SUPPLEMENTARY MATERIALS

NF-ICP: NEURAL FIELD ICP FOR ROBUST 3D HUMAN REGISTRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this document, we provide further technical details of our method. We also provide an extensive collection of results from the data presented in the paper and a discussion about failure cases, pointing to interesting challenges for future works. We remark here that our method works considering only the point cloud and does not consider the mesh of the target, which we visualize when available for clarity. We show the target mesh only for visualization purposes.

1 TECHNICAL DETAILS

1.1 BACKBONE DETAILS

All the trained NF and Uni- baselines use the same backbone network of Corona et al. (2022), which is composed of an IFNet (Chibane et al. (2020)) to extract global features of the target shape, and an MLP network to query $\mathbb{R}^3$ points and output offsets (or the universal embedding, in case of Uni- baselines).

IFNet Network. We start from a $64 \times 64 \times 64$ voxelization of the point cloud. Each voxel is represented as the distance from its center to the nearest point of the input. We also compute the discrete gradient of the functions along the three coordinates, and we apply this information to the input. Then, we use twelve 3D Convolutional layers with the following numbers of filters: $(64, 64, 128, 128, 192, 192, 256, 256, 256, 256, 256, 256)$. We assign to each point a set of local and global features considering the output of intermediate layers (one every two) for a total of 1152 features.

MLPs Query Network. This network takes the coordinates of a point with the global features extracted by IFNet and predicts all the displacements towards 690 locations, which are the ground truth registered template vertices. In the proposed LVD_{16}, we divided this network into 16 MLPs, each one composed of 6 layers of dimensions $(256, 512, 512, 512, 512, n)$ where n is the number of template vertices belonging to that local part. The local region is obtained by performing spectral clustering on the template described in the next paragraph. The activation function for the first five layers is the ReLu (Nair & Hinton (2010)).

Template Segmentation. While we could segment the SMPL model using its skinning weights of the 24 parts, this would need setting thresholds for vertices affected by multiple joints, and grouping them in a different number (e.g., 10 or 16) would require further manual design. Instead, we rely on spectral clustering as described in Liu & Zhang (2004). This approach lets us automatically set a number of segments and divide the template relying on its geometry. To divide the template into m segments, this algorithm requires computing the Laplace-Beltrami Operator for the surface, collecting its m eigenvectors associated with the m smallest eigenvalues, and using them as features for K-Means clustering. We use the Laplace-Beltrami Operator discretization provided by the Robust Laplacian library in Python Sharp & Crane (2020).

Training. The backbone network has in total 36776428 parameters, and it is trained end-to-end. We use a batch size of 8, a learning rate of $1e-4$, and an Adam optimizer (Kingma & Ba (2014)). During training, for each training sample, we query 2200 points: 400 uniformly sampled in $\mathbb{R}^3$, and 1800 near the input point cloud, by perturbing the points with random Gaussian noise of standard-deviation 0.05. Hence, the network predicts for each of the 2200 points the offsets towards the ground truth template vertices positions. Before computing the loss, we rescale the offsets with a maximum norm of 0.05, so the network learns to converge in small steps.

1.2 Template fitting with NF-ICP - Details

NF-ICP iterations For each iteration of our procedure, we consider the vertices of the target point cloud, pass them toward the query network, compute the loss described in Equation 5 of the paper, and backpropagate it to update the weights of the backbone network. This is done for 20 steps using a learning rate of $1e-5$ and an Adam optimizer Kingma & Ba (2014). If the input point cloud is particularly dense, we sample 20000 random points at each iteration.

Neural Field Evaluation. To evaluate the LVD NF, we initialize 690 points at $(0, 0, 0)$ coordinates, and we update their positions 50 times. For the OneShot baseline, we initialize 690 points near the target surface and update their position once. After this, we fit a full 6890 SMPL model to the predicted points, optimizing its parameters for 2000 steps and a learning rate of $1e-1$, using an Adam optimizer (Kingma & Ba (2014)). The loss is a standard L1, plus the statistical pose prior of Pavlakos et al. (2019) (weighted for $1e-8$) and an L2 regularization on the shape parameters magnitude (weighted for $1e-2$).

Chamfer Refinement. The Chamfer refinement optimizes the SMPL parameters using the bidirectional Chamfer Distance between the obtained SMPL and the target point cloud. We perform 500 iterations, with a learning rate of $2e-2$ using an Adam optimizer (Kingma & Ba (2014)). We generally considered the bidirectional Chamfer loss when the input point cloud is a full human model. Otherwise, we optimize only for one direction. We keep the regularizers' weights similar to the previous step.

SMPL+D. If the shape contains details that the SMPL model cannot express, we compute the displacements $\mathbf{O} \in \mathbb{R}^{6890\times 3}$ to fit the target point cloud. To do so, for each of the SMPL vertices, we optimize the bidirectional Chamfer distance, plus an L2 regularization on the offsets magnitudes and a Laplacian regularization as designed in Gao et al. (2020):

$$L_{lap} = \frac{1}{6890}\sum_{i=1}^{6890}(\Delta(\mathbf{v}_i + \mathbf{O}_i) - \Delta\mathbf{v}_i)^2, \tag{1}$$

Where $\mathbf{v}_i$ the i-esim vertex the SMPL template resulting from the previous step, and Δ the Laplace-Beltrami Operator (in our case, obtained using the Robust Laplacian Python library Sharp & Crane (2020)). This loss promotes the offsets to preserve the smoothness of the input surface. We rely on an Adam optimizer (Kingma & Ba (2014)).

2 Further Ablations

Number of segments. We trained our backbone network considering different number of segments: 1 (standard LVD), 10, 16, and 24. In all cases, we resized the query MLPs such that they have a comparable number of parameters. Table 1 reports evaluation results. The localization produces a significant gain, while its excess causes performance degradation. Given that using 10 or 16 segments performs similarly, we opted for 16 since it performs better on real scans.

Ablation on registration steps. To validate the contribution of our registration method, we report in Table 2 the quantitative results after enabling different components. We appreciate the significant improvement provided by NF-ICP, upgrading LVD_{16} prediction by a margin of 18% on real scans. To highlight the nature of the contribution, we report in Figure 1 a visualization of the normalized error following the protocol of Kim et al. (2011) (*i.e.*, error on X-Axis, percentage of correspondences

	FAUST$_R$	FAUST$_S$
LVD$_1$	4.78	3.54
LVD$_{10}$	**4.27**	3.13
LVD$_{16}$	4.35	**3.11**
LVD$_{24}$	7.92	5.58

Table 1: Ablation study on the number of local MLPs. Increasing the number of components helps, but an excess is also detrimental. Using 10 and 16 segments lead to a good tradeoff, and our full model relies on the second, which performs better on real data.

	FAUST$_R$	FAUST$_S$
LVD$_1$	4.78	3.54
LVD$_{16}$	4.35	3.11
LVD$_{16}$+NF-ICP (Ours)	2.97	2.55
Ours+R	**1.76**	**1.85**

Table 2: Ablation results for different elements of our pipeline. NF-ICP largely improves the results of the backbone, promoting the better convergence of the refinement.

below that error on Y-Axis). We emphasize that NF-ICP increases the number of correct matches (doubling the points with 0 error), and is more robust (the curve saturates faster). The similar saturation of curves for our approach before and after the refinement suggests that our method provides a good initialization for the convergence of Chamfer, which uses the geometry to align local features.

3 Further Results

3.1 Real scan registrations - DFAUST

DFAUST	v2v
LVD$_{16}$+R	3.26
Ours+R	**3.08**

Table 3: Evaluation on 417 DFAUST shapes compared to the ground-truth registration. NF-ICP improves the initialization for Chamfer Distance refinement, leading to better final registration.

To further validate the relevance of our NF-ICP, we validate its impact on the real scans of DFAUST Bogo et al. (2017). Such scans contain noise, missing geometry due to occlusion, and in particular, identities significantly far from the AMASS distribution Mahmood et al. (2019). We select 11 sequences of different subjects, and we run our method with and without NF-ICP on a frame every ten for a total 417 scans. We evaluate our error using the distance from the provided ground truth. We report the results in Table 3. The results confirm that despite the strong initialization provided by the data-driven backbone and the robust SMPL refinement using Chamfer distance, our method provides a further improvement, obtaining a more precise registration. A qualitative comparison is reported in Figure 2.

3.2 NF-ICP on Point clouds

Here we would emphasize the role of NF-ICP refinement in the whole pipeline when the input is a partial point cloud. In Figure 3 we report some results of our method on point cloud coming from DSFN Burov et al. (2021) (first two rows) and CAPE Ma et al. (2020) (last two rows), enabling and disabling the NF-ICP inside the full pipeline. The use of NF-ICP refinement is crucial and lets the network recover the correct pose of the subject. Without it, the prediction of the backbone is not good enough to initialize the Chamfer refinement, leading to catastrophic failures. We discuss failure cases for our method in the next Section.

