Supplementary Material
Surface Extraction from Neural Unsigned Distance Fields

1. Error characteristics of MLP-encoded UDF in 3D

To supplement the discussions of Sec. 3 in the main text, in this section, we provide more information about the error characteristics of a neural UDF in the 3D case, which is most relevant to the setting of our algorithm.

Fig. 1(b) shows the GT UDF of the Fandisk model at a cross-section (Fig. 1(a)). Fig. 1(c) and (d) show the close-up views of the neural UDF values and the GT UDF values of the Fandisk model around a flat region, and Fig. 1(e) and (f) show another comparison between the neural and GT UDF values around a sharp edge. Fig. 1(g)-(j) present the color-coded error maps of the neural UDF values ((g) and (h)) and of the normalized gradient directions of the neural UDF ((i) and (j)), for the same two local regions. The GT UDF values (blue curves) and the neural UDF values (orange curves) are plotted regarding the same flat region in Fig. 1(k), regarding the same sharp edge in Fig. 1(l), and regarding the entire surface in Fig. 1(m). Histograms are plotted to show the overall distributions of the neural UDF errors in Fig. 1(n) and its gradient vector errors in Fig. 1(o), respectively, against the GT UDF values (the horizontal axes). Finally, in Fig. 1(p) and (q), we visualize the distributions against neural UDF values being the horizontal axes.

The observations about the error characteristics of the neural UDF made in this 3D case align well with the 2D case provided in the main text, which are:

(1) The errors of the neural UDF are concentrated around the target surface, *i.e.*, the zero-level set of the ideal UDF;

(2) The errors around the sharp edges are more pronounced than those around a flat region; and

(3) The errors of the neural UDF are too large to be reliable within the distance of 0.002 from the surface around a flat region and within the distance of 0.005 from the surface around a sharp edge. [1]

In our algorithm, we choose the sampling threshold to be 0.002, based on the overall error characteristics as reflected in the histograms in Fig. 1(p) and (q). That is no sampling points are selected with their neural UDF values less than 0.002. While the qualitative observations (1) and (2) hold generally for all kinds of neural UDFs, the numerical characterization in observation (3) is specific to the particular MLP architecture and the training strategy used in our experimental setup. Different MLP architectures or training strategies might lead to more or less accurate neural UDFs, as we will show later in this Supplementary Materials.

2. MLP-encoded UDF training details

We will introduce the loss function, sampling strategy, and more detailed training configuration in this section.

The overfitting network that we used in the main text is a 9-layer MLP with each layer having 512 neurons. The activation functions are the Sine activation, except for the last one which is a SoftPlus ($\beta = 100$) activation to ensure output values are non-negative.

Loss function. The loss function to train an overfitting network is presented as follows:

$$\mathcal{L} = \int_{\Omega} |f_\theta(\mathbf{x}) - \mathrm{UDF}(\mathbf{x})| d\mathbf{x} \tag{1}$$

where $f_\theta(\mathbf{x})$ and $\mathrm{UDF}(\mathbf{x})$ denote the neural and GT unsigned distance values at point $\mathbf{x}$, respectively.

