Supplementary Material for
Unsigned Orthogonal Distance Fields (UODFs): An Accurate Neural Implicit Representation for Diverse 3D Shapes

A. Detailed Configurations

A.1. Training Details in Our method

All reconstruction results shown in this manuscript are free of fine-tuning parameters in the network and the post-processing method, thus are easy to reproduce. Each shape is normalized with a radius of 0.9. In our training, the UODFs for each model are trained for 100 epochs. The method for point sampling during training is detailed in Sec. 3.4 in the main manuscript. We adopt a batch size of 1024 points during training. The optimization is performed using the Adam optimizer, starting with an initial learning rate of 0.001, which is halved every 20 epochs.

A.2. Post-processing Details in Our method

In this section, we detail the post-processing meshing process for UODFs. Our process begins with the employment of the VCGlib library [1], a comprehensive toolbox also utilized by MeshLab [12], to compute normal for the reconstructed grid edge points (GEP). The number of neighboring point clouds required for merging when calculating normal is set to the default value of 10. Subsequently, we use screened Poisson surface reconstruction (SPSR) [7] for meshing the point clouds. Within SPSR, we adhere to several default parameters: the maximum depth of the octree used for reconstruction is set at 8, the minimum number of samples within each octree node is 1.5, and the interpolation weight between point clouds is 4. It is worth emphasizing that all models presented in our paper strictly follow these default VCGlib and SPSR settings, with no additional modifications post-reconstruction. This consistent use of default parameters ensures a standardized and reproducible approach across all our models.

A.3. Configurations of Other Methods

SIREN [10]. We configure SIREN following the original paper. Notably, in our experimental setup, we deliberately abstain from utilizing the normal information of SIREN for supervision. This exclusion aims to maintain uniform supervision conditions across all methods. Furthermore, our loss setup incorporates the Eikonal regularizer, denoted as $\nabla|f| = 1$. This inclusion is specifically designed to align with our $\mathcal{L}_{der}$ term, thereby maintaining fairness and consistency in the comparative analysis of different methods.

NGLOG [11]. The official implementation of NGLOD in accordance with the original paper is adopted. To attain a more precise reconstruction model, we opt for a dense voxel version of NGLOD. This variation permits NGLOD to encompass global SDF details, resulting in a more complete surface reconstruction, thereby facilitating a fairer comparison among different reconstruction methods.

NDF [3]. While the original NDF is designed to learn and reconstruct a class of objects, including unseen ones, modifications to NDF are needed to align with single object reconstruction. To this end, we replicate 20 instances of a single training model as input. Since the input point clouds of each model consist of 50k points near the surface, our configuration yields approximately 1 million sample training points, ensuring a similar scale to the sample inputs used for UODFs. Moreover, to make a fair reconstruction comparison with UODFs, the point cloud estimated from NDF prediction also undergoes SPSR with masking. This revision is non-trivial as it bypasses the use of BPA [2], whose parameter configuration is tailored to different models.

GIFS [14]. Similar to NDF's adjustment, GIFS is adapted for single object reconstruction. The training process involves replicating a single model 20 times to form a class of identical objects. This approach generates 1 million learning points per object. For mesh reconstruction, we employ GIFS' default mesh post-processing extraction algorithm, ensuring consistency with the original paper.

HSDF [13]. HSDF also undergoes modifications to focus on learning and reconstructing a single model. Similar to NDF and GIFS, the sampling entails repeating a single model 20 times, resulting in a mesh model extraction using the masked marching cube (MC) algorithm proposed by HSDF.

A.4. Parameter Amount

Table A.1 presents a comprehensive comparison of the total number of parameters utilized in various methods. In our method, we employ a configuration of three $10{*}256$ and $3{*}256$ MLPs to separately regress UODF in each orthogonal direction. Despite this seemingly large network setup, our parameter amount (1.84M) is notably smaller than those in three UDF regression methods and the state-of-the-art SDF regression method NGLOD5.

Table A.1. Comparison of parameter amount.

Method	SDF			UDF			UODFs
	SIREN [10]	NGLOD3 [11]	NGLOD5 [11]	NDF [3]	HSDF [13]	GIFS [14]	**Ours**
Paras. (M)	0.199	0.199	10.1	4.62	6.60	3.68	1.84

B. Unstable Cases of SDF Sign

SDF presents unique challenges in determining the sign at each sampling position, especially for complex shapes, unlike UDF and UODFs that only require specific distance values. This section discusses the instability issues associated with SDF computation using the following two categories of approaches.

