Research Article

NumGrad-Pull: Numerical Gradient Guided Tri-plane Representation for Surface Reconstruction from Point Clouds

Ruikai Cui[1], Shi Qiu[2], Jiawei Liu[1], Saeed Anwar[1], Nick Barnes[1]

1. Australian National University, Australia; 2. The Chinese University of Hong Kong, Hong Kong, Hong Kong

Reconstructing continuous surfaces from unoriented and unordered 3D points is a fundamental challenge in computer vision and graphics. Recent advancements address this problem by training neural signed distance functions to pull 3D location queries to their closest points on a surface, following the predicted signed distances and the analytical gradients computed by the network. In this paper, we introduce NumGrad-Pull, leveraging the representation capability of tri-plane structures to accelerate the learning of signed distance functions and enhance the fidelity of local details in surface reconstruction. To further improve the training stability of grid-based tri-planes, we propose to exploit numerical gradients, replacing conventional analytical computations. Additionally, we present a progressive plane expansion strategy to facilitate faster signed distance function convergence and design a data sampling strategy to mitigate reconstruction artifacts. Our extensive experiments across a variety of benchmarks demonstrate the effectiveness and robustness of our approach. Code is available at https://github.com/CuiRuikai/NumGrad-Pull.

Corresponding author: Ruikai Cui, ruikai.cui@anu.edu.au
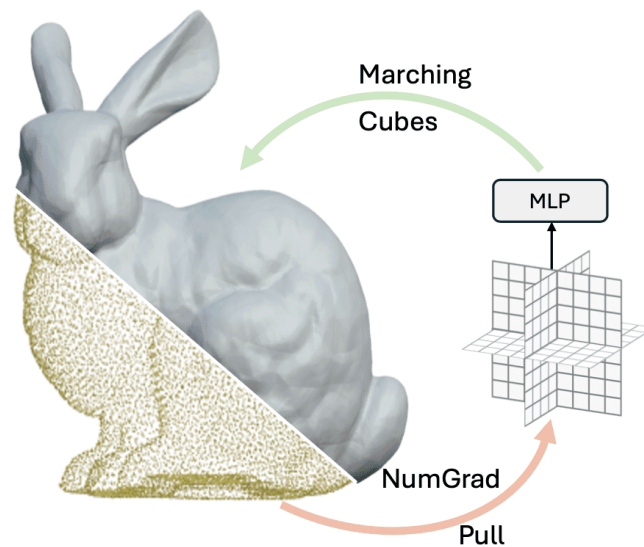
1. Introduction

Figure 1. We present NumGrad-Pull, a tri-plane-based framework that enables efficient and robust surface reconstruction from unoriented point sets.

The signed distance function (SDF) has become a fundamental 3D representation in computer vision and graphics, widely adopted for its utility in various applications. Recent advances[1][2][3][4][5] have increasingly focused on leveraging SDFs to reconstruct continuous surfaces from 3D point clouds[6][7][8][9]. By mapping each 3D coordinate to a corresponding signed distance from the shape surface, SDFs implicitly define surfaces as zero-level sets, providing a powerful surface representation framework that offers distinct advantages in capturing high-fidelity 3D shapes with complex, arbitrary topologies.

Recent efforts[1][4][10][2][11][12] in learning SDFs from point clouds tend to directly train neural networks on point sets, relying on extensive distance computations between query and surface points to guide the network towards accurate distance field convergence. A pioneering approach in this direction is Neural-Pull[1], which trains a neural network to iteratively project query points onto surfaces, using the learned signed distances and gradients to guide each time of adjustment. Subsequent works have extended this method by incorporating additional constraints for improved pulling target estimation[2] [10][4] or enhanced scalability of the approach[3]. In general, existing methods often utilize deeper neural architectures and complex loss constraints to model signed distance fields, yet they still produce overly smoothed surfaces lacking fine-grained details and suffer from slow query speeds.