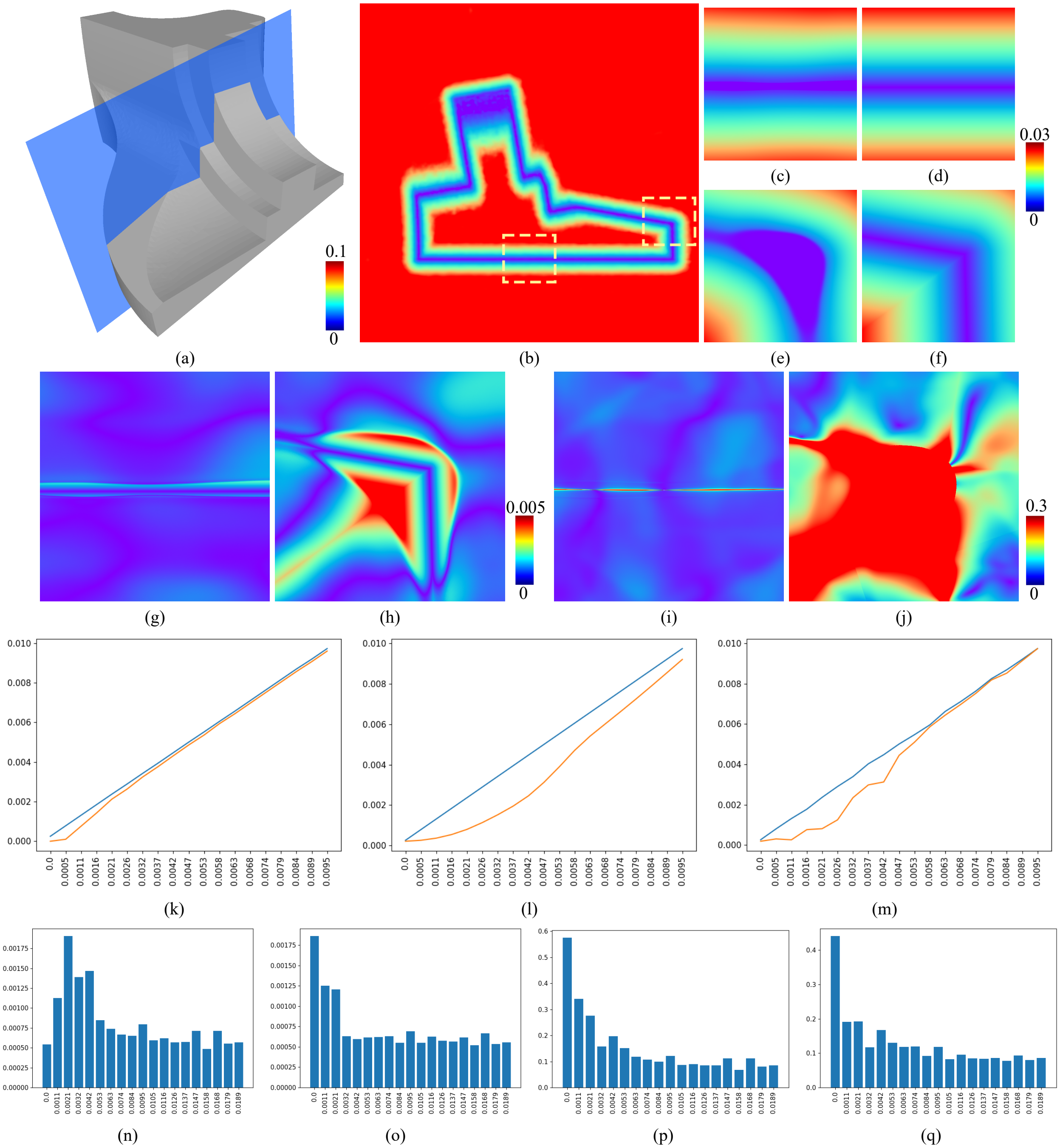
Sampling strategy. For each shape, we normalize the shape so that it is centered at the origin and bounded in a domain $\Omega = [-1, 1]^3$. The training samples of each shape consist of 600k points uniformly sampled on the shape surface, 1200k points within the distance of 0.05, 800k points within the distance of 0.3 following Gaussian distribution, and 400k points uniformly sampled in the bounding domain Ω.

Training configuration. The mini-batch size is set to 30k samples. The initial learning rate is set to 1×10^{-4} and is decayed by 0.3 after 1500 and 2300 iterations.

3. Supplementary experiments

We present additional experiments and discussions on 1) visualizing the mesh extraction results of different methods

[1] All distance thresholds are given based on shapes normalized to a bounding region of $[-1, 1]^3$.