- *First Category:* Implemented in the *libigl* library [6], used by NI [4] and HSDF [13]. *libigl* employs the generalized winding number to determine the inside or outside positioning of points relative to a given mesh.
- *Second Category:* Adopted in NGLOD [11] and Instant-NGP [9]. NGLOD initially computes the unsigned distance, then determines the sign through ray stabbing in 13 pre-defined directions. Instant-NGP follows a similar approach but uses BVH acceleration and ray stabbing with 32 uniformly distributed directions.

Figure B.1 illustrates the marching cubes (MC) reconstruction results based on the ground truth SDF computed by NI and NGLOD, highlighting noticeable artifacts and detail errors. In contrast, UODFs do not exhibit these issues in ground truth computation, as evidenced by the comparative reconstruction metrics presented in Table B.2. The table reveals that MC reconstructions based on ground truth SDF are significantly less accurate and NGLOD tends to diminish high-frequency details in its regression to the ground truth SDF.

Table B.2. Comparison of reconstruction for the birdcage.

Metric \ Method	NGLOD3	NGLOD5	SDF GT	UODFs **Ours**
CD-GEP(*10^5) ↓	9.91	23.7	32.7	**1.62**
CD-Mesh(*10^5) ↓	14.0	27.7	39.2	**5.90**
NC-Mesh ↑	95.9	94.5	93.1	**97.7**

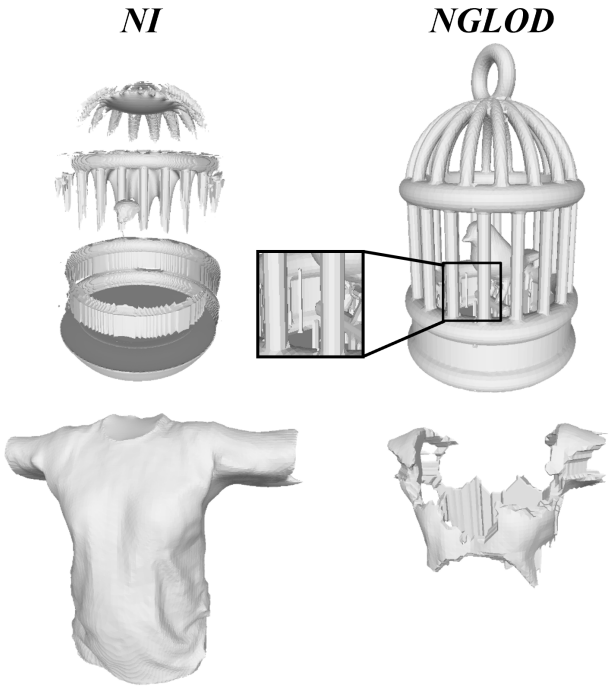

Figure B.1. Reconstructions of the watertight birdcage and the non-watertight garment for two choices of SDF computation. Due to the unstable SDF sign, most of the shape surface may not be reconstructed.

C. Additional Details of UODFs Fitting and Reconstruction

C.1. 2D UODF Results

In this section, we present the additional details of UODFs fitting and reconstruction using two distinct instances: the closed 'Rabbit' from the Thingi32 dataset and the 'Dragon' from the Stanford Scanning dataset.

Rabbit Instance:

- The fitting and reconstruction results at 257^3 grid corners are displayed in Fig. C.2. The first row depicts the predicted masks on three orthogonal planes, accompanied by the number of rays (NoR) in each mask.
- The subsequent row details the maximum errors in fitted UODF values along each orthogonal direction, along with the number of outliers (defined as points deviating more than 5 grids from the shape surface).
- A significant reduction of outliers is observed from the third row, due to averaging operations in estimating GEP, as discussed in Sec. 3.5. Moreover, the fusion operation results in only 4 final outliers, illustrating that most outliers in one orthogonal direction are isolated.
- Fine-tuning efforts can be seen in the last two rows, involving over-sampling rays with high prediction errors and additional training epochs. This leads to a noticeable decrease in outliers of the fitted UODF values, the estimated GEP, and the fused GEP.

The right part of Fig. C.2 illustrates the results for the 'Dragon' instance, being more complex. Consequently, the second to fifth rows show higher numbers of rays with outliers. Similarly, the number of final outliers in fused points reduced from 32 to 13 after fine-tuning, affirming the effectiveness of our approach. As the complement of Fig. 3 in the main manuscript, Fig. C.3 shows a slice of UODF in the other two orthogonal directions for the 'Dragon' model. There is a subtle difference between ground truth and our prediction, highlighted by black dotted circles.