3.3 Animation

Rigging noise acquisition. The recent advancements in NeRF representations proposed by Mildenhall et al. (2020) enabled several tools that provide 3D reconstructions of objects from monocular videos. Among these, Luma AI provides a web interface that automatically segments and outputs a textured geometry of the object of interest. Automatic character rigging would allow the use of these reconstructions for a number of downstream applications. However, it often requires watertight meshes and manual annotations from the user. Figure 4 shows a user's 3D reconstruction where we register and automatically rig a user, enabling its usage in tasks such as animation. We show a

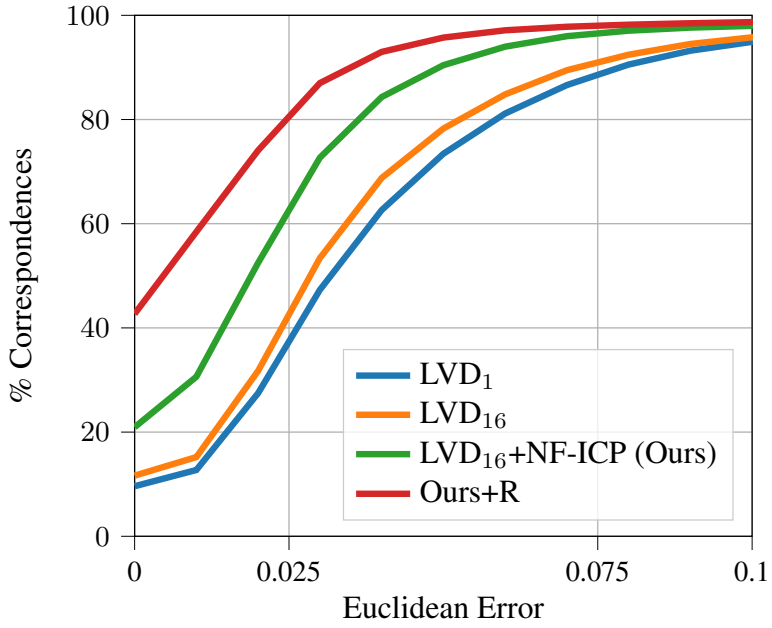


Figure 1: Error curves for ablation study of the pipeline components. We observe that our NF-ICP refinement produces a significant impact, doubling the number of exact matches and providing better correspondences.

similar example with heavier cloths in Figure 5. Despite the clutter and geometric gluing caused by the garments, our registration is precise enough to provide coherent rigging of the target shape.

3.4 Failure Cases

During our experiments, we observed some recurrent failure modes for our method. While we show robustness to clutter when it constitutes a significant part of the scene (*e.g.*, large objects), our method cannot distinguish between what is human or not. Our method performs well with disparate identities even significantly far from the training distribution, even on non-humanoids like the ones in Figure 6. Still, when this is combined with unusual poses, arms and legs might be wrongly located in space. Finally, our method can also recover the human posture in the presence of partial point clouds. However, if the input does not contain enough information to define the position of all the human parts, the registration may not find the correct locations for the missing ones attracted by the input point cloud. Examples of these failures are reported in Figure 7, where we can appreciate that some of the body parts are correctly located even in failure cases. We believe that data augmentation combined with segmentation prediction of the input to remove the clutter or highlight what template parts have an image in the input could be a promising direction to address these challenges.

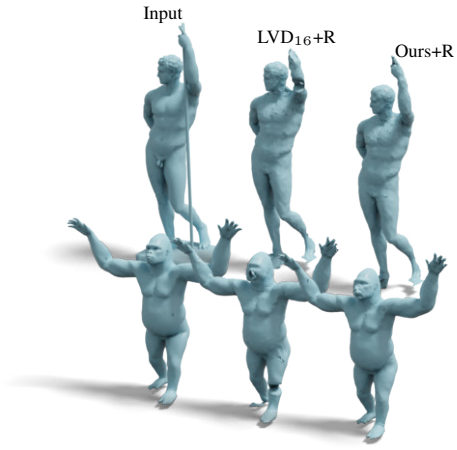


Figure 6: Some results on non-humans models. NF-ICP opens to promising generalization results in the presence of highly non-isometries and heavy clutter.

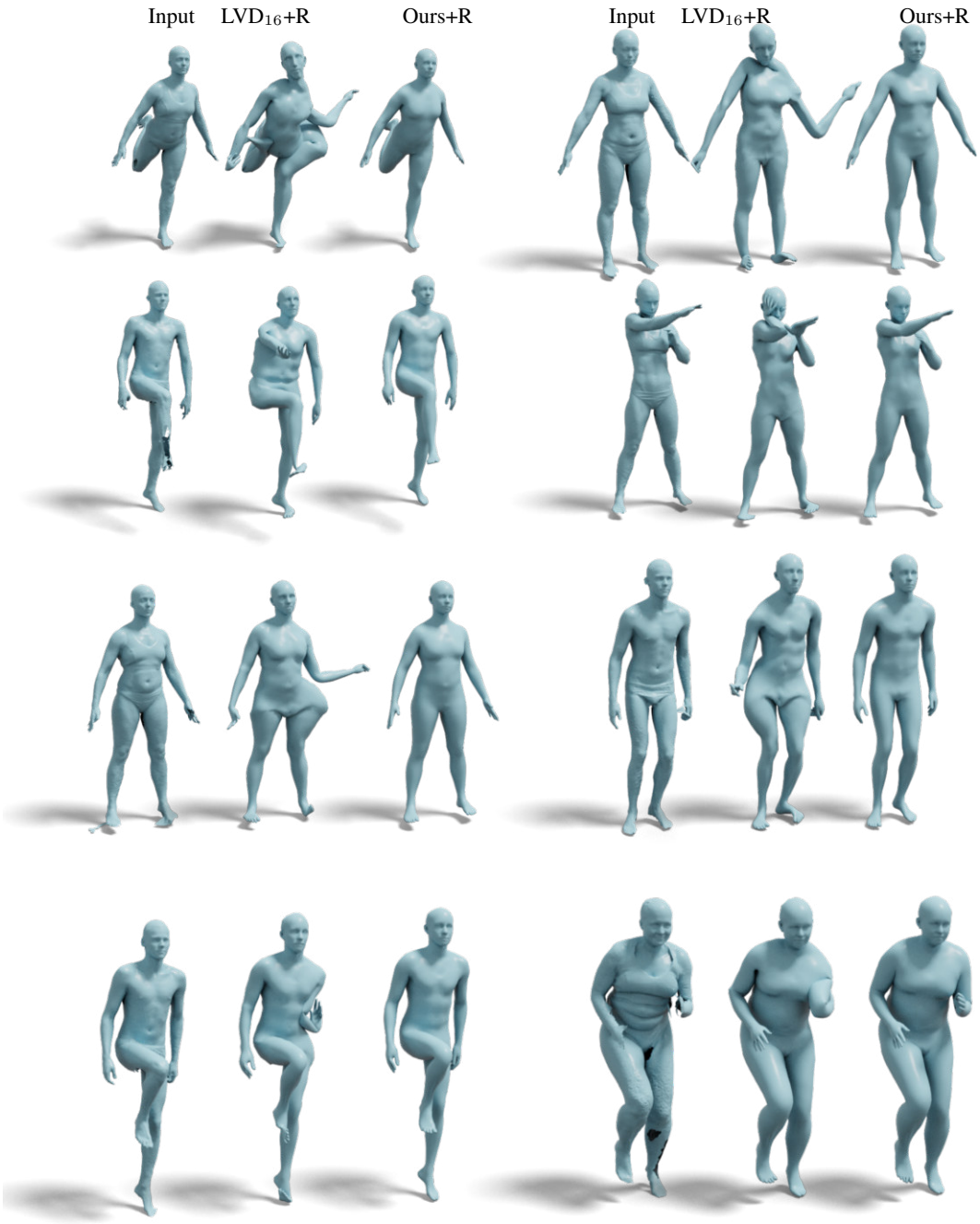


Figure 2: Comparison of results from the full pipeline without and with NF-ICP. DFAUST may contain unreferenced vertices, which significantly change the input implicit representation. NF-ICP helps to refine these results bringing the deformation of the backbone towards the majority of points, i.e., the target human.

Figure 3: Comparison of results from the full pipeline without and with NF-ICP. Partial point clouds are significantly far from the training distribution seen at training time by the backbone network. NF-ICP enables this challenging scenario, improving the geometry fitting.

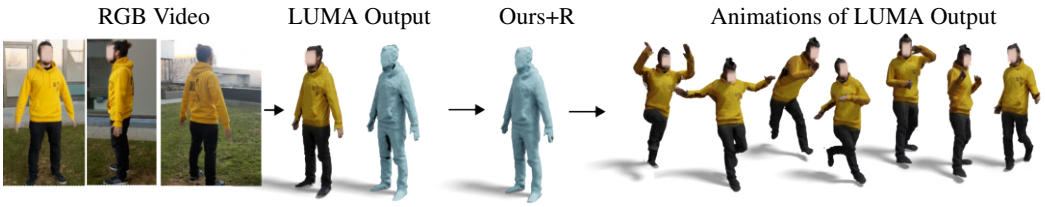


Figure 4: Animatable avatar from Luma AI. From left to right: Images from the smartphone video; the textured geometry obtained by Luma AI NeRF; our registration; motion capture animation applied to the input data.