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j)

(k) (l) (m)

(n) (o) (p) (q)

Figure 1: The errors of a neural UDF on *Fandisk*. **(a)** A plane cutting across the Fandisk model; **(b)** The color map of an MLP-encoded UDF of the Fandisk on the cross-section in (a); **(c)-(f)** The closeup views of the neural UDF ((c), (e)) and the corresponding GT UDF ((d), (f)) in the two framed local regions shown in (b), respectively; **(g)-(j)** The color maps of the errors of the neural UDF ((g), (h)) and the errors of the neural UDF's *normalized* gradient vector fields in the same two local regions ((i), (j)), respectively; **(k)-(m)** The plots of the averages of the GT UDF values (blue curve) and the neural UDF values (orange curve) in the same two local regions (around plane (k), around edge (l)), and around the overall surface (m), respectively, with the GT UDF value being the horizontal axess; **(n)-(o)** The histogram of the errors of the neural UDF (n) and the histogram the errors of normalized gradient vector fields (o), respectively, with *the GT UDF values* being the horizontal axes. **(p)-(q)** The histogram of the errors of the neural UDF and the histogram of the errors of normalized gradient vector fields, respectively, with *the neural UDF values* being the horizontal axes.

on GT UDFs; 2) performance on neural UDFs trained with another frequently used setting, *i.e*. using the positional encoding and replacing ReLU activation layers with Softplus; 3) our method under different grid resolutions; and 4) showcases of our reconstructed mesh quality.

3.1. Evaluation on ground-truth UDFs

We evaluate our method, MeshUDF [2], CAP-UDF [4], and NDC [1] on a set of ground truth UDFs that are directly generated by 3D meshes. In Fig. 2, we present four models for visual comparison. Our method preserves both sharpness and smoothness more faithfully when compared to the other three methods.

3.2. Generalization to MLPs trained with different settings

We have reported all quantitative comparisons in the main text using the Sine activation [3] for the hidden layers. In this supplementary material, we also tested our method on neural UDFs learned with other network settings and compared our results with those of the other three competing methods. Specifically, we follow the setting in MeshUDF [2] where the positional encoding is used before passing the coordinate queries into the network, and all activations are replaced with Softplus ($\beta = 100$). The hyperparameters of the proposed are unchanged during this experiment.

We compare our method with MeshUDF [2], CAP-UDF [4], and NDC [1] on our shape-overfitting neural UDF dataset including 354 shapes in total as we introduced in our main text. These comparisons show that our results are superior to all three methods in terms of approximation errors, visual smoothness, and preservation of sharp edges and smooth surface boundary curves. This validates the applicability of the proposed method to neural UDFs trained with other network settings.

Specifically, in Fig. 3, we visualize the distribution of approximation errors over the reconstructed meshes, measured by the distances from the GT mesh to the reconstructed mesh in each case. All the color-coded error maps are ranged in $[0, 0.0015]$ (from cool to warm) with errors larger than 0.0015 clamped at 0.0015. The results validate that our method produces smaller reconstruction errors overall, especially around the geometric features, than the other methods.

3.3. Experiment on different resolutions

We validate our method's capability at a wide range of resolutions. Since other competing methods are based on regular grids, evaluating them on finer resolutions will exceed our memory limit. In this experiment, we test our method under different octree max depths (i.e., different grid resolutions) on several complex shapes for GT UDF and our neural UDF datasets that we introduced in our main text, respectively. We show the visual comparison in Fig. 4 for GT UDFs and Fig. 5 for neural UDFs, respectively. It is recommended to zoom in for better viewing. In Table. 2, we report the quantitative results of 64^3, 128^3, 256^3, and 512^3 resolutions on the neural UDF datasets. We note that the edge length of a cell under the grid resolution of 512^3 is around 0.004, which is twice the threshold δ_1 of 2×10^{-3} set for sample filtering in Sec. 4, hence there may be insufficient samples (i.e., less than 3 valid sample points) within this cell after filtering. To address this issue, whenever a cell at this resolution level has less than 3 valid sample points, we lower δ_1 to 0.001 to collect enough samples to solve the QEF problem and estimate the surface point in this cell. As we mentioned in our main text, when cells become too small, the sample points within these cells will lie in the unreliable region of the neural UDF, which will eventually lead to unstable estimation.

3.4. Mesh quality demonstration

Our surface point positioning strategy and mesh surface generation method result in satisfactory mesh quality. The mesh connectivity of two of our results is visualized in Fig. 6.