These instances exemplify the effectiveness of our UODFs fitting and reconstruction process. **We emphasize that the experimental results in our manuscript and video are free of fine-tuning, although this can further improve the visual effect and accuracy metrics.**

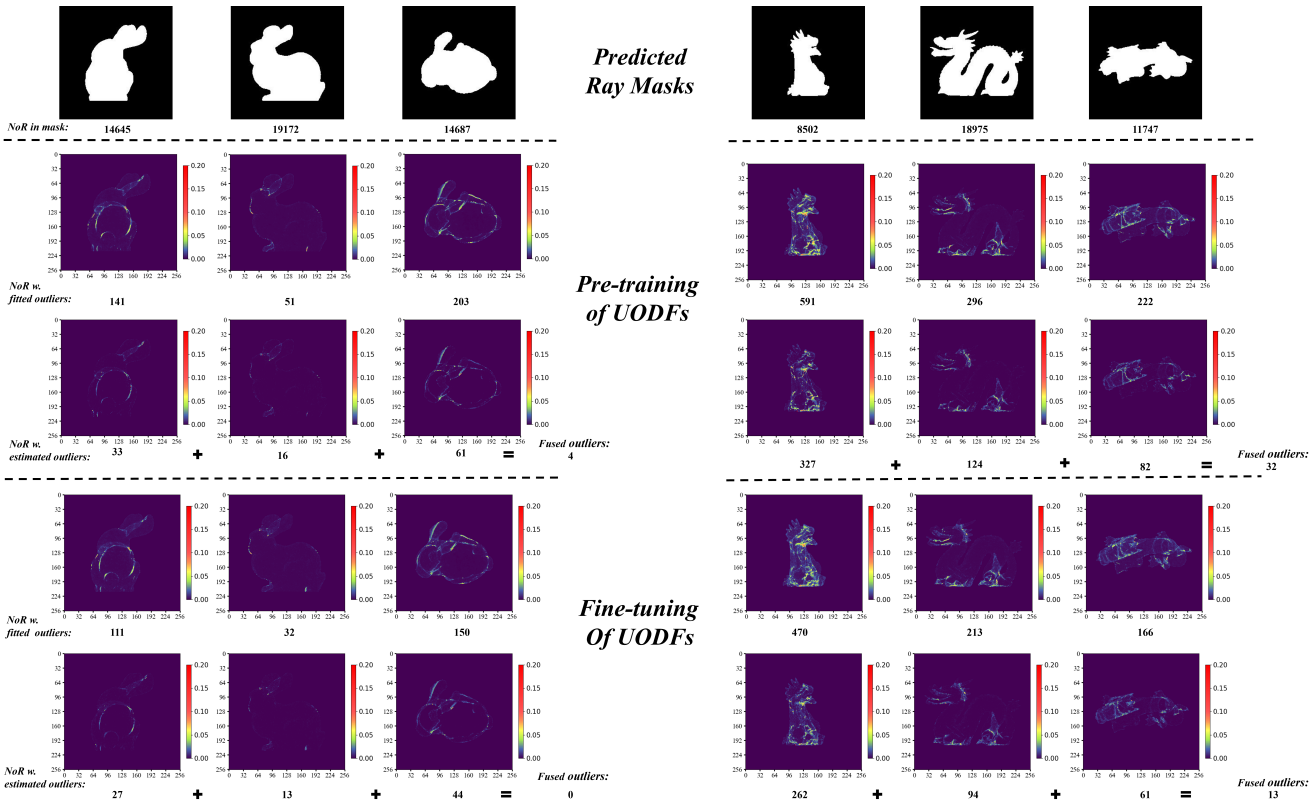

Figure C.2. Detailed results of UODF fitting and reconstruction of grid edge points for 'Rabbit' and 'Dragon'.

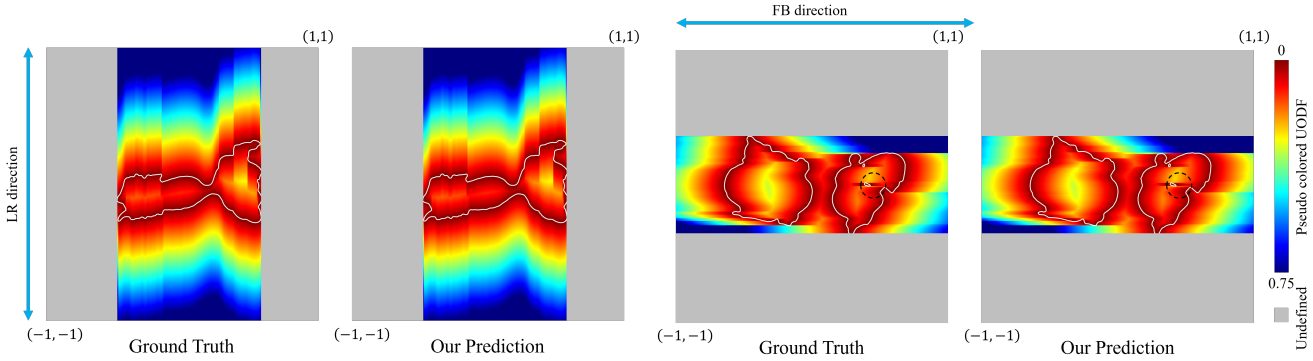

Figure C.3. One slice of the $UODF_{LR}$ and $UODF_{FB}$ of the 'Dragon' model. A subtle difference between ground truth and our prediction can be seen in the areas drawn by black dotted circles in the $UODF_{FB}$ (with zoomed-in views).

