Neural Vector Fields: Implicit Representation by Explicit Learning (Supplementary Materials)

Xianghui Yang[1], Guosheng Lin[2], Zhenghao Chen[1], Luping Zhou[1]
[1]The University of Sydney, [2]Nanyang Technological University
{xianghui.yang,zhenghao.chen,luping.zhou}@sydney.edu.au, gslin@ntu.edu.sg

A. Data Preprocessing

We create the training point samples (i.e., query points) with their ground truth displacement vectors and precompute 100k points following NDF [4]. For all experiments on the ShapeNet [2] dataset, we sample points in the vicinity of the surface to capture details. Specifically, we uniformly randomly sample a surface point $\hat{\mathbf{q}}$ on the ground-truth mesh and move it by a displacement $\mathbf{d}$ sampled from a Gaussian distribution $\mathbf{d} \sim N(0, \Sigma))$ to form a 3D query point $\mathbf{q} = \hat{\mathbf{q}} + \mathbf{d}$. We employ a diagonal covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ with entries $\Sigma_{i,i} = \sigma$. For the training procedure, we use a subset of samples including 50k points, with 1% of samples from $\sigma = 0.08$ to learn to regress query points far away, with 49% of samples from $\sigma = 0.02$ to learn to regress surface distances within σ and with 50% of samples from $\sigma = 0.003$ to learn to approximate the detailed surface boundary. The detailed data preprocessing script and split will be released.

B. Implementation of Baseline Methods

For those baseline methods in comparison, we download their publicly released codes and rerun them under the same experimental setting as our method for a fair comparison. For the hyper-parameters used in these methods, we follow the optimal values suggested in their original papers. For watertight shape reconstruction, the shapes are pre-processed by DISN [7] and we assign inside/outside labels to the sampled query points according to codes released by OccNet [5] and IF-Nets [3] for training. The three comparing methods *i.e.*, IF-Nets [3], NDF [4], GIFs [9] take the same 3D convolution network as the backbone and their input point clouds are all voxelized with the resolution of 128^3, same as the watertight car reconstruction in the original papers of NDF [4] and GIFS [9]. For other non-watertight shape reconstruction, *i.e.*, category-specific, category-agnostic, category-unseen, and cross-domain reconstructions, the input point clouds of the above three comparing methods are all voxelized with the resolution 256^3. We adopt the Adam optimizer for training with the $lr = 10^{-3}$, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and weight decay $\lambda = 0$.

	Codebook	CD↓	EMD↓	$F1_{1\times10^{-5}}$	$F1_{2\times10^{-5}}$
Base	single	0.094	1.198	75.292	90.167
	multi	**0.085**	**1.197**	**75.372**	**90.266**
Novel	single	0.146	1.366	79.460	91.228
	multi	**0.078**	**1.340**	**79.723**	**91.576**

Table S1. Reconstruction results of our method using a multi-head codebook and its variant using a single-head codebook for the category-agnostic and category-unseen tasks on the ShapeNet dataset.

C. Effectiveness of Multi-head Codebook

In our NVF, we employ a multi-head structured codebook to extend the feature space significantly and enhance the codebook's representation capacity (see Sec. 4 in our paper for more details). To demonstrate the benefits of using a multi-head codebook, we provide a quantitative comparison with a single-head codebook in Tab. S1 for both category-agnostic and category-unseen reconstruction tasks on the ShapeNet dataset (the setting is the same as our ablation study in the paper, please refer to Sec. 5 for more details).

Specifically, we set the codebook size as 512 for our single-head codebook. Our multi-head codebook uses 4 heads and the sub-codebook size of each head is 128, yielding the total size same as that of a single-head cookbook. The code dimensions are all 64. It is observed that the multi-head codebook consistently achieves better performance than the single-head codebook for all metrics. Especially, on novel class reconstruction, the CD of the multi-head codebook almost halves that of the single-head codebook, demonstrating the necessity of employing a multi-head codebook.

