GIFS: Neural Implicit Function for General Shape Representation Supplementary Materials

In this supplementary material, we provide additional details of the implementation and more visualization results.

A. Implementation Details

Padding Issue. In the original IF-Net implementation, the size of the 3D encoding grid is the same as the normalized mesh. In experiments, we find that the lack of padding is prone to generate artifacts on the boundaries, which significantly degrades the reconstruction accuracy. In our implementation, the size of the normalized mesh is 0.9 of the encoding grid. Moreover, in 3D convolution, we find that the zero padding outperforms the border padding used in IF-Net.

Training procedure. During training, the number of training pairs is 50000 per instance and the batch size is 8. We employ the Adam optimizer with a learning rate of 1×10^{-4} The watertight and the general shape experiments take 200 and 300 epochs respectively.

Mesh refinement. The initial mesh produced by our adapted Marching Cubes is further refined by minimizing the UDF values on the mesh surface. We employ an RMSprop optimizer with an initial learning rate of 2×10^{-4}. In each iteration, a random point is sampled on each face of the mesh. Given a trained GIFS model, we take the sampled points as input, query, and minimize their UDF values. The total number of iterations is 30.

B. Surface Extraction Algorithm

In this section, we provide the detailed surface extraction algorithm flow. Our algorithm consists of three steps: (i) Locate cubes that intersect the surface in a coarse-to-fine paradigm; (ii) Generate mesh triangles in final intersecting cubes with our adapted Marching Cubes; (iii) Refine mesh with the UDF branch.

First, we introduce our coarse-to-fine intersecting cubes localization algorithm. In our implementation, the initial resolution of the grid is 20^3 and is subdivided 3 times. The final resolution is 160^3. We show the detailed process in Algorithm 1. Among the inputs of the algorithm, the initial intersecting indices $\boldsymbol{I}$ are integer indices of all 20^3 cubes,

Algorithm 1 Locate intersecting cubes

Input: Initial intersecting indices $\boldsymbol{I} \in \mathbb{Z}_0^{+N\times 3}$, initial cube size s_0, point embedding layer $g_{\boldsymbol{\theta}_1}$, UDF layer $h_{\boldsymbol{\theta}_3}$, intersecting threshold τ, total number of stages T.
Output: Intersecting indices $\boldsymbol{I} \in \mathbb{Z}_0^{+M\times 3}$

- **for** stage $t \in range(T)$ **do**
  - Cube size $s \leftarrow s_0/2^t$
  - New empty intersecting indices $\boldsymbol{I}_n \leftarrow \{\}$
  - **for** intersecting index $\boldsymbol{i} \in \boldsymbol{I}$ **do**
    - Center of the intersecting cube $\boldsymbol{p}_c \leftarrow s\boldsymbol{i}$
    - Predicted UDF of the center $\boldsymbol{u} \leftarrow h_{\boldsymbol{\theta}_3}(g_{\boldsymbol{\theta}_1}(\boldsymbol{p}_c))$
    - **if** $\boldsymbol{u} < s\tau$ **then**
      - Subdivide current cube and add new indices to $\boldsymbol{I}_n$
    - **end if**
  - **end for**
  - $\boldsymbol{I} \leftarrow \boldsymbol{I}_n$
- **end for**

Algorithm 2 Adapted Marching Cubes

Input: Intersecting indices $\boldsymbol{I} \in \mathbb{Z}_0^{+M\times 3}$, cube size s, point embedding layer $g_{\boldsymbol{\theta}_1}$, decoder $f_{\boldsymbol{\theta}_2}$, all possible assignments $\boldsymbol{A}$
Output: Mesh $\boldsymbol{M} = (\boldsymbol{V}, \boldsymbol{F})$

- Empty mesh $\boldsymbol{M} \leftarrow \{\}$
- **for** intersecting index $\boldsymbol{i} \in \boldsymbol{I}$ **do**
  - Calculate 8 vertices of the intersecting cube using $\boldsymbol{i}$ and s.
  - Predict 28 binary flags between 8 vertices using Eq.3
  - Minimal cost $l_{min} \leftarrow +\infty$
  - **for** possible assignment $\boldsymbol{a} \in \boldsymbol{A}$ **do**
    - Calculate cost l for assignment $\boldsymbol{a}$ using Eq.8
    - **if** $l < l_{min}$ **then**
      - $l_{min} \leftarrow l$, $\boldsymbol{a}_{min} \leftarrow \boldsymbol{a}$
    - **end if**
  - **end for**
  - Query the vertices and faces in the lookup table according to assignment $\boldsymbol{a}_{min}$ and add to $\boldsymbol{M}$
- **end for**

the initial cube size $s_0 = 1.0/20 = 0.05$, the total number of stages $T = 3$ and the intersecting threshold $\tau = 2$.

After obtaining intersecting indices $\boldsymbol{I}$, the next step is to generate triangles using our adapted Marching Cubes. In each cube, we first use our model to predict all binary flags between 8 vertices, then assign binary labels (0/1) to 8 ver-