D. Ablation of Point Merging Threshold

This section presents an ablation study on the point merging threshold τ, initially introduced in Sec. 3.5. While the threshold τ is set to $1/512$ (a quarter of the grid size at a 256^3 resolution) in our primary experiments, we explore its impact on reconstruction accuracy by adjusting its value. τ is additionally set to $1/256$ (half of the grid size) and $1/128$ (equal to the grid size). Along with the 'Nandi the Bull' and 'Hilbert Cube' from the main manuscript, we include the 'Lamp' from ShapeNet, the 'Birdcage' from Thingi10K, and the 'Armadillo' from the Stanford dataset.

Table D.3 shows the three metrics across five shapes. It indicates that the reconstruction accuracy remains largely unaffected by variations in the point merging threshold. Even at a threshold of $1/128$, the accuracy metrics show minimal sensitivity to this parameter. This finding suggests a robustness in our method's performance relative to the point merging threshold.

Table D.3. Ablation study of the point merging threshold τ. The metrics in each cell are CD-GEP($*10^5$) ↓, CD-Mesh($*10^5$) ↓, and NC-Mesh ↑, respectively.

Shape / τ	'Nandi the Bull'	'Hilbert Cube'	Lamp	Birdcage	'Armadillo'
1/512	0.228 / 2.77 / 98.5	0.00532/12.0/92.1	0.0399/2.43/98.8	1.62/5.90/97.7	0.684/2.37/97.8
1/256	0.217 / 2.77 / 98.5	0.00523/12.0/92.0	0.0398/2.43/98.7	1.66/5.93/97.6	0.726/2.39/97.8
1/128	0.210 / 2.76 / 98.5	0.00523/12.0/92.0	0.0405/2.44/98.8	1.69/6.02/97.7	0.707/2.39/97.8

E. Additional Reconstruction Results

This section presents additional reconstruction results not included in the main manuscript.

First, we elaborate on the three reconstruction metrics for the ten non-watertight garments from the MGN dataset in Table E.4. These results show that our UODFs based NIR method significantly outperforms three other UDF regression methods. We have already showcased the 'TshirtNoCoat-1' garment in Fig. 9 in the main manuscript and the results for the remaining nine garments are illustrated in Fig. E.4.

Second, for the watertight shapes, additional results on the Thingi32 dataset are displayed in Fig. E.5, demonstrating consistency with the findings shown in Fig. 6 and Table 1 in the main manuscript.

Third, we present the outcomes for the additional four complex shapes in Fig. E.6. The zoomed-in views further underscore the high-quality shape reconstruction achievable with our proposed UODFs.

Finally, more reconstruction experiments at various grid resolutions from 32^3 to 256^3 are conducted. Fig. E.7 shows the results of four distinct shapes: two watertight and two non-watertight. To highlight the reconstruction accuracy of our UODF-based NIR method, we compare it with the marching cube (MC) algorithm [8] and MeshUDF [5], that utilize ground truth SDF and UDF as input, respectively. Therefore, the MC and MeshUDF results do not suffer from fitting errors of neural network, but are only affected by interpolation errors. For these two watertight models, our method achieves performance comparable to the MC method across all grid resolutions. In the case of non-watertight models, our method significantly outperforms MeshUDF using the ground truth UDF as input, at all grid resolutions. MeshUDF has some spurious meshes at the edge of shapes, especially under lower resolutions. In addition, our meshes generated by SPSR [7] are smoother than those extracted by MC and MeshUDF.

References

[1] Visualization and computer graphics library. https://vcg.isti.cnr.it/vcglib/. 1

[2] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics (TVCG)*, 5(4):349–359, 1999. 1

[3] J. Chibane, A. Mir, and G. Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Neural Information Processing Systems (NeurIPS)*, 2020. 1, 5

[4] T. Davies, D. Nowrouzezahrai, and A. Jacobson. On the effectiveness of weight-encoded neural implicit 3d shapes. *arXiv preprint arXiv:2009.09808*, 2020. 2

[5] B. Guillard, F. Stella, and P. Fua. Meshudf: Fast and differentiable meshing of unsigned distance field networks. In *European Conference on Computer Vision (ECCV)*, 2022. 4

[6] A. Jacobson, D. Panozzo, et al. libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 2

[7] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):1–13, Jul. 2013. 1, 4

[8] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987. 4

[9] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4):102:1–102:15, Jul. 2022. 2

