
MultiPull: Detailing Signed Distance Functions by Pulling Multi-Level Queries at Multi-Step

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Reconstructing a continuous surface from a raw 3D point cloud is a challenging
2 task. Latest methods employ supervised learning or pretrained priors to learn a
3 signed distance function (SDF). However, neural networks tend to smooth local
4 details due to the lack of ground truth signed distances or normals, which limits
5 the performance of learning-based methods in reconstruction tasks. To resolve this
6 issue, we propose a novel method, named MultiPull, to learn multi-scale implicit
7 fields from raw point clouds to optimize accurate SDFs from coarse to fine. We
8 achieve this by mapping 3D query points into a set of frequency features, which
9 makes it possible to leverage multi-level features during optimization. Meanwhile,
10 we introduce optimization constraints from the perspective of spatial distance and
11 normal consistency, which play a key role in point cloud reconstruction based on
12 multi-scale optimization strategies. Our experiments on widely used object and
13 scene benchmarks demonstrate that our method outperforms the state-of-the-art
14 methods in surface reconstruction.

15

1 Introduction

16 Reconstructing surfaces from 3D point clouds is an important task in computer vision. It is widely
17 used in various real-world scenarios such as autonomous driving, 3D scanning and other applications,
18 as well as in downstream tasks.

19 Recently, using neural networks to learn signed distance functions from 3D point clouds has made
20 huge progress [1, 2, 3, 4, 5]. An SDF represents a 3D surface as the zero-level set of a continuous
21 field in 3D space, and the surface can be further extracted using the marching cubes algorithm [6].
22 In supervised methods [7, 8, 9, 10], a continuous field in 3D space is learned using signed distance
23 supervision. Some methods employ multi-level representations [11, 12], such as Fourier layers
24 and level of detail (LOD) [13, 14], to learn detailed geometry. However, these methods require
25 3D supervision, including ground truth signed distances or point normals, by converting a mesh
26 to a watertight manifold. To address this issue, several unsupervised methods [15, 16, 17, 18, 19]
27 have been designed to directly learn an SDF from a single point cloud without requiring ground
28 truth signed distances and point normals. They usually adopt various strategies, including geometric
29 constraints [15, 16, 17] and consistency constraints [19, 20], to guide neural networks to fit a signed
30 distance field. However, the raw point cloud is a highly discrete approximation of the surface, learning
31 SDFs directly from the point cloud is often inaccurate and highly ambiguous. This makes it hard for
32 the network to learn accurate SDFs on local details, resulting in over-smooth reconstruction.

33 To address this issue, we propose *MultiPull*, to learn a precise SDF with multi-scale frequency
34 features. It enables network to predict SDF from coarse to fine, significantly enhancing the accuracy
35 of the predictions. Furthermore, to optimize the SDF at various scales simultaneously, we design
36 constraints to learn accurate implicit functions through query point stride constraints.

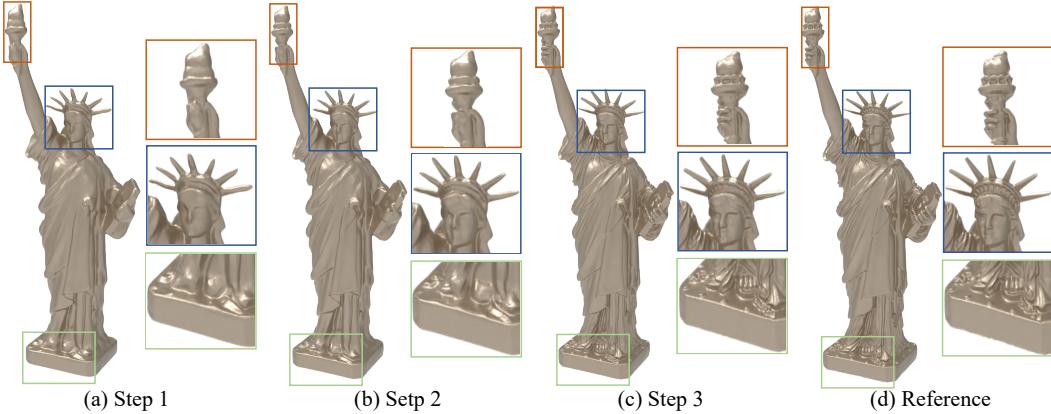


Figure 1: Visualization of the 3D shape reconstruction. In (a), (b) and (c), SDFs are learned from a point cloud by optimizing multi-level query points at multi-step. At each step, we optimize query points at one level with frequency features at this specific level as conditions. This enables the network to progressively recover coarse-to-fine geometry details.

37 Specifically, given query points sampled around 3D space as input, we use a Fourier transform
 38 network to represent them as a set of Fourier features. Next, we design an SDF prediction network
 39 that leverages multi-scale Fourier features to learn a series of SDF fields from coarse to fine. To
 40 precisely optimize the multi-scale signed distance fields, we introduce a loss function based on
 41 gradient consistency and distance awareness. Compared with Level of Detail (LOD) methods, we
 42 optimizes the signed distance fields uniformly and effectively without the need for real supervision,
 43 achieving more accurate geometric details. Evaluations on widely used benchmarks show that our
 44 method outperforms the state-of-the-art methods. Our contribution can be summarized as follows.

- 45 • We propose a novel framework that can directly learn SDFs with details from raw point
 46 clouds, progressing from coarse to fine. This provides a new perspective for recovering 3D
 47 geometry details.
- 48 • We introduce an effective joint multi-level loss function based on gradient consistency and
 49 distance awareness, enabling the network to adaptively perform more accurate inferences in
 50 challenging regions.
- 51 • Our method outperforms state-of-the-art methods in surface reconstruction in terms of
 52 accuracy under widely used benchmarks.

53 2 Related Work

54 Traditional studies for geometric modeling [21, 22, 23, 24] have attempted to analyze the geometric
 55 surface morphology of objects, which do not require large-scale datasets. With the advent of
 56 extensive and intricate 3D datasets like ShapeNet [25] and ABC [26], learning-based methods
 57 have achieved significant advancements [9, 27, 28, 19, 29, 30, 31, 32]. These approaches acquire
 58 implicit representations through various inputs, including multi-views [33, 34, 35, 36], point clouds
 59 [37, 38, 39], and voxels [40, 41, 42]. Notably, surface reconstruction from 3D point clouds stands
 60 out as one of the most demanding tasks in the realm of graphics and 3D vision.

61 **Learning Implicit Function with Supervision.** Supervised methods have made significant progress
 62 in recent years. These methods leverage deep learning network models to learn priors from datasets
 63 or use real data for supervision [7, 8, 43] to improve surface reconstruction performance. Typically,
 64 supervised approaches learn real signed distances and point normals as prior information, or leverage
 65 actual occupancy grids to guide the network’s learning process. In order to improve the generalization
 66 ability of neural networks and learn more geometric details, some studies learn geometry prior of
 67 shapes through supervised learning. For example, some methods use signed distance fields (SDF) or
 68 occupancy fields to learn the occupancy or grids to achieve precise results.

69 **Learning Implicit Function under Condition.** To alleviate the dependence on supervised infor-
 70 mation, recent studies focus on unsupervised implicit reconstruction methods. These methods do

not require priors or normals during optimization. For example, NeuralPull (NP) [18] learns SDF by pulling query points in nearby space onto the underlying surface, which relies on the gradient field of the network. CAP [44] further complements this by forming a dense surface by additionally sampling dense query points. GridPull [20] generalizes this learning method to the grid, by pulling the query point using interpolated signed distances on the grid. In addition, there are some studies that explore surface reconstruction more deeply and innovative methods such as utilizing differentiable Poisson solutions [45] or leaning signed [46, 47] or unsigned functions [48, 44] with priors. However, inferring implicit functions without 3D supervision requires a very long convergence process, which limits the performance of unsupervised methods in large-scale point cloud data.

Learning Implicit Function under LOD. Level-Of-Detail (LOD) models [13, 14, 49] are used to simplify code complexity and refine surface details through the architecture of multi-level outputs. The creation of 3D shapes LOD usually depends on mesh decimation, which has difficulty in blending between LODs. In terms of shape and scene representation, several studies have explored multi-scale architectures. For example, NGLOD [13] uses octree-based feature volumes to represent implicit surfaces, which can adapt to shapes with multiple discrete levels of detail and enable continuous level-of-detail switching through SDF interpolation. MFLOD [14] applies Fourier layers to LOD, which can offer better feasibility in Fourier analysis.

