boldsymbol{r}_i^R = R\boldsymbol{r}_i, \qquad \boldsymbol{s}_j^R = R\boldsymbol{s}_j, \tag{4}$$

$$\boldsymbol{f}'_{i,j} = \frac{\boldsymbol{r}_i^R + \boldsymbol{s}_j^R}{\|\boldsymbol{r}_i^R - \boldsymbol{s}_j^R\|} = R\boldsymbol{f}_{i,j} \tag{5}$$

Therefore, the dot product in equation 3 is rotation invariant as shown bellow

$$\begin{aligned} (\boldsymbol{q}_i^R)^\top \boldsymbol{f}'_{i,j} &= (R\boldsymbol{q}_i)^\top (R\boldsymbol{f}_{i,j}) = \boldsymbol{q}_i^\top R^\top R\boldsymbol{f}_{i,j} & (6)\\ &= \boldsymbol{q}_i^\top \boldsymbol{f}_{i,j}. & (7)\end{aligned}$$

Hence, equation 3 is rotation invariant.

B. Dataset

Generating a dataset with multi-layered garments is non-trivial – interpenetration between garments should be avoided, and their dynamics should obey the physics rules. Thanks to recent developments in physics-based methods, several software, such as Blender[1], can infer the interactions

[1]https://www.blender.org/

Table 2: We compare D-LAYERS with existing 3D datasets. Our dataset is composed of multi-layered clothes, with unique attribute data, such as stiffness and friction, attached to each garment. Moreover, we include data of wind with its strength and direction randomly sampled. $*^1$: 3DPeople [7] does not specify the exact number of garments, while it claims to dress each subject with different outfits. $*^2$: The multi-layered garments in Layered-Garment do not follow physics laws and the penetrated vertices are forced to move out of inner garments in hard-coded manner.

Dataset	**Dynamics**	**Subjects**	**Garments**	**Multi-Layered**	**Attributes**	**Wind**
3DPeople [7]		80	$*^1$		N/A	
TailorNet [6]		9	20		N/A	
Cloth3D [2]	✓	8.5K	11.3K		4	
Layered-Garment [1]		142	101	$*^2$	N/A	
D-LAYERS (Ours)	✓	4.9K	9.9K	✓	9.9K	✓

