

Supplementary Material for Learning Bijective Surface Parameterization for Inferring Signed Distance Functions from Sparse Point Clouds with Grid Deformation

Takeshi Noda¹ Chao Chen^{1*} Junsheng Zhou¹ Weiqi Zhang¹

Yu-Shen Liu^{1†} Zhizhong Han²

¹School of Software, Tsinghua University, Beijing, China

²Department of Computer Science, Wayne State University, Detroit, USA

{yeth24, zhou-js24, zwq23}@mails.tsinghua.edu.cn

chenchao19@tsinghua.org.cn liuyushen@tsinghua.edu.cn h312h@wayne.edu

1. Implementation Details

We further describe the network architecture of our method. The encoder Φ receives a sparse point cloud with dimensions $N \times 3$ as input and transforms it into feature maps with dimensions $N \times 256$ and a set of parameterized surfaces, each of which is with a dimension of $N \times 3$. For each parameterized surface point, we use KNN (K-Nearest Neighbors) sampling on the spherical surface to select 10 neighboring points, resulting in local patches. Similarly, the decoder Ψ consists of a four-head attention layer with MLP layers. It takes a feature map and position encodings with 256 dimensions as input, and predicts the coordinates of the local patches. All local patches share the same parameters, which further enhances the efficiency and performance of BSP. The neural network g consists of linear layers with ReLU activation functions, where the first 5 layers have a dimension of 256, and the final layer has a dimension of 1 to predict the signed distance functions.

We utilize the Adam optimizer during training, initializing the learning rate of 0.0001. We conduct the training over 40,000 epochs as our default setting. With learned signed distances from each shape, we employ the Marching Tetrahedra algorithm to reconstruct the surface. All of our experiments are conducted on a single GeForce RTX-3090 GPU.

2. Additional Experiments and Visualization

2.1. Robustness to Noise

We report the experimental results under different noise levels in Tab. 1. We add Gaussian noise with standard deviations including 1%, 2% and 3% to create noisy inputs. Although our metrics slightly decrease under high level noise, we can still predict the complete geometry. We further provide the visual comparisons in Fig. 1.

2.2. Comparison of Densification Strategies

To validate the effectiveness of different densification strategies for processing sparse point clouds, we select GradPU [3] and LDI [4] as the baselines for comparative

Noise Level	0%	1%	2%	3%
$CD_{L1} \times 10$	0.077	0.083	0.090	0.098
$CD_{L2} \times 100$	0.043	0.047	0.056	0.064
NC	0.914	0.883	0.832	0.780

Table 1. Robustness of noise.

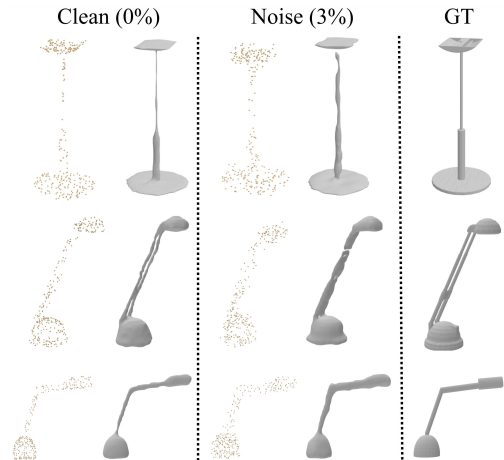


Figure 1. Robustness of noise.

analysis. GradPU generates dense point clouds through linear interpolation, while LDI employs local indicators to achieve more precise results. We compare the densification effects of their point clouds following the default settings of TPS [2] on ShapeNet. As shown in Fig. 2, neither GradPU nor LDI is able to produce a uniform distribution, as both are hindered by the extremely sparse input. Conversely, BSP achieves the smallest error (Bluer) with a continuous distribution. Furthermore, we replace BSP with LDI and GradPU for optimization. We also provide CD error maps for comparison. Both LDI and GradPU struggle to provide reliable supervision for inferring continuous surfaces. In contrast, BSP significantly improves the accuracy of the reconstruction in challenging local regions.