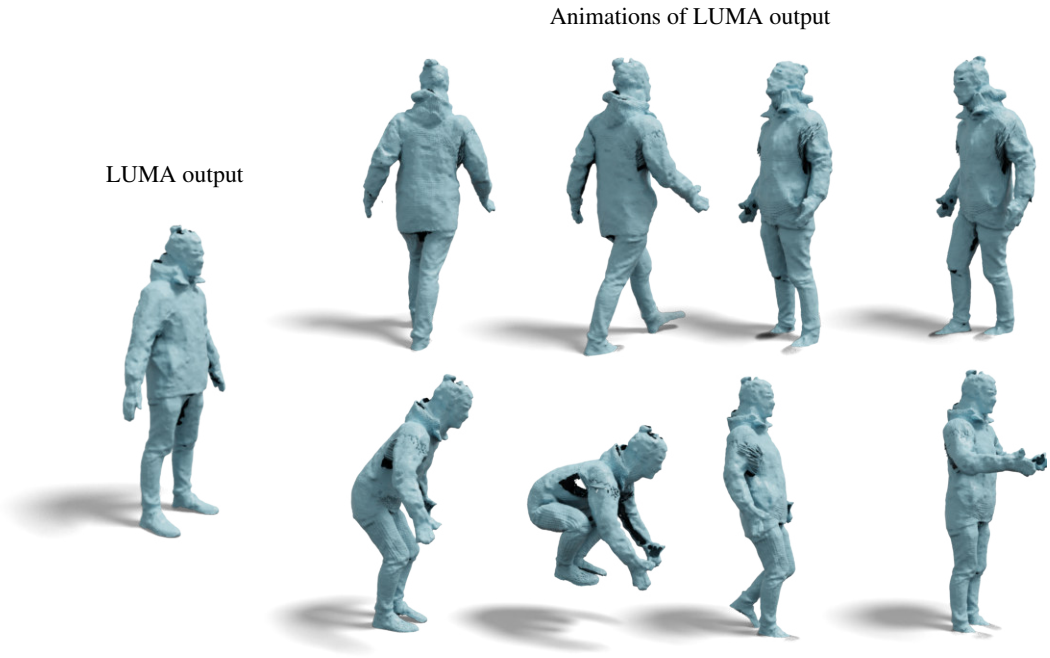


Figure 5: Animatable avatar from Luma AI. On the left, the geometry extracted from LUMA. On the right, our animation results. Our animation results show semantically coherent motions despite the heavy clothing and gluing that ruined the geometry extracted by LUMA.

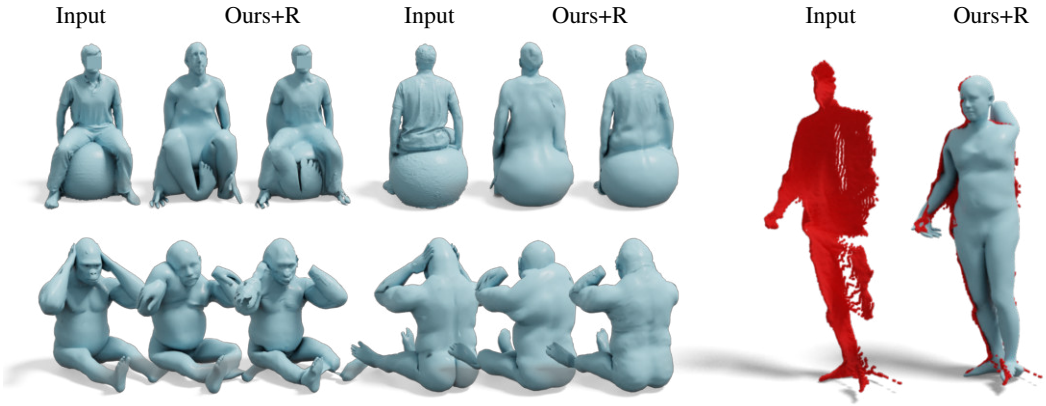


Figure 7: Some failure cases for our pipeline. The presence of heavy clutter, highly unnatural poses, and the complete absence of limbs are among the main failure cases we observed.

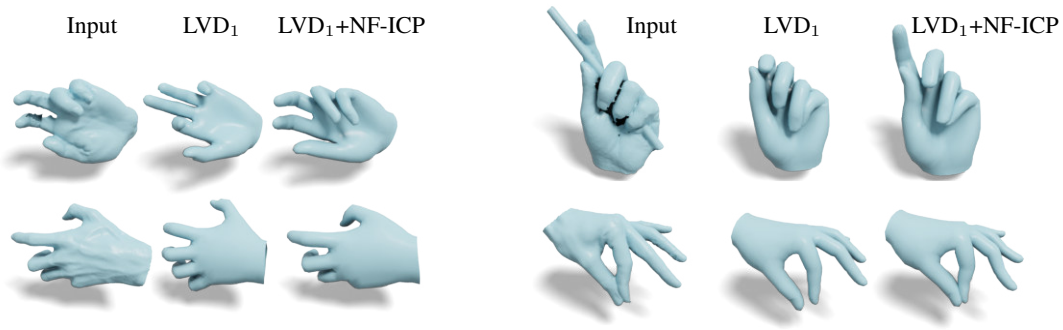


Figure 8: Further results on hand fitting.