References

[1] Zhiqin Chen, Andrea Tagliasacchi, Thomas Funkhouser, and Hao Zhang. Neural dual contouring. *ACM Trans. Graph.*, 41(4), jul 2022. 3, 4

[2] Benoît Guillard, Federico Stella, and Pascal Fua. Meshudf: Fast and differentiable meshing of unsigned distance field networks. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 576–592, Cham, 2022. Springer Nature Switzerland. 3, 4

[3] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7462–7473. Curran Associates, Inc., 2020. 3

[4] Junsheng Zhou, Baorui Ma, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Learning consistency-aware unsigned distance functions progressively from raw point clouds. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 3, 4

It is recommended to zoom in for better viewing

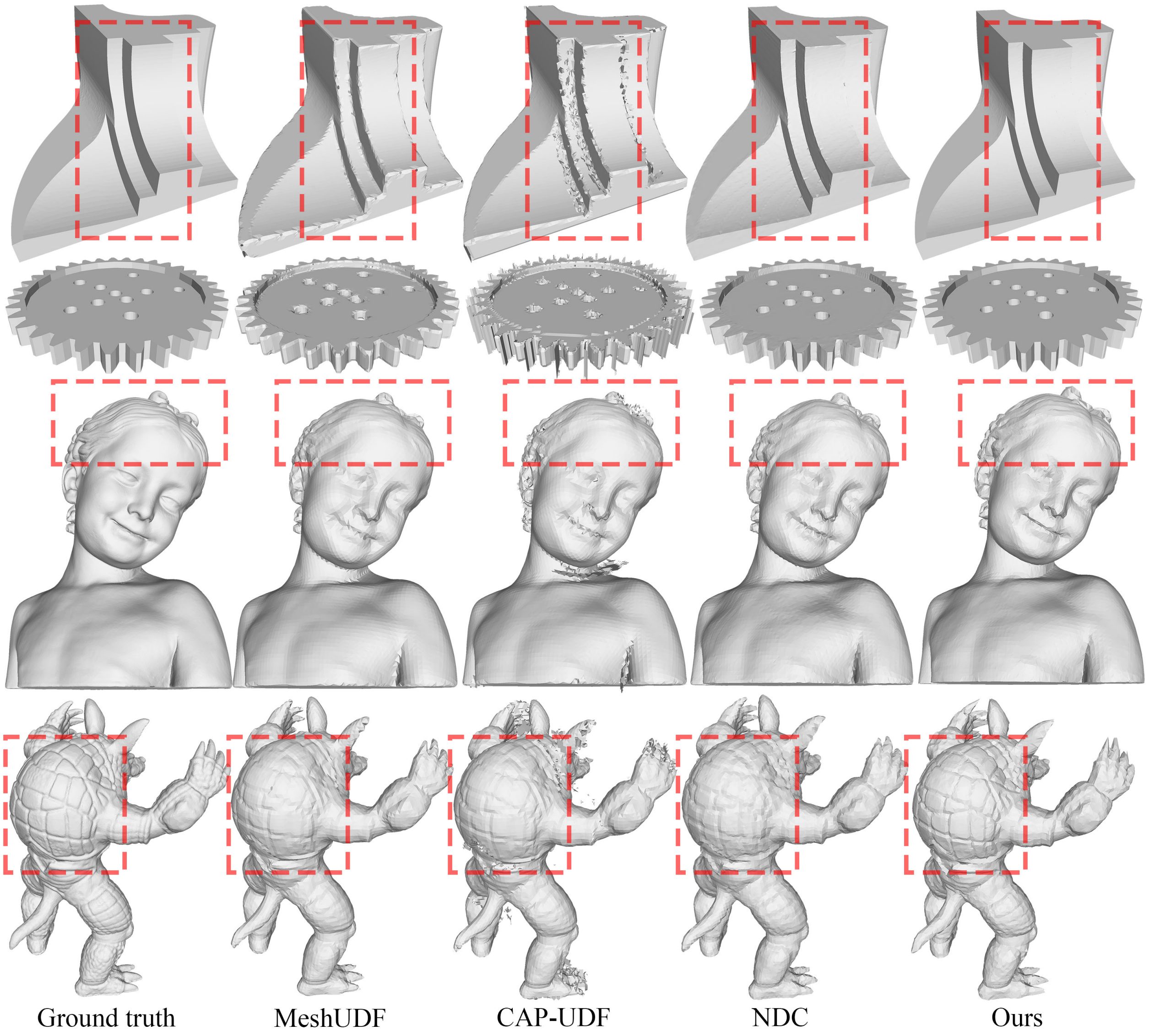


Figure 2: Meshes extracted from ground truth UDFs. We show four examples and compare our results to MeshUDF [2], CAP-UDF [4], and NDC [1]. Our method preserves sharp geometric features better (top two rows), and yields smoother results on organic models (bottom two rows).