D. Comparsion with generation with related works.

ShapeGF [1], AutoSDF [6] and ShapeFormer [8] focus on the 3D generation rather than our task, *i.e.*, surface re-

Dataset		Method	CD↓	EMD↓	$F1_{1\times10^{-5}}$ ↑	$F1_{2\times10^{-5}}$ ↑
Non-watertight	Base	ShapeGF [8]	312.159	14.147	2.523	4.040
		Ours	**0.085**	**1.197**	**75.372**	**90.266**
	Novel	ShapeGF [8]	365.742	15.391	1.076	1.745
		Ours	**0.078**	**1.340**	**79.723**	**91.576**
Watertight	Base	AutoSDF [6]	5.670	1.991	0.905	1.688
		ShapeFormer [8]	0.799	1.509	47.028	67.210
		Ours	**0.091**	**1.079**	**78.503**	**91.408**
	Novel	AutoSDF [6]	8.193	2.256	2.270	3.895
		ShapeFormer [8]	1.877	1.973	38.979	58.051
		Ours	**0.144**	**1.145**	**80.883**	**91.836**

Table S2. Quantitative comparison with related works ShapeGF, AutoSDF, and ShapeFormer.

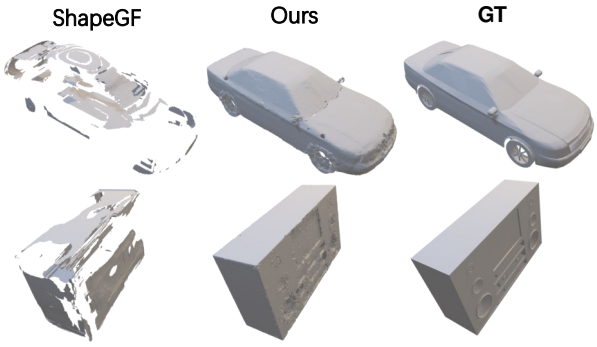

Figure S1. Examples of non-watertight mesh reconstruction.

construction. We add performance comparison in Tab. S2 and Fig. S1. Ours performs consistently best, while ShapeGF [1] leaves large holes on most non-watertight meshes and AutoSDF [6] & ShapeFormer [8] could not work on non-watertight meshes due to their SDF representation. Note that AutoSDF is a voxelized-SDF method limited by its predefined resolution 64^3, leading to larger errors. As for the use of gradient, ShapeGF predicts gradients to attain a Gaussian mixture model to estimate point distribution density, while these gradients point to high-density areas instead of the underlying surface, which is completely different from our motivation and implementation. As for the codebook, AutoSDF and Shapeformer utilize VQ for denoising and bridging cross-modality gaps, different from our purpose of encoding cross-object priors. Ours also differs in using a multi-head codebook and encoding the relation of query points to local geometry in each code. Note that we use VQ as an example to demonstrate that the differentiation-free property of NVF provides more flexibility in model design.

E. Surface Extraction Speed

In the main manuscript, we present the inference comparison for obtaining distance and direction. In this regard, the branch in GIFS [9] for distance prediction is akin to NDF [4]. However, it reduces the time for surface extraction, by incorporating an intersection classification branch. We present an evaluation of the overall reconstruction process speed, encompassing surface extraction. Remarkably, our NVF surpasses GIFS [9] by approximately 25%, even when the latter purposely optimizes surface extraction as shown in Tab. S3.

Methods	GIFs [9]	Ours
Runtime	23.89s	17.74s

Table S3. Surface extraction speed comparison. The runtime are average time cost for 100 times surface extraction.

F. Codebook Visualization

We further provide the codebook visualization in Fig. S2 to demonstrate the information (*i.e.*, the learned code) stored by our codebook. We select the 1^{st} head as an example from our multi-head codebook to visualize 4 learned VQ codes, in which we show all the query points using this code and their corresponding displacements against the ground-truth surfaces. It is observed that across different objects, the query points corresponding to the same code display roughly the same displacement (in both magnitude and direction), which will further provide cross-object priors.

G. Explicit Deformation Visualization

In order to demonstrate that NVF could serve as both an implicit neural function and an explicit deformation function (*i.e.*, displacement output of the function could be directly used to deform source meshes), we provide some deformation results in Fig. S3, which indicates that our NVF can directly deform the convex hull meshes of the input point cloud to fit the surface.