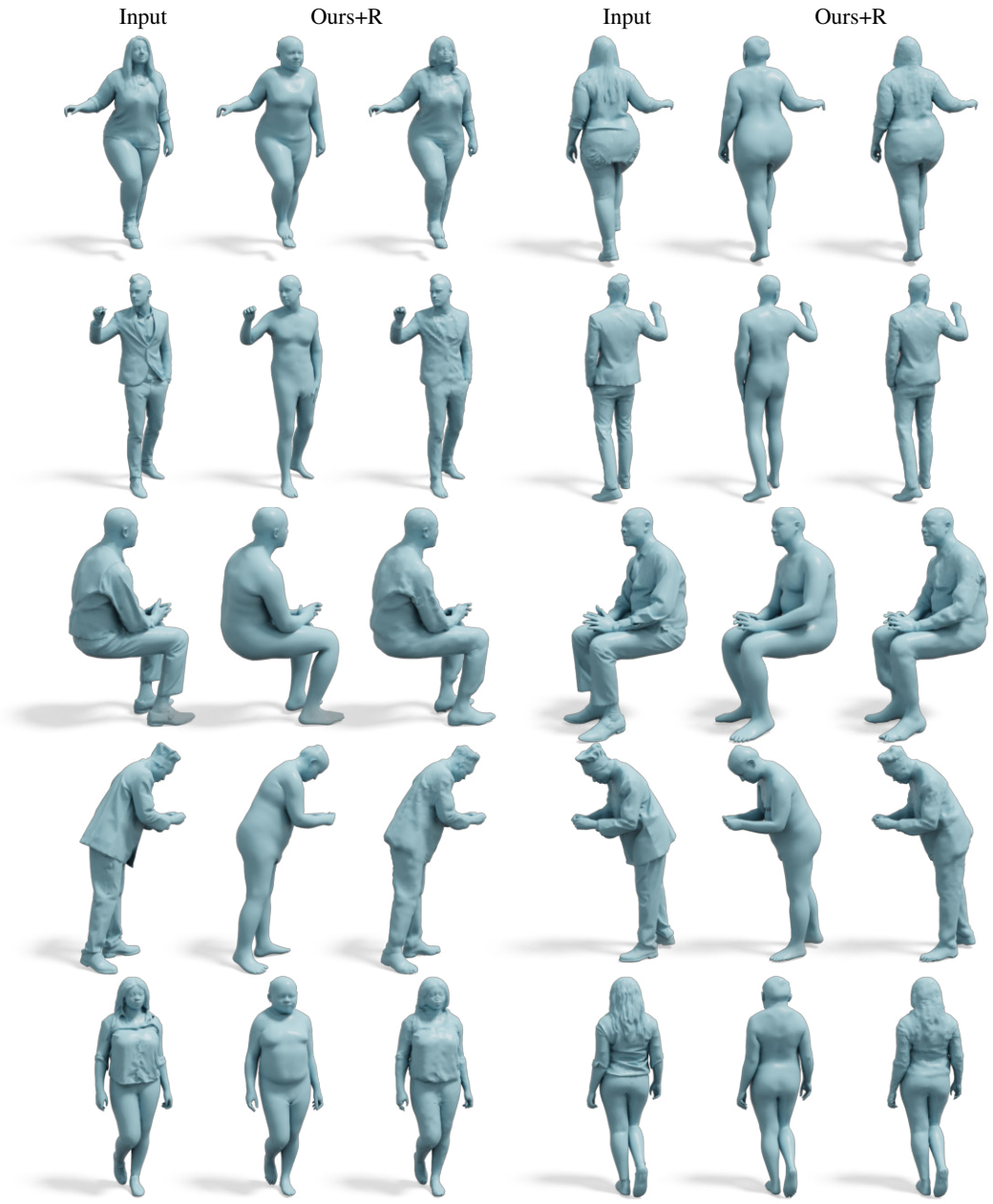
3.5 Further Qualitative results

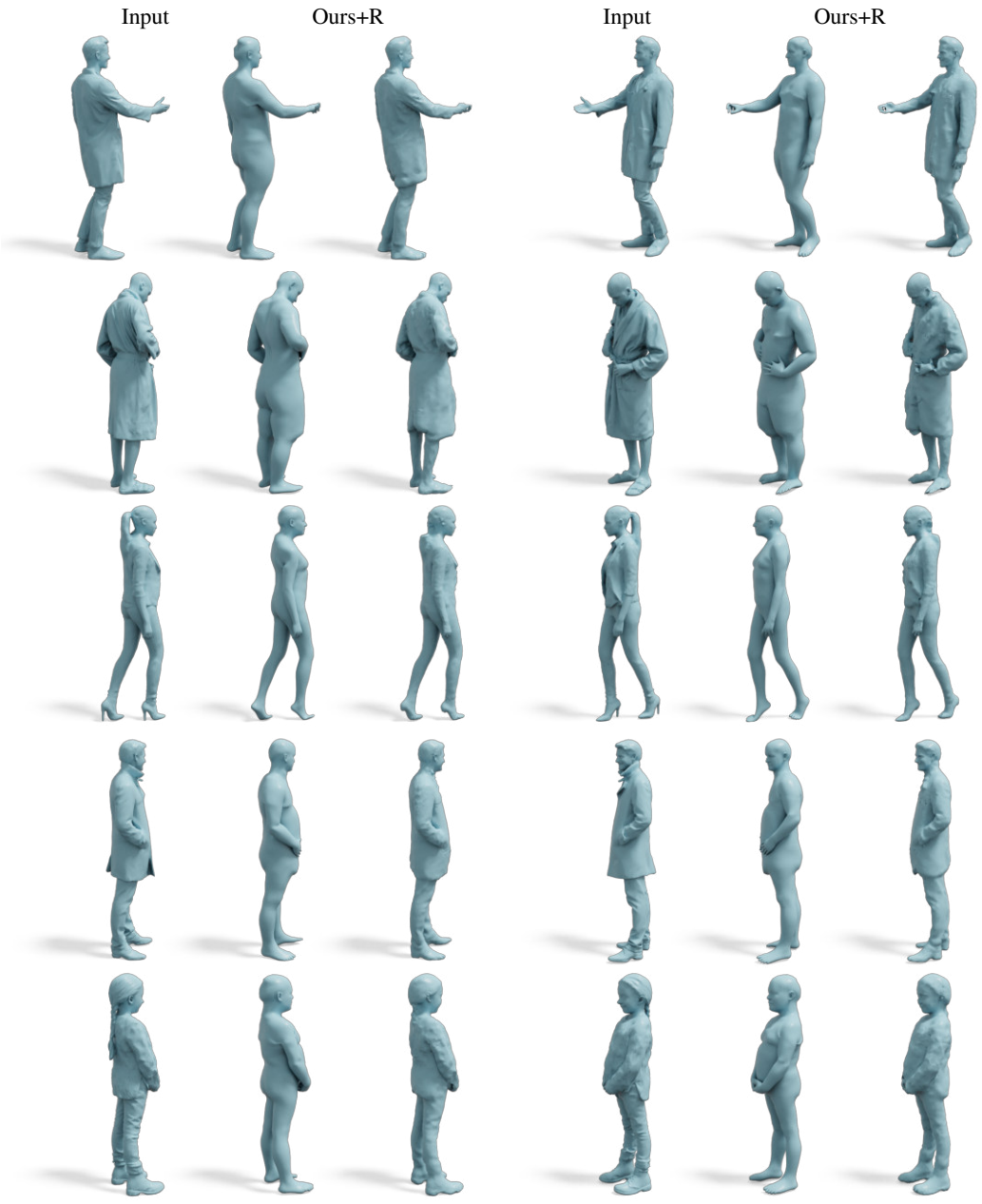
In the last pages of this document, we show many qualitative results from different datasets. In order:

- Pages 9 to 12: RenderPeople (renderpeople)
- Pages 13 to 17: D-Faust (Bogo et al. (2017))
- Pages 18 to 21: Test shapes from FAUST challenge (Bogo et al. (2014))
- Pages 22 to 24: HuMMan (Cai et al. (2022))
- Pages 25 to 27: BEHAVE (Cai et al. (2022))

For each row, we will display the input and the result of our Ours+R pipeline, both with and without the SMPL+D refinement on the left-hand side. However, for HuMMan and BHEAVE, we will not report the SMPL+D since the shapes in the former do not have significant details, and in the latter, the high-frequency features are primarily noise. It is important to note that we only visualize the meshes for clarity, and our method works solely from the point cloud and does not consider the target mesh in any way. We highlight the variety of poses, clothes, clutter, holes, and identities our method can solve. We also report some further qualitative results on hands fitting in Figure 8.