Table 1: Quantitative comparison between the results obtained by our method and those by the competing methods, *i.e*. MeshUDF, CAP-UDF, and NDC. Different from Table 1 in the main text, this experiment follows MeshUDF [2] to use the positional encoding but replace all ReLU activations with SoftPlus activations. The average performance on the dataset containing 354 shapes is reported. The Chamfer distance (CD) and the Hausdorff Distance (HD) are scaled by 10^{-4} and 10^{-3}, respectively.

	MGN			Thingi10K			ABC		
	CD↓	F-score↑	HD ↓	CD↓	F-score↑	HD ↓	CD↓	F-score↑	HD ↓
Ours	**4.25**	**87.95**	**10.75**	**4.58**	**88.24**	**13.99**	**5.19**	**88.23**	**10.63**
MeshUDF [2]	11.31	54.10	17.99	10.46	58.66	17.65	13.47	68.63	17.12
CAP-UDF [4]	18.22	52.09	37.78	19.98	53.47	34.88	35.49	62.98	37.20
NDC [1]	6.26	73.24	11.09	6.92	72.54	16.77	7.45	80.29	15.04

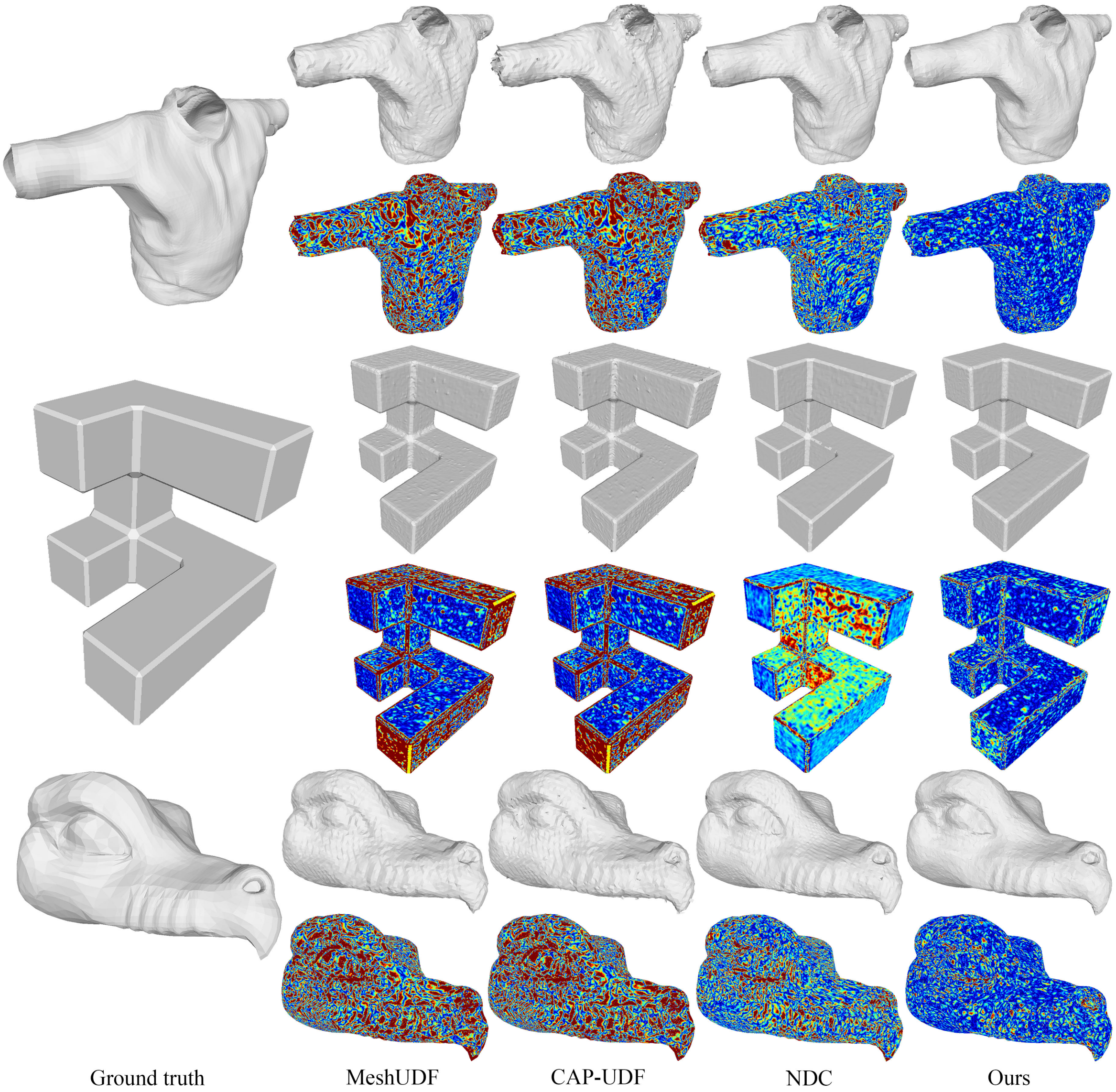


Figure 3: Comparisons of extracted meshes and the corresponding color-coded error maps obtained by our method and the competing methods. The warmer color indicates a larger error. Our