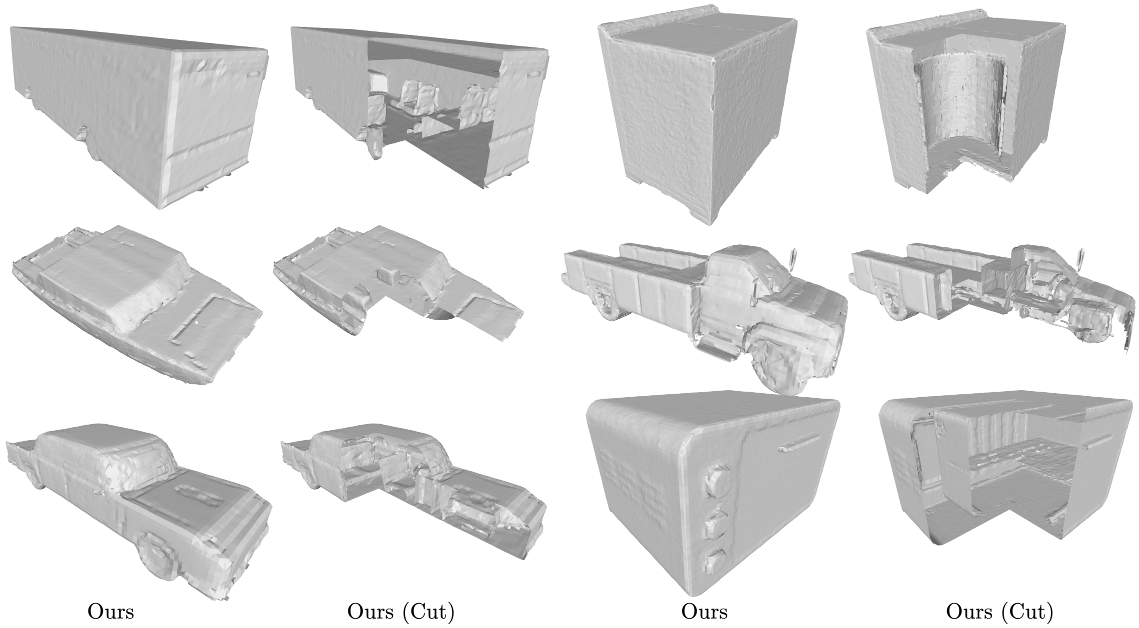

Figure 1. **Reconstruction results of multi-layer shapes.** Our method can reconstruct internal structures of various shapes.

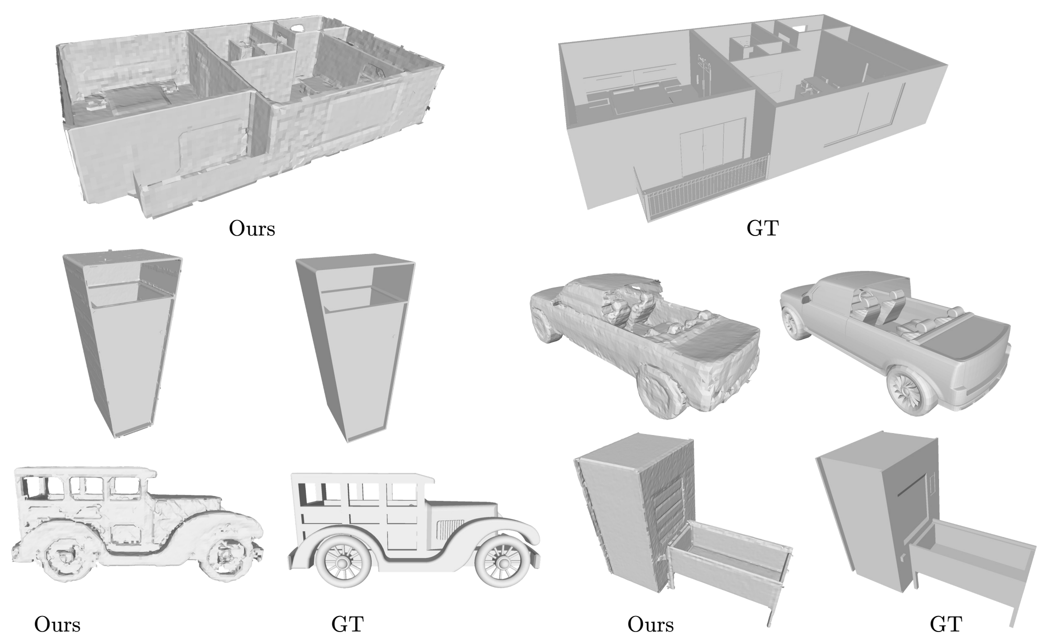

Figure 2. **Reconstruction results of non-watertight shapes.** The non-watertight shapes are difficult for traditional neural implicit functions to reconstruct.

tices based on the binary flags, and finally generate triangles with the lookup table provided by the original Marching Cubes. We show the detailed process in Algorithm 2. Among the inputs of the algorithm, $\boldsymbol{A} = \{0, 1\}^8$ is the all possible binary assignments for 8 vertices.

Finally, we utilize the UDF branch to refine the mesh $\boldsymbol{M} = (\boldsymbol{V}, \boldsymbol{F})$. We sample points on each face and refine mesh vertices by minimizing the UDF values of sam-

Algorithm 3 Mesh refinement

Input: Mesh $\boldsymbol{M} = (\boldsymbol{V}, \boldsymbol{F})$, point embedding layer $g_{\boldsymbol{\theta}_1}$, UDF layer $h_{\boldsymbol{\theta}_3}$, number of iteration N

Output: Refined mesh $\boldsymbol{M} = (\boldsymbol{V}, \boldsymbol{F})$

1: **for** iteration $n \in range(N)$ **do**

2: Sample points $\boldsymbol{P}$ from each face, each sampled point is a linear combination of 3 mesh vertices

3: Optimize mesh vertices $\boldsymbol{V}$ by minimizing Eq.9

4: **end for**

pled points. We show the detailed process in Algorithm 3. Among inputs, the number of iteration N is 30.

C. Qualitative Evaluation

Reconstruction results of multi-layer shapes and non-watertight shapes are shown in Figure 1 and Figure 2.