In this paper, we introduce NumGrad-Pull, a novel method for fast and high-fidelity surface reconstruction from point clouds. NumGrad-Pull models SDFs using a hybrid explicit–implicit representation, which combines a tri-plane structure that stores spatial information, with a shallow multi-layer perceptron (MLP) that maps feature embeddings on the tri-plane to a real signed distance value. Particularly, this design enables faster distance queries[13] compared to previous implicit methods[1][2][10], while maintaining strong geometric and shape expressive capabilities. Although the tri-plane structure improves reconstruction speed and fidelity, it poses challenges for stable training, since its grid-based format lacks feature interaction between adjacent grids, conflicting with the continuous nature of 3D surfaces. To improve the training stability of our approach, we innovate by using numerical gradients instead of analytical ones, allowing queries to back-propagate more effectively to relevant grid entities. Moreover, we introduce a progressive tri-plane expansion scheme, starting the training with a low-resolution tri-plane and gradually increasing the resolution. This approach accelerates convergence and helps prevent the network from getting into local minima.

To assess the effectiveness of our proposed method, we evaluate it on widely used benchmarks that consist of both synthetic and real-world objects. Our main contributions are as follows:

- We introduce a hybrid explicit–implicit tri-plane representation that significantly improves both speed and fidelity in learning signed distance functions from unoriented point clouds.
- We propose to utilize numerical gradients in place of traditional analytical gradients for back-propagating supervision on each query, significantly improving the stability and convergence of the signed distance function.
- We design a progressive tri-plane expansion scheme that further accelerates the convergence of tri-plane learning, enhancing both computational efficiency and reconstruction quality.
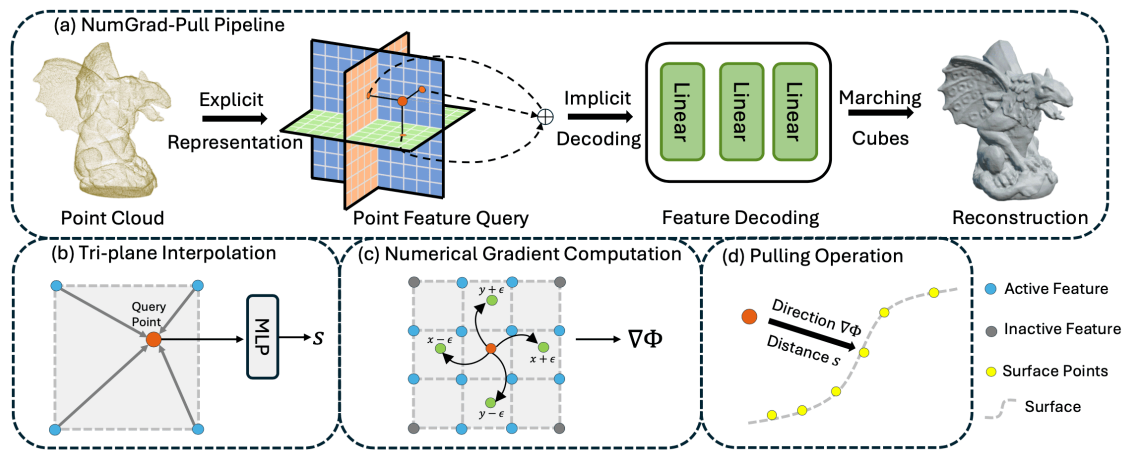

Figure 2. Illustration of our method. (a) The NumGrad-Pull framework parameterizes a neural signed distance function using a hybrid explicit-implicit representation. We employ a tri-plane structure to store explicit spatial information and a shallow MLP to decode features extracted from the tri-plane implicitly, enabling efficient and robust surface reconstruction from unoriented point sets. (b) For a given query point, our method extracts features from each of the three orthogonal planes via bilinear interpolation. (c) To address locality issues and stabilize the training process, we introduce a numerical gradient computation strategy, which involves adjacent grid entities in back-propagation, ensuring smoother feature propagation across the tri-plane. (d) Using the signed distance and numerical gradients obtained from the tri-plane-based SDF, our method trains the network by pulling the query point toward its nearest neighbor on the surface.

2. Related Work

Recent advancements in deep learning have highlighted the significant potential of neural networks for surface reconstruction from 3D point clouds. Below, we provide a brief review of methods relevant to point cloud-based surface reconstruction and surface representation.