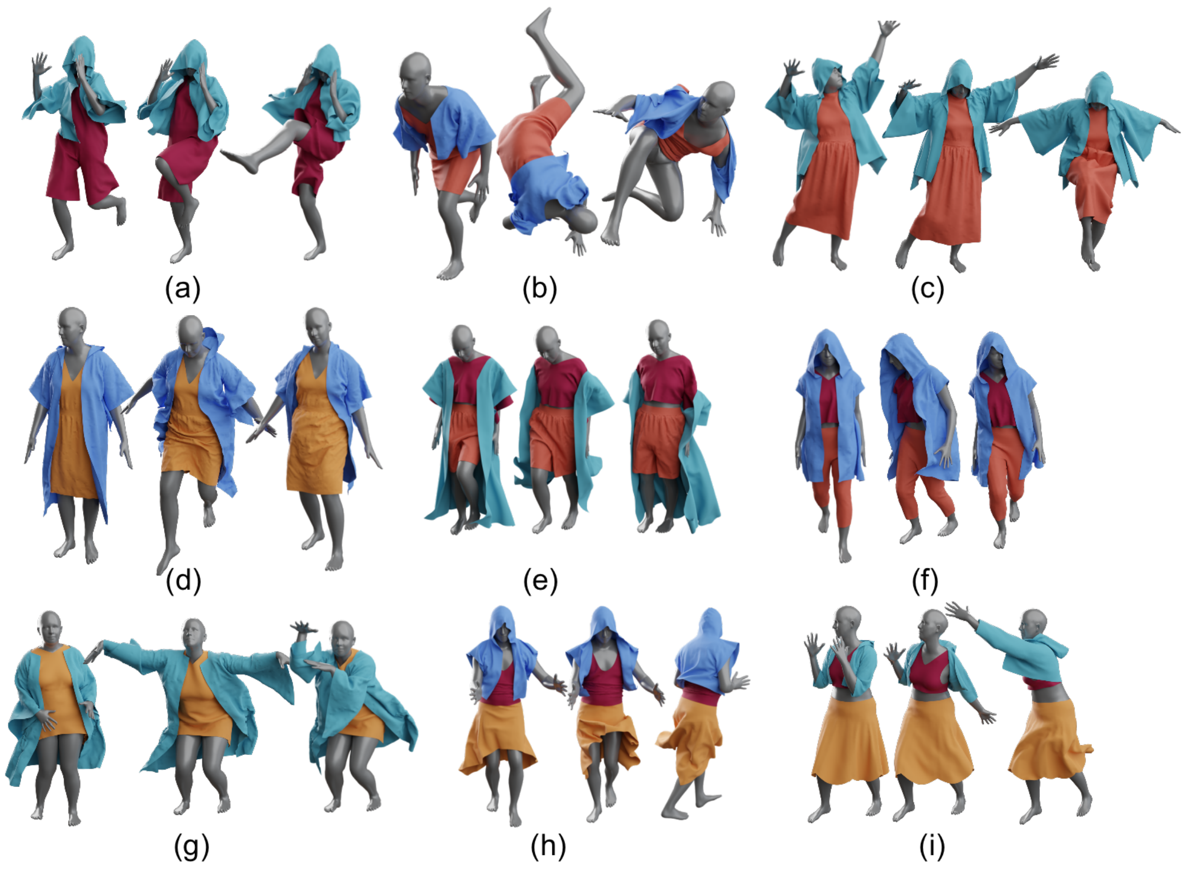

Figure 1: Samples from our D-LAYERS. Our D-LAYERS focuses on the general scenarios of garment animations, which include multi-layered garments with diverse topologies driven by various external forces, such as human bodies and wind. The garments include jumpsuits (a), different dresses (b,c,d,g) and skirts (h,i), pants (e,f), T-shirts (e,f,h,i), jackets (b,e,g) and jackets with hood (a,c,d,f,h,i). The topologies of the garments are also diverse and close to real-life cases. Different layers of garments interact with each other constrained by physics law, leading to rich dynamics in our dataset.

among different clothes and generate faithful garments with multiple layers. We show various samples in Figure 1, including diverse garments with realistic dynamics driven by complex human motions and sampled wind.

B.1. Generating Details

Although Blender does not support collisions among multiple objects, it is able to solve collisions within one object. Thus, by merging multiple garments as a single mesh, we regard the collisions among different garments as the interactions within one mesh, which can be solved by Blender. In other words, different layers of garments will interact with each other following the physics rules.

Before simulation, we properly dress the human body in T-pose and scale up all objects 10 times the real-world size, which can preserve more details of the garments such as wrinkles.

B.2. Dataset Settings

We compare our dataset with existing datasets in Table 2.

Multi-Layered Garments. The main challenge is to have interpenetration-free simulations for multiple objects. To achieve that, we first drape the multi-layered outfit to SMPL human body in T-pose, followed by a warm-up simulation to solve the interpenetrations among garments. We adopt a large collision distance to solve the interpenetrations. Afterwards, we merge the garments into one garment mesh and conduct a simulation driven by the human body and wind. Since all garments belong to one mesh after merging, the interactions among garments are computed through the self-collision mechanism in Blender, which generates interpenetration-free results in simulation.

Garments' Attributes. Existing datasets cover a limited choice for garment attributes, e.g., cotton or fabric. Tasks like physics parameter estimations could barely benefit from those datasets. To make garments animations more diverse and flexible, we uniformly sample different garments' attributes, such as mass, stiffness, and friction. In addition, we introduce different attributes to the inner and outer outfits, leading to more varieties. Specifically, we uniformly sample the following attributes: vertex mass from 0.2 to 0.8; stiffness of tension, compression, shear, and bending from 15 to 100; friction from 40 to 80. In this way, our dataset include various dynamics of garments and can be used for many tasks, such as physics parameter estimation.