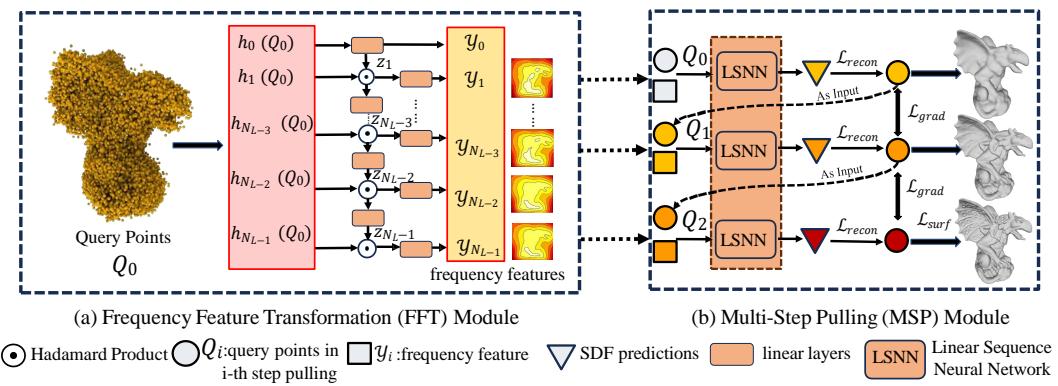


Figure 2: Overview of our method: (a) Frequency Feature Transformation (FFT) module and (b) Multi-Step Pulling (MSP) module. In (a), we learn Fourier bases $h_i(Q)$ from query points Q using the Fourier layer and obtain multi-level frequency features y_i through Hadamard product. In (b), using multi-level frequency features from (a) and a linear network **LSNN** with shared parameters, we calculate the distance(D) of Q_i to its corresponding surface target point Q_t to predict a more accurate surface. We visualize the predicted SDF distribution map corresponding to the frequency features in (a) and the reconstruction from each step of SDF predictions on the right side of (b).

3 Method

Overview. The overview of MultiPull is shown in Fig. 2. We design a neural network to learn an implicit function f from a single 3D point cloud by progressively pulling a set of query points Q_0 onto the underlying surface, where Q_0 is randomly sampled around the raw point cloud S . Our network mainly consists of two parts as follows.

(1) The Frequency Feature Transformation (FFT) Module (Fig. 2(a)) aims to convert the query points Q_0 into a set of multi-level frequency features $Y = \{y_i, i \in [0, N_L - 1]\}$. The key insight for introducing frequency features lies in a flexible control of the degree of details. (2) The Multi-Step Pulling (MSP) Module (Fig. 2(b)) is designed to predict f with coarse-to-fine details under the guidance of frequency features Y . At the i -th step, we pull Q_i to Q_{i+1} with its feature y_i using the predicted signed distances $s_i = f(Q_i, y_i)$ and the gradients at Q_i . To this end, we constrain query points to be as close to their nearest neighbor on S .

3.1 Frequency Feature Transformation (FFT) Module

We introduce a neural network to learn frequency features Y from point clouds. The network manipulates input Q_0 through several linear layers to obtain an initial input z_0 and a set of Fourier

103 basis $h_i(Q_0)$, $i \in [0, N_L - 1]$, formulated as follows.

$$\begin{cases} h_i(Q_0) = \sin(\omega_i Q_0 + \phi_i), \\ z_0 = h_0(Q_0), \end{cases} \quad (1)$$

104 where ω_i and ϕ_i are the parameters of the network, and N_L is the number of layers of the network.

105 To effectively represent the expression of the raw input in the frequency space, we choose the sine
106 function as the activation function and employ the Hadamard product to compute the intermediate
107 frequency feature output. Since the Hadamard product allows the representation of frequency
108 components as the product of two feature inputs, denoted as a and b , which can be formulated as:

$$\sin(a)\sin(b) = \frac{1}{2}(\sin(a + b - \frac{\pi}{2}) + \sin(a - b + \frac{\pi}{2})). \quad (2)$$

109 Through Eq. (2), we can calculate the frequency component z_i of $h_i(Q_0)$, and then obtain the output
110 y_i of the i -th layer, formulated as:

$$\begin{cases} z_i = h_i(Q_0) \odot (W_i z_{i-1} + b_i), i \in [1, N_L - 1] \\ y_i = W_i z_i + b_i, \end{cases} \quad (3)$$

111 where \odot indicates the Hadamard product, W_i, b_i are parameters of the network.

112 Frequency networks based on the Multiplication Filter Network (MFN) [11] typically employ uniform
113 or fixed-weight initialization for network parameters in practice. This approach overlooks the issue
114 of gradient vanishing in deep network layers during the training process, leading to underfitting and
115 making the network overly sensitive to changes in hyperparameters. To address this challenge, we
116 propose a new initialization scheme that thoroughly considers the impact of network propagation,
117 aiming at ensuring a uniform distribution of initial parameters. Specifically, we dynamically adjust
118 initial weights, which can be formulated as:

$$\Psi_i = \sqrt{\eta \times \sin(i\pi/N_L)}, i \in [1, N_L - 1]. \quad (4)$$

119 We leverage the standard deviation as the initialization range to ensure that the parameters are within
120 a reasonable range. As shown in Fig.3, we compare the parameter distributions of different linear
121 layers. The initialization scheme based on MFN results in gradient vanishing and small activations in
122 deeper linear layers. In contrast, our initialization scheme ensures that the parameters of each linear
123 layer follow a standard normal distribution.

124 3.2 Multi-Step Pulling (MSP) Module

125 In Fig. 2(b), we demonstrate our idea of learning
126 an accurate implicit function f with multiple
127 frequency features. Given a set of frequency fea-
128 ture Y , we use frequency features y_i in Y as the
129 input with query points Q_i for the MSP module.
130 And we calculate the stride and direction of the
131 query points Q_i at i -th step moving to the target
132 surface point with y_i to predict the current
133 signed distance $f(Q_i, y_i)$ where $i \in [1, N_L - 1]$.
134 Furthermore, we set the direction of the gradient
135 as $\nabla f(Q_i, y_i)$, it represents the fastest in-
136 crease in signed distance in 3D space, pointing
137 away from the surface direction. Therefore,
138 $Q_i = Q_{i-1} - f(Q_i, y_i) \times \nabla f(Q_i, y_i) / \|$
139 $\nabla f(Q_i, y_i) \|_2$. For each step of the query points Q_i , it corresponds to a nearest point q_i on the surface
140 S , and the distance between query points and surface points can be described as $D_i = \|Q_i - q_i\|_2^2$.
141 Above this, we establish the fundamental optimization form between query points Q_i and target
142 points q_i . Therefore, we can naturally obtain the combined loss \mathcal{L}_{pull} under optimal conditions:

$$\mathcal{L}_{pull} = f(Q_1, y_1) + \sum_{i=2}^I D_i, i \in [1, I] \quad (5)$$

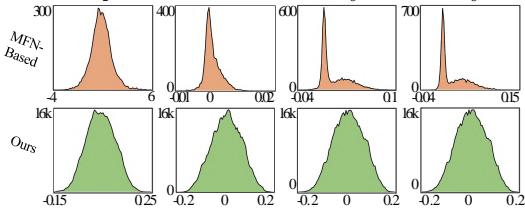


Figure 3: Comparison of parameter distributions of different linear layers especially in (L_2, L_4, L_6, L_8) . We show the different initialization strategies on the results of the reconstruction task and the visualization effects in Appendix B.

143 However, optimizing all query points accurately through this equation alone is challenging when
 144 considering only surface point constraints. This is because Q_i locate at different scales in space,
 145 making it difficult to simultaneously account for the optimization step sizes at different scales with the
 146 inconsistent gradient directions caused by continuous movement. Consequently, some outlier points
 147 may not be effectively optimized. Additionally, for sampling points near target points, some surface
 148 constraints are required to enable the network to accurately predict their corresponding zero-level-set
 149 to avoid optimization errors. Therefore, we will further advance Eq. (5) from the perspectives of
 150 distance constraints, gradient consistency, surface constraints in 3.3, to enhance network performance.

151 **3.3 Loss Function**

152 **Distance-Aware Constraint.** Inspired by FOCAL[50], we design a novel constraint with
 153 distance-aware attention weights α to ensure that the network pays more attention to the optimization
 154 of underfitting query points in space and optimizes the SDFs simultaneously. This allows query
 155 points at different distances from the surface to be optimized properly, and assigns higher attention
 156 weights for outliers, formulated as:

$$\begin{cases} \alpha = \text{softmax}(D_1, D_2), \\ \mathcal{L}_{\text{recon}} = \alpha D_1 + (1 - \alpha)^{\gamma} D_2 + D_3, \end{cases} \quad (6)$$

158 where α is calculated from D_1, D_2 by the softmax activation, γ is a scaling coefficient we set to 2 by
 159 default. Here, we only consider 3 steps, which is a trade-off between performance and efficiency.

160 **Consistent Gradients.** We additionally introduce consistency constraints in the gradient direction.
 161 This loss encourages close level sets to keep parallel, which reduces the artifacts off the surface and
 162 smooths the surface. We add a cosine gradient consistency loss function to encourage the query
 163 points with the same gradient direction with target surface points and improve the continuity of the
 164 gradient during the multi-step pulling. We use Q_1, Q_2 and Q_3 to represent the query points that have
 165 been continuously optimized by the multiple steps. We take the one with the lowest similarity score
 166 to measure the overall similarity.

$$\begin{cases} L_{\nabla}(Q_i) = 1 - \cos(\nabla f(Q_i), \nabla f(q)), \\ \mathcal{L}_{\text{grad}} = 1 - \text{argmin}\{L_{\nabla}(Q_1), L_{\nabla}(Q_2), L_{\nabla}(Q_3)\}, \end{cases} \quad (7)$$

167 where $L_{\nabla}(Q_i)$ represents the loss of cosine similarity between query points Q and target surface
 168 points q .

169 **Surface Constraint.** We introduce the surface constraint for the implicit function f , aiming to assist
 170 the network in approaching the zero-level-set on the surface at final step. Hence, we can constrain the
 171 query points Q on the surface as follow:

$$\mathcal{L}_{\text{surf}} = \|f(Q)\|. \quad (8)$$

172 **Joint Loss Function.** Overall, we learn the SDFs by minimizing the following loss function \mathcal{L} .