2.1. Traditional Surface Reconstruction

Pioneering surface reconstruction methods like Delaunay triangulation[14] introduce tetrahedral meshes to connect points, producing accurate surfaces for dense data but struggling with sparse or noisy inputs due to their computational intensity. Building on this, Poisson surface reconstruction[15] formulates surface inference as a volumetric problem by solving the Poisson equation. This approach effectively produces smooth, watertight surfaces and handles noise well, but it often smooths out fine details. Moving least squares[16] is often used as pre-processing that smooths noisy data by fitting local planes; however, it also oversmooths high-frequency details. The interpolation-based approaches, such as Radial Basis Functions[17], employ continuous functions to interpolate across sparse points, providing robust surfaces for smooth geometries, though computationally demanding for large datasets. By leveraging graph connectivity, graph-based methods address surface reconstruction by organizing point clouds through denoising, outlier removal, and robust normal estimation steps[18][17][19]. These methods effectively handle sparse or non-uniform data, but can be computationally expensive, especially for high-resolution point clouds.

2.2. Neural Surface Reconstruction

More recently, neural networks have enabled learning-based approaches to surface reconstruction, where implicit functions trained on large datasets facilitate detailed surface modeling even with sparse inputs[20][21]. However, estimating surfaces from unoriented point sets remains challenging without ground truth signed distance values. Point2Surf[22] addresses this issue by learning a prior over local patches with detailed structures and coarse global information. Moreover,[1] introduce the first pulling operation-based surface reconstruction scheme, observing that the gradient of a neural SDF represents the normal direction of a query point. By using this gradient and the estimated distance from the SDF, any query point can be pulled to the surface, considering the neural SDF as a close approximation of the ground truth. Building on this scheme, Grid-Pull[3] presents a discrete grid-based representation to improve the scalability of pull-based methods. Subsequent work by[10] applies guiding points for incrementally changing optimizations toward the true surfaces. More recently,[2] propose a bilateral filter to smooth the implicit field, retaining high-frequency geometric details. Different from existing studies, we are pioneering the exploration of hybrid explicit–implicit representations[13] for fundamental surface reconstruction problems, where our approach shows promise in accelerating query speeds and preserving geometric details across various reconstruction tasks[23][24].

2.3. Neural Surface Representation

Implicit representations have become widely used for surface reconstruction from point clouds, e.g., Neural-Pull[1] devises a deep fully-connected network to represent the distance field as a continuous function. However, this approach results in slow query speeds, as each distance query requires a complete pass through the network. In contrast, explicit representations, such as discrete voxel grids[3], offer faster query times but face memory limitations, with memory consumption scaling as $O(n^3)$ as the discretization resolution increases. Given complementary advantages provided by explicit and implicit representations, motivated to develop a hybrid approach for surface reconstruction that combines the strengths of both paradigms. In this work, we present the first hybrid explicit–implicit method for neural surface reconstruction from point clouds, leveraging the tri-plane structure[13][25][26][27][24][28][29] to integrate the efficiency of explicit storage with the flexibility of implicit decoding. Particularly, our proposed tri-plane method stores features on axis-aligned planes and exploys a lightweight implicit decoder to aggregate these features for efficient querying, achieving computational and memory efficiency.

3. Method

NumGrad-Pull reconstructs continuous surfaces from unoriented point sets by parameterizing a signed distance function (SDF) with a hybrid explicit-implicit tri-plane structure (Sec. 3.2). The network is trained with numerical gradients (Sec. 3.3), which help stabilize the training process. Additionally, we introduce a progressive tri-plane expansion strategy (Sec. 3.4) and a novel query location sampling method (Sec. 3.5) to enhance both convergence speed and reconstruction quality. The overall pipeline of our proposed method is illustrated in Fig. 2.

3.1. Preliminaries

A signed distance function is a function $\Phi : \mathbb{R}^3 \mapsto \mathbb{R}$ that maps a spatial position to a real value representing the signed distance to the shape surface. Using this function, the shape surface can be defined as the zero-level set of Φ, allowing for surface extraction through methods such as marching cubes[30]. To learn a neural SDF Φ from a given point cloud P without normal information,[1] propose to query the function at various positions, and train a network to "pull" each query position q toward the shape surface, approximated by the nearest point on P to q. Specifically, this pulling operation is defined as:

$$q' = q - |\Phi(q)| \times \frac{\nabla\Phi(q)}{\|\nabla\Phi(q)\|},\tag{1}$$

where $\frac{\nabla\Phi(q)}{\|\nabla\Phi(q)\|}$ denotes the normalized gradient of q and $|\Phi(q)|$ is its distance from the shape surface, both predicted by the SDF. In practice, neural SDFs are often implemented as deep fully-connected networks, which can be slow to query due to the need for a full forward pass regarding each time of a distance query. This limitation highlights the critical need for a faster SDF parameterization that achieves high reconstruction quality without compromising query efficiency.

