Supplementary Material:
GarSim: Particle Based Neural Garment Simulator

Lokender Tiwari   Brojeshwar Bhowmick
TCS Research, India
lokender.tiwari@tcs.com, b.bhowmick@tcs.com

We present the supplementary materials in both pdf and a video. The supplementary material contains the following items.

1. Textured results using random unseen body shapes, poses and garments for CLOTH3D [4] test sequences in Sec. S1.
2. Textured results on the unseen poses from YouTube sequences, unseen cloth types, and unseen body shapes in Sec. S2
3. *GarSim* detailed architecture in Sec. S3
4. Sample use cases of *GarSim* in Sec. S4
5. Video Results: Rendering unseen garment on unseen pose sequences from test set of CLOTH3D and youtube video with varying body shapes, poses and the garments. We also render the draped human sequence in a virtual environment and animate. *Please watch the supplementary video*.

S1. Additional results on CLOTH3D test set

We show additional results using texture and sequence. We use the poses and garments from the CLOTH3D test set. Fig. S1 and Fig. S2(bird's eye view) show that the deformation of the garments by our *GarSim* is visually plausible according to the poses on unseen garments and body shapes. Also Fig. S3 shows a sequence of frames rendered using our method which demonstrate that even if the garment is not present in training (dress, T-shirt, shirt), it is deforming well according to the underlying body motion over time.

S2. Textured results on YouTube video frames

In addition to the results shown in Fig. 4 and Fig. 5 in the main paper. We additionally show textured results on another frame of YouTube video in Fig. S4. This shows the generalization of *GarSim* on unseen cloth types. We show the textured results from various viewpoints.

S3. *GarSim* Architecture

In Fig.2 of the main paper, we have shown the overall architecture with the input and the output. A detailed architecture of all the learnable blocks are shown in Fig. S5. The two update functions of the message-passing block are implemented using MLPs whose architecture is also shown in Fig. S5 . We use PyTorch-Geometric [3] library to implement PointNet++[6] based geometry encoders and the message-passing block. The geometry encoder encodes the local geometry around a garment vertex and append it with the global garment geometry. The concatenated learned local and global garment geometric features become the final geometric feature of each garment vertex. We use same geometric encoder architecture for both body and the garment.

S4. *GarSim* use cases

There could be many possible use cases of *GarSim* . We mention two most important ones.

S4.1. Metaverse usecase

Animating garments based on the body shape and poses in an AR/VR setup is one of the important task under the recent metaverse concept. Avatar and its motion capture imitating real human is one of the important entity in the metaverse along with a virtual environment. To make the presence of avatar realistic, various kind of clothing on top of it is necessary. We use SMPL[5] avatar to be draped by *GarSim* and situate in the VR to show the complete rendering quality in the virtual world in Fig. S6.

S4.2. Garment Authoring

An average designer takes hours to drape a garment on a person in an arbitrary pose using a designer tool such as Maya[2], Blender[1] etc. This effort can significantly be reduced to a few minutes by *GarSim* . While *GarSim* can directly predict the garment on the human body in the arbitrary pose, a designer can use the predicted garment to give some final touches which would take only a few minutes. To validate this we asked a few designers to

Figure S1: **Additional textured results on the test samples with varying skirt sizes:** Unseen body shape, height and poses.

Figure S2: **Bird's eye view rendering of textured sample results on the test samples with varying skirt sizes:** Unseen body shape, height and poses.