Human Motion Sequences. We adopt the SMPL-based human motion sequences from CMU MoCap in AMASS [5], which includes 2,600 sequences with 30FPS in total. During simulation, we randomly sample the human shapes and genders for each sequence and extract sub-sequences with a maximum of 600 frames. To accurately simulate garments on human body, collision-free human meshes are required to avoid invalid simulations. Since our dataset focuses on garments generations, we adopt linear regressions to solve the self-collisions from SMPL models [4] and leave a minimum gap of 0.004 meters before scaling up the human mesh. We skip the unresolvable collisions and discard the corresponding frames.

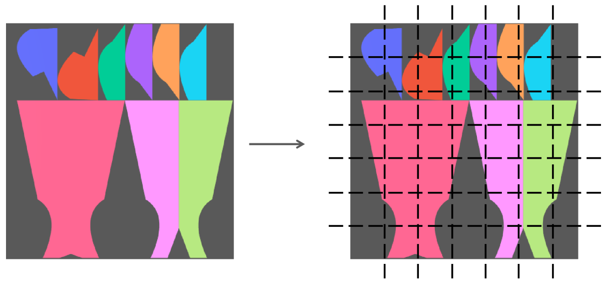

Figure 2: We grouped garment's vertices into patches given the corresponding UV mapping. Based on the coordinates in UV, we group the vertices according to their positions. Subsequently, we divide the 3D garment mesh into patches based on the groups.

C. Experiments

C.1. Implementation Details

Patched Garment Model. We group the particles in garment's mesh given corresponding UV mapping. Specifically, we divide the UV mapping into square patches according to the UV coordinates, as shown in Figure 2. Then, we group the garment's vertices in 3D space given the grouped UV mapping. When building the connections $\boldsymbol{E}^M$ of the patched garments, we connect patch i and patch j iff there is at least one pair of vertices within the patches are connected in 3D space.

LayersNet details. To obtain the world space edges $\boldsymbol{E}^W$, we adopt $R = 0.4$ to calculate the neighbors from the human mesh and $R = 0.6$ for different layers of garments. We adopt $h = 1$ for the inputs of all objects' states. The hyperparameters λ_m, λ_n in our loss term are set to 1. When trained with collision loss, we set $\lambda_b = 1, \lambda_g = 0.1$ by default. The optimal combination for the weight of the loss term is $\lambda_b = 1.3, \lambda_g = 0.1$. In practice, we train LayersNet without collision loss for one epoch as warmup, and continue to train another 9 epochs using the collision loss. We adopt Adam optimizer with an initial learning rate of 0.001 and a decreasing factor of 0.5 every two epochs. The batch size is set to 4.

We adopt four different encoders for meshes, garment attributes, wind attributes, and gravity. This is because the four components belong to different domain space and have different dimensions. Since the states of garments mesh and human bodies mesh are from the same domain, we share the encoder for them. All the encoders are two-layer MLPs with dimensions 128. The only difference is the input dimensions. We adopt 4 blocks of modified Transformer block in LayersNet, with hidden dimensions 128 for each block. The number of head in multi-head attention is set to 8, while 4 heads apply the attention mask generated by

Table 3: We verify our implementation of DeePSD on Cloth3D accroding to official paper [3]. The results are similar to original paper, suggesting that our implementation of DeePSD is similar to official one.