H. Qualitative Visualization

Finally, We provide more visualization results in Figs. S4 to S9 to further demonstrate the reconstruction performance

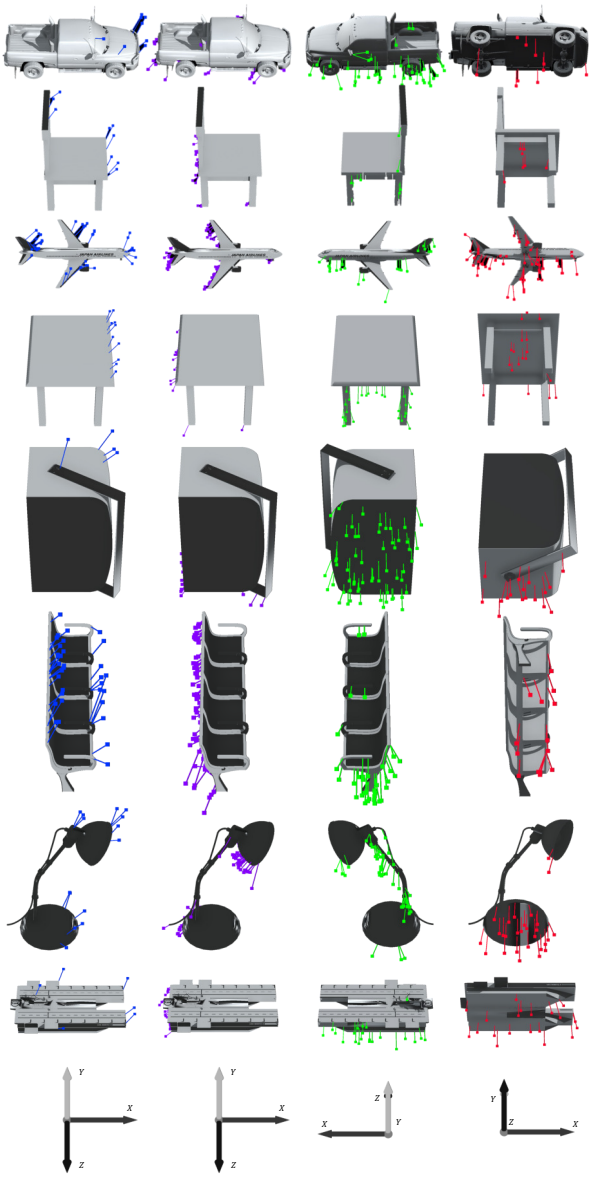

Figure S2. Visualization of four codes from the 1^{st} head learned by our multi-head codebook. Each column represents a learned code. Each row represents an object, except the last row indicates the views (coordinate system) of the object.

of our NVF on different reconstruction tasks. Please refer to the main paper for a detailed description of each task.

References

[1] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2

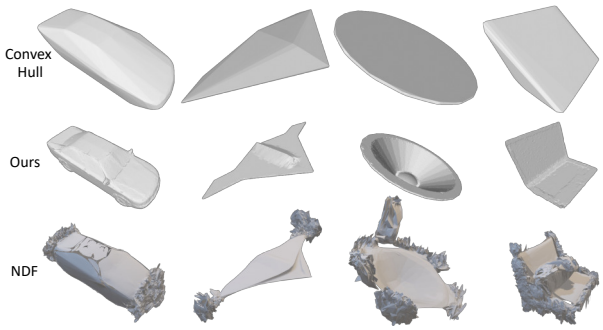

Figure S3. Mesh deformation visualization. The 1^{st} row represents the source meshes that are the convex hull of the input point clouds, and the 2^{nd} row represents the shapes after deformation by our NVF.

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[3] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. 1

[4] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2020. 1, 2

[5] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[6] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2022. 1, 2

[7] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 492–502. Curran Associates, Inc., 2019. 1

[8] Xingguang Yan, Liqiang Lin, Niloy J. Mitra, Dani Lischinski, Danny Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation, 2022. 1, 2

[9] Jianglong Ye, Yuntao Chen, Naiyan Wang, and Xiaolong Wang. Gifs: Neural implicit function for general shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2