3.2. Tri-plane Formulation

Unlike prior methods[1][2] that rely solely on deep fully-connected networks to learn a neural SDF, we leverage tri-plane embeddings to efficiently incorporate explicit 3D spatial information. Basically, our method estimates the signed distance of a query position by interpolating within the tri-plane structure and decoding through a shallow MLP to produce a real-valued signed distance output. By combining tri-plane structures with finite differentiation for gradient computation, our end-to-end trainable neural SDF model captures a high-fidelity 3D shape representation of a given point cloud.

To formulate our hybrid explicit-implicit representation for a neural SDF, we integrate a tri-plane structure for capturing geometric features, with an MLP parameterized by θ for signed distance decoding. Specifically, the tri-plane structure consists of three orthogonal planes (XY, YZ, and XZ) that collectively factorize a dense 3D volume grid. For any query point $q \in \mathbb{R}^3$, the signed distance function $\Phi : \mathbb{R}^3 \to \mathbb{R}$ is computed as follows:

$$\Phi(q) = \text{MLP}_\theta \left(\bigoplus_{i \in \{xy, xz, yz\}} \text{Interp}_{\gamma_i} \left(\text{Proj}_{\gamma_i}(q) \right) \right),\tag{2}$$

where $\gamma = \{\gamma_i | \gamma_i \in \mathbb{R}^{N^2 \times C}, i \in \{xy, xz, yz\}\}$ denotes the three feature planes with N^2 spatial resolution and C feature channels, $\text{Proj}(\cdot)$ represents an orthogonal projection of the query point onto each respective plane, $\text{Interp}(\cdot)$ denotes bilinear interpolation that extracts feature vectors from each plane, and $\bigoplus$ indicates element-wise addition.

3.3. Numerical Gradient

While tri-plane representations enhance geometric feature learning and accelerate training, they introduce a locality issue because spatial features are stored in a grid-based structure. Accordingly, each forward pass only considers neighboring features within a small spatial region, limiting the optimization of tri-plane features to adjacent grids. Following previous works[1][3][2][10], we initially compute the gradient term $\nabla\Phi(\cdot)$ in Eq. 1 using analytical gradients. However, this attempted approach proves unstable as it only propagates guidance to limited features in one forward pass, preventing proper convergence of the tri-plane-based neural SDF. The detailed reasons are explained as below:

Given a query point $q = (x, y, z)$ and a plane resolution N^2, the feature extraction for the tri-plane encoding from the XY plane can be achieved via bilinear interpolation. First, we scale the query point by the plane resolution: $q_{xy} = (x, y) \cdot N$, and define the coefficients for bilinear interpolation at the four corners of the grid as:

$$\beta_x = q_{xy,x} - \lfloor q_{xy,x} \rfloor, \quad \beta_y = q_{xy,y} - \lfloor q_{xy,y} \rfloor,\tag{3}$$

where β_x and β_y are the interpolation coefficients along the x- and y-axes, respectively. The resulting feature vector $\gamma_{xy}(q_{xy})$ from the XY plane is then given by:

$$\begin{aligned}
\gamma_{xy}(q_{xy}) = \ &\gamma_{xy}(\lfloor q_{xy,x} \rfloor, \lfloor q_{xy,y} \rfloor) \cdot (1 - \beta_x) \cdot (1 - \beta_y) + \\
&\gamma_{xy}(\lceil q_{xy,x} \rceil, \lfloor q_{xy,y} \rfloor) \cdot \beta_x \cdot (1 - \beta_y) + \\
&\gamma_{xy}(\lfloor q_{xy,x} \rfloor, \lceil q_{xy,y} \rceil) \cdot (1 - \beta_x) \cdot \beta_y + \\
&\gamma_{xy}(\lceil q_{xy,x} \rceil, \lceil q_{xy,y} \rceil) \cdot \beta_x \cdot \beta_y.
\end{aligned}\tag{4}$$

After performing bilinear interpolation on each of the three planes (XY, XZ, and YZ), the final feature vector is obtained by aggregating the features from each plane:

$$\gamma(q) = \gamma_{xy}(q_{xy}) \oplus \gamma_{xz}(q_{xz}) \oplus \gamma_{yz}(q_{yz}).\tag{5}$$

However, the feature extraction in the forward pass involves non-differentiable rounding operations, such as $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$. As a result, the gradient of the tri-plane encoding is inherently local: when the query point q crosses grid boundaries, the corresponding grid corner entities change discontinuously. Consequently, the supervision loss is only back-propagated to the grid entities involved in the current forward pass, i.e., the four corner points involved in the interpolation. This local gradient propagation becomes problematic when learning neural SDFs from sparse point clouds. In such cases, the sparsity of the points can confuse the network about surface connectivity as the discrete grid features do not imply continuous gradient changes, leading to corrupted learning.