Method	**T-shirt**	**Top**	**Trousers**	**Skirt**	**Jumpsuit**	**Dress**
DeePSD	25.01±20.94	16.90±15.38	20.02±8.50	20.43±31.10	24.31±6.36	42.10±21.41

Table 4: Components of different splits and models' corresponding Euclidean errors (mm). Notice that in our D-LAYERS, all objects are scaled up 10 times than real-world size. To analyze the influence of different combinations in our D-LAYERS, we sample four splits from our dataset: inner garments are tight clothes (jumpsuit) without wind (**T**); inner garments are tight clothes (jumpsuit) with strong wind (**T+W**); inner garments are loose clothes (dress) without wind (**L**); inner garments are loose clothes (dress) with strong wind (**L+W**). The models trained with only inner garments are marked by $*$. Notice that MGNet has worse generalization abilities due to garment-specific design. LayersNet achieves superior and robust performance in most cases especially those with multi-layered garments.

Splits	**Methods on Inner Garment**				**Methods on Layered Garments**		
	DeePSD*	MGNet*	GarSim*	LayersNet*(Ours)	DeePSD	GarSim	LayersNet(Ours)
T	225.3±106.4	5219.2±1565.8	284.2±145.0	**220.0±195.3**	1072.6±694.7	920.2±608.9	**489.9±447.7**
T+W	**239.5±103.9**	5186.8±1754.8	280.5±124.5	258.0±312.8	1121.0±731.7	999.5±686.2	**508.5±562.7**
L	501.3±300.1	4432.7±1438.0	534.6±333.0	**344.7±244.5**	887.8±460.5	929.6±509.7	**378.0±293.0**
L+W	577.5±373.9	4595.0±1215.2	621.1±403.9	**347.4±288.9**	1083.1±492.2	1050.0±523.4	**467.8±403.7**

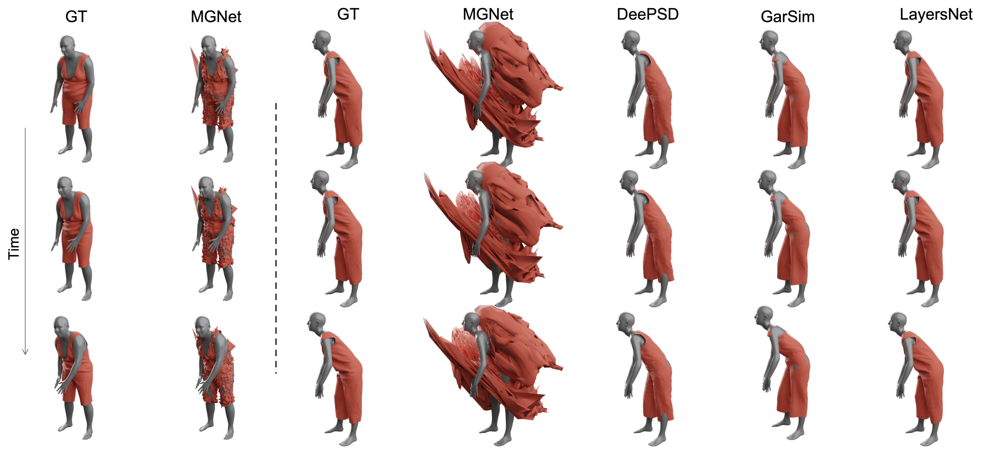

Figure 3: Visualized samples on split T, where we train and test models with only inner garments (jumpsuit). The left are the training samples while the right is test samples. MGNet is able to generate 3D garments on training examples on the left while has difficulties to generalize to unseen examples in test set due to the garment-specific design. DeePSD and GarSim has faithful predictions on single-layered garments. Our LayersNet faithfully rollouts 3D garments when only with inner clothes.