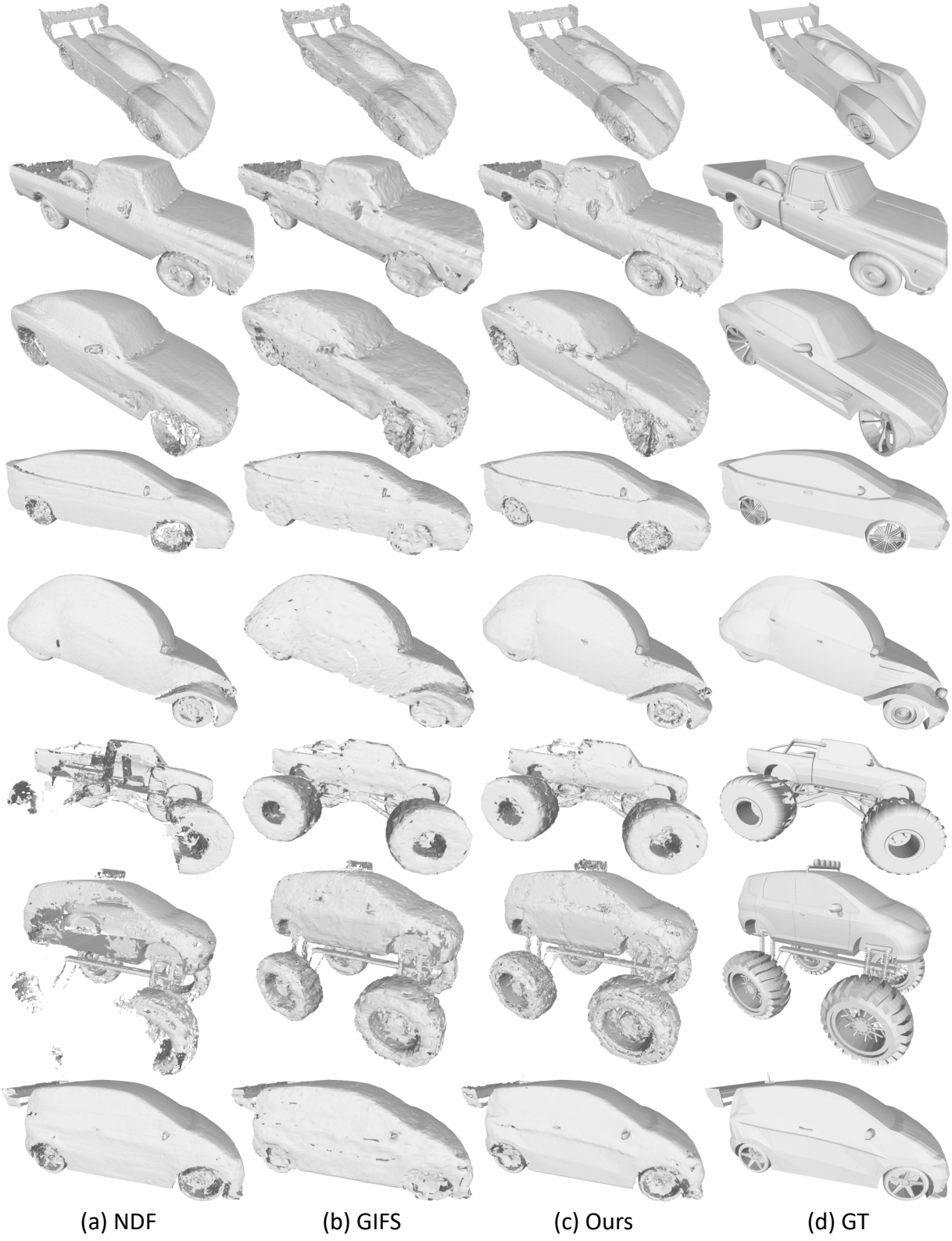

Figure S4. Visualization of category-specific reconstruction task on car samples from the ShapeNet dataset.