To address the locality issue of the analytical gradient in tri-plane encoding and enable joint optimization across neighboring grids, we employ finite differentiation for gradient computation. Specifically, for each query point $q = (x, y, z)$, we estimate the gradient by sampling additional points around q at a small offset ϵ along each axis of the canonical coordinate system. For instance, the x-component of the gradient is computed as:

$$\nabla_x \Phi(q) = \frac{\Phi(q + \epsilon_x) - \Phi(q - \epsilon_x)}{2\epsilon},$$

(6)

where $\epsilon_x = [\epsilon, 0, 0]$ is the perturbation along the x-axis; $q + \epsilon_x$ and $q - \epsilon_x$ represent two sample points along the x-axis. Using the same approach for $\nabla_y \Phi(q)$ and $\nabla_z \Phi(q)$, we sample a total of six points around q to construct the complete gradient $\nabla \Phi(q) = (\nabla_x \Phi(q), \nabla_y \Phi(q), \nabla_z \Phi(q))$.

This finite differentiation approach enables gradient computation to incorporate information from adjacent grids when a query point is near a grid boundary. By involving multiple tri-plane entries in the gradient calculation, the subsequent back-propagation updates multiple grid regions simultaneously, thereby enhancing the stability of the learning process.

3.4. Progressive Tri-Plane Expansion

We further propose a coarse-to-fine optimization scheme for surface reconstruction, which progressively expands tri-plane resolution over n stages to prevent the neural SDF's convergence to local minima. Starting with a low-resolution tri-plane facilitates efficient learning of the global shape context and provides a good initialization for subsequent higher-resolution refinements of surface details.

Let the finest tri-plane resolution be N^2. We begin the optimization with a tri-plane of resolution $R = \left(\frac{N}{2^n}\right)^2$, where R denotes the current tri-plane resolution, and increase the resolution by a factor of 2 at each stage. We achieve the resolution upsampling via bilinear interpolation. Furthermore, to ensure that the queries for numerical gradients remain within a valid vicinity, we dynamically adjust the perturbation value ϵ to be inversely correlated with the current resolution: $\epsilon = \frac{1}{2R}$. This progressive refinement of the tri-plane resolution helps stabilize training, allowing the network to capture both global and fine-grained surface details while avoiding local minima.

3.5. Query Locations Sampling

To learn a neural SDF from point clouds, we need to sample query-target point pairs[1]. For an extensive sampling of random query positions around each point p_j in P, we follow an isotropic Gaussian distribution $\mathcal{N}(p_j, \sigma^2)$ to sample query locations. We randomly sample 25 query points based on this distribution, where σ^2 controls the sampling range around the surface. The value of σ^2 is adaptively set as the squared distance between p_j and its 50th nearest neighbor, indicating the local point density around p_j.

However, this data sampling strategy is insufficient for the learning of our tri-plane-based hybrid SDF representations. The abovementioned procedure guides the object's surface, while neglecting distant regions. Since our method encodes spatial information on a tri-plane structure, this lack of sufficient guidance can lead to inaccurate or random predictions for the signed distances in distant areas. To address this issue, we introduce another complementary sampling strategy of randomly sampling points in a unit cube $[-1, 1]^3$, aligning with the scale of training data. Accordingly, we not only ensure that every tri-plane entity is trained, but also prevent untrained regions of the tri-plane grid from negatively affecting the learning process.