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \beta \mathcal{L}_{\text{grad}} + \delta \mathcal{L}_{\text{surf}}, \quad (9)$$

173 where β and δ are balance weights for each term. In the subsequent ablation experiments, we
 174 validated the efficacy of different loss functions.

175 **4 Experiments**

176 In this section, we evaluate the surface reconstruction effectiveness of MultiPull by conducting
 177 numerical and visual comparisons with state-of-the-art methods on both synthetic and real-scan
 178 datasets. Specifically, in Sec. 4.1, we experiment on synthetic shape datasets with diverse topological
 179 structures to validate the effectiveness of our approach. Furthermore, in Sec. 4.2, we verify the
 180 performance of our method across various scales using real large-scale scene datasets. Meanwhile, we
 181 consider FAMOUS as the verification dataset in the ablation experiments to compare the effectiveness
 182 of each module in MultiPull in Sec. 4.3.

183 **4.1 Surface Reconstruction for Shapes**

184 **Datasets and Metrics.** For the single shape surface reconstruction task, we perform evaluation on
 185 multiple datasets including ShapeNet [25], FAMOUS [7], Surface Reconstruction Benchmark (SRB)
 186 [37] Thingi10K [51] and D-FAUST [52]. We conduct validation experiments on 8 subcategories
 187 within the ShapeNet dataset, while the remaining datasets are experimented on the complete dataset.
 188 For metric comparison, we leverage Chamfer Distance CD_{L1} and CD_{L2} , Normal Consistency (NC),
 189 and F-Score as evaluation standards.

190 **ShapeNet.** We evaluate our approach on the ShapeNet[25] according to the experimental settings of
 191 GP [20] . We compare our methods with methods including ATLAS [53], DSDF [43], NP [18], PCP
 192 [54], GP [20], as shown in Tab. 1. We report CD_{L2} , NC and F-Score metrics for ShapeNet, where
 193 we randomly sample 10,000 points on the reconstructed object surface for evaluation. MultiPull
 194 outperforms the state-of-the-art methods. Compare to previous gradient-based methods, which often
 195 suffer from unsMOOTH or missing reconstructions, our method performs better on local details of these
 complex structures. We provide detailed results in Appendix C.

Table 1: Reconstruction accuracy under ShapNet in terms of CD_{L2} , NC and F-Score with thresholds
 of 0.002 and 0.004.

Methods	$CD_{L2} \times 100$	NC	$F\text{-Score}^{0.002}$	$F\text{-Score}^{0.004}$
ATLAS	1.368	0.695	0.062	0.158
DSDF	0.766	0.884	0.212	0.717
NP	0.038	0.939	0.961	0.976
PCP	0.0136	0.9590	0.9871	0.9899
GP	0.0086	0.9723	0.9896	0.9923
Ours	0.0075	0.9737	0.9906	0.9932

196

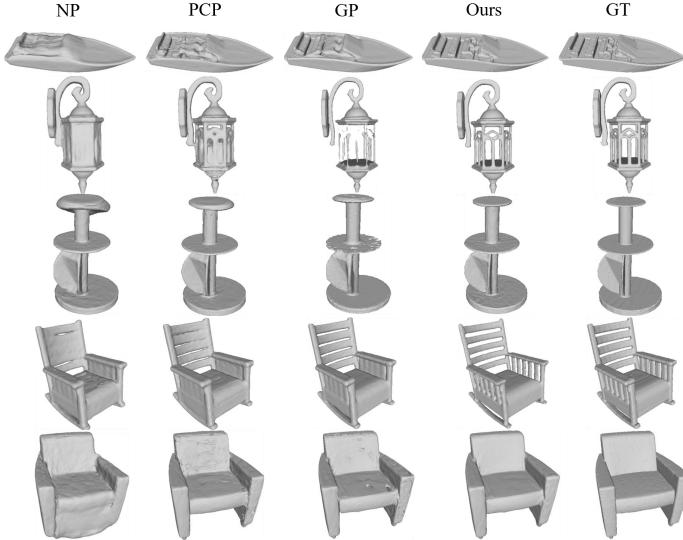


Figure 4: Visual comparison under ShapeNet.

197 **FAMOUS.** We evaluate the performance of our method on the FAMOUS dataset according to the
 198 experimental settings of PCP [54] and NP [18]. Our method demonstrates superiority over recent
 199 approaches, including GP [20], PCP [54], GenSDF [55], FGC [56], NP [18], and IGR [46]. As
 200 shown in Tab. 2, we compare the recent methods using CD_{L2} and NC metrics, and our method
 201 exhibits outstanding performance. To demonstrate the effectiveness of our method in reconstruction
 202 accuracy, we visualize the error-map for comparison in Fig. 4.1. Compare to the the state-of-art
 203 methods, our method has better overall reconstruction accuracy (bluer).

204 **SRB.** We validate our method on the real scanned dataset SRB, following the experimental settings of
 205 VisCo [57] and GP [20]. In Tab. 6, we compare our approach with recent methods including SPSR
 206 [22], IGR [46], P2M [58], SAP [59], NP [18], NDF [60], BACON [12], CAP [44], GP [20]. We

Table 2: Reconstruction accuracy under FAMOUS in terms of CD_{L2} and NC.

Methods	$CD_{L2} \times 100$	NC
IGR	1.65	0.911
GenSDF	0.668	0.909
NP	0.220	0.914
FGC	0.055	0.933
PCP	0.044	0.933
GP	0.040	0.945
Ours	0.035	0.953

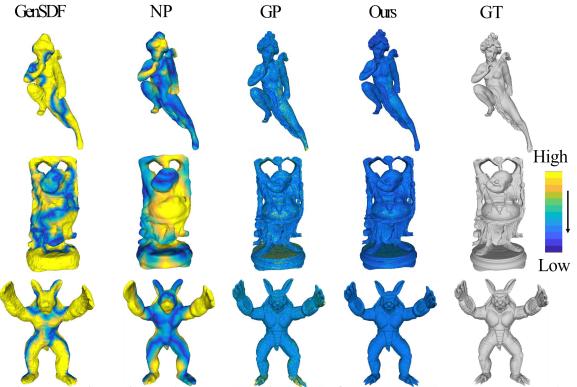


Figure 5: Visual comparison of CD error maps under FAMOUS.

Figure 6: Reconstruction accuracy under SRB in terms of CD_{L1} and F-Score with a threshold of 0.01.

Methods	CD_{L1}	F-Score ^{0.01}
P2M	0.116	64.8
SAP	0.076	83.0
NP	0.106	79.7
BACON	0.089	82.7
CAP	0.073	84.5
GP	0.070	85.1
Ours	0.068	85.7

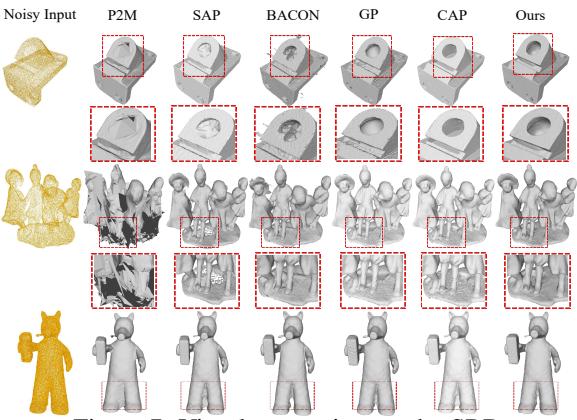


Figure 7: Visual comparison under SRB.

use CD_{L1} and F-Score to evaluate performance , and we surpass all others in terms of these metrics.
As depicted in Fig. 7, our method excels at reconstructing more complete and smoother surfaces.

D-FAUST. We evaluate our method on the D-FAUST dataset with SAP [59] settings. As indicated in Tab. 3, we compare our approach with recent methods including IGR [46], SAP [59], GP [20]. Our method excels in CD_{L1} , F-Score and NC. As illustrated in Fig. 8, compare to other methods, our approach demonstrates superior accuracy in recovering human body shapes.

Table 3: Reconstruction accuracy under D-FAUST in terms of CD_{L1} and F-Score with a threshold of 0.01.

Methods	CD_{L1}	F-Score ^{0.01}	NC
IGR	0.235	0.805	0.911
SAP	0.043	0.966	0.959
GP	0.015	0.975	0.978
Ours	0.009	0.986	0.988

Table 4: Reconstruction accuracy under Thingi10K in terms of CD_{L1} and F-Score with a threshold of 0.01.

Methods	CD_{L1}	F-Score ^{0.01}	NC
IGR	0.440	0.505	0.692
SAP	0.054	0.940	0.947
BACON	0.053	0.946	0.961
GP	0.051	0.948	0.965
Ours	0.048	0.953	0.968

Thingi10K. We assess the performance of our approach on the Thingi10K dataset, following the experimental setup of SAP [59]. We compare our approach with recent methods including IGR [46], SAP [59], BACON [12], GP [20]. As indicated in Tab. 4, our method surpasses existing methods across in CD_{L1} , F-Score and NC metrics. As illustrated in Fig. 9, our method can reconstruct surfaces with more accurate details.