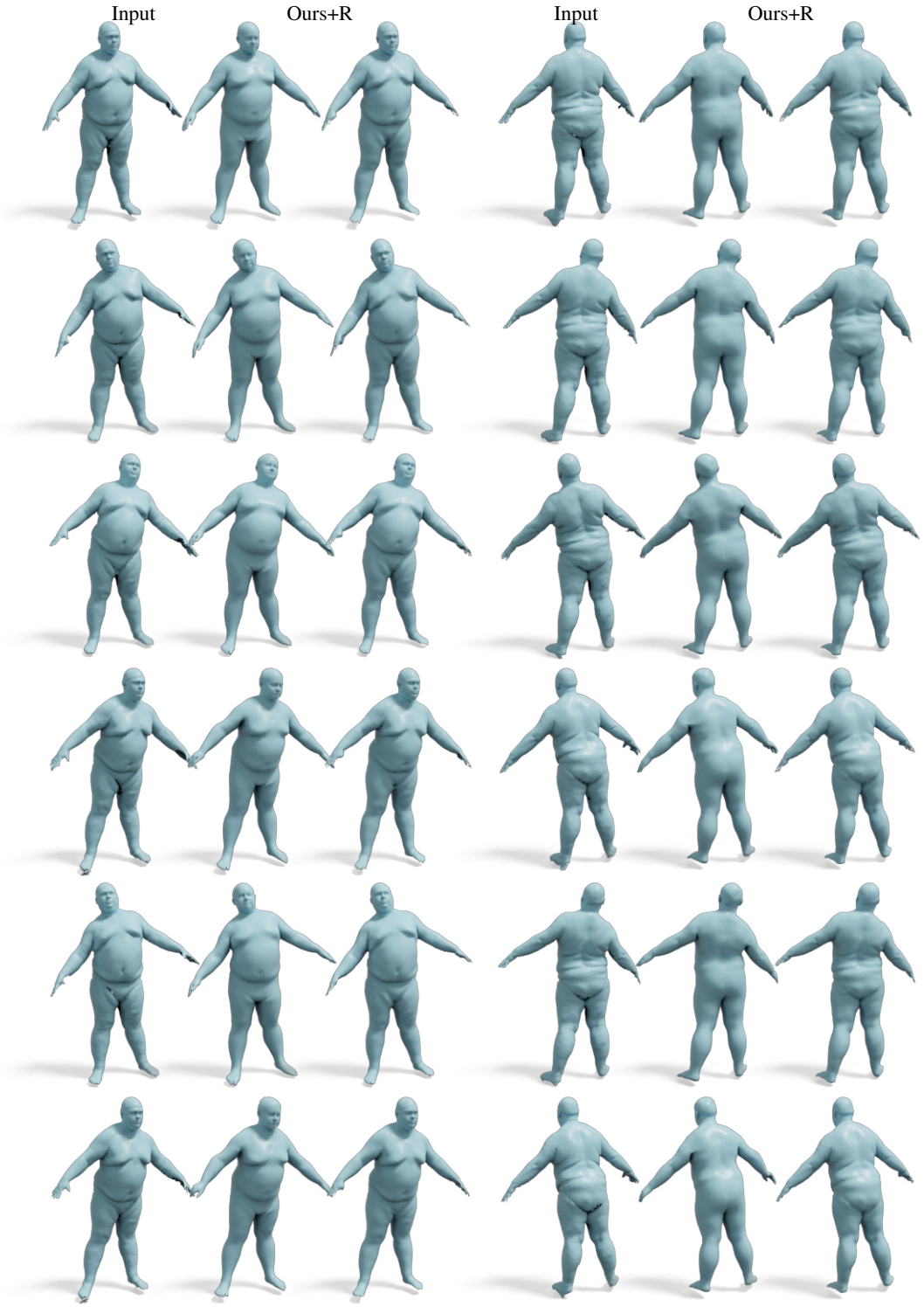

RenderPeople
Input
Ours+R
Input
Ours+R


RenderPeople
Input
Ours+R
Input
Ours+R


RenderPeople
Input
Ours+R
Input
Ours+R


RenderPeople
Input
Ours+R
Input
Ours+R

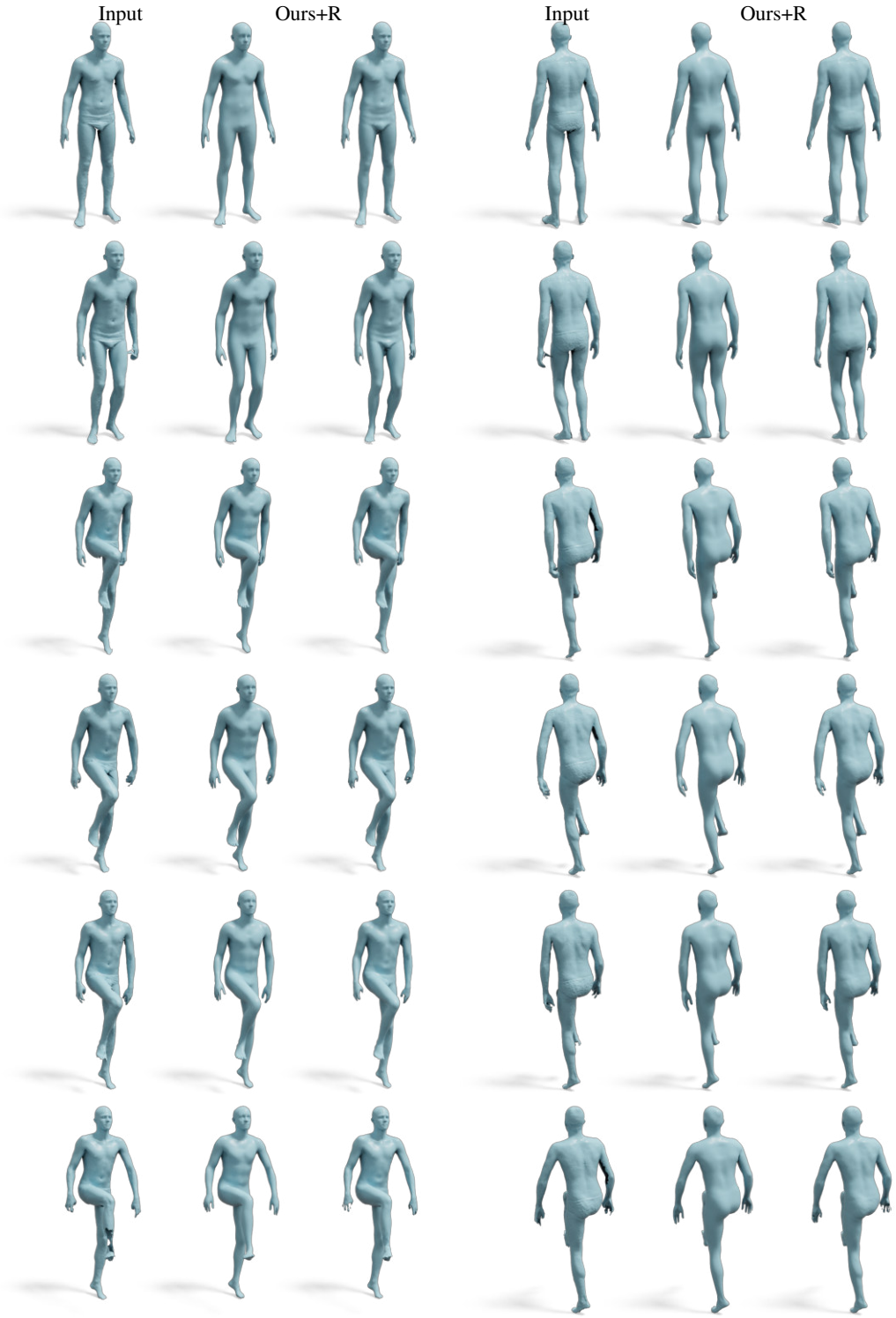
DFAUST



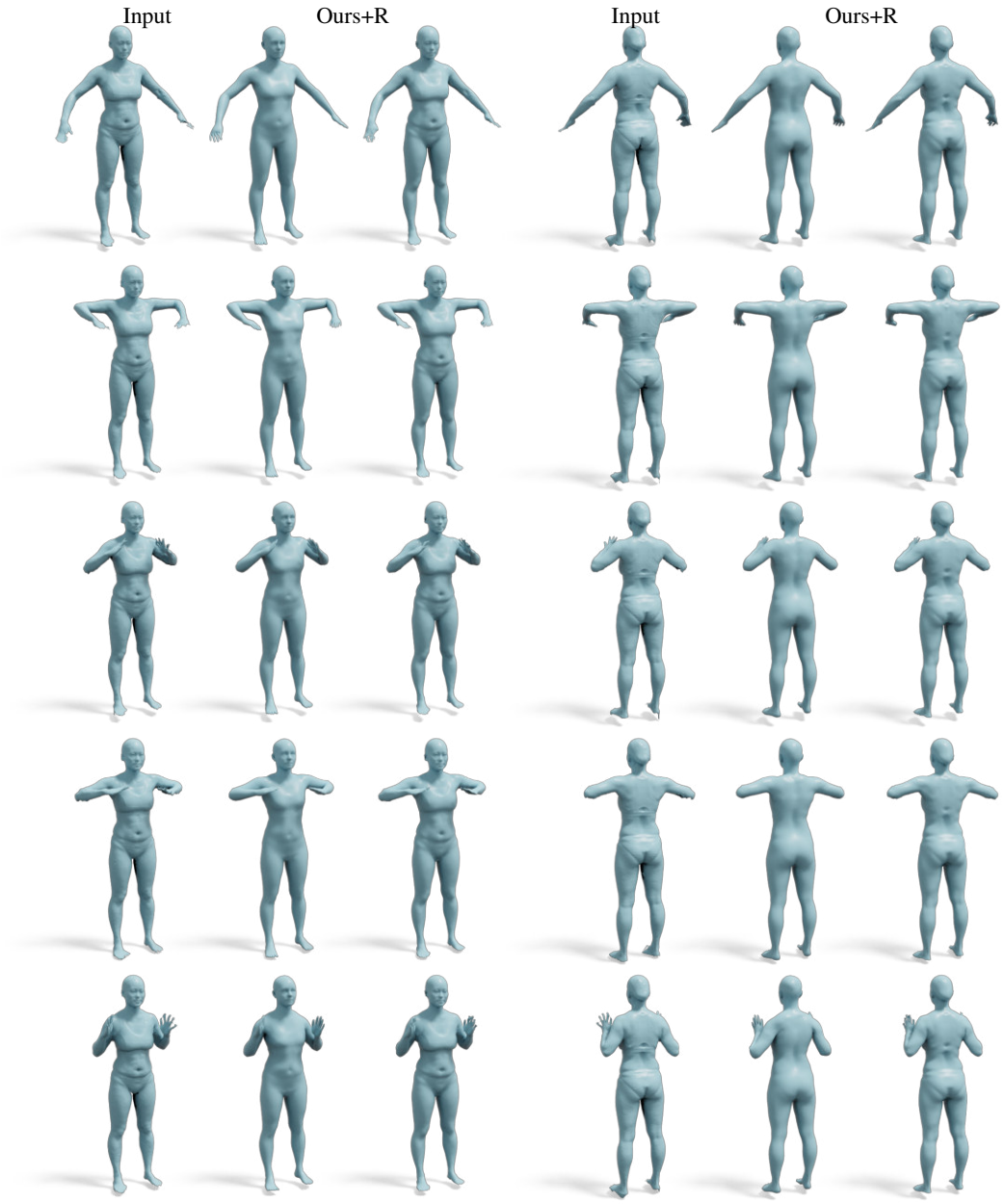
DFAUST

DFAUST

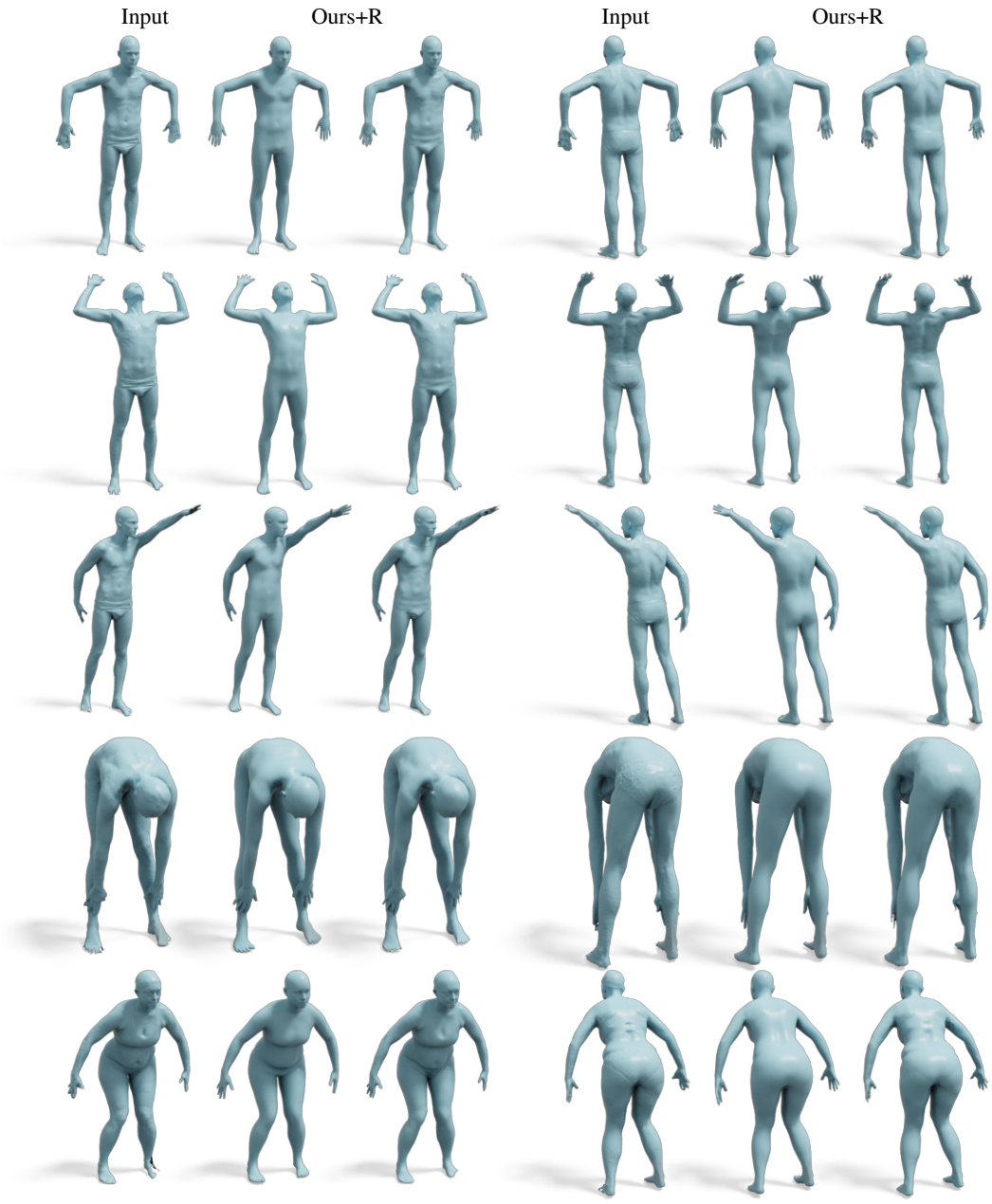
DFAUST



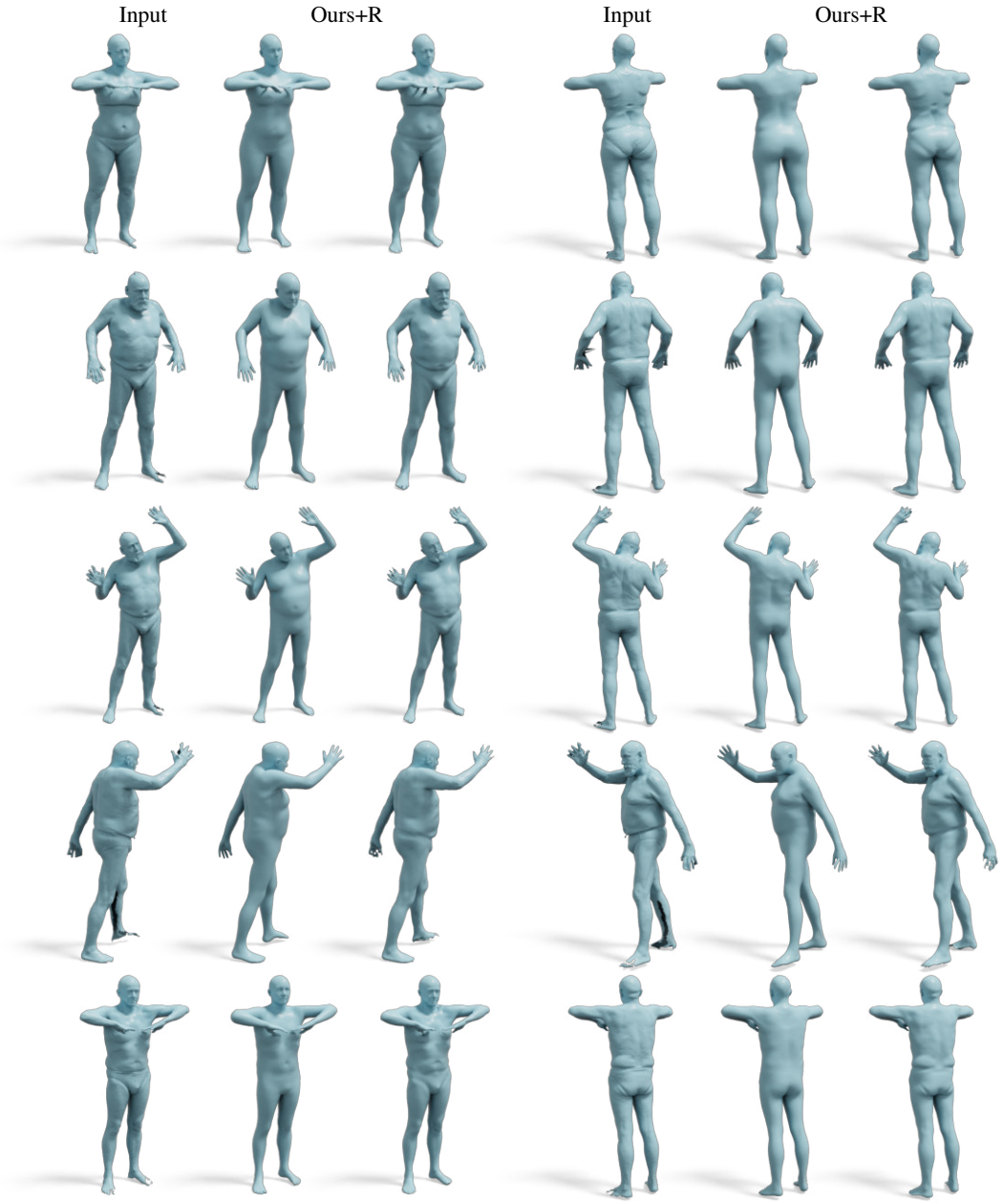
DFAUST



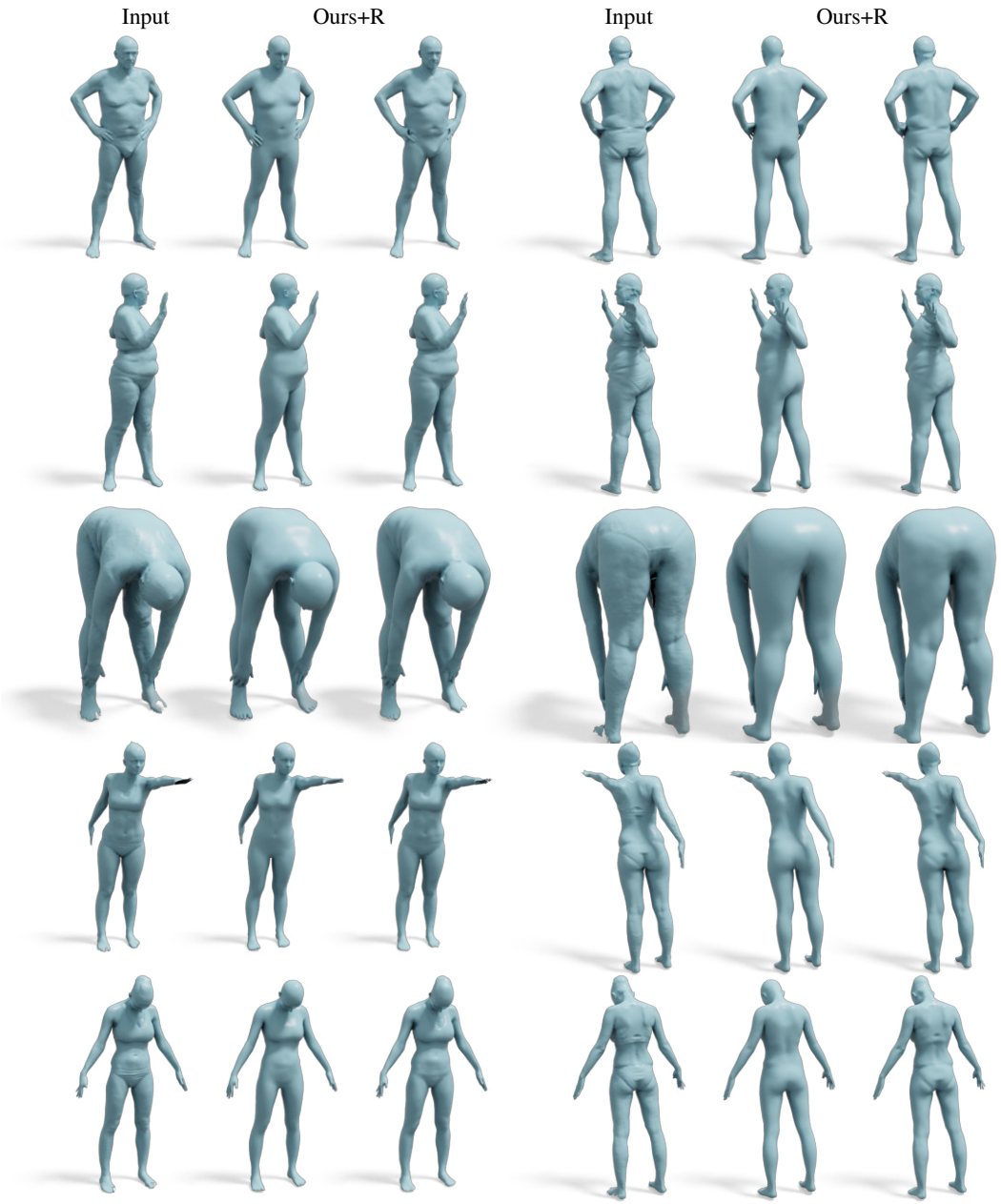
FAUST test set



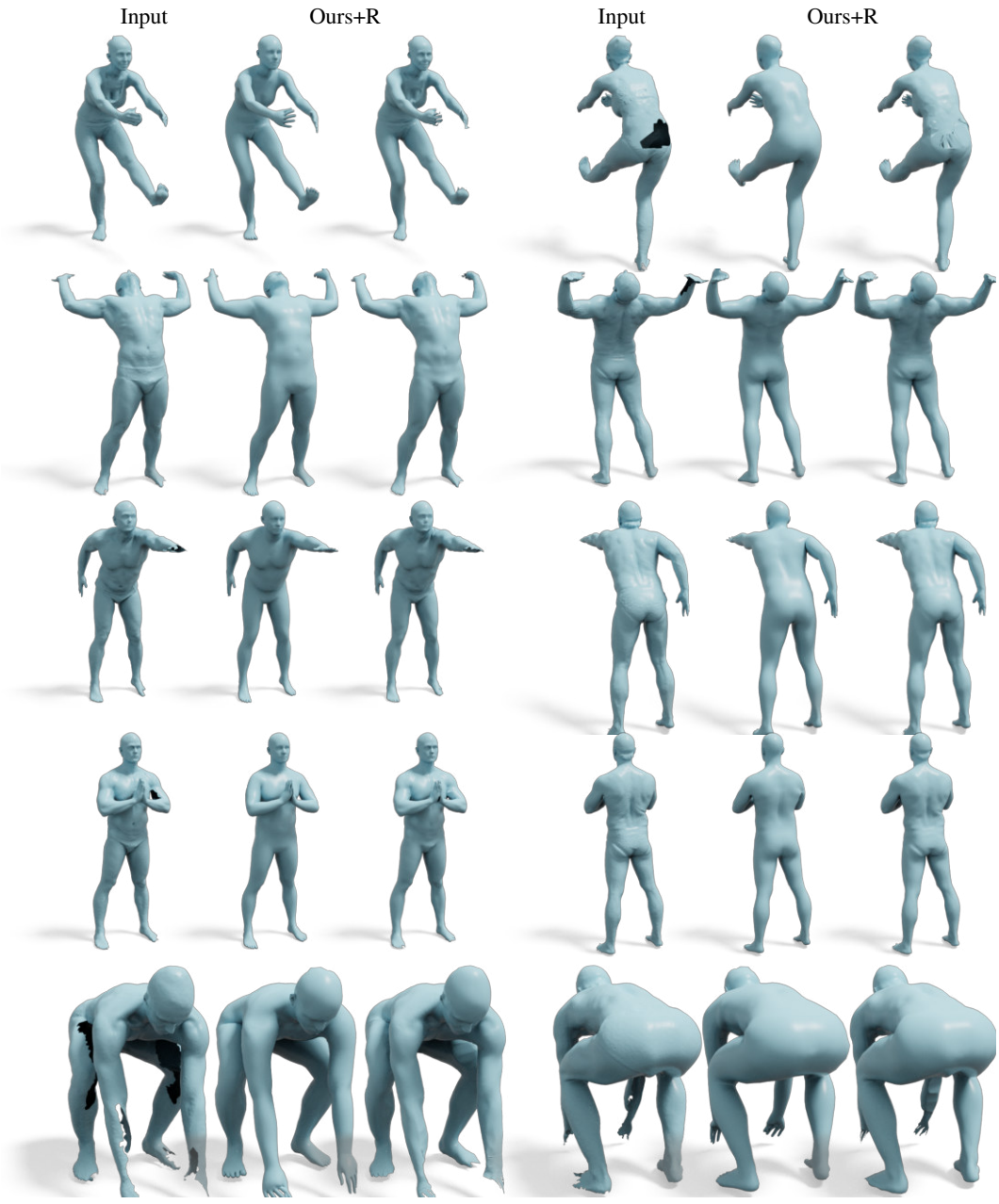
FAUST test set



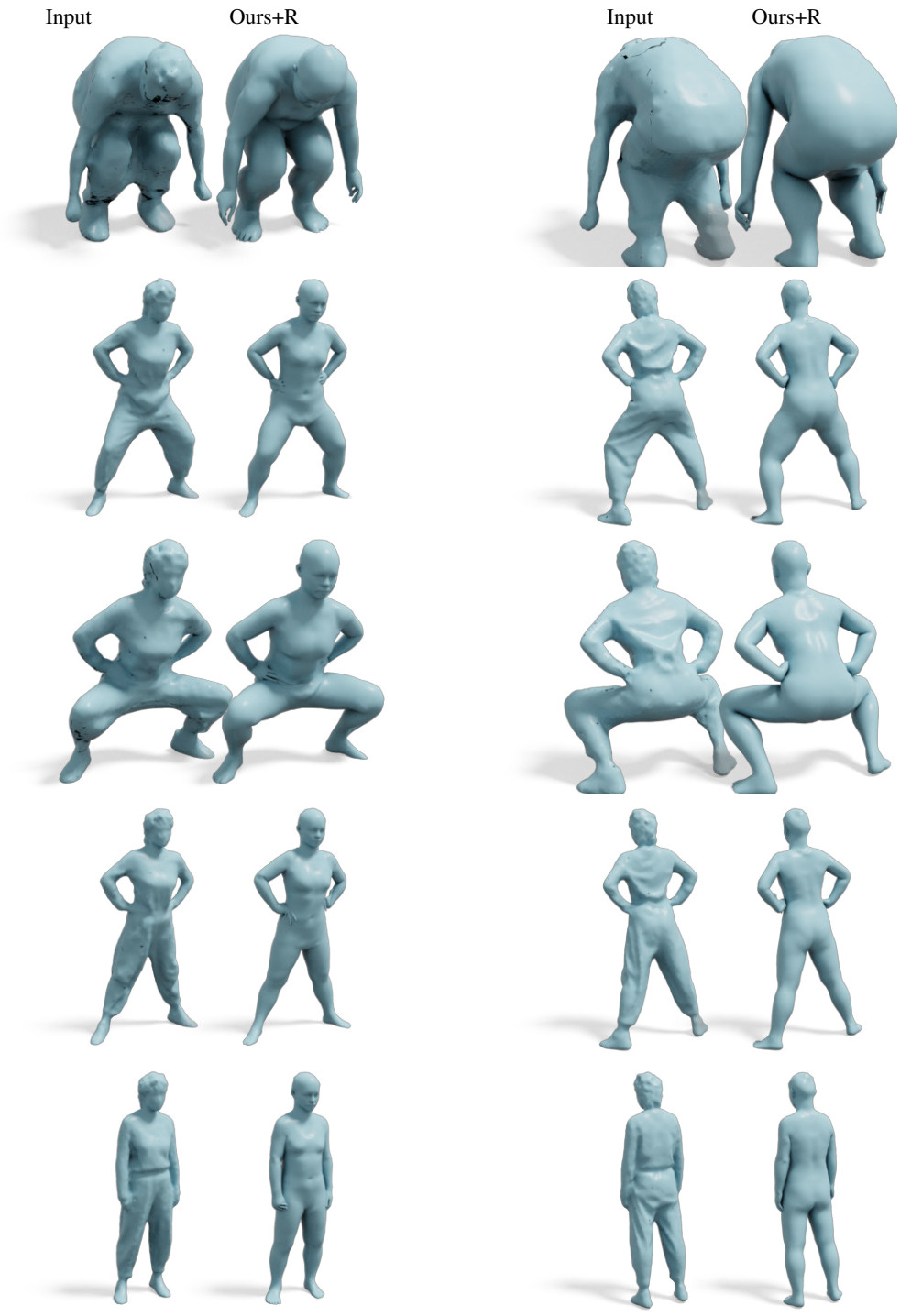
FAUST test set



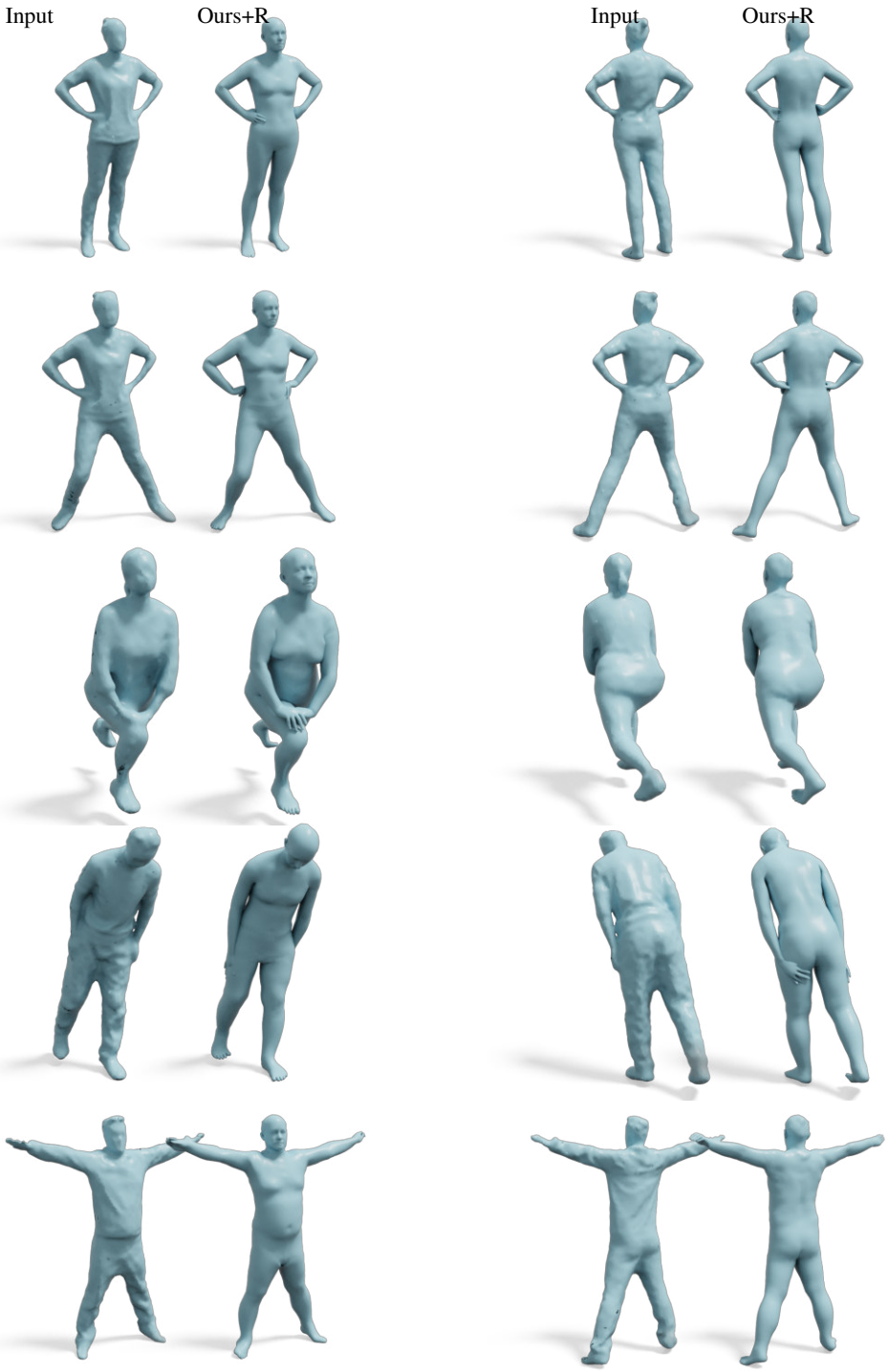
FAUST test set



