# Supplementary Material for UNIF: United Neural Implicit Functions for Clothed Human Reconstruction and Animation

Shenhan Qian<sup>1,2</sup><sup>®</sup> Jiale Xu<sup>2</sup><sup>©</sup> Ziwei Liu<sup>3</sup><sup>®</sup> Liqian Ma<sup>1</sup><sup>†</sup><sup>®</sup> Shenghua Gao<sup>2,4,5</sup><sup>®</sup> {qianshh, xujl1, gaoshh}@shanghaitech.edu.cn zwliu.hust@gmail.com liqianma.scholar@outlook.com

<sup>1</sup> ZMO AI Inc.

<sup>2</sup> ShanghaiTech University

<sup>3</sup> S-Lab, Nanyang Technological University

<sup>4</sup> Shanghai Engineering Research Center of Intelligent Vision and Imaging

 $^5\,$ Shanghai Engineering Research Center of Energy Efficient and Custom AI IC

## 1 Network Architecture

Our method uses multiple neural implicit functions to model a shape. All the networks share the same structure as shown in Fig. 1. The SDF networks are initialized with geometric initialization [1], and the pose condition networks are initialized with Kaiming initialization [2]. To relieve the conflict between the pose condition and geometric initialization [1], we initialize the last layer of each pose condition network with weights sampled from the distribution  $\mathcal{N}(0, 1e-5)$ . We use weight normalization [5] on SDF networks for stable training.



Fig. 1: The architecture of each neural implicit function. Every gray box represents a linear layer with its output dimension marked on the box. Every linear layer except the last one is followed by an activation layer (Softplus [7] with  $\beta = 100$ ).

<sup>\*</sup> Work conducted during an internship at ZMO AI Inc.

<sup>&</sup>lt;sup>†</sup> Corresponding author.

2 S. Qian et al.

#### 2 The Union Operation

During shape learning, taking the minimum means that supervising a data point only updates one network. This makes the model more sensitive to initialization and extreme body poses because once a body part is captured by a false network, it is hard for other networks to fight back due to the lack of chances to update. An alternative way is by using the smooth minimum function

$$d = \frac{\sum_{n=1}^{N} d^{(n)} e^{-\beta d^{(n)}}}{\sum_{n=1}^{N} e^{-\beta d^{(n)}}},\tag{1}$$

where  $\beta$  controls the steepness of the weighting ratio. However, this function makes steeper blending for points farther away from the surface. Since we only care about the difference between  $\min_{1 \le n \le N} d^{(n)}$  and other outputs instead of their concrete values, we use the following function in practice:

$$d = \min_{1 \le n \le N} d^{(n)} + \frac{\sum_{n=1}^{N} \Delta d^{(n)} e^{-\beta \Delta d^{(n)}}}{\sum_{n=1}^{N} e^{-\beta \Delta d^{(n)}}},$$
(2)

where  $\Delta d^{(n)} = d^{(n)} - \min_{1 \le n \le N} d^{(n)}$ , and  $\beta = 200$ .

### 3 Minimal Perimeter Loss

Our inspiration comes from PHASE [3], a method that bridges the signed distance field and the occupancy field with a logarithm transformation. To encourage learning a tight surface, PHASE [3] applies a punishment on the norm of the gradient of the occupancy field while maintaining the unit-gradient-norm property in the SDF field. This works because the norm of the gradient for occupancy achieves the highest at the zero level and minimizing the norm of gradient is equivalent to suppressing all the surfaces.

Since an SDF has a constant norm of gradient, we apply the sigmoid function  $(\sigma(x) = \frac{1}{1+e^{-\beta x}})$  on it to construct a peak of norm of gradient at the zero level. Then we can suppress surfaces by minimizing the norm of the gradient in the constructed field while maintaining the unit-gradient-norm property in the SDF field.

Our experiments also validate that this minimal perimeter loss is essential to suppress periodic extra surfaces when Fourier features [6] are used. However, since Fourier features can hurt the initialization of our method, we do not use it unless specified.

## 4 Blending weights from "Competing Bones".

#### 4.1 Derivation

In the main text, we have defined the tendency of a point to stay static on a part as rigidness. Now we present the derivation of the concrete definition.



(a) Rigidness defined by the (b) Rigidness defined by the (c) Rigidness defined by the projection of a point on projection of a point on projection of a point on the bones with a large angle. bones with a small angle. connecting line.