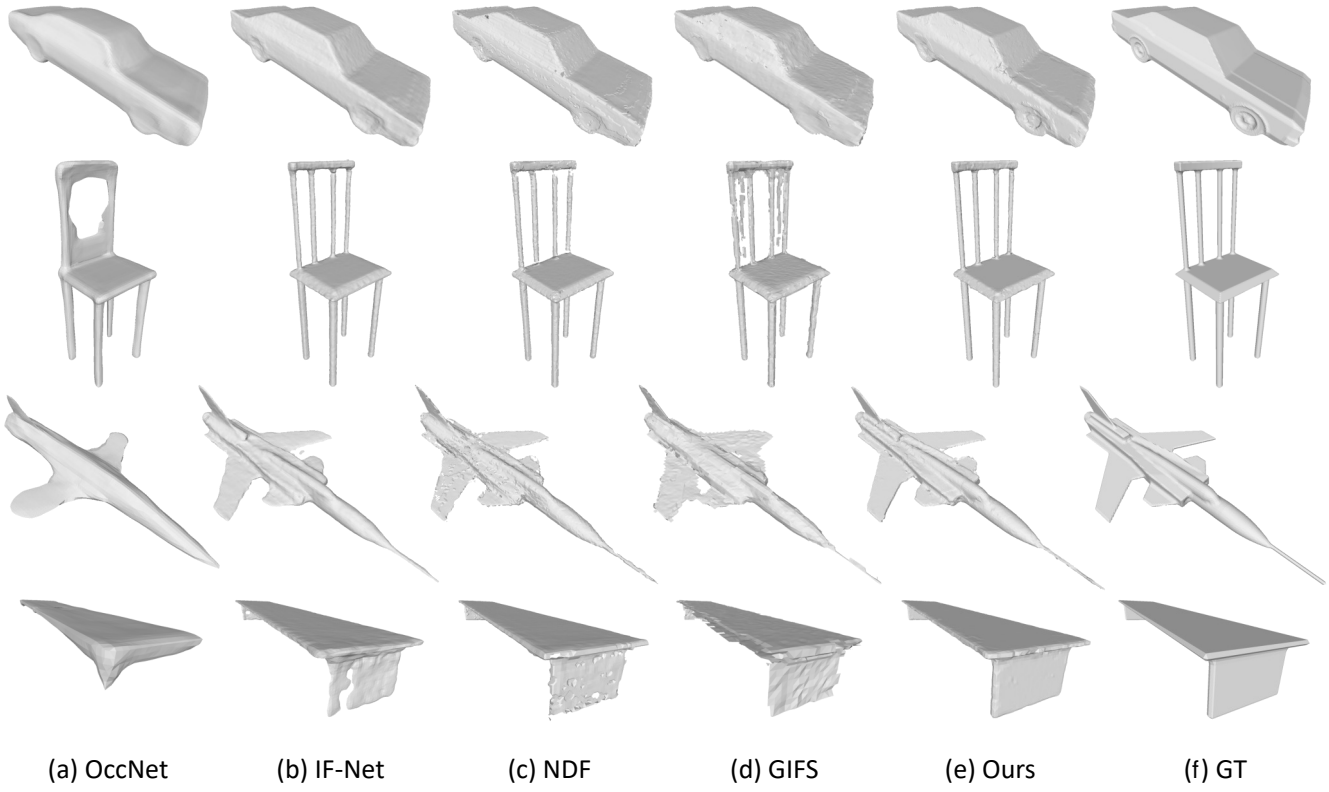

Figure S5. Visualization of category-agnostic reconstruction task on watertight shapes from the ShapeNet dataset.