method consistently outperforms the competing methods in all examples, achieving lower errors, preserving sharp features, and reproducing smooth surfaces and geometric details. MeshUDF and CAP-UDF cannot preserve the sharp features of the shapes (the 3rd row). The three competing methods find it difficult to cope with smooth transitions in the surface, yielding staircase artifacts.

Table 2: Quantitative results of 64^3, 128^3, 256^3, and 512^3 resolutions. The Chamfer distance (CD) and the Hausdorff Distance (HD) are scaled by 10^{-4} and 10^{-3}, respectively. T1 and T2 stand for the time spent (seconds) on the mesh extraction and that on the UDF query, respectively.

	MGN			Thingi10K			ABC			Running Time	
	CD↓	F-score↑	HD ↓	CD↓	F-score↑	HD ↓	CD↓	F-score↑	HD ↓	T1↓	T2↓
64^3	5.55	86.23	23.56	5.70	90.57	19.18	13.54	81.99	23.20	0.098	0.324
128^3	2.38	98.09	11.91	1.97	97.51	9.21	3.69	93.41	11.47	0.297	1.184
256^3	**2.03**	**98.96**	7.35	**1.60**	98.29	**6.78**	2.22	**96.57**	**8.20**	1.492	5.508
512^3	2.28	98.48	**7.01**	1.74	**98.88**	6.91	**2.00**	95.68	9.22	9.362	28.56