Fig. 2: 2D toy examples to illustrate rigidness computation.

Based on the common sense and the visualization of LBS weights of SMPL [4], we adopt a prior that the farther a point is away from the joint the more rigid it is. Therefore, we define a rigidness field by the projection of a point on the bone. For the example in Fig. 2a, the rigidness of bone  $b_1$  and  $b_2$  with respect to joint O are defined as

$$r_1 = \exp(\frac{\overrightarrow{OC} \cdot \overrightarrow{OA}}{\|\overrightarrow{OA}\|^2}), \quad r_2 = \exp(\frac{\overrightarrow{OD} \cdot \overrightarrow{OB}}{\|\overrightarrow{OB}\|^2}), \quad (3)$$

For the point X in Fig. 2a, when it moves closer to the farther end of bone  $b_1$  than  $b_2$ , its rigidness on bone  $b_1$  increases while its rigidness on bone  $b_2$  decreases. However, when the angle between two linked bones is less than 90 degree (Fig. 2b), moving along a bone leads to increase of both bones, which is incorrect. Therefore, we correct the rigidness definition as shown in Fig. 2c. In the corrected version, we connect the end points of both bones and split it with point Q by the ratio of bone lengths. Then we can compute the rigidness of point X with respect to either bone by

$$r_1 = \exp(\alpha_1 \frac{\overrightarrow{QP} \cdot \overrightarrow{QA}}{\|\overrightarrow{QA}\|^2} + \beta_1), \quad r_2 = \exp(\alpha_2 \frac{\overrightarrow{QP} \cdot \overrightarrow{QB}}{\|\overrightarrow{QB}\|^2} + \beta_2), \tag{4}$$

where P is the projection point of X onto the connecting line;  $\alpha_1$ ,  $\alpha_2$ ,  $\beta_1$ , and  $\beta_2$  are learnable parameters to adjust the rigidness of each bone. The corrected definition of rigidness works well regardless of the angle between the bones. Interestingly, when the angle between bones is 180 degree, the corrected version is equivalent to the first one.

#### 4.2 Rigidness Coefficients Learning

Since the rigidness coefficients, the scaling factor  $\alpha$  and the bias factor  $\beta$ , are defined for every pair of adjacent parts, we store them in matrices. Empirically, we init the matrices for  $\alpha$  to 2 and the matrices for  $\beta$  to 0. After the models have converged, we plot the rigidness coefficient matrices in Fig. 3. We observe that the matrices for the scaling factor  $\alpha$  are symmetric, while the matrices for the bias factor  $\beta$  are skew-symmetric. Note that only partial elements deviate from the initialization values because the connections between parts are sparse.





Fig. 3: The learned rigidness coefficient matrices. The first row corresponds to the scaling factor  $\alpha$ , and the second row contains the bias factor  $\beta$ .

# 5 Illustration of Adjacent Part Seaming

To illustrate the effect of adjacent part seaming (APS), we present a zoom-in example in Fig. 4. When bone  $b_1$  and bone  $b_2$  undergo a relative rotation, the part on bone  $b_1$  deforms so that its section across the joint can still align with the section of bone  $b_2$ . Furthermore, we compare the results under novel poses with and without APS in Fig. 5. Disabling APS leaves the model almost rigid, resulting in artifacts like cracks at elbows and knees.



Fig. 4: Non-rigid deformation produced by adjacent part seaming. When the bones  $b_1$  and  $b_2$  undergo a relative rotation, our adjacent part seaming algorithm guarantees the alignment of their sections. We disconnect adjacent parts to better visualize the sections and to confirm that parts are not overlapped.

UNIF 5



Fig. 5: Results of our method under novel poses with (green boxes) and without (red boxes) adjacent part seaming.

# Bibliography

- Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2565–2574 (2020)
- [2] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
- [3] Lipman, Y.: Phase transitions, distance functions, and implicit neural representations. arXiv preprint arXiv:2106.07689 (2021)
- [4] 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)
- [5] Salimans, T., Kingma, D.P.: Weight normalization: A simple reparameterization to accelerate training of deep neural networks. Advances in neural information processing systems 29 (2016)
- [6] Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. NeurIPS (2020)
- [7] Zheng, H., Yang, Z., Liu, W., Liang, J., Li, Y.: Improving deep neural networks using softplus units. In: 2015 International Joint Conference on Neural Networks (IJCNN). pp. 1–4. IEEE (2015)