[10] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Neural Information Processing Systems (NeurIPS)*, 2020. 1

[11] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[12] M. Tarini, N. Pietroni, P. Cignoni, D. Panozzo, and E. Puppo. Practical quad mesh simplification. *CGF*, 29(2), 407–418, 2010. 1

[13] L. Wang, J. Yang, W. Chen, X. Meng, B. Yang, J. Li, and L. Gao. Hsdf: Hybrid sign and distance field for modeling surfaces with arbitrary topologies. In *Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2, 5

[14] J. Ye, Y. Chen, N. Wang, and X. Wang. Gifs: Neural implicit function for general shape representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 5

Table E.4. Detailed reconstruction metrics for each of ten garments from the MGN dataset. The two CD metrics ↓ are multiplied by a factor of 10^5.

Method	UDF regression									UODFs regression		
	NDF [3]			HSDF [13]			GIFS [14]			**Ours**		
Metric / Shape	CD-GEP	CD-Mesh	NC-Mesh	CD-GEP	CD-Mesh	NC-Mesh	CD-GEP	CD-Mesh	NC-Mesh	CD-GEP	CD-Mesh	NC-Mesh
TshirtNoCoat-1	146	145	94.4	8.52	10.1	90.0	2.13	3.32	97.7	**0.212**	**1.87**	**99.8**
ShortPants-1	154	155	93.6	13.5	17.7	93.4	2.04	5.05	97.4	**0.162**	**3.60**	**99.6**
Pants-1	81.7	82.0	94.0	14.4	15.8	90.1	2.0	3.49	95.4	**0.284**	**2.04**	**99.6**
Pants-2	148	145	91.5	17.9	19.4	88.3	3.78	5.07	90.9	**0.182**	**2.05**	**99.7**
ShirtNoCoat-1	34.9	34.8	93.7	12.0	11.8	81.7	16.5	16.1	84.6	**0.311**	**0.929**	**99.7**
TshirtNoCoat-2	158	156	91.5	7.98	9.40	91.0	2.12	3.35	95.6	**0.156**	**1.86**	**99.8**
ShortPants-2	245	248	93.4	13.2	17.2	91.6	2.46	5.21	97.3	**0.143**	**3.43**	**99.7**
ShirtNoCoat-2	107	104	77.2	13.6	13.1	85.2	8.24	8.0	86.7	**0.183**	**0.777**	**99.7**
LongCoat-1	135	141	88.1	10.4	11.4	88.4	6.62	7.24	86.0	**0.368**	**1.69**	**98.9**
LongCoat-2	66.9	65.0	90.7	18.0	18.3	80.5	3.61	3.88	88.5	**0.264**	**1.07**	**99.7**