$\boldsymbol{E}^M$, and 4 heads adopt the attention mask generated by $\boldsymbol{E}^W$. For the decoder, we adopt a three-layer MLPs with a forward dimension of 128 and an output dimension of 3. When concatenating the nearest point on human mesh, we mask the point $\boldsymbol{v}_{b,i}^{t+1}$ to zeros if the there is no edge $e_{b,i} \in \boldsymbol{E}^W$ connecting the human body point $\boldsymbol{v}_{b,i}^{t+1}$ and garment point $\boldsymbol{v}_i^{t+1}$. For the inputs of garment mesh and human body mesh, we adopt relative positions to the root of human body mesh. Gravity inputs can be naturally transformed to relative coordinates by subtracting the acceleration of the body's root. Since the wind's attributes are still measured in global coordinates, we also convert them to the relative coordinates in implicit manner, i.e. convert the value of strength in global coordinates to local coordinates defined by the root of human body. Specifically, we concatenate the position, velocity, and acceleration of the human body's root point $\boldsymbol{v}_r^t$ as extra features $\boldsymbol{w}^t = \{\boldsymbol{q}^t, s^t, \boldsymbol{v}_r^t\}$ and let LayersNet learns the wind's features in relative coordinates. We apply $\Delta t = 1$ in our experiments, which is independent from the real time interval between each frames, which is 0.33s. When training LayersNet, we normalize the meshes' states across the whole training set before feeding into the model, which is a commonly adopted processing in literature.

During training, we gradually enlarge the maximum rollouts steps as mentioned in main text (Section 3.2) before back-propagate the gradient. Specifically, the maximum steps increase by 1 after each epoch. In practice, we set the maximum steps as 0 at the first epoch and 9 at the 10-th epoch. When training LayersNet with garment-to-garment loss $\mathcal{L}_{c,g}$, we penalize $\mathcal{L}_{c,g}$ only after the first epoch, since $\mathcal{L}_{c,g}$ would enlarge the errors brought by unstable predictions.

DeePSD details. We implement DeePSD according to the original paper [3] and verify the performance of DeePSD on Cloth3D as shown in Table 3. Our implementation of DeePSD achieves similar performance compared with the original paper.

C.2. Comparisons

Influence of Multi-Layered Garments and Wind. For convenience, we copy the table in main text for reference as shown in Table 4 which shows the quantitative results from four splits of our dataset. We show the qualitative results in Figure 3 Specifically, the results on the first three rows, where models are marked by *, are obtained by training models with only inner garments. The remaining results are obtained by training on both inner and outer garments on all splits.

Notice that we scale up the human mesh and garment mesh 10 times the real-world size, the corresponding errors are also scaled up. Thus, DeePSD*, which is trained with only inner single-layered garments, achieves similar results on both datasets: the Euclidean errors of jumpsuit and dress are similar on both D-LAYERS and Cloth3D, suggesting that the quality of our dataset is not worse than Cloth3D. We also implement GarSim given the original paper [8]. And the reuslts, especially the loose garments, are close to the original paper.

We further visualize some samples from split T within only inner garments in Figure 3. When trained with only inner garments (jumpsuit), DeePSD [3] is able to predict faithful rollouts. MGNet [10] is able to generate 3D garments on training samples while struggles to generalize to unseen garments from test examples due to the garment-specific design. In original paper of MGNet, they train

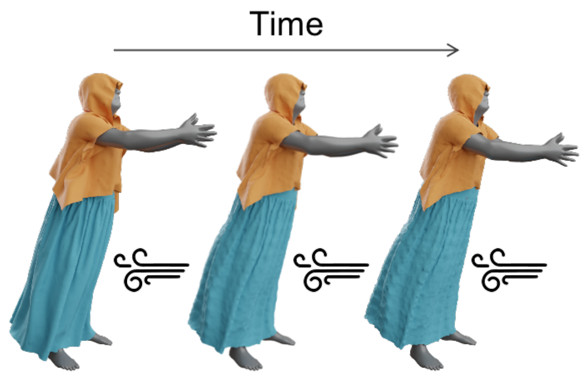

Figure 4: An animation sample by LayersNet. Specifically, the garments are driven by both human body and wind. As the wind blowing the garments, the clothes gradually and faithfully wave in the air, suggesting that our LayersNet is able to extend to various types of outer forces.

MGNet with only on 300 frames of data with the same garment topology, while in our dataset each garment is unique with different topology. In contrast, our LayersNet achieves faithful predictions in this simplified case.