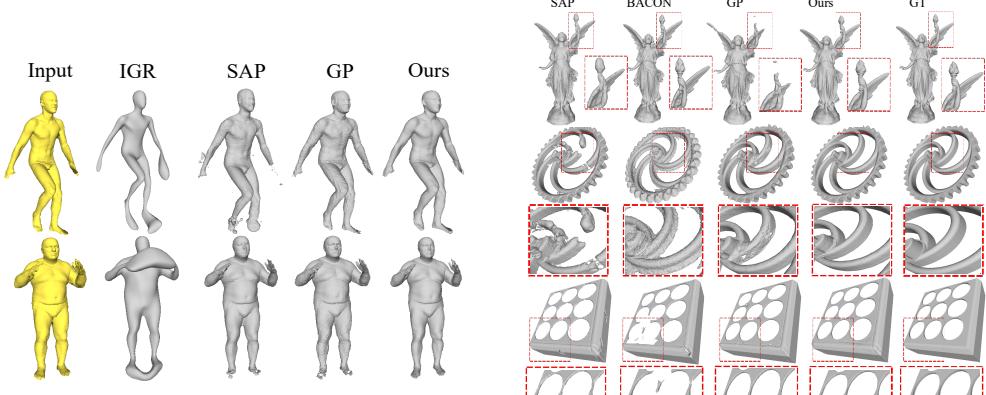


Figure 8: Visual comparison under D-FAUST.

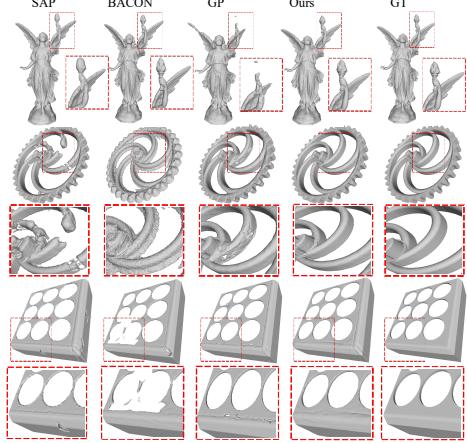


Figure 9: Visual comparison under Thing10K.

218 4.2 Surface Reconstruction for Real-Scan Scenes

219 **Datasets and Metrics.** For the scene reconstruction task, we validate our method on the 3DScene
 220 [61] and KITTI [62] datasets to assess the performance on large-scale datasets. We keep the same
 221 evaluation metrics as those used for shape reconstruction in Sec. 4.1.

222 **3DScene.** In accordance with the experimental settings of PCP [54], we compare our approach with
 223 recent methods including ConvOcc [33], NP [18], PCP [54] and GP [20]. We report the evaluation
 224 results of CD_{L1} , CD_{L2} and NC, and compare our method with the latest approaches listed in Tab. 5.
 225 As illustrated in Fig. 10, our method outperforms prior-based methods and overfitting based methods.
 226 We will present a more detailed comparison of individual scene metrics and visual comparisons in
 227 Appendix C.

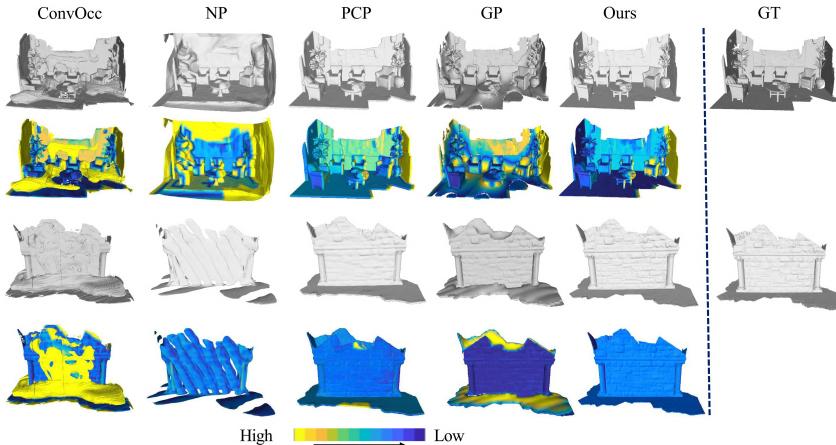


Figure 10: Visual comparison of CD error maps under 3DScene.

Table 5: Reconstruction accuracy under 3DScene in terms of CD_{L1} , CD_{L2} and NC.

Methods	$CD_{L2} \times 100$	CD_{L1}	NC
ConvOcc	13.32	0.049	0.752
NP	8.350	0.0194	0.713
PCP	0.11	0.007	0.886
GP	0.10	0.006	0.903
Ours	0.094	0.006	0.918

228 **KITTI.** We validate our method on a large-scale scanned point cloud dataset KITTI [62], which
 229 contains 13.8 million points. We will provide the complete scene reconstruction results in Appendix
 230 C.

231 **4.3 Ablation Experiments**

232 **Effect of Frequency Layers.** We denote the j -th layer of the frequency network as L_j , a specific
 233 combination of frequency feature layers can be formulated as $\{L_i, L_j, L_k\}$, where $\{i, j, k\} \in$
 234 $[1, N_L - 1]$. We evaluate the effectiveness of the frequency transformer layers in Tab. 6 with CD_{L2}
 235 and NC, replacing the frequency network with linear layers results in a decrease in the performance
 236 of the CD_{L2} and NC metrics. The performance of using only one layer(L_4) surpasses linear layers.
 237 With the increase of the frequency layers, $\{L_4, L_6, L_8\}$ produces best results.

Table 6: Effect of frequency features.

Layer	$CD_{L2} \times 100$	NC
<i>Linear</i>	0.042	0.920
L_4	0.040	0.926
L_4, L_6	0.037	0.933
L_4, L_6, L_8	0.036	0.948

238 **Effect of MSP Module.** We report comparisons with different features in Tab. 7. The 'Layer'
 239 column denotes the combination of frequency features obtained by the FFT module. For instance,
 240 $\{L_4, L_6, L_8\}$ represent the frequency features from the 4th, 6th, and 8th layers guiding the pulling of
 241 the query point in the MSP network, respectively. We find that the accuracy of the network increases
 242 with the number of steps. After considering both performance metrics and time efficiency, we have
 243 set Step=3 by default.

Table 7: Effect of MSP Module.

Step	$CD_{L2} \times 100$	NC
1	0.040	0.926
2	0.037	0.933
3	0.036	0.948
4	0.036	0.942
5	0.357	0.955

244 **Effect of Loss Functions.** We compare CD_{L2} metric under different loss strategies in Tab. 8. As
 245 shown in the table, Weighting query points at different scales effectively enhances reconstruction
 246 accuracy and the reconstruction loss allows the network to obtain a complete shape with local details.
 247 Furthermore, The gradient loss improves the surface continuity of the object. And the surface
 248 supervision loss facilitates the learning of more precise zero-level sets, which also improves the
 249 accuracy.

Table 8: Effect of loss functions.

Loss	$CD_{L2} \times 100$	NC
\mathcal{L}_{pull}	0.0443	0.937
\mathcal{L}_{recon}	0.0383	0.946
$\mathcal{L}_{recon} + \mathcal{L}_{sim}$	0.0367	0.948
$\mathcal{L}_{recon} + \mathcal{L}_{sim} + \mathcal{L}_{sdf}$	0.0352	0.954

250 **5 Conclusion**

251 We propose a novel method to learn detailed SDFs by pulling multi-level queries onto the surface
 252 at multi-step. We leverage the multi-level features to predict signed distances, which recovers high
 253 frequency details. Through optimization, our method is able to gradually restore the coarse-to-fine
 254 structure of reconstructed objects, thereby generating accurate 3D models. Compared to supervised
 255 methods, our approach still demonstrates competitive performance in our evaluation.

256 **References**

- 257 [1] Matthew Tancik, Vincent Casser, Xincheng Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan,
258 Jonathan T Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In
259 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258,
260 2022.
- 261 [2] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and
262 radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference
263 on Computer Vision*, pages 5589–5599, 2021.
- 264 [3] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. Drwr: A differentiable renderer without
265 rendering for unsupervised 3D structure learning from silhouette images. In *International Conference on
266 Machine Learning*, 2020.
- 267 [4] Chao Chen, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Unsupervised learning of fine structure
268 generation for 3D point clouds by 2D projections matching. In *Proceedings of the IEEE/CVF International
269 Conference on Computer Vision*, pages 12466–12477, 2021.
- 270 [5] Junsheng Zhou, Baorui Ma, Shujian Li, Yu-Shen Liu, and Zhizhong Han. Learning a more continuous
271 zero level set in unsigned distance fields through level set projection. In *Proceedings of the IEEE/CVF
272 International Conference on Computer Vision*, 2023.
- 273 [6] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction
274 algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169, 1987.
- 275 [7] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2Surf
276 learning implicit surfaces from point clouds. In *European Conference on Computer Vision*, pages 108–124.
277 Springer, 2020.
- 278 [8] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser,
279 et al. Local implicit grid representations for 3D scenes. In *Proceedings of the IEEE/CVF Conference on
280 Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.
- 281 [9] Julien NP Martel, David B Lindell, Connor Z Lin, Eric R Chan, Marco Monteiro, and Gordon Wetzstein.
282 ACORN: adaptive coordinate networks for neural scene representation. *ACM Transactions on Graphics
283 (TOG)*, 40(4):1–13, 2021.
- 284 [10] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. Curriculum deepsdf. In
285 *European Conference on Computer Vision*, pages 51–67. Springer, 2020.
- 286 [11] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In
287 *International Conference on Learning Representations*, 2020.
- 288 [12] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. BACON: Band-limited
289 coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF Conference on
290 Computer Vision and Pattern Recognition*, pages 16252–16262, 2022.
- 291 [13] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai,
292 Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering
293 with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
294 Recognition*, pages 11358–11367, 2021.
- 295 [14] Yishun Dou, Zhong Zheng, Qiaoqiao Jin, and Bingbing Ni. Multiplicative fourier level of detail. In
296 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1808–1817,
297 2023.
- 298 [15] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proceedings
299 of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020.
- 300 [16] Matan Atzmon and Yaron Lipman. SALD: Sign agnostic learning with derivatives. In *International
301 Conference on Learning Representations*, 2020.
- 302 [17] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit
303 functions for 3D shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
304 Recognition*, pages 4857–4866, 2020.
- 305 [18] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-Pull: Learning signed distance
306 function from point clouds by learning to pull space onto surface. In *International Conference on Machine
307 Learning*, pages 7246–7257. PMLR, 2021.