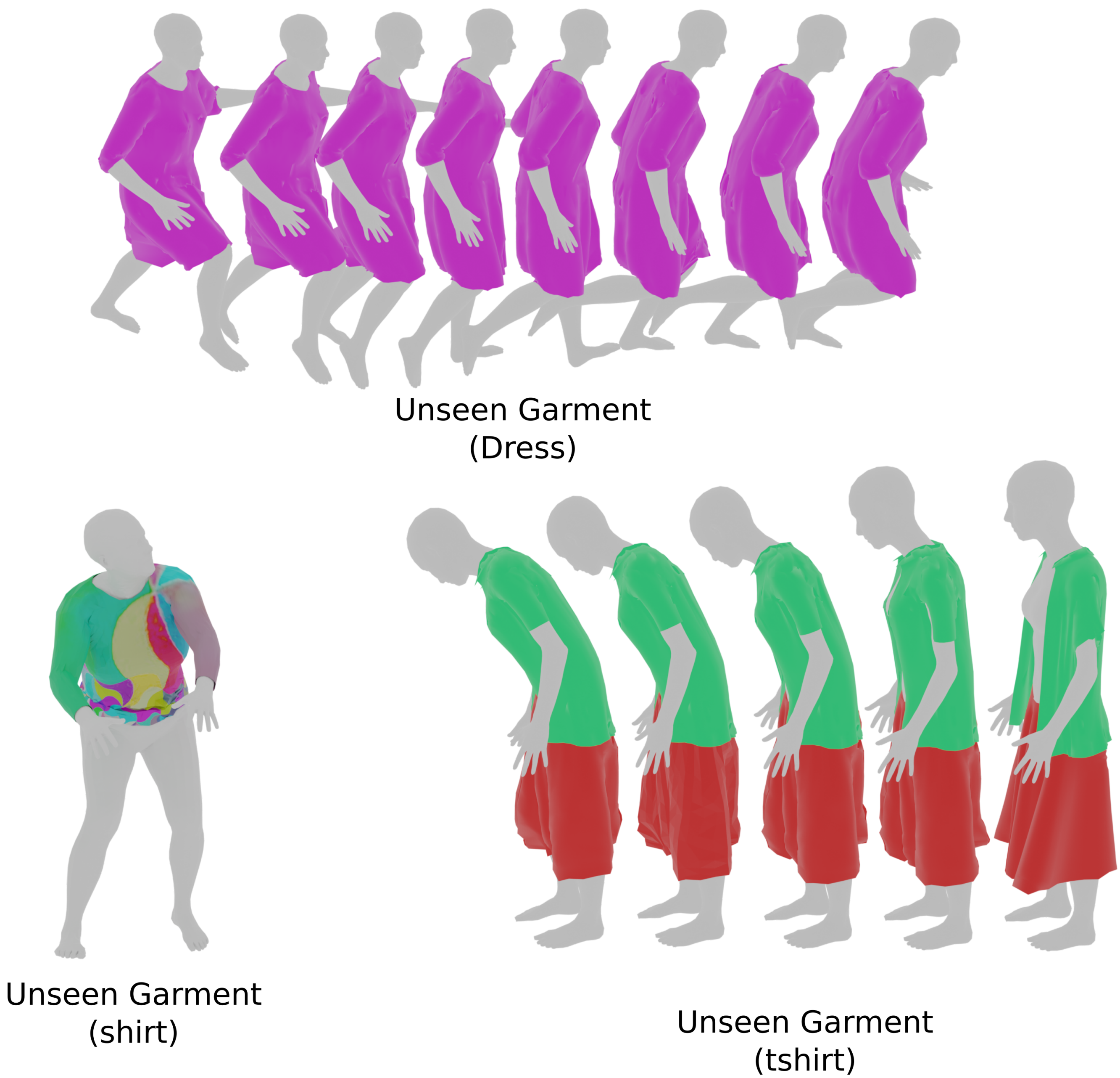

Figure S3: Sample results on unseen garments on unseen body shape, and poses.

take *GarSim* outputs and template garments as two starting points and drape the garment on the desired body pose. The outcome of this exercise was, designers took a few minutes when *GarSim* outputs were used as a starting point as compared to the template garments.

References

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

[2] Maya: A professional 3d software for creating realistic characters and blockbuster-worthy effects. `https://www.autodesk.com/products/maya/overview`.

[3] Pytorch geometric. `https://pytorch-geometric.readthedocs.io/`.

[4] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020.

[5] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[6] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J