General Garment Animations. We report detailed comparisons in Table 5. Besides the comparisons in main text, we visualize several samples by DeePSD and GarSim for reference in Figure 5. As discussed in the main text, the garments in our dataset D-LAYERS, especially the outer clothes, are more flexible and are able to respond to various garment attributes. Consequently, the high flexibility brings more challenges to DeePSD and GarSim, leading to difficulties in convergence. The collision loss for DeePSD further introduces noises due to inaccurate garment meshes proposed by DeePSD without collsision loss, and leads to higher euclidean errors. In contrast, our simulation-based LayersNet animates garments in topology-independent and unified manners and successfully animates garments on general scenarios.

Additionally, we display more qualitative results of our LayersNet in Figure 4 and Figure 6. Specifically, We apply customized wind in Figure 4 and animate the garments driven mainly by wind. We animate garments with diverse topologies in various scenarios in Figure 6. Surprisingly, we find that LayersNet is still able to successfully animate garments for long sequences even on complex scenarios.

Limitations. Simulating particle-wise interactions enables LayersNet to animate garments with diverse topologies driven by various external forces in a unified manner, leading to highly generalizable abilities in unseen scenarios. Since the predictions of future frames are based on

Table 5: Euclidean error (mm) on sampled D-LAYERS with maximum sequence length of 35 frames. The collision rates between different layers of garments are shown under **L-Collision**, while the collision rates between garments and human bodies are shown under **H-Collision**. Models trained with collision loss $\mathcal{L}_{c,b}$, $\mathcal{L}_{c,g}$ are marked by +. Our LayersNet achieves superior results in all cases.

Methods	**Jacket**	**Jacket + Hood**	**Dress**	**Jumpsuit**	**Skirt**
DeePSD	1385.3±886.6	1087.8±564.5	736.8±466.6	535.2±224.7	1107.3±769.2
DeePSD+	1830.1±803.3	1566.0±527.1	1333.0±349.2	1219.0±186.8	1194.7±311.2
GarSim+	1412.1±886.8	1139.1±653.5	674.4±451.8	317.8±157.4	689.9±386.7
LayersNet(Ours)	571.9±451.9	493.9±354.2	397.2±342.2	264.0±200.2	301.3±79.3
LayersNet+(Ours)	**567.3±425.5**	**491.4±361.3**	**379.1±299.7**	**260.1±222.2**	**299.5±92.3**

Methods	**Pants**	**T-shirt**	**Overall**	**L-Collision(%)**	**H-Collision(%)**
DeePSD	498.8±109.5	613.1±338.2	1049.8±549.7	10.11±5.31	23.89±7.89
DeePSD+	1185.7±213.3	1202.9±233.6	1563.4±486.8	8.78±5.12	19.47±6.38
GarSim+	317.8±150.1	447.6±303.8	1028.3±581.0	6.03±4.23	15.11±7.11
LayersNet(Ours)	234.4±206.3	273.3±169.0	472.8±343.5	**3.13±2.22**	10.68±4.53
LayersNet+(Ours)	**200.9±140.1**	**267.8±189.6**	**467.2±330.7**	3.77±2.60	**2.16±1.46**

previous rollouts by the model, the errors inevitably accumulate as the length of predictions, which is a common problem in simulation. As shown in Figure 4 and Figure 6, LayersNet rollouts some artifacts when the sequences are too long or external forces are too fierce. We will explore more strategies in the future to alleviate the problem, such as adding physics constrains to penalize the inertia.