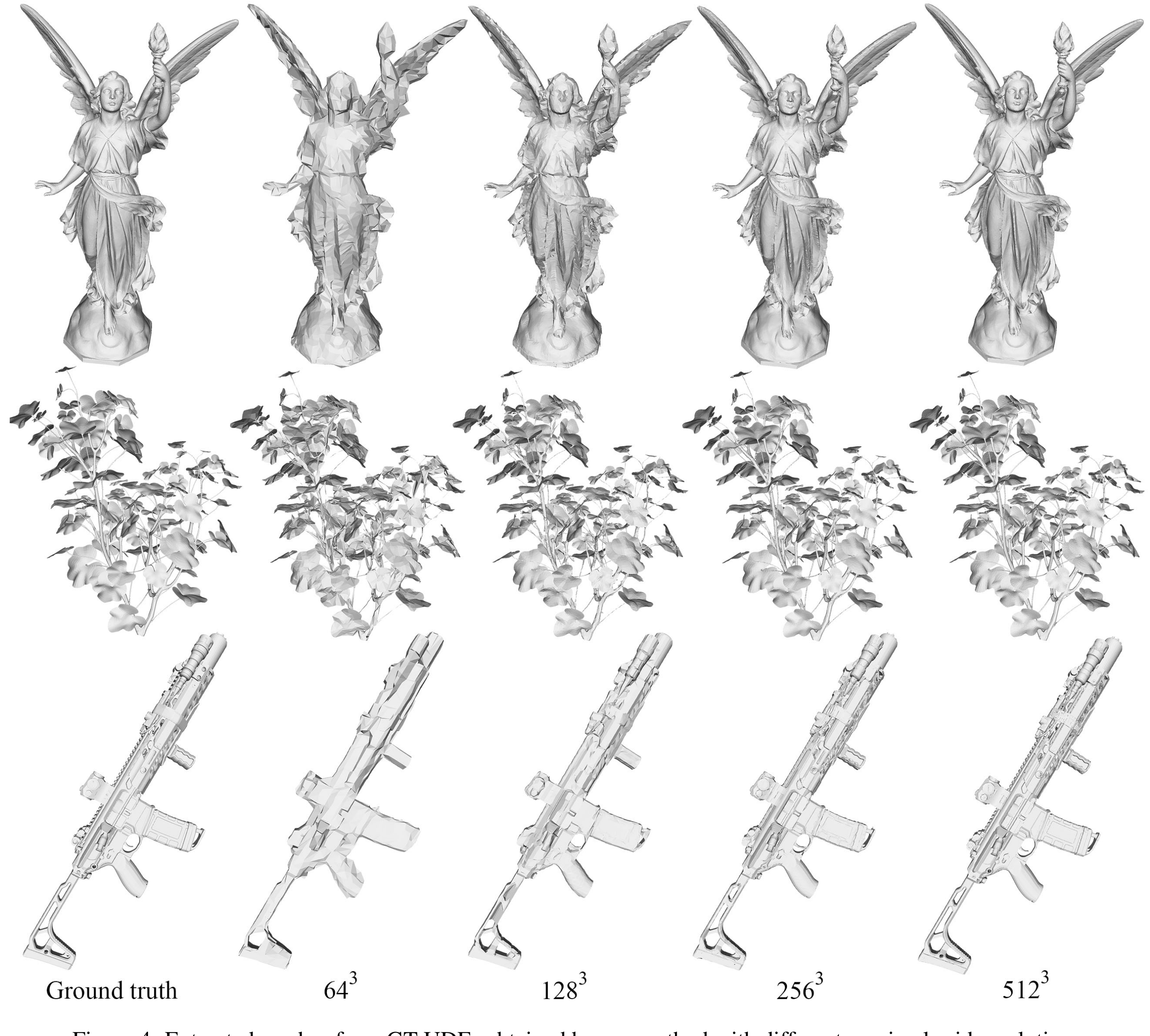


Figure 4: Extracted meshes from GT UDFs obtained by our method with different maximal grid resolutions.