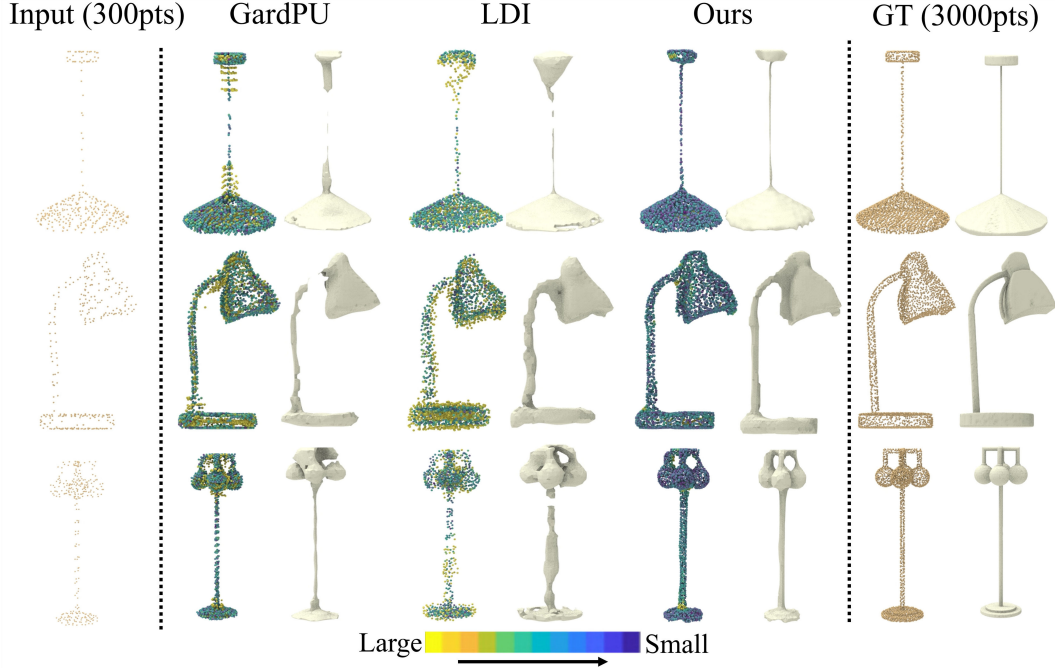


Figure 2. Comparison of densification strategies, the input contains 300 points. The color indicates the point distance error to ground truth surface.

2.3. Effect of Sample Strategy

To validate the effect of the spherical projection U_{proj} on the center point Q , we replace U_{proj} with normalization, resulting in an approximate spherical surface $U_{\text{norm}}(Q)$. However, the center points $U_{\text{norm}}(Q)$ do not guarantee that local queries lie on the same plane. As shown in Tab. 2, the normalization strategy $U_{\text{norm}}(Q)$ leads to an increase in the CD metric, indicating that spherical projection sampling is effective for learning bidirectional mappings.

	$CD_{L1} \times 10$	$CD_{L2} \times 100$	NC
U_{norm}	0.081	0.046	0.911
Ours (U_{proj})	0.077	0.043	0.914

Table 2. Effect of sample strategy.

2.4. Comparison of Mapping Strategy

BSP aims to densify sparse point clouds in latent space, it relies on an Encoder-Decoder (ED) architecture to map sparse 3D points into codes in the latent space and then decode newly sampled codes into dense 3D points. Our novelty lies in how we can sample codes that are consistent to the codes of sparse points so that we can not only densify points but also preserve the geometry and topology of sparse points without learning data-driven priors. Specifically, we introduce BSP within the ED, which projects codes onto a sphere manifold and samples new codes on the manifold to decode densified points. Tab. 3 and Fig. 3 show

that our method significantly constrains the sampling space, preserve geometry of the shape and recover more consistent geometry.

Methods	$CD_{L1} \times 10$	$CD_{L2} \times 100$	NC
ED+DMTet	0.091	0.055	0.873
ED+GDO	0.083	0.048	0.881
BSP+GDO (Ours)	0.077	0.043	0.914

Table 3. Comparison of ED and BSP on ShapeNet Lamp class.