- 308 [19] Baorui Ma, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Towards better gradient consistency for
 309 neural signed distance functions via level set alignment. In *Proceedings of the IEEE/CVF Conference on*
 310 *Computer Vision and Pattern Recognition*, pages 17724–17734, 2023.
- 311 [20] Chao Chen, Yu-Shen Liu, and Zhizhong Han. GridPull: Towards scalability in learning implicit represen-
 312 tations from 3D point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer*
 313 *Vision*, pages 18322–18334, 2023.
- 314 [21] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-
 315 pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*,
 316 5(4):349–359, 1999.
- 317 [22] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on*
 318 *Graphics (TOG)*, 32(3):1–13, 2013.
- 319 [23] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface repre-
 320 sentations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages
 321 2916–2925, 2018.
- 322 [24] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition
 323 of unity implicits. *ACM Transactions on Graphics*, 22(3):463–470, 2003.
- 324 [25] ShapeNet. <https://www.shapenet.org/>.
- 325 [26] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev,
 326 Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning.
 327 In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611,
 328 2019.
- 329 [27] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. Sharf: Shape-conditioned radiance
 330 fields from a single view. In *International Conference on Machine Learning*, 2021.
- 331 [28] Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Learning signed distance functions from noisy 3d point
 332 clouds via noise to noise mapping. In *Advances in Neural Information Processing Systems (NeurIPS)*,
 333 2023.
- 334 [29] Shujuan Li, Junsheng Zhou, Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Neaf: Learning neural angle
 335 fields for point normal estimation. In *Proceedings of the AAAI conference on artificial intelligence*,
 336 volume 37, pages 1396–1404, 2023.
- 337 [30] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint*
 338 *arXiv:2305.02463*, 2023.
- 339 [31] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing
 340 instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,
 341 2023.
- 342 [32] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-
 343 Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer*
 344 *Vision and Pattern Recognition*, 2023.
- 345 [33] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional
 346 occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020.
- 347 [34] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3D
 348 supervision. In *Advances in Neural Information Processing Systems*, 2019.
- 349 [35] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In
 350 *Advances in Neural Information Processing Systems*, 2021.
- 351 [36] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman.
 352 Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural*
 353 *Information Processing Systems*, 33, 2020.
- 354 [37] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep
 355 geometric prior for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer*
 356 *Vision and Pattern Recognition*, pages 10130–10139, 2019.

- 357 [38] Minghua Liu, Xiaoshuai Zhang, and Hao Su. Meshing point clouds with predicted intrinsic-extrinsic ratio
 358 guidance. In *European Conference on Computer vision*, 2020.
- 359 [39] Zhenxing Mi, Yiming Luo, and Wenbing Tao. SSRNet: Scalable 3D surface reconstruction network. In
 360 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 970–979,
 361 2020.
- 362 [40] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Improved direct voxel grid optimization for radiance fields
 363 reconstruction. *arXiv preprint arXiv:2206.05085*, 2022.
- 364 [41] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox. Orientation-boosted voxel nets for 3D object recognition.
 365 In *British Machine Vision Conference*, 2017.
- 366 [42] Hai Li, Xingrui Yang, Hongjia Zhai, Yuqian Liu, Hujun Bao, and Guofeng Zhang. Vox-Surf: Voxel-based
 367 implicit surface representation. *CorR*, abs/2208.10925, 2022.
- 368 [43] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF:
 369 Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF*
 370 *Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.
- 371 [44] Junsheng Zhou, Baorui Ma, Liu Yu-Shen, Fang Yi, and Han Zhizhong. Learning consistency-aware
 372 unsigned distance functions progressively from raw point clouds. In *Advances in Neural Information
 373 Processing Systems (NeurIPS)*, 2022.
- 374 [45] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as
 375 points: A differentiable poisson solver. *Advances in Neural Information Processing Systems*, 34, 2021.
- 376 [46] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization
 377 for learning shapes. In *International Conference on Machine Learning*, pages 3789–3799. PMLR, 2020.
- 378 [47] Wenbin Zhao, Jiabao Lei, Yuxin Wen, Jianguo Zhang, and Kui Jia. Sign-agnostic implicit learning of
 379 surface self-similarities for shape modeling and reconstruction from raw point clouds. In *Proceedings of*
 380 *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10256–10265, 2021.
- 381 [48] Yu-Tao Liu, Li Wang, Jie Yang, Weikai Chen, Xiaoxu Meng, Bo Yang, and Lin Gao. Neudf: Leaning
 382 neural unsigned distance fields with volume rendering. In *Proceedings of the IEEE/CVF Conference on*
 383 *Computer Vision and Pattern Recognition*, pages 237–247, 2023.
- 384 [49] Danhang Tang, Mingsong Dou, Peter Lincoln, Philip Davidson, Kaiwen Guo, Jonathan Taylor, Sean
 385 Fanello, Cem Keskin, Adarsh Kowdle, Sofien Bouaziz, et al. Real-time compression and streaming of 4d
 386 performances. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.
- 387 [50] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object
 388 detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988,
 389 2017.
- 390 [51] Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint*
 391 *arXiv:1605.04797*, 2016.
- 392 [52] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic FAUST: Registering
 393 human bodies in motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern*
 394 *Recognition*, pages 6233–6242, 2017.
- 395 [53] Qian Yu, Chengzhan Yang, and Hui Wei. Part-wise atlasnet for 3d point cloud reconstruction from a
 396 single image. *Knowledge-Based Systems*, 242:108395, 2022.
- 397 [54] Baorui Ma, Yu-Shen Liu, Matthias Zwicker, and Zhizhong Han. Surface reconstruction from point clouds
 398 by learning predictive context priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision*
 399 *and Pattern Recognition*, 2022.
- 400 [55] Gene Chou, Ilya Chugunov, and Felix Heide. GenSDF: Two-stage learning of generalizable signed distance
 401 functions. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2022.
- 402 [56] Mulin Yu and Florent Lafarge. Finding good configurations of planar primitives in unorganized point
 403 clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*,
 404 pages 6367–6376, June 2022.
- 405 [57] Albert Pumarola, Artsiom Sanakoyeu, Lior Yariv, Ali Thabet, and Yaron Lipman. Visco grids: Surface
 406 reconstruction with viscosity and coarea grids. In *Advances in Neural Information Processing Systems*,
 407 2022.

- 408 [58] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2Mesh: a self-prior for deformable
409 meshes. *ACM Transactions on Graphics (TOG)*, 39(4):126–1, 2020.
- 410 [59] Songyou Peng, Chiyu “Max” Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger.
411 Shape as points: A differentiable poisson solver. In *Advances in Neural Information Processing Systems*,
412 2021.
- 413 [60] Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning.
414 *Advances in Neural Information Processing Systems*, 33:21638–21652, 2020.
- 415 [61] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions
416 on Graphics (TOG)*, 32(4):1–8, 2013.
- 417 [62] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision
418 benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361.
419 IEEE, 2012.
- 420 [63] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives
421 with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

422 **A Implementation Details.**

423 Our network consists of two main parts: frequency feature transformer and multi-step pulling modules
 424 in Fig 2 (a) and (b), respectively. For frequency feature transformer, we transform the raw point
 425 cloud into frequency features of size $N \times D$, where D is initialized to 256. Same for the multi-step
 426 pulling module, we train a linear sequence neural network (**LSNN**) with shared parameters and we
 427 fix intermediate layer output dimension to 512. In the construction of query points, we establish the
 428 corresponding pairs between query points and their nearest points on surfaces. Specifically, for an
 429 input point cloud, we construct 40 queries for each point of the point cloud, following NeuralPull [18].
 430 The construction of these query points follows a Gaussian distribution. During the reconstruction
 431 process, we use the Marching Cubes algorithm [23] to extract the mesh surface.

432 During the training process, we default to 40,000 iterations, with an average time of 28 minutes for
 433 single-object reconstruction. We uniformly utilize a single NVIDIA RTX 3090 GPU for both training
 434 and testing.

435 **B Additional Experiments**

436 **Effect of Frequency Features.** To further validate the superiority of frequency features, we exclude
 437 multi-step pulling, and only use single-layer frequency feature for performance verification against
 438 linear layers. We note the frequency feature in the i -th layer as L_i . We compare the CD_{L_2} and NC
 439 of specific layers(L_2, L_4, L_6, L_8) with the linear layers. As shown in Tab. 9 the performance of
 440 frequency features at different layers is superior to the linear layers, and with an increase in the
 441 number of layers, higher-level frequency conditions enhance the network’s performance.

Table 9: Effect of frequency features.