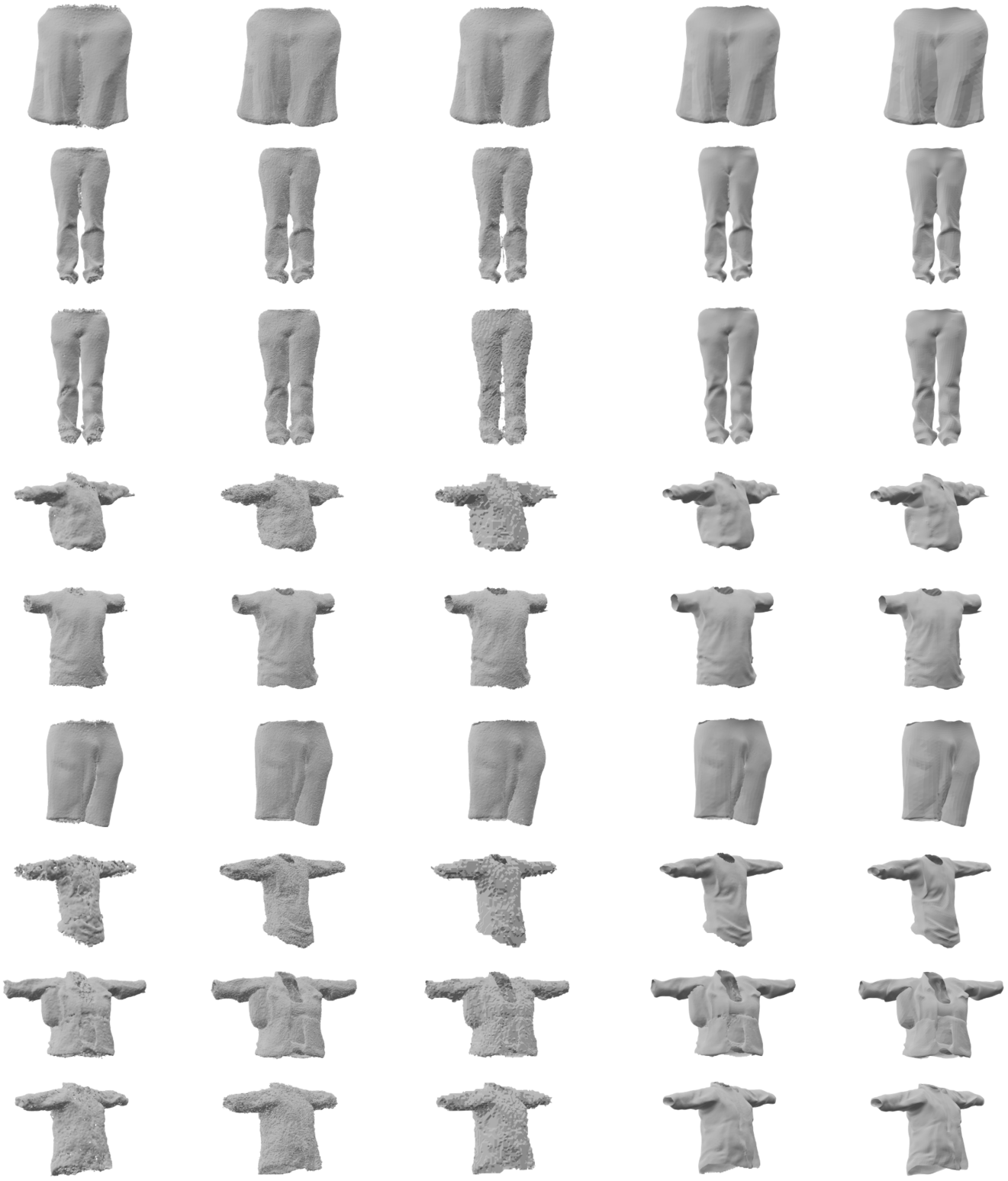

Figure E.4. Reconstruction results for other nine shapes in the MGN10 dataset.