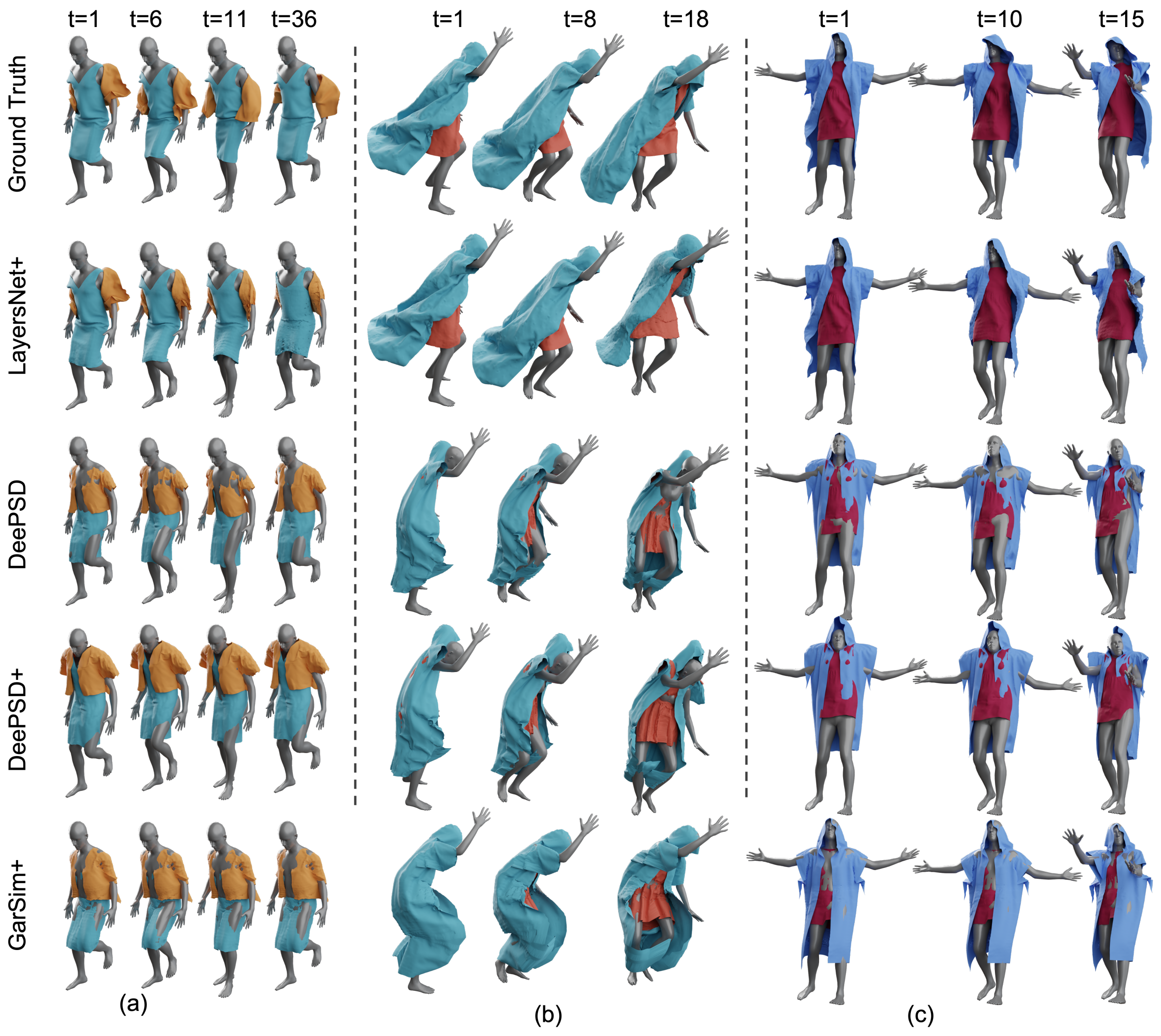

Figure 5: The outer garments in our dataset are more flexible, especially when interacting with inner garments, and are able to respond to different garment attributes, such as friction, leading to rich dynamics. DeePSD has difficulites in handling the rich dynamics in our dataset leading to low quality predictions. DeePSD+, which is finetuned on DeePSD using collision loss, is slightly better than DeePSD and solve part of garment-to-human collisions. However, DeePSD+ gets lower performance and higher euclidean errors, suggesting that DeePSD has limited abilities to handle general scenarios in D-LAYERS, such as the complex dynamics introduced by multi-layered garments. While GarSim is also a static garment model sharing many similarities with DeePSD, it suffers from similar problems and struggles with the highly flexible dynamics of garments in D-LAYERS. In contrast, our LayersNet achieves faithful rollouts of garments with various topologies in a unified manner.