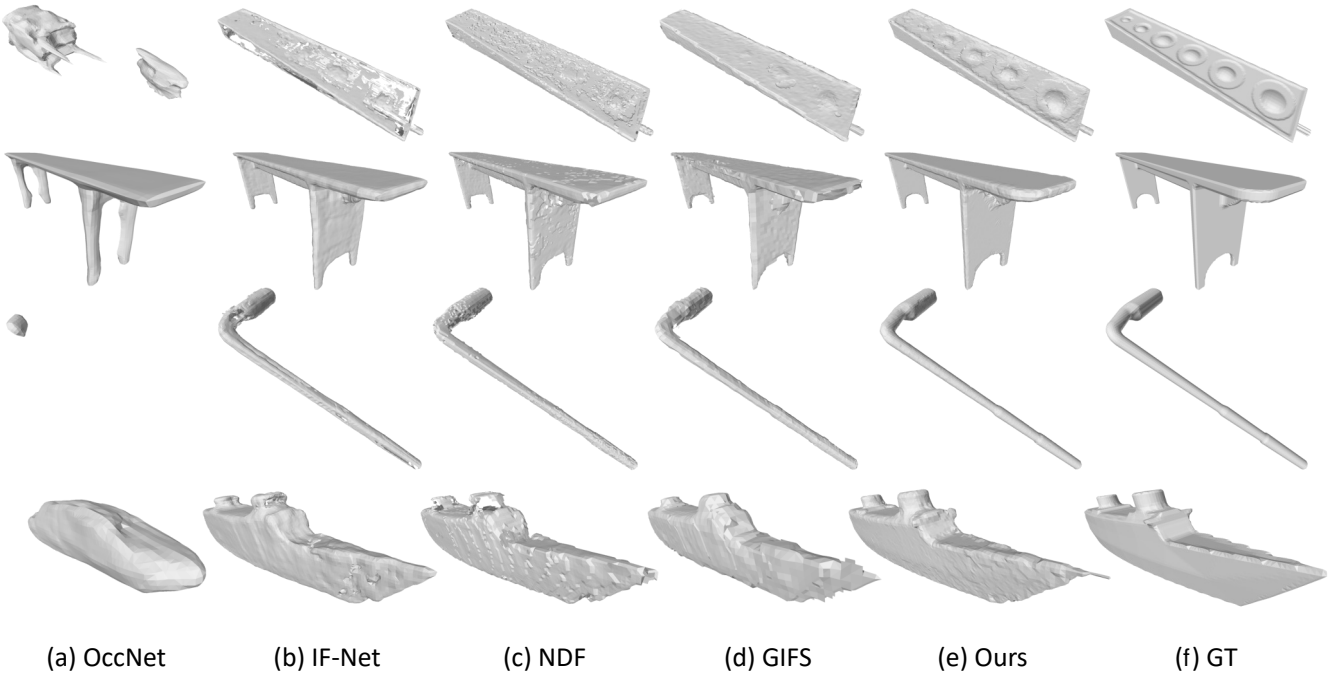

Figure S6. Visualization of category-unseen reconstruction task on watertight shapes from the ShapeNet dataset.

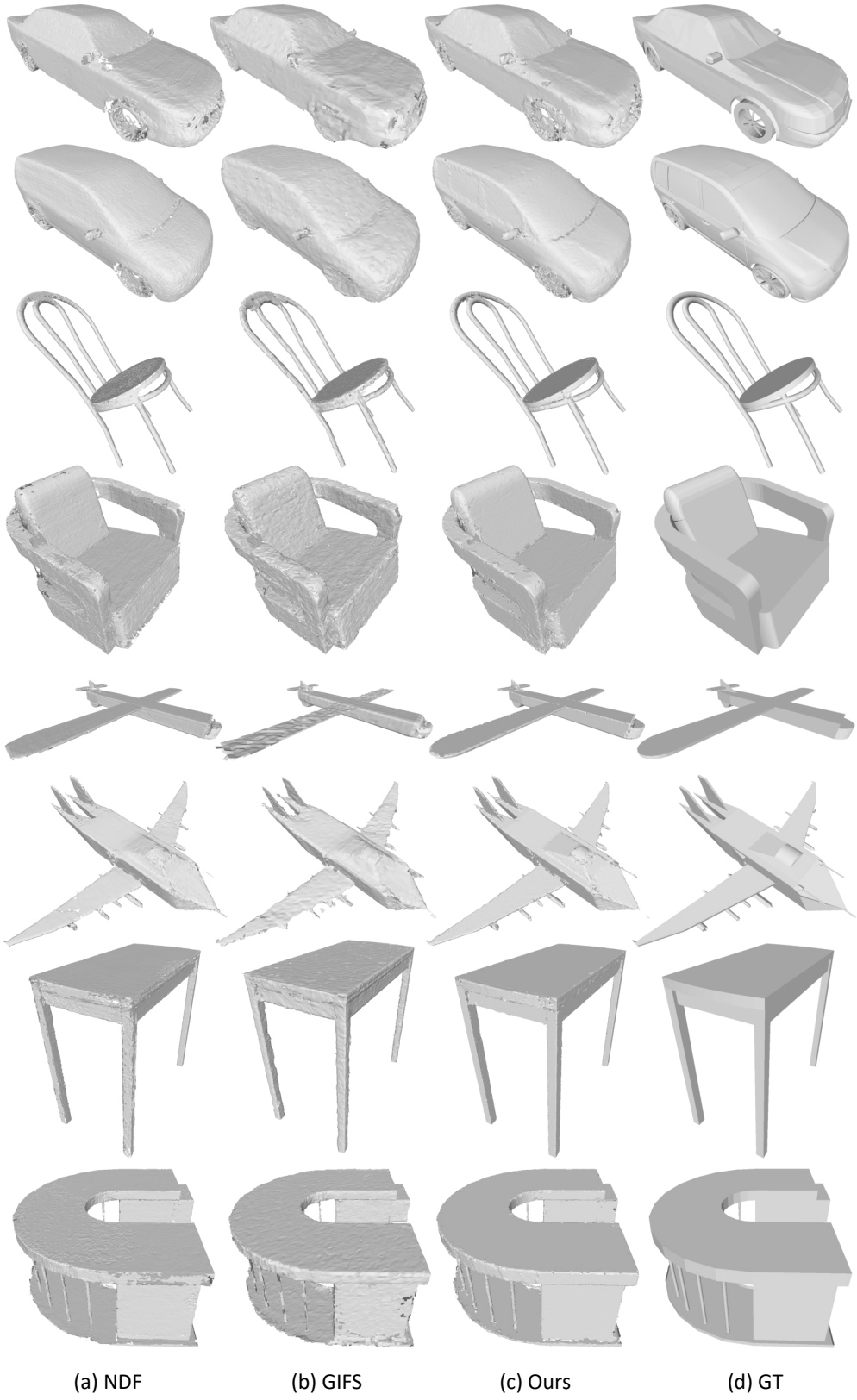

Figure S7. Visualization of category-agnostic reconstruction task on non-watertight shapes from the ShapeNet dataset.

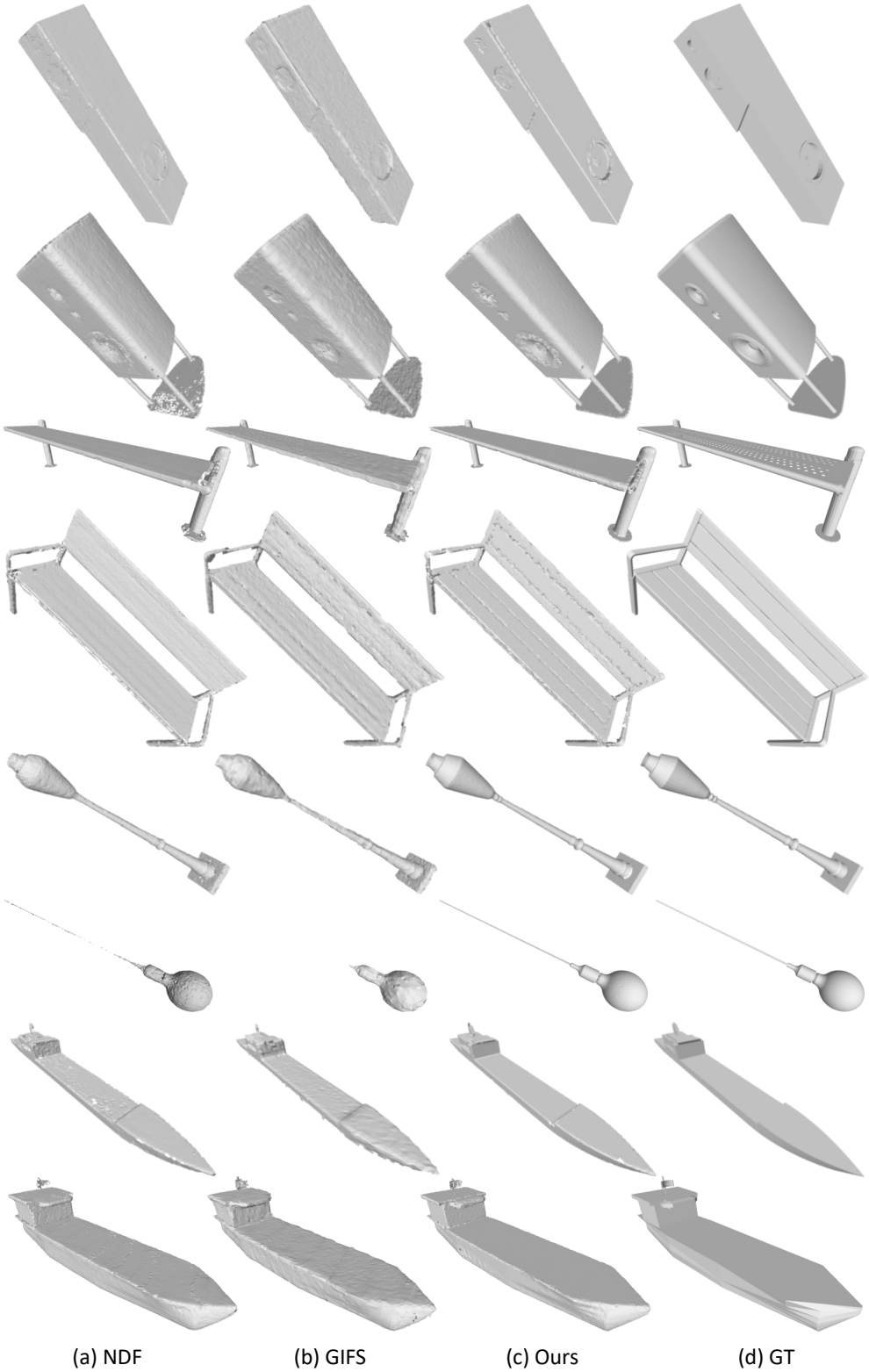

Figure S8. Visualization of category-unseen reconstruction task on non-watertight shapes from the ShapeNet dataset.

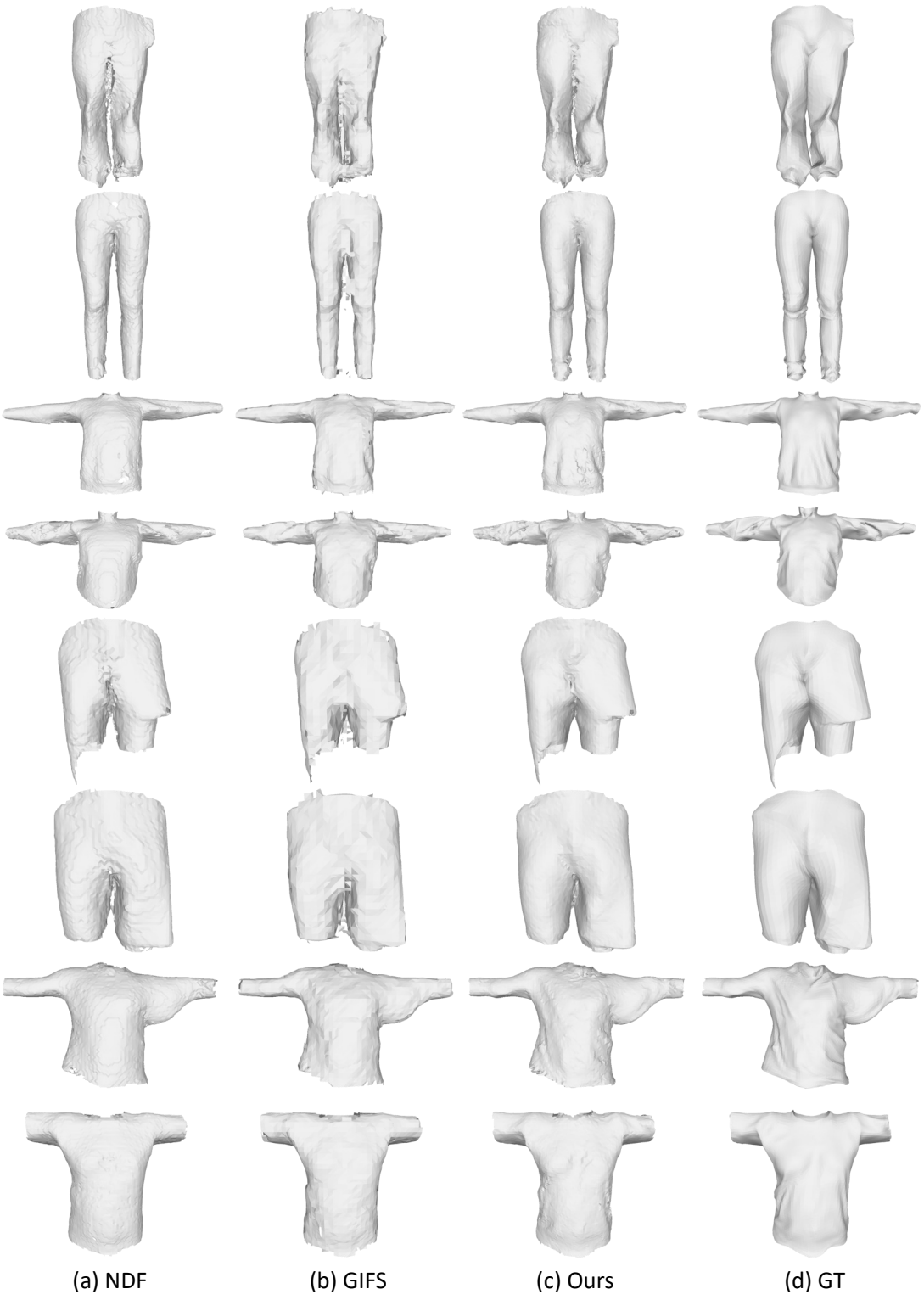

Figure S9. Visualization of cross-domain reconstruction task on the MGN dataset. All models are trained based on the base classes from the ShapeNet dataset, and then evaluated on the MGN dataset.