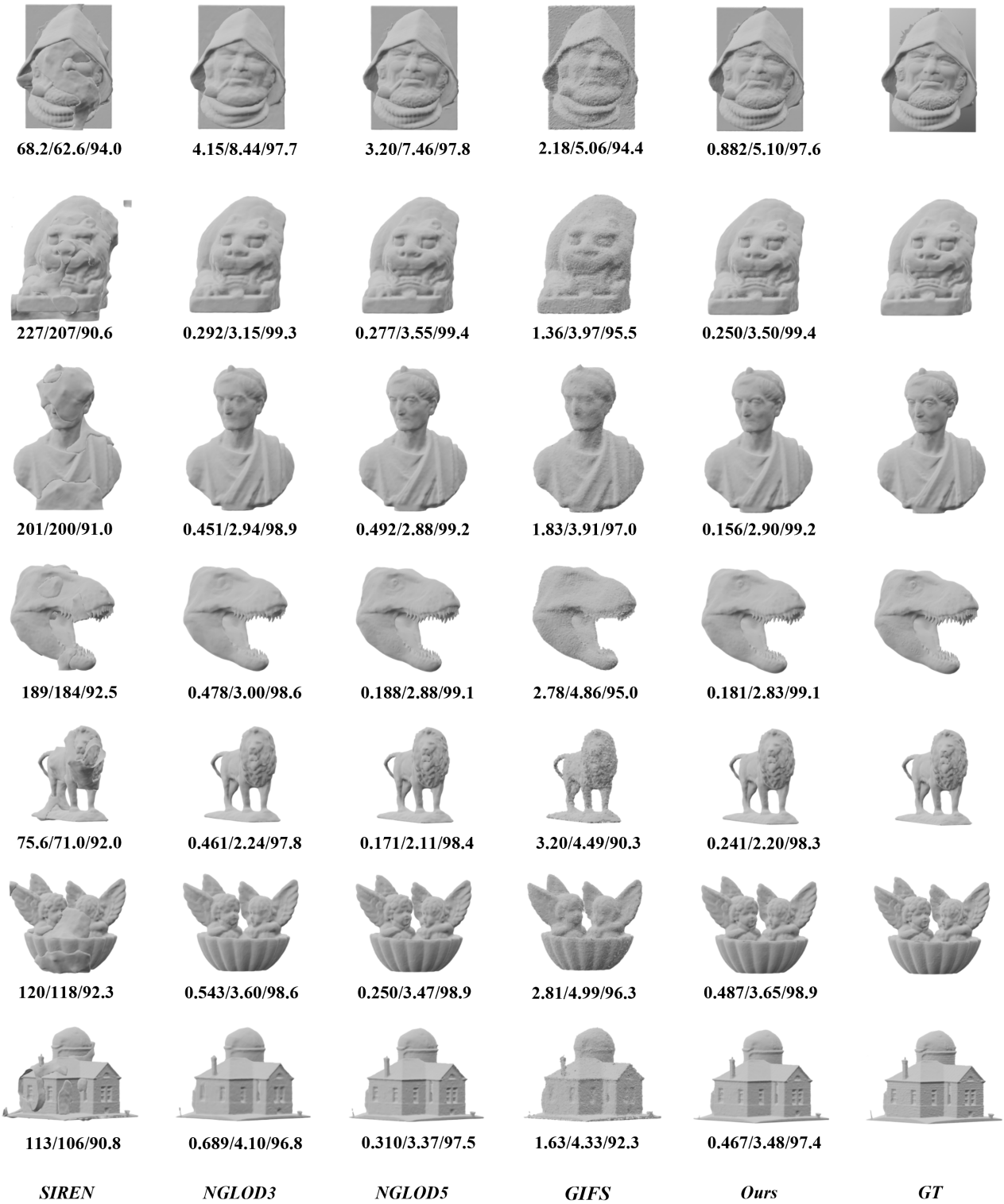

Figure E.5. Reconstruction results for more shapes in the Thingi32 dataset. The metrics from left to right below each shape are CD-GEP(*10^5) ↓, CD-Mesh(*10^5) ↓, and NC-Mesh ↑, respectively.

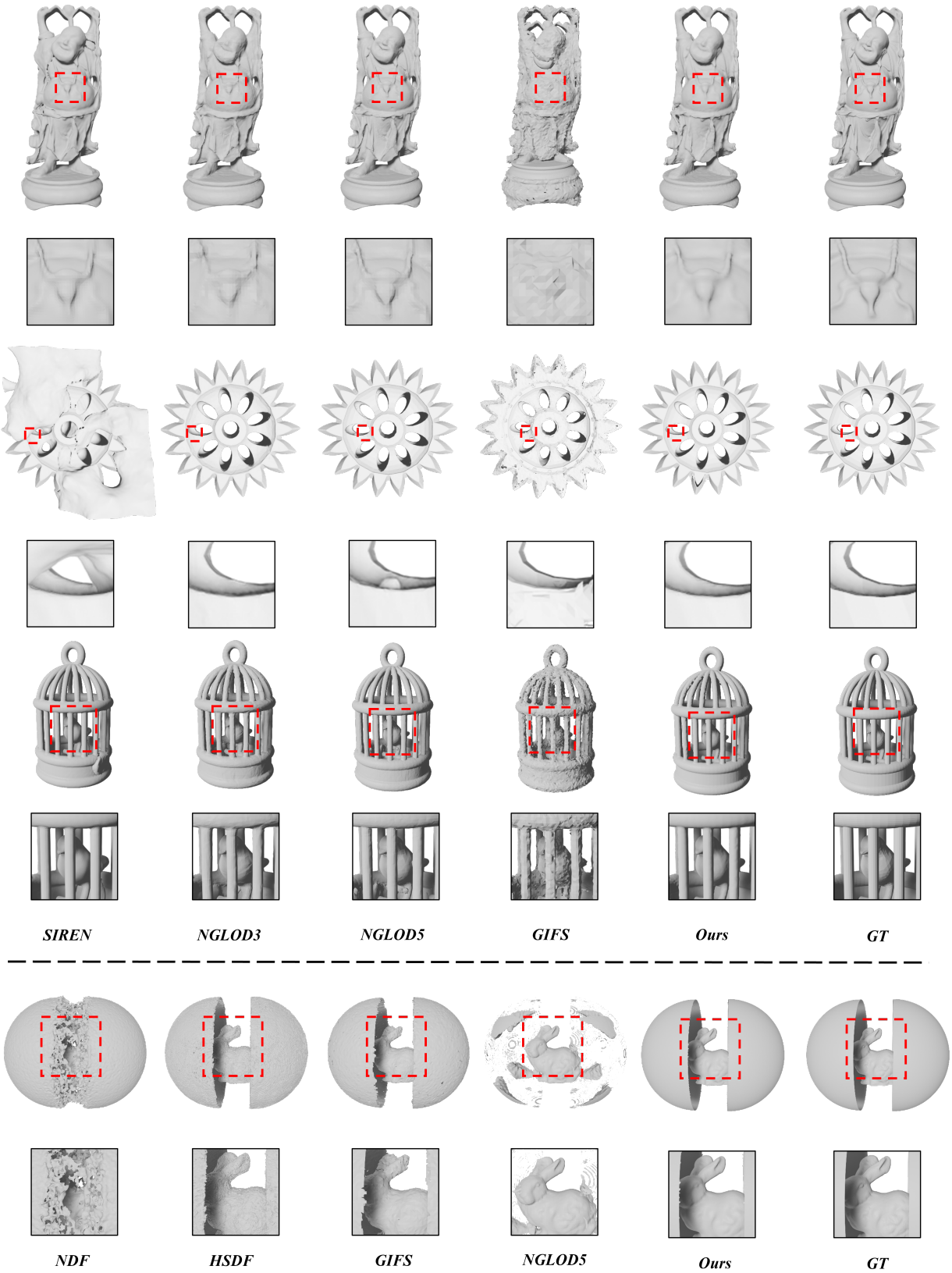

Figure E.6. Reconstruction results for additional four complex shapes.