Layer	$CD_{L_2} \times 100$	NC
<i>Linear</i>	0.042	0.920
L_4	0.040	0.926
L_6	0.038	0.931
L_8	0.037	0.935

442 **Effect of Different Levels of Noise.** We evaluate the reconstruction performance of our method on
 443 the Famous dataset under two levels of noise: Mid-Level and Max-Level noise. As shown in Tab. 10,
 444 our method outperforms the majority of approaches even in the presence of noisy inputs.

Table 10: Effect of different levels of noise.

Noise level	NP	PCP	GP	Ours
Mid-Noise	0.280	0.071	0.044	0.037
Max-Noise	0.310	0.298	0.060	0.058

445 **Effect of Initialization Strategies.** We compare our initialization strategy with random initialization
 446 and MFN-based method (BACON [12]) as example. We compare the metrics of these initialization
 447 methods in Tab. 11, which shows that combining random or BACON initialization with our approach
 448 does not yield satisfactory results. To further demonstrate the advantages of our initialization method,
 449 we visually compared SDF with random initialization and BACON initialization strategies. As
 450 shown in the Fig. 11, our method significantly outperforms other initialization methods in terms of
 451 convergence speed. In addition, our reconstruction results also indicate that a reasonable initialization
 452 method can enable the network to learn more accurate signed distance field. In addition, we compare
 453 the results with the same iterations and the final results under the default settings for different methods
 454 (Final) in Fig. 12.

455 **Effect of Parameters on Networks.** We compare the parameter quantities of the methods listed
 456 in Tab. 12 below. It shows that the parameter number of PCP is the largest among all the three
 457 methods, while NP has the least parameters. To further investigate the performance of networks with
 458 the similar amount of parameters, we increase the parameters of NP and MultiPull to match PCP.
 459 The comparison in the Tab. 13 indicates that both NP(NeuralPull) and MultiPull show the improved

Table 11: Effect of initialization strategies.

Initialization	$CD_{L2} \times 100$	NC
Random	0.042	0.938
BACON	0.038	0.946
Ours	0.035	0.950

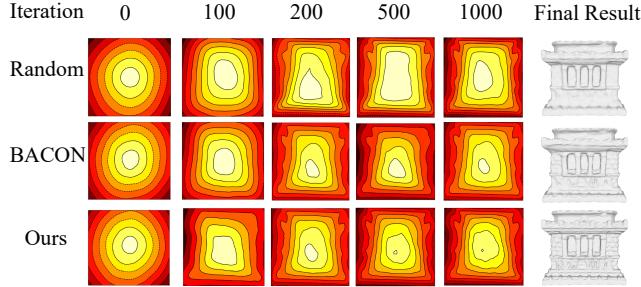


Figure 11: Comparison of signed distances in optimization

460 performance. This demonstrates that more parameters are beneficial to improve the performance, but
461 our performance comes from our novel methods rather than more parameters.

Table 12: Comparison at different parameter levels.

Method	Parameters	$CD_{L2} \times 100$	NC
NP	1843708	0.220	0.914
PCP	7894022	0.044	0.933
Ours	2648094	0.035	0.953

Table 13: Comparison at a uniform parameter levels.

Method	Parameters	$CD_{L2} \times 100$	NC
NP	7907441	0.1966	0.917
PCP	7894022	0.044	0.933
Ours	7856620	0.0317	0.957

462 C Additional Results

463 **Comparison Details for ShapeNet.** The complete comparison under all the five scenes of the
464 ShapNet dataset. The results are shown in Tab. 14 to Tab.16. We use Chamfer Distance (CD_{L1} , CD_{L2})
and NC as evaluation metrics.

Table 14: Reconstruction accuracy under ShapeNet in terms of CD_{L2}

Class	ATLAS	DSDF	NP	PCP	GP	Ours
Display	1.094	0.317	0.039	0.0087	0.0082	0.0074
Lamp	1.988	0.955	0.080	0.0380	0.0347	0.0301
Airplane	1.011	1.043	0.008	0.0065	0.0007	0.0006
Cabinet	1.611	0.921	0.026	0.0153	0.0112	0.0105
Vessel	0.997	1.254	0.022	0.0079	0.0033	0.0028
Table	1.311	0.660	0.060	0.0131	0.0052	0.0047
Chair	1.575	0.483	0.054	0.0110	0.0043	0.0036
Sofa	1.307	0.496	0.012	0.0086	0.0015	0.0009
Mean	1.368	0.766	0.038	0.0136	0.0086	0.0075

465

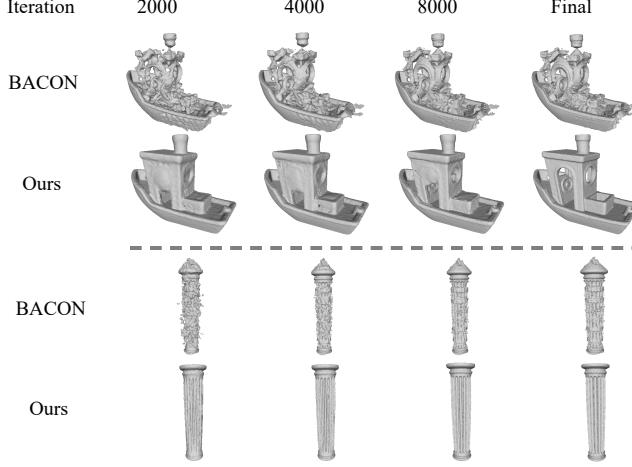


Figure 12: Comparison of the initialization strategies in optimization.

Table 15: Reconstruction accuracy under ShapeNet in terms of NC.

Class	ATLAS	DSDF	NP	PCP	GP	Ours
Display	0.828	0.932	0.964	0.9775	0.9847	0.9855
Lamp	0.593	0.864	0.930	0.9450	0.9693	0.9710
Airplane	0.737	0.872	0.947	0.9490	0.9614	0.9633
Cabinet	0.682	0.872	0.930	0.9600	0.9689	0.9693
Vessel	0.671	0.841	0.941	0.9546	0.9667	0.9671
Table	0.783	0.901	0.908	0.9595	0.9755	0.9758
Chair	0.638	0.886	0.937	0.9580	0.9733	0.9757
Sofa	0.633	0.906	0.951	0.9680	0.9792	0.9822
Mean	0.695	0.884	0.939	0.9590	0.9723	0.9737

Table 16: Reconstruction accuracy under ShapeNet in terms of F- Score with thresholds of 0.002 and 0.004.

	F-Score ^{0.002}						F-Score ^{0.004}					
	ATLAS	DSDF	NP	PCP	GP	Ours	ATLAS	DSDF	NP	PCP	GP	Ours
Display	0.071	0.632	0.989	0.9939	0.9963	0.9966	0.179	0.787	0.991	0.9958	0.9963	0.9980
Lamp	0.029	0.268	0.891	0.9382	0.9455	0.9473	0.077	0.478	0.924	0.9402	0.9538	0.9561
Airplane	0.070	0.350	0.996	0.9942	0.9976	0.9989	0.179	0.566	0.997	0.9972	0.9989	0.9991
Cabinet	0.077	0.573	0.980	0.9888	0.9901	0.9913	0.195	0.694	0.989	0.9939	0.9946	0.9969
Vessel	0.058	0.323	0.985	0.9935	0.9956	0.9962	0.153	0.509	0.990	0.9958	0.9972	0.9979
Table	0.080	0.577	0.922	0.9969	0.9977	0.9982	0.195	0.743	0.973	0.9958	0.9990	0.9988
Chair	0.050	0.447	0.954	0.9970	0.9979	0.9980	0.134	0.665	0.969	0.9991	0.9990	0.9994
Sofa	0.058	0.577	0.968	0.9943	0.9974	0.9981	0.153	0.734	0.974	0.9987	0.9992	0.9992
Mean	0.062	0.212	0.961	0.9871	0.9896	0.9906	0.158	0.717	0.976	0.9899	0.9923	0.9932

466 **Comparison Details for 3DScene.** We also provide detailed metrics for single scenes in 3DScene
467 dataset. We evaluate it by CD_{L1} , CD_{L2} and NC. As shown in the Tab. 17, our approach achieves the
468 best performance across all scenes.

Table 17: CD_{L1} , $CD_{L2} \times 100$ and NC comparison under 3DScene.

		ConvOcc	NP	PCP	GP	Ours
Burghers	$CD_{L2} \times 100$	26.69	1.76	0.267	0.246	0.228
	CD_{L1}	0.077	0.010	0.008	0.008	0.008
	NC	0.865	0.883	0.914	0.926	0.934
Lounge	$CD_{L2} \times 100$	8.68	39.71	0.061	0.055	0.048
	CD_{L1}	0.042	0.059	0.006	0.005	0.005
	NC	0.857	0.857	0.928	0.922	0.949
Copyroom	$CD_{L2} \times 100$	10.99	0.051	0.076	0.069	0.064
	CD_{L1}	0.045	0.011	0.007	0.006	0.006
	NC	0.848	0.884	0.918	0.929	0.923
Stonewall	$CD_{L2} \times 100$	19.12	0.063	0.061	0.058	0.043
	CD_{L1}	0.066	0.007	0.0065	0.006	0.005
	NC	0.866	0.868	0.888	0.893	0.936
Totepole	$CD_{L2} \times 100$	1.16	0.19	0.10	0.093	0.087
	CD_{L1}	0.016	0.010	0.008	0.007	0.007
	NC	0.325	0.765	0.784	0.847	0.851

469 **Comparison Details for KITTI.** As shown in Fig. 13, our approach is capable of reconstructing
 470 more complete and accurate surfaces compared to GP [20]. GP fails to reconstruct more continuous
 471 surfaces such as walls and streets. In contrast, our method achieves complete reconstruction of objects
 472 at various scales in large scenes.