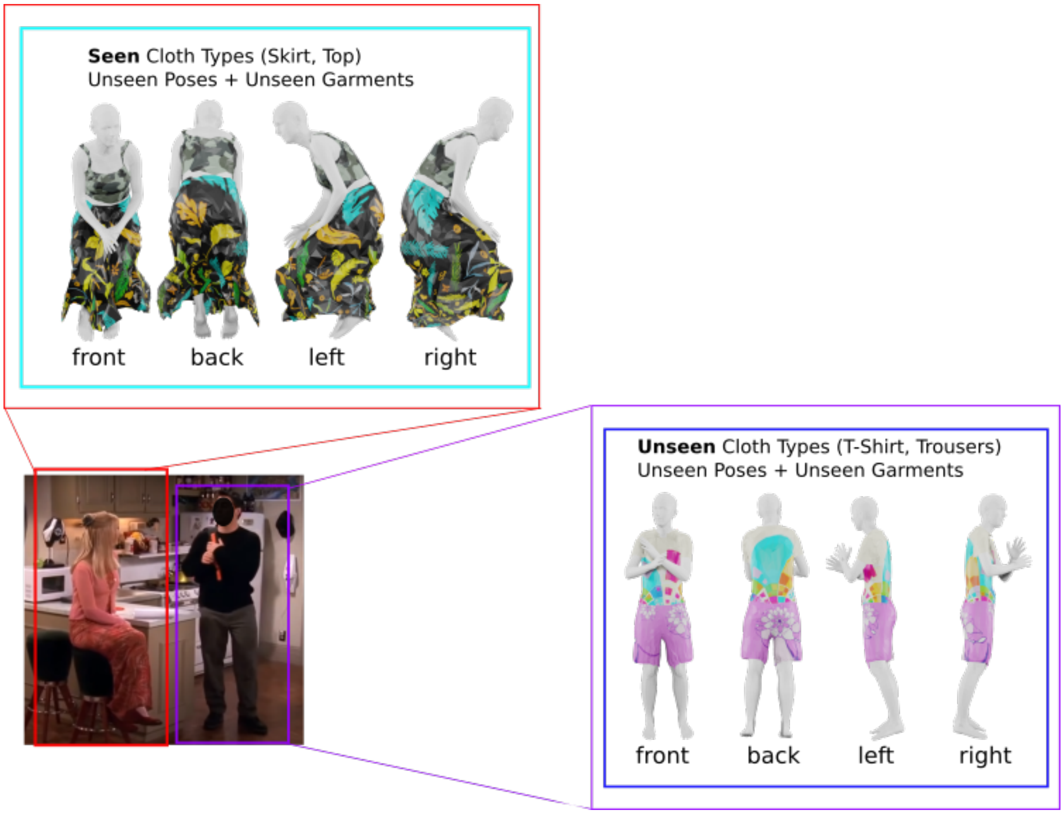

Figure S4: Textured outputs: Sample results on YouTube video frames. Here unknowns are the body shapes, the poses, and the garments (skirt and top). The T-shirt and Trousers results show the generalization of *GarSim* on unseen cloth types.