3.6. Optimization

Our proposed method trains a neural SDF by learning to pull a query location q_i towards its nearest neighbor, i.e., the target surface point t_i in the point cloud P. Following the computations in Eq. 1, after each training iteration, a query point q_i will be moved to a new position q_i' closer to the point cloud surface, guided by the learned distance value and numerical gradients. Particularly, we train the neural layers of NumGrad-Pull in an end-to-end manner by minimizing the squared Euclidean distances (ℓ_2 norm) between the pulled query locations and their corresponding nearest neighbor targets. The overall loss is formulated as follows:

$$\mathcal{L} = \frac{1}{I} \sum_{i=1}^{I} \|q_i' - t_i\|^2,$$

(7)

where I is the number of sampled query points.

4. Experiments

We conduct experiments to evaluate the performance of our NumGrad-Pull for surface reconstruction from raw point clouds, presenting results on both synthetic point clouds and real scanned data. We also perform ablation studies to validate the main components of NumGrad-Pull and analyze the impact of various design choices.

4.1. Implementation Details

We initialize the tri-plane with a resolution of 8^2 and a feature dimension of 32. The tri-planes are then expanded over three stages at the 3k, 8k, and 12k iterations, ultimately reaching a resolution of 32^2. To decode the tri-plane features, we use a three-layer fully connected network with a hidden dimension of 128. Additionally, we initialize the network parameters using the geometric network initialization scheme proposed in[31], which approximates the signed distance function of a sphere. For optimization, we use the Adam optimizer[32], with an initial learning rate of 0.001 for the MLP and 0.05 for the tri-plane parameters. All the experiments are conducted on an Nvidia GeForce RTX 3090 GPU.

Dataset	NP[1]	PCP[12]	SIREN[33]	DIGS[4]	IF[2]	Ours
ABC	0.95	2.52	0.22	0.21	0.11	**0.09**
FAMOUS	1.00	0.37	0.25	0.15	0.08	**0.04**
Mean	0.98	1.45	0.24	0.18	0.10	**0.07**

Table 1. Comparisons on ABC[34] and FAMOUS[22] datasets. The metric is chamfer distance (CD_{ℓ_2}) scaled by 10^3.

Method	Venue	Display	Lamp	Plane	Cabinet	Vessel	Table	Chair	Sofa	Avg.
SPSR[15]	SGP 2006	2.730	2.270	2.170	3.630	2.540	3.830	2.930	2.760	2.860
NP[1]	ICML 2021	0.390	0.800	0.080	0.260	0.220	0.600	0.540	0.120	0.380
LPI[35]	ECCV 2022	0.080	0.172	0.060	0.179	0.092	0.436	0.187	0.164	0.171
PCP[12]	CVPR 2022	0.887	0.380	0.065	0.153	0.079	0.131	0.110	0.086	0.136
GP[3]	ICCV 2023	0.082	0.347	0.007	0.112	0.033	0.052	0.043	0.015	0.086
IF[2]	ECCV 2024	**0.009**	0.019	0.045	0.055	**0.005**	0.025	0.070	0.027	0.032
Ours	N/A	0.024	**0.013**	**0.012**	**0.032**	0.012	**0.022**	**0.024**	**0.020**	**0.020**

Table 2. Comparisons on the ShapeNet dataset[36]. The metric is chamfer distance (CD_{ℓ_2}) scaled by 10^3.