473 **Computational Complexity.** We report our computational complexity in Tab. 18, we present
 474 numerical comparisons with the latest overfitting-based methods, including NP and PCP, using
 475 different point counts, such as 20K and 40K. The benchmark rounds for both NP and PCP are set
 476 at 40K. NP does not require learning priors, resulting in the highest operational efficiency. PCP
 477 needs to learn priors, which requires additional time. To achieve more refined results, we dedicate
 478 extra time to learning the frequency features of point clouds and computing the sampling point
 479 strides. Consequently, our speed is slower compared to NP. However, it is noteworthy that our method
 480 outperforms PCP and operates faster even without using local priors.

Table 18: Comparison of computational complexity.

Time/GPU Memory	20K	40K
NP	12min/1.5G	15min/2.3G
PCP	14min/1.9G	18min/2.7G
Ours	13min/1.8G	16min/2.5G

481 D Limitation

482 We propose a method that approximates the accurate signed distance field through multi-step opti-
 483 mization, achieving more precise results. However, there is still room for further optimization in
 484 terms of time and computational efficiency as shown in Tab. 7 and Tab. 13. In future work, we will
 485 continue to explore how to integrate multi-resolution (such as NGLOD [13] and Instant-NGP [63])
 486 features effectively to balance computational efficiency and accuracy.

487 E Video

488 We provide a video containing the visualization of optimization process, the visualization under
 489 synthetic and real-captured datasets as a part of our supplementary materials.

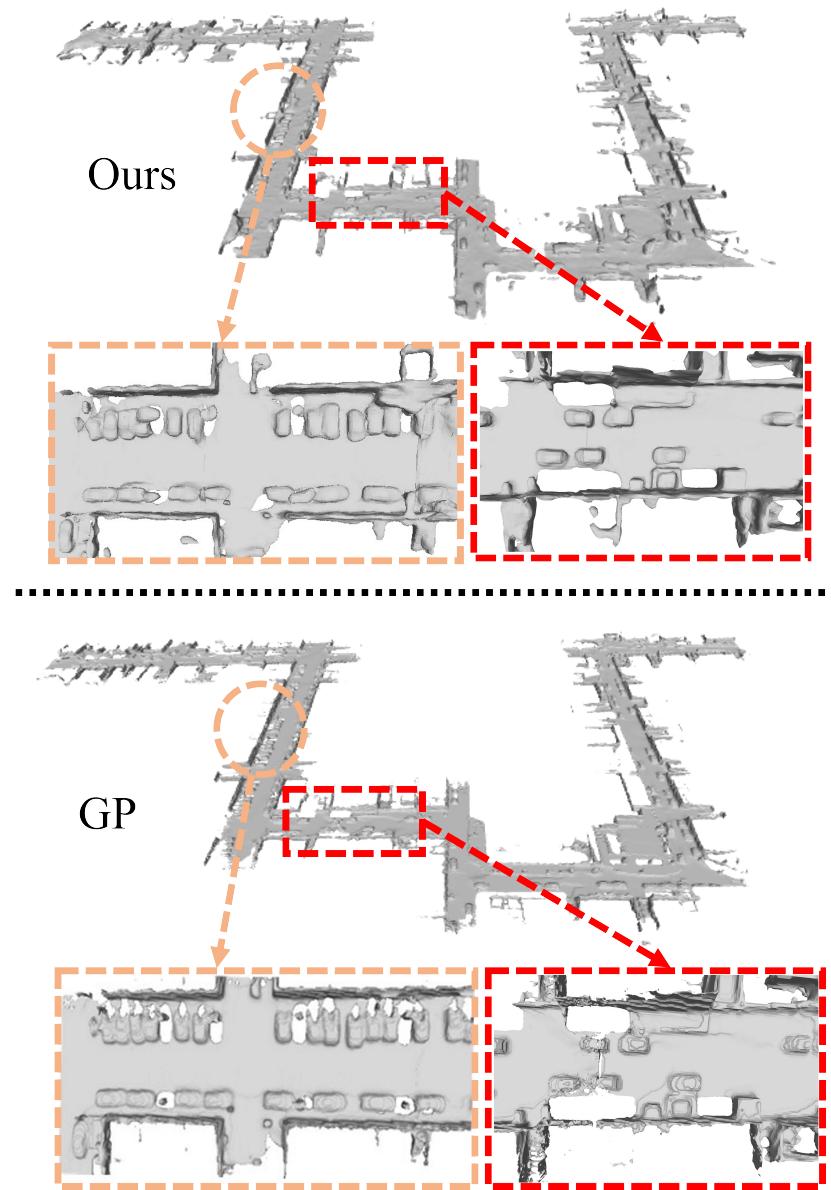


Figure 13: Visual comparison under KITTI.

490 **F Code**

491 We provide our demonstration code as a part of our supplementary materials. We will release the
492 source code, data and instructions upon acceptance.

493 **NeurIPS Paper Checklist**

494 **1. Claims**

495 Question: Do the main claims made in the abstract and introduction accurately reflect the
496 paper's contributions and scope?

497 Answer: [Yes]

498 Justification: Our main claims made in the abstract and introduction accurately reflect the
499 paper's contributions and scope.

500 Guidelines:

- 501 • The answer NA means that the abstract and introduction do not include the claims
502 made in the paper.
- 503 • The abstract and/or introduction should clearly state the claims made, including the
504 contributions made in the paper and important assumptions and limitations. A No or
505 NA answer to this question will not be perceived well by the reviewers.
- 506 • The claims made should match theoretical and experimental results, and reflect how
507 much the results can be expected to generalize to other settings.
- 508 • It is fine to include aspirational goals as motivation as long as it is clear that these goals
509 are not attained by the paper.

510 **2. Limitations**

511 Question: Does the paper discuss the limitations of the work performed by the authors?

512 Answer: [Yes]

513 Justification: We analyze the limitations of our method in the Appendix. D

514 Guidelines:

- 515 • The answer NA means that the paper has no limitation while the answer No means that
516 the paper has limitations, but those are not discussed in the paper.
- 517 • The authors are encouraged to create a separate "Limitations" section in their paper.
- 518 • The paper should point out any strong assumptions and how robust the results are to
519 violations of these assumptions (e.g., independence assumptions, noiseless settings,
520 model well-specification, asymptotic approximations only holding locally). The authors
521 should reflect on how these assumptions might be violated in practice and what the
522 implications would be.
- 523 • The authors should reflect on the scope of the claims made, e.g., if the approach was
524 only tested on a few datasets or with a few runs. In general, empirical results often
525 depend on implicit assumptions, which should be articulated.
- 526 • The authors should reflect on the factors that influence the performance of the approach.
527 For example, a facial recognition algorithm may perform poorly when image resolution
528 is low or images are taken in low lighting. Or a speech-to-text system might not be
529 used reliably to provide closed captions for online lectures because it fails to handle
530 technical jargon.
- 531 • The authors should discuss the computational efficiency of the proposed algorithms
532 and how they scale with dataset size.
- 533 • If applicable, the authors should discuss possible limitations of their approach to
534 address problems of privacy and fairness.
- 535 • While the authors might fear that complete honesty about limitations might be used by
536 reviewers as grounds for rejection, a worse outcome might be that reviewers discover
537 limitations that aren't acknowledged in the paper. The authors should use their best
538 judgment and recognize that individual actions in favor of transparency play an impor-
539 tant role in developing norms that preserve the integrity of the community. Reviewers
540 will be specifically instructed to not penalize honesty concerning limitations.

541 **3. Theory Assumptions and Proofs**

542 Question: For each theoretical result, does the paper provide the full set of assumptions and
543 a complete (and correct) proof?

544 Answer: [NA]

545 Justification: The paper does not include theoretical results. method in the main paper.

546 Guidelines:

- 547 • The answer NA means that the paper does not include theoretical results.
- 548 • All the theorems, formulas, and proofs in the paper should be numbered and cross-
- 549 • referenced.
- 550 • All assumptions should be clearly stated or referenced in the statement of any theorems.
- 551 • The proofs can either appear in the main paper or the supplemental material, but if
- 552 • they appear in the supplemental material, the authors are encouraged to provide a short
- 553 • proof sketch to provide intuition.
- 554 • Inversely, any informal proof provided in the core of the paper should be complemented
- 555 • by formal proofs provided in appendix or supplemental material.
- 556 • Theorems and Lemmas that the proof relies upon should be properly referenced.

557 4. Experimental Result Reproducibility

558 Question: Does the paper fully disclose all the information needed to reproduce the main ex-

559 perimental results of the paper to the extent that it affects the main claims and/or conclusions

560 of the paper (regardless of whether the code and data are provided or not)?

561 Answer: [Yes]

562 Justification: We provide the detailed information in reproducing our methods in Sec. 3-4 of

563 the main paper and the appendix. We also provide a demonstration code of our method in

564 the supplementary materials.