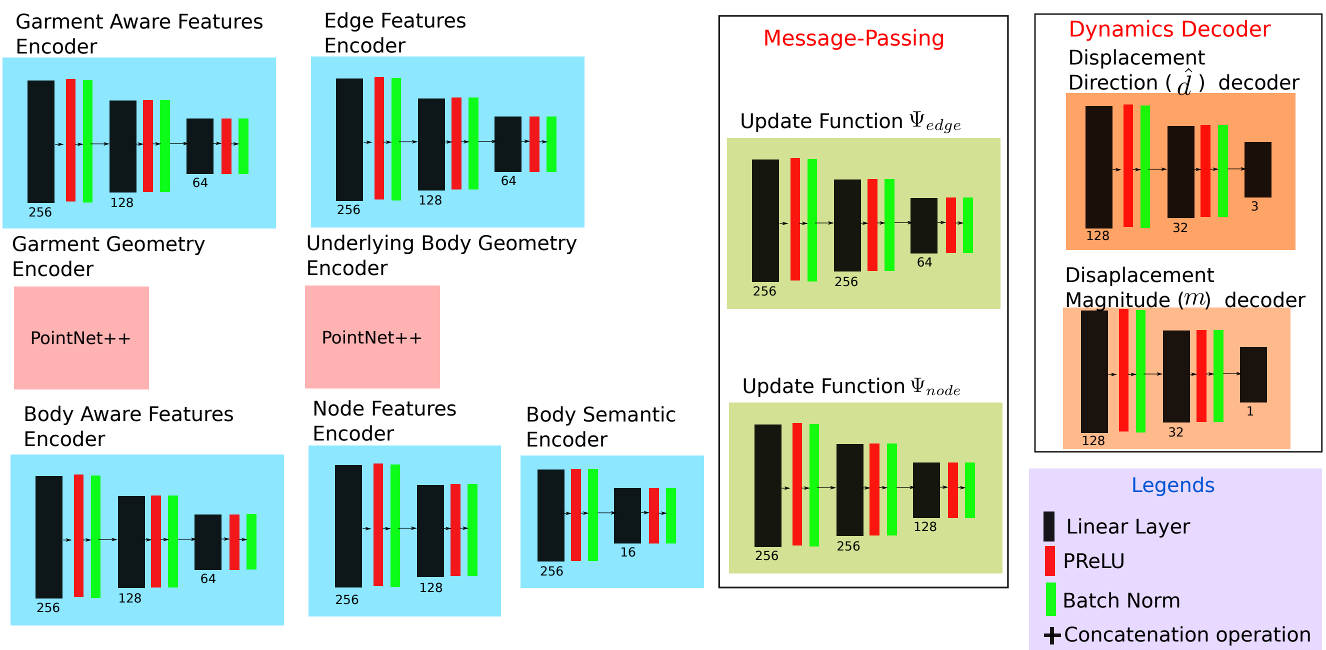

Figure S5: Architecture of all learnable blocks of *GarSim* . The number of neurons of linear layers are shown below them. **Refer main paper Fig. 2 for the input output flows and specifications**. The input to the garment and body geometry encoders are the mesh vertices of garments and the partial body respectively.

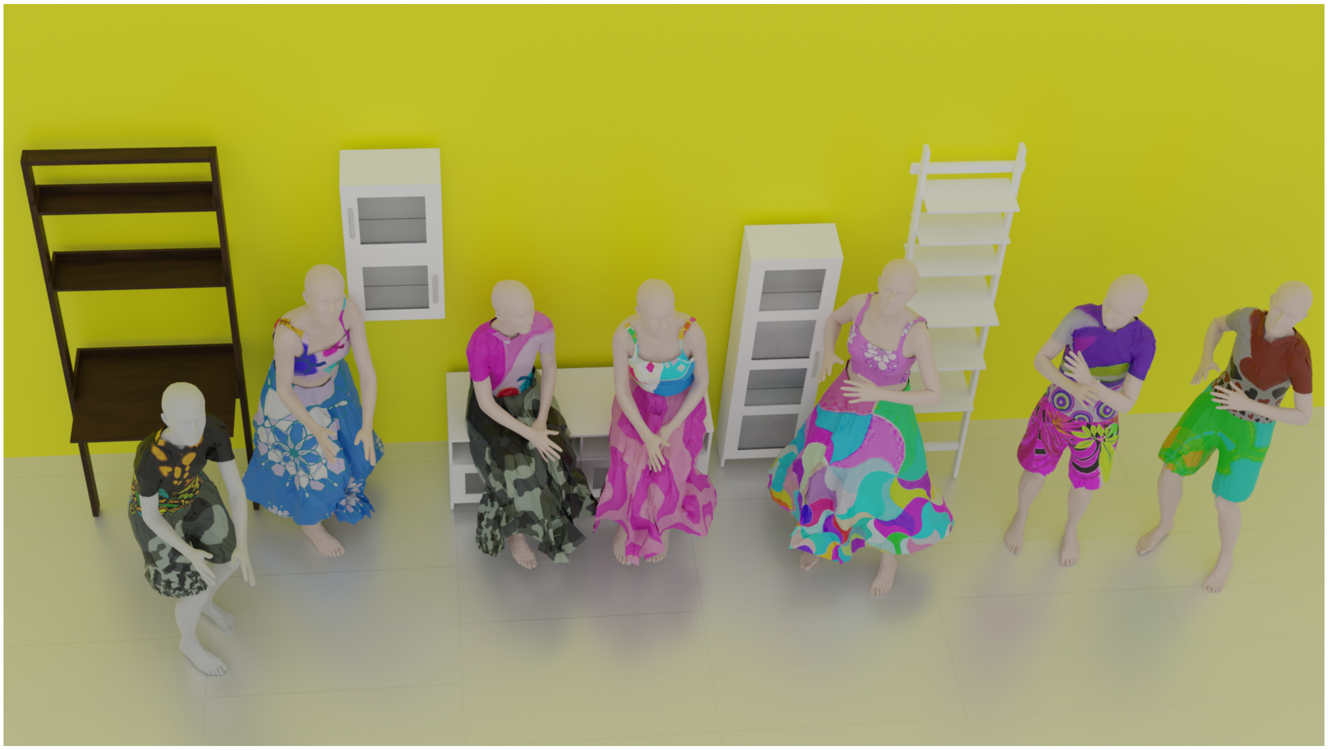

Figure S6: Rendering garments types (top, skirt, T-shirt, Trouser) on top of unseen body shape and poses. **Note:** Garment types T-shirt and Trousers **were not** present in the training set. This shows *GarSim* generalizes to unseen garment types. Here, body shapes and poses are retrieved from a YouTube video. Augmenting the avatars draped with variety of cloth types in a virtual environment will further enable in accomplishing various use cases served by a recent metaverse concept. *Please watch supplementary video for another set of rendering (Jumping and Running)*

Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.