Method	Anchor				Daratech				DC				Gargoyle				Lord Quas			
	CD_{ℓ_1}	HD	$d_{\vec{C}}$	$d_{\vec{H}}$	CD_{ℓ_1}	HD	$d_{\vec{C}}$	$d_{\vec{H}}$	CD_{ℓ_1}	HD	$d_{\vec{C}}$	$d_{\vec{H}}$	CD_{ℓ_1}	HD	$d_{\vec{C}}$	$d_{\vec{H}}$	CD_{ℓ_1}	HD	$d_{\vec{C}}$	$d_{\vec{H}}$
SPSR[15]	0.60	14.89	0.60	14.89	0.44	7.24	0.44	7.24	0.27	3.10	0.27	3.10	0.26	6.80	0.26	6.80	0.20	4.61	0.20	4.61
IGR[31]	0.22	4.71	0.12	1.32	0.25	4.01	0.08	1.59	0.17	2.22	0.09	2.61	0.16	3.52	0.06	0.81	0.12	1.17	0.07	0.98
SIREN[33]	0.32	8.19	0.10	2.43	0.21	4.30	0.09	1.77	0.15	2.18	0.06	2.76	0.17	4.64	0.08	0.91	0.17	0.82	0.12	0.76
VisCo[37]	0.21	3.00	0.15	1.07	0.21	4.06	0.14	1.76	0.15	2.22	0.09	2.76	0.17	4.40	0.11	0.96	0.12	1.06	0.07	0.64
SAP[38]	0.12	2.38	0.08	0.83	0.26	0.87	0.04	0.41	0.07	1.17	0.04	0.53	0.07	1.49	0.05	0.78	0.05	0.98	0.04	0.51
NP[1]	0.122	3.243	0.061	3.208	0.375	3.127	0.746	3.267	0.157	3.541	0.242	3.523	0.080	1.376	0.063	0.475	0.064	0.822	0.053	0.508
GP[3]	0.093	1.804	0.066	0.460	0.062	**0.648**	0.039	0.293	0.066	1.103	0.036	0.539	0.063	1.129	0.045	0.700	0.047	0.569	0.031	0.370
DIGS[4]	0.063	1.447	0.030	0.270	**0.049**	0.858	**0.025**	0.441	0.042	0.667	0.022	0.729	0.047	0.971	0.028	0.271	0.031	0.496	0.017	0.181
IF[2]	0.052	1.232	**0.025**	0.265	0.051	0.751	0.028	0.423	0.041	0.815	**0.019**	0.724	0.044	1.089	**0.022**	0.246	**0.030**	0.554	**0.014**	0.230
Ours	**0.051**	**1.194**	0.041	**0.113**	0.050	0.730	0.036	**0.236**	**0.026**	**0.215**	0.032	**0.286**	**0.033**	**0.416**	0.040	**0.127**	0.036	**0.293**	0.039	**0.127**

Table 3. Comparison on the Surface Reconstruction Benchmark[39].

Metric	8	16	32	64	NP[1]	IF[2]
$CD_{\ell_2} \downarrow$	9.31	0.45	0.39	**0.35**	1.00	0.80
Speed (iter/s) $\uparrow$	219	**231**	227	225	122	42

Table 4. Effect of different tri-plane configurations on surface reconstruction for the FAMOUS dataset[22] compared with recent methods, measured by CD_{ℓ_2} and scaled by 10^4.
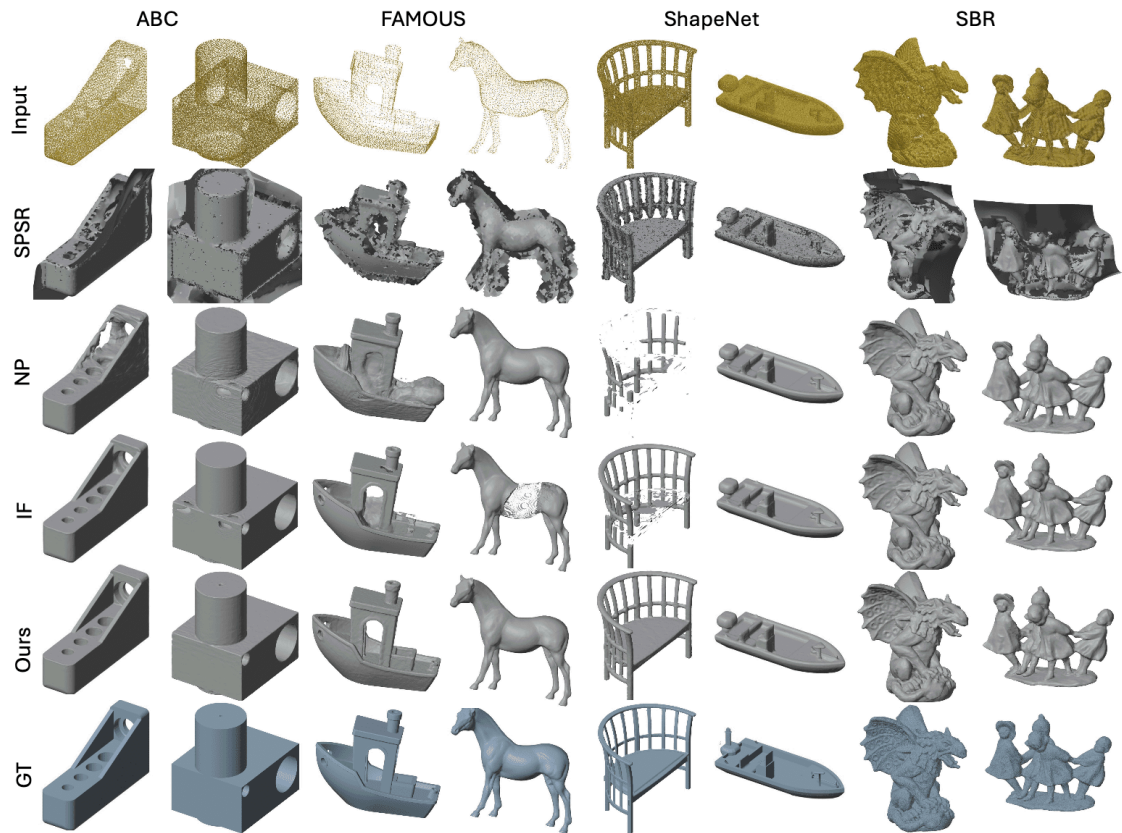
Figure 3. Visual comparisons of surface reconstruction quality on the synthetic datasets (ABC[34], FAMOUS[22], ShapeNet[36]) and the real-world scan dataset (SBR[39]).