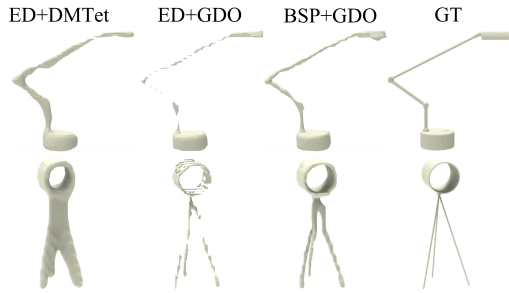


Figure 3. Comparison of mapping strategy.

2.5. Effect of the Gradient of GDO

GDO leverages the gradient of SDF to constrain the offset direction of nodes, which pushes all nodes move along the directions of the greatest signed distance increase to the zero-level set. This significantly improves the performance of DMTet in surface inference from sparse point

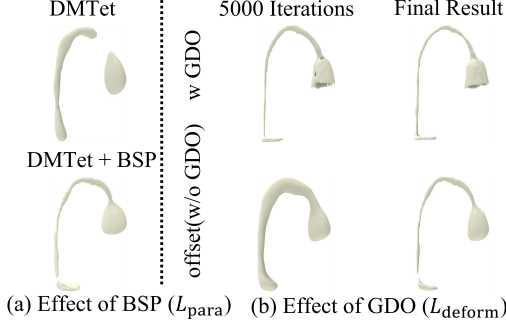


Figure 4. Effect of the gradient of GDO.

clouds. Meanwhile, GDO also improves both convergence efficiency and performance as shown in Fig. 4.

2.6. Optimal Setting of KNN.

BSP samples k code candidates in the spherical latent space rather than in the 3D space, where the sampled codes will be decoded into 3D points by the decoder. The k does not significantly affect the locations of the densified points, since the code candidates are densely and uniformly distributed on the spherical latent space. With the input as supervision, the densified points are uniformly distributed on the surface if k is large enough to set to 10. We further provide visual comparisons in Fig. 5.

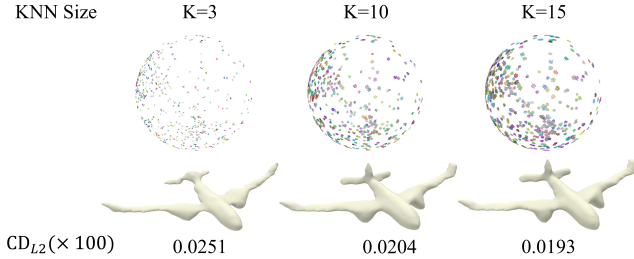


Figure 5. Optimal setting of KNN.

2.7. Robustness of BSP.

BSP encodes points into features which are then projected to the spherical latent space. The impact of sparsity and noises on topology can be minimized in the latent space during the sampling since we do not directly sample points in 3D. This can be well illustrated in Fig. 6 where we can reveal surfaces well with the same topology even from sparse and noisy points which are almost mixed with each other.

3. More Visualizations

To further validate the effectiveness of our method, we provide additional visualization and comparisons in Fig. 7 and Fig. 8, respectively. We select lamps, planes, and vessels from the ShapeNet dataset for visual comparison. We also compare our approach with more recent methods, including

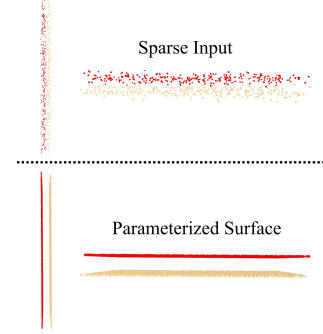


Figure 6. Robustness of BSP.

NS [6], IMLS [5], and GP [1]. The results in Fig. 8 demonstrate that our method offers competitive performance in reconstruction tasks.

4. Limitation and Future Works

We observe that recent parameterization based methods can produce dense representations, but the uniform distribution of the parameterized surface is unconstrained, making the reconstruction results highly sensitive. We aim to address this issue to further improve the accuracy of the representation in future work.