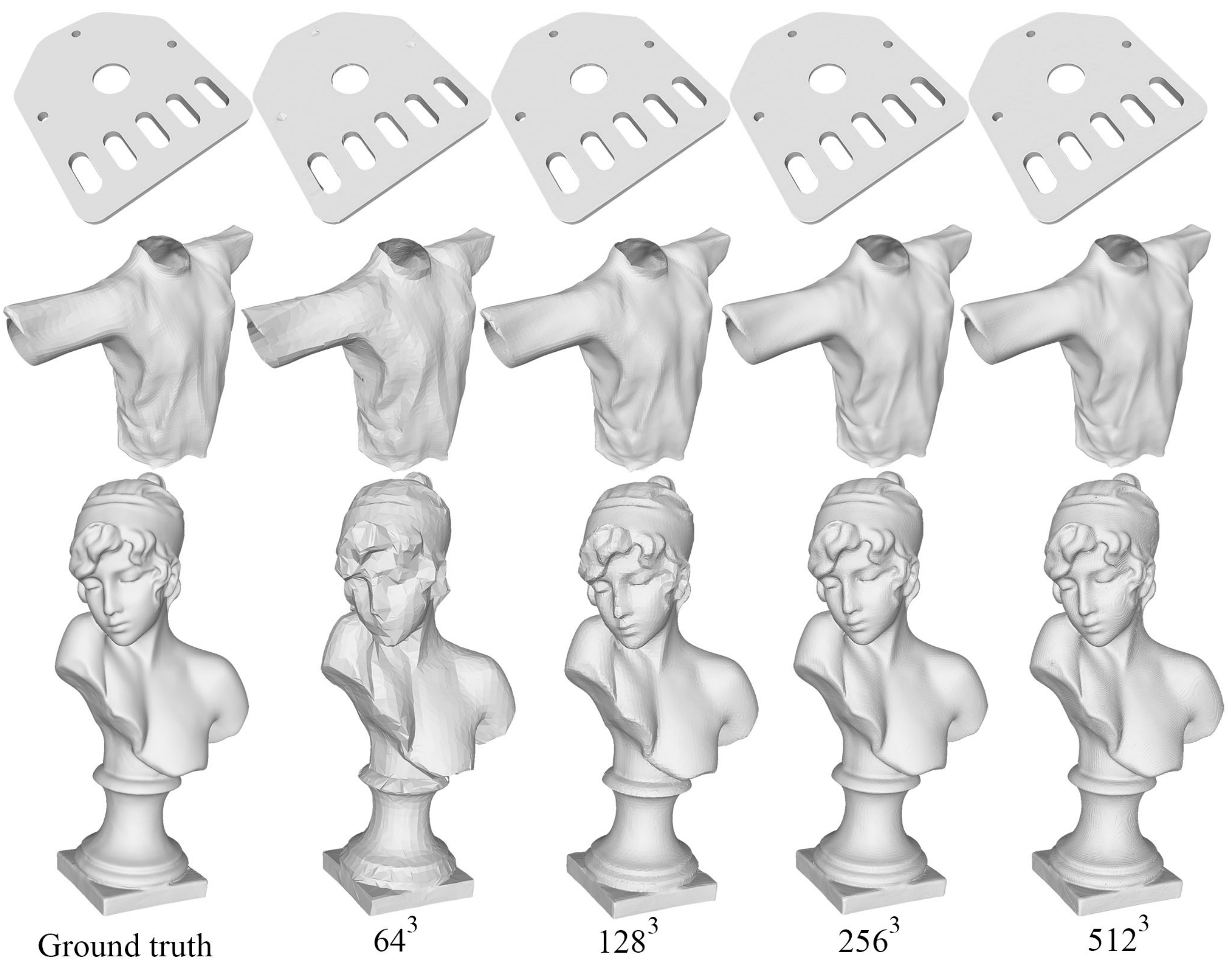


Figure 5: Extracted meshes from neural UDFs obtained by our method with different grid resolutions.

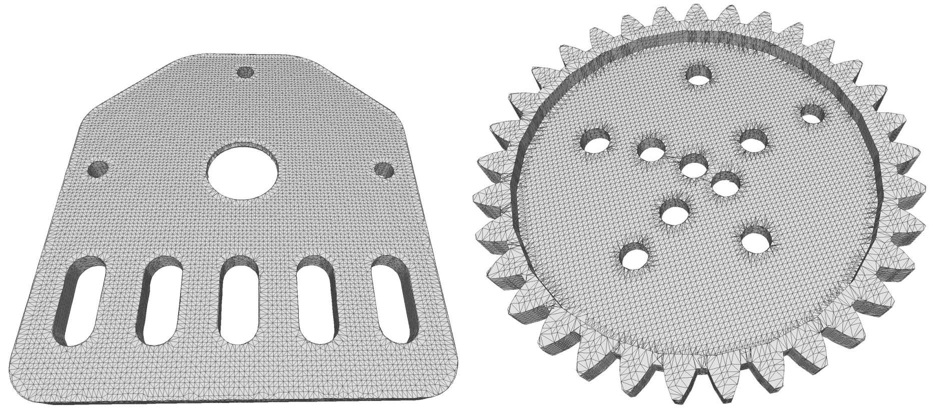


Figure 6: The visualization of the mesh quality of our reconstruction results.