565 Guidelines:

- 566 • The answer NA means that the paper does not include experiments.
- 567 • If the paper includes experiments, a No answer to this question will not be perceived
- 568 • well by the reviewers: Making the paper reproducible is important, regardless of
- 569 • whether the code and data are provided or not.
- 570 • If the contribution is a dataset and/or model, the authors should describe the steps taken
- 571 • to make their results reproducible or verifiable.
- 572 • Depending on the contribution, reproducibility can be accomplished in various ways.
- 573 • For example, if the contribution is a novel architecture, describing the architecture fully
- 574 • might suffice, or if the contribution is a specific model and empirical evaluation, it may
- 575 • be necessary to either make it possible for others to replicate the model with the same
- 576 • dataset, or provide access to the model. In general, releasing code and data is often
- 577 • one good way to accomplish this, but reproducibility can also be provided via detailed
- 578 • instructions for how to replicate the results, access to a hosted model (e.g., in the case
- 579 • of a large language model), releasing of a model checkpoint, or other means that are
- 580 • appropriate to the research performed.
- 581 • While NeurIPS does not require releasing code, the conference does require all submis-
- 582 • sions to provide some reasonable avenue for reproducibility, which may depend on the
- 583 • nature of the contribution. For example
 - 584 (a) If the contribution is primarily a new algorithm, the paper should make it clear how
 - 585 • to reproduce that algorithm.
 - 586 (b) If the contribution is primarily a new model architecture, the paper should describe
 - 587 • the architecture clearly and fully.
 - 588 (c) If the contribution is a new model (e.g., a large language model), then there should
 - 589 • either be a way to access this model for reproducing the results or a way to reproduce
 - 590 • the model (e.g., with an open-source dataset or instructions for how to construct
 - 591 • the dataset).
 - 592 (d) We recognize that reproducibility may be tricky in some cases, in which case
 - 593 • authors are welcome to describe the particular way they provide for reproducibility.
 - 594 • In the case of closed-source models, it may be that access to the model is limited in
 - 595 • some way (e.g., to registered users), but it should be possible for other researchers
 - 596 • to have some path to reproducing or verifying the results.

597 5. Open access to data and code

598 Question: Does the paper provide open access to the data and code, with sufficient instruc-
599 tions to faithfully reproduce the main experimental results, as described in supplemental
600 material?

601 Answer: [Yes]

602 Justification: We provide our demonstration code as a part of our supplementary materials.
603 We will release the source code, data and instructions upon acceptance.

604 Guidelines:

- 605 • The answer NA means that paper does not include experiments requiring code.
- 606 • Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- 607 • While we encourage the release of code and data, we understand that this might not be
608 possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not
609 including code, unless this is central to the contribution (e.g., for a new open-source
610 benchmark).
- 611 • The instructions should contain the exact command and environment needed to run to
612 reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- 613 • The authors should provide instructions on data access and preparation, including how
614 to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- 615 • The authors should provide scripts to reproduce all experimental results for the new
616 proposed method and baselines. If only a subset of experiments are reproducible, they
617 should state which ones are omitted from the script and why.
- 618 • At submission time, to preserve anonymity, the authors should release anonymized
619 versions (if applicable).
- 620 • Providing as much information as possible in supplemental material (appended to the
621 paper) is recommended, but including URLs to data and code is permitted.

624 6. Experimental Setting/Details

625 Question: Does the paper specify all the training and test details (e.g., data splits, hyper-
626 parameters, how they were chosen, type of optimizer, etc.) necessary to understand the
627 results?

628 Answer: [Yes]

629 Justification: We provide the training and testing details in the experiment section (Sec.4)
630 and the Appendix .A.

631 Guidelines:

- 632 • The answer NA means that the paper does not include experiments.
- 633 • The experimental setting should be presented in the core of the paper to a level of detail
634 that is necessary to appreciate the results and make sense of them.
- 635 • The full details can be provided either with the code, in appendix, or as supplemental
636 material.

637 7. Experiment Statistical Significance

638 Question: Does the paper report error bars suitably and correctly defined or other appropriate
639 information about the statistical significance of the experiments?

640 Answer: [No]

641 Justification: We report the average performance as the experimental results.

642 Guidelines:

- 643 • The answer NA means that the paper does not include experiments.
- 644 • The authors should answer “Yes” if the results are accompanied by error bars, confi-
645 dence intervals, or statistical significance tests, at least for the experiments that support
646 the main claims of the paper.
- 647 • The factors of variability that the error bars are capturing should be clearly stated (for
648 example, train/test split, initialization, random drawing of some parameter, or overall
649 run with given experimental conditions).

- 650 • The method for calculating the error bars should be explained (closed form formula,
 651 call to a library function, bootstrap, etc.)
 652 • The assumptions made should be given (e.g., Normally distributed errors).
 653 • It should be clear whether the error bar is the standard deviation or the standard error
 654 of the mean.
 655 • It is OK to report 1-sigma error bars, but one should state it. The authors should
 656 preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis
 657 of Normality of errors is not verified.
 658 • For asymmetric distributions, the authors should be careful not to show in tables or
 659 figures symmetric error bars that would yield results that are out of range (e.g. negative
 660 error rates).
 661 • If error bars are reported in tables or plots, The authors should explain in the text how
 662 they were calculated and reference the corresponding figures or tables in the text.

663 8. Experiments Compute Resources

664 Question: For each experiment, does the paper provide sufficient information on the com-
 665 puter resources (type of compute workers, memory, time of execution) needed to reproduce
 666 the experiments?

667 Answer: [Yes]

668 Justification: The computer resources needed to reproduce the experiments are provided in
 669 the Appendix .A.

670 Guidelines:

- 671 • The answer NA means that the paper does not include experiments.
- 672 • The paper should indicate the type of compute workers CPU or GPU, internal cluster,
 673 or cloud provider, including relevant memory and storage.
- 674 • The paper should provide the amount of compute required for each of the individual
 675 experimental runs as well as estimate the total compute.
- 676 • The paper should disclose whether the full research project required more compute
 677 than the experiments reported in the paper (e.g., preliminary or failed experiments that
 678 didn't make it into the paper).

679 9. Code Of Ethics

680 Question: Does the research conducted in the paper conform, in every respect, with the
 681 NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

682 Answer: [Yes]

683 Justification: The research conducted in the paper conform, in every respect, with the
 684 NeurIPS Code of Ethics.

685 Guidelines:

- 686 • The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- 687 • If the authors answer No, they should explain the special circumstances that require a
 688 deviation from the Code of Ethics.
- 689 • The authors should make sure to preserve anonymity (e.g., if there is a special consid-
 690 eration due to laws or regulations in their jurisdiction).

691 10. Broader Impacts

692 Question: Does the paper discuss both potential positive societal impacts and negative
 693 societal impacts of the work performed?

694 Answer: [Yes]

695 Justification: We discuss the applications and potential impacts of our method in the
 696 introduction and contribution sections.

697 Guidelines:

- 698 • The answer NA means that there is no societal impact of the work performed.
- 699 • If the authors answer NA or No, they should explain why their work has no societal
 700 impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use the open-sourced datasets under their licenses.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- 755 • For existing datasets that are re-packaged, both the original license and the license of
756 the derived asset (if it has changed) should be provided.
757 • If this information is not available online, the authors are encouraged to reach out to
758 the asset's creators.

759 **13. New Assets**

760 Question: Are new assets introduced in the paper well documented and is the documentation
761 provided alongside the assets?

762 Answer: [NA]

763 Justification: The paper does not release new assets.

764 Guidelines:

- 765 • The answer NA means that the paper does not release new assets.
766 • Researchers should communicate the details of the dataset/code/model as part of their
767 submissions via structured templates. This includes details about training, license,
768 limitations, etc.
769 • The paper should discuss whether and how consent was obtained from people whose
770 asset is used.
771 • At submission time, remember to anonymize your assets (if applicable). You can either
772 create an anonymized URL or include an anonymized zip file.

773 **14. Crowdsourcing and Research with Human Subjects**

774 Question: For crowdsourcing experiments and research with human subjects, does the paper
775 include the full text of instructions given to participants and screenshots, if applicable, as
776 well as details about compensation (if any)?

777 Answer: [NA]

778 Justification: The paper does not involve crowdsourcing nor research with human subjects.

779 Guidelines:

- 780 • The answer NA means that the paper does not involve crowdsourcing nor research with
781 human subjects.
782 • Including this information in the supplemental material is fine, but if the main contribu-
783 tion of the paper involves human subjects, then as much detail as possible should be
784 included in the main paper.
785 • According to the NeurIPS Code of Ethics, workers involved in data collection, curation,
786 or other labor should be paid at least the minimum wage in the country of the data
787 collector.

788 **15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human
789 Subjects**

790 Question: Does the paper describe potential risks incurred by study participants, whether
791 such risks were disclosed to the subjects, and whether Institutional Review Board (IRB)
792 approvals (or an equivalent approval/review based on the requirements of your country or
793 institution) were obtained?

794 Answer: [NA]

795 Justification: The paper does not involve crowdsourcing nor research with human subjects.

796 Guidelines:

- 797 • The answer NA means that the paper does not involve crowdsourcing nor research with
798 human subjects.
799 • Depending on the country in which research is conducted, IRB approval (or equivalent)
800 may be required for any human subjects research. If you obtained IRB approval, you
801 should clearly state this in the paper.
802 • We recognize that the procedures for this may vary significantly between institutions
803 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
804 guidelines for their institution.
805 • For initial submissions, do not include any information that would break anonymity (if
806 applicable), such as the institution conducting the review.