5. Related Metrics

L1-Chamfer Distance (CD_{L1}). CD_{L1} measures the average of the nearest distance errors between two point sets with L1-norm (Manhattan distance). Given two point sets P_1 and P_2 , the CD_{L1} can be calculated as:

$$CD_{L1}(P_1, P_2) = \frac{1}{|P_1|} \sum_{\mathbf{p}_i \in P_1} \min_{\mathbf{p}_j \in P_2} \|\mathbf{p}_i - \mathbf{p}_j\|_1 + \frac{1}{|P_2|} \sum_{\mathbf{p}_j \in P_2} \min_{\mathbf{p}_i \in P_1} \|\mathbf{p}_j - \mathbf{p}_i\|_1. \quad (1)$$

L2-Chamfer Distance (CD_{L2}). CD_{L2} measures the average nearest distance errors between two point sets using the L2-norm (Euclidean distance). Given two point sets P_1 and P_2 , which do not necessarily have the same number of points, CD_{L2} is defined as:

$$CD_{L2}(P_1, P_2) = \frac{1}{|P_1|} \sum_{\mathbf{p}_i \in P_1} \min_{\mathbf{p}_j \in P_2} \|\mathbf{p}_i - \mathbf{p}_j\|_2 + \frac{1}{|P_2|} \sum_{\mathbf{p}_j \in P_2} \min_{\mathbf{p}_i \in P_1} \|\mathbf{p}_j - \mathbf{p}_i\|_2. \quad (2)$$

Normal Consistency (NC). NC is used to evaluate the accuracy of the normal vectors between two point clouds P_1 and P_2 . Here, $\langle p_i, P_2 \rangle$ denotes finding the point in point cloud P_2 that is closest to p_i , and the similarity of the normal vectors at the positions of the two points is measured

by calculating the dot product of the normal vector of that point with the normal vector of p_i . The higher this metric, the better. The formula is as follows:

$$\begin{aligned} \text{NC}(P_1, P_2) = & \frac{1}{|P_1|} \sum_{p_i \in P_1} \langle p_i, P_2 \rangle \\ & + \frac{1}{|P_2|} \sum_{p_j \in P_2} \langle p_j, P_1 \rangle. \end{aligned} \quad (3)$$

Hausdorff Distance (HD). HD measures the maximum distance from a point p_i in point set P_1 , to the nearest point p_j in another point set P_2 , which can be described as:

$$\begin{aligned} \text{HD}(P_1, P_2) = \max \left(\max_{p_i \in P_1} \min_{p_j \in P_2} \|p_i - p_j\|_2, \right. \\ \left. \max_{p_j \in P_2} \min_{p_i \in P_1} \|p_j - p_i\|_2 \right). \end{aligned} \quad (4)$$