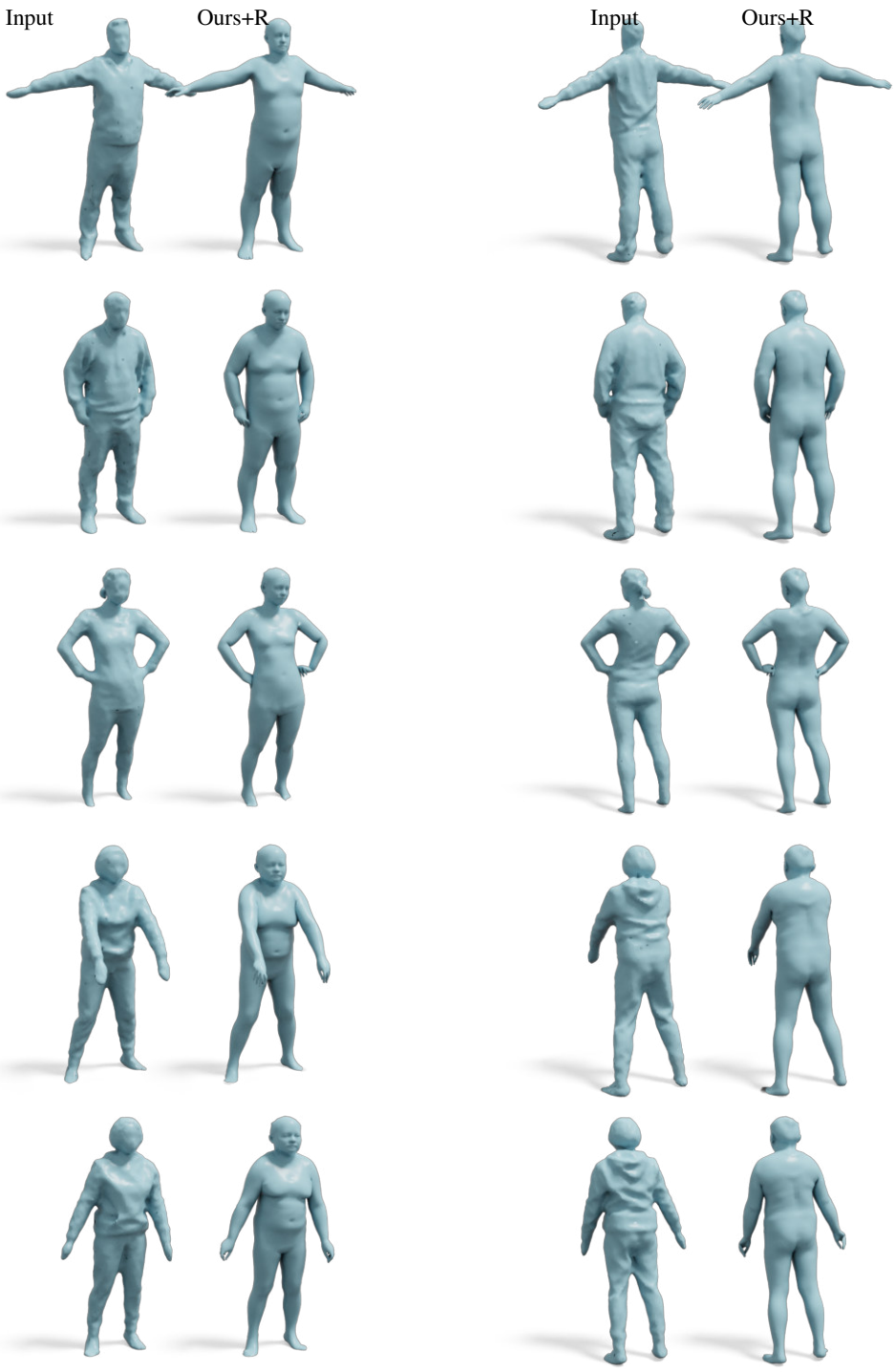
HuMMan



HuMMan

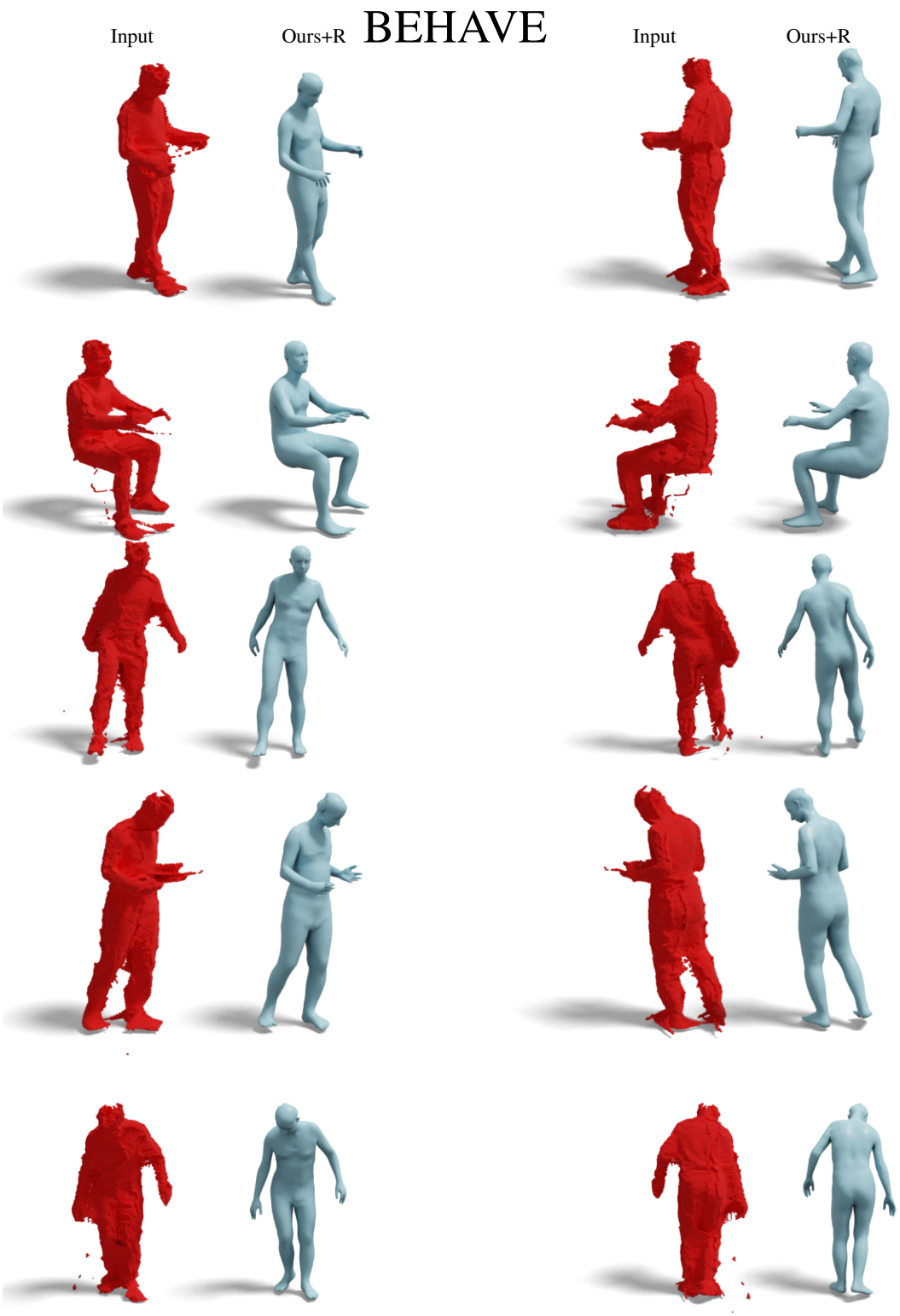


HuMMan




BEHAVE
Input
Ours+R
Input
Ours+R

BEHAVE


BEHAVE
Input
Ours+R
Input
Ours+R

REFERENCES

Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3794–3801, 2014.

Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6233–6242, 2017.

Andrei Burov, Matthias Nießner, and Justus Thies. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction, 2021.

Zhongang Cai, Daxuan Ren, Ailing Zeng, Zhengyu Lin, Tao Yu, Wenjia Wang, Xiangyu Fan, Yang Gao, Yifan Yu, Liang Pan, et al. Humman: Multi-modal 4d human dataset for versatile sensing and modeling. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pp. 557–577. Springer, 2022.

Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6970–6981, 2020.

Enric Corona, Gerard Pons-Moll, Guillem Alenyà, and Francesc Moreno-Noguer. Learned vertex descent: a new direction for 3d human model fitting. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pp. 146–165. Springer, 2022.

Jun Gao, Wenzheng Chen, Tommy Xiang, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Learning deformable tetrahedral meshes for 3d reconstruction. *Advances In Neural Information Processing Systems*, 33:9936–9947, 2020.

Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM transactions on graphics (TOG)*, 30(4):1–12, 2011.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Rong Liu and Hao Zhang. Segmentation of 3d meshes through spectral clustering. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, pp. 298–305. IEEE, 2004.

Luma AI. Luma ai. `https://lumalabs.ai/`. Accessed: 2023-09-28.

Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6469–6478, 2020.

Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5442–5451, 2019.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

renderpeople. renderpeople. `https://renderpeople.com/`. Accessed: 2023-09-28.

Nicholas Sharp and Keenan Crane. A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum (SGP)*, 39(5), 2020.