4.2. Surface Reconstruction for Shapes

4.2.1. Datasets and Metrics

For surface reconstruction of general shapes from raw point clouds, we follow the evaluation protocol used in prior work[1][2] and assess our method on three widely used datasets: ShapeNet[36], ABC[34], and FAMOUS[22]. For ShapeNet, we use the test split defined in[40] and the pre-processed data provided in[20]. Following the footsteps of[22], we employ the same train/test split and pre-processed data for the ABC and FAMOUS datasets.

For performance evaluation, we follow the approach of previous works[1][2], sampling 1×10^5 points from both the reconstructed surfaces and the ground truth meshes for the ShapeNet dataset and 1×10^4 points for the ABC and FAMOUS datasets. We use Chamfer distance (CD), measured by ℓ_2 norm to quantify the distance between our reconstructions and the ground truth surfaces.

4.2.2. Comparison

We compare the performance of our proposed NumGrad-Pull with several state-of-the-art methods, including SPSR[15], PCP[12], LPI[35], SIREN[33], Neural-Pull[1], DIGS[4], Grid-Pull[3], and the recent IF[2]. Results on the ABC and FAMOUS datasets are presented in Tab. 1, highlighting the superiority of our method: NumGrad-Pull outperforms the latest state-of-the-art by margins of 0.02 and 0.04 on the ABC and FAMOUS datasets, respectively. Furthermore, our method shows strong performance in Tab. 2 on the ShapeNet dataset, which is a challenging and comprehensive benchmark comprising over 3,000 testing objects. Our NumGrad-Pull achieves the best average surface reconstruction quality across all methods, outperforming the state-of-the-art IF[2] method by 0.012 and achieving the best results in six out of eight categories. These quantitative results demonstrate the effectiveness of our approach in improving surface reconstruction quality.

In Fig. 3, we present qualitative comparisons with other approaches[15][1][2], showcasing that our method effectively reconstructs object surfaces from unoriented point clouds across varying levels of sparsity and noise. Notably, our method reliably captures detailed structures, as seen in the chair sample from the ShapeNet dataset. Similarly, our approach produces intact surfaces without the breaks observed in other methods, as illustrated in the second example from the ABC dataset. These results further highlight the effectiveness and robustness of our approach.

4.3. Surface Reconstruction for Real Scans

4.3.1. Datasets and Metrics

For surface reconstruction from real point cloud scans, we follow the evaluation protocol of VisCo[37] on the Surface Reconstruction Benchmark (SRB)[39], which includes five challenging real-world scans. We adopt Chamfer and Hausdorff distances (CD_{ℓ_1} and HD) between the reconstruction meshes and the ground truth as the quality metrics. Following the state-of-the-art method[2], we also report the corresponding uni-directional distances ($d_{\vec{C}}$ and $d_{\vec{H}}$) between the reconstructed meshes and the input noisy point clouds to evaluate the performance in preserving scanned geometries.

4.3.2. Comparison

We compare our method with various prior methods using the specified evaluation metrics, with per-object results presented in Tab. 3. Our method performs well across multiple metrics and objects, particularly achieving superior reconstruction quality on the Anchor and the Gargoyle object. To further substantiate the results, we provide visualizations of the Gargoyle (7th column) and DC (8th column) samples in Fig. 3. Our method captures finer details, such as the rings on the