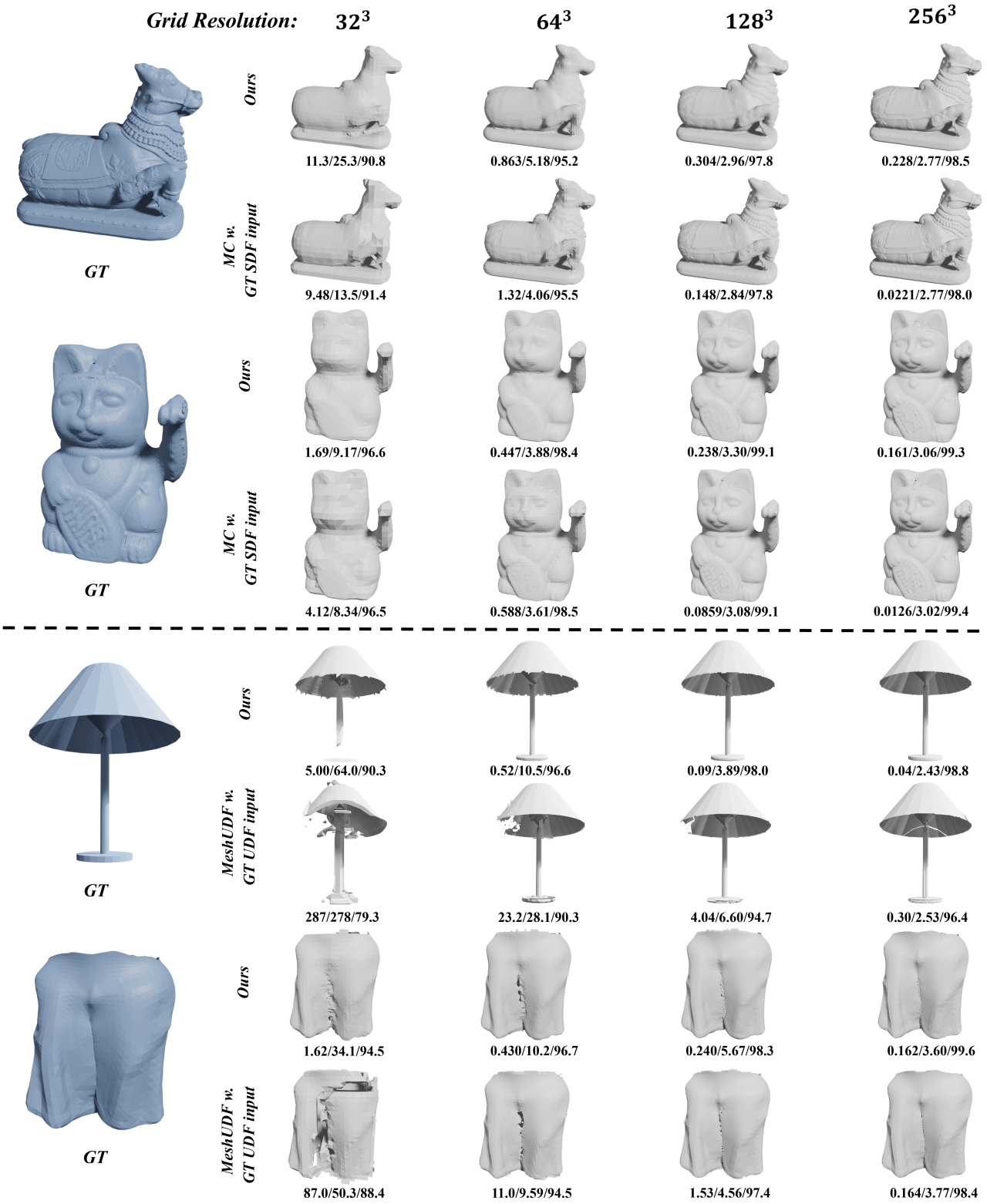

Figure E.7. Additional reconstruction results at various resolutions of grids. The metrics from left to right below each shape are CD-GEP($*10^5$) ↓, CD-Mesh($*10^5$) ↓, and NC-Mesh ↑, respectively.