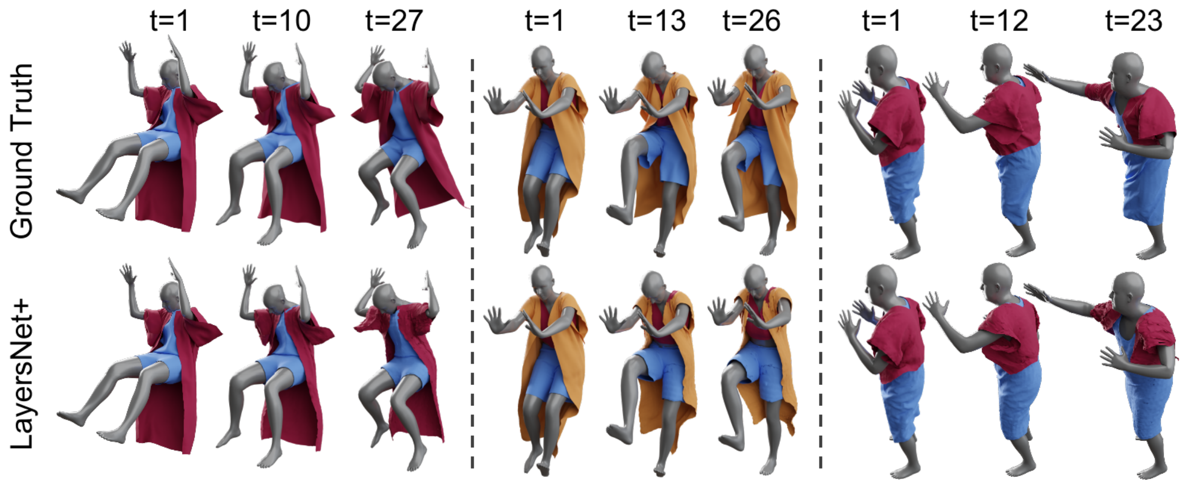

Figure 6: Additional qualitative results by LayersNet on our D-LAYERS. Our LayersNet is able to animate diverse garments driven by various external forces and achieves faithful rollouts with long sequences. We recommend readers to go through the attached videos for better views of dynamics.

References

[1] Alakh Aggarwal, Jikai Wang, Steven Hogue, Saifeng Ni, Madhukar Budagavi, and Xiaohu Guo. Layered-garment net: Generating multiple implicit garment layers from a single image. In *ACCV*, 2022. 2

[2] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. CLOTH3D: Clothed 3D humans. In *ECCV*, 2020. 2

[3] Hugo Bertiche, Meysam Madadi, Emilio Tylson, and Sergio Escalera. DeePSD: Automatic deep skinning and pose space deformation for 3D garment animation. In *ICCV*, 2021. 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. Graph.*, 2015. 3

[5] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019. 3

[6] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. TailorNet: Predicting clothing in 3D as a function of human pose, shape and garment style. In *CVPR*, 2020. 2

[7] Albert Pumarola, Jordi Sanchez, Gary P. T. Choi, Alberto Sanfeliu, and Francesc Moreno. 3DPeople: Modeling the geometry of dressed humans. In *ICCV*, 2019. 2

[8] Lokender Tiwari and Brojeshwar Bhowmick. Garsim: Particle based neural garment simulator. In *WACV*, 2023. 5

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[10] Meng Zhang, Duygu Ceylan, and Niloy J. Mitra. Motion guided deep dynamic 3D garments. *ToG*, 2022. 5