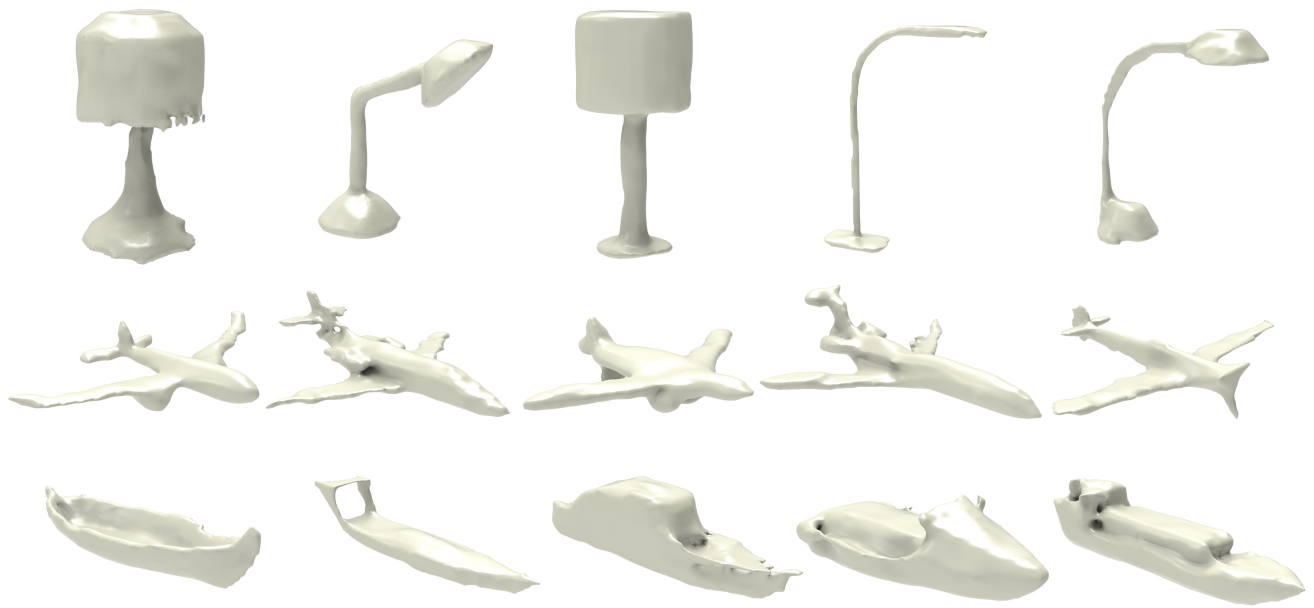


Figure 7. Visual presentations for ShapeNet dataset.

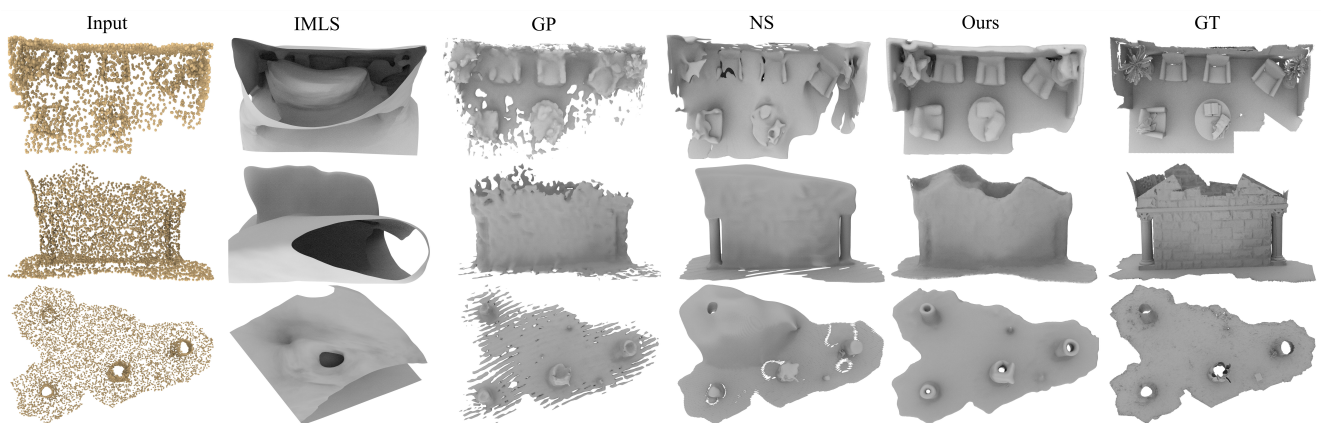


Figure 8. Visual comparison of state-of-art methods.

References

- [1] Chao Chen, Yu-Shen Liu, and Zhizhong Han. GridPull: Towards scalability in learning implicit representations from 3D point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18322–18334, 2023. [3](#)
- [2] Chao Chen, Yu-Shen Liu, and Zhizhong Han. Unsupervised inference of signed distance functions from single sparse point clouds without learning priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17712–17723, 2023. [1](#)
- [3] Yun He, Danhang Tang, Yinda Zhang, Xiangyang Xue, and Yanwei Fu. Grad-pu: Arbitrary-scale point cloud upsampling via gradient descent with learned distance functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2023. [1](#)
- [4] Shujuan Li, Junsheng Zhou, Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Learning continuous implicit field with local distance indicator for arbitrary-scale point cloud upsampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3181–3189, 2024. [1](#)
- [5] Zixiong Wang, Pengfei Wang, Pengshuai Wang, Qiujie Dong, Junjie Gao, Shuangmin Chen, Shiqing Xin, Changhe Tu, and Wenping Wang. Neural-impls: Self-supervised implicit moving least-squares network for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2023. [3](#)
- [6] Zixiong Wang, Yunxiao Zhang, Rui Xu, Fan Zhang, Pengshuai Wang, Shuangmin Chen, Shiqing Xin, Wenping Wang, and Changhe Tu. Neural-singular-hessian: Implicit neural representation of unoriented point clouds by enforcing singular hessian. *ACM Transactions on Graphics (TOG)*, 42(6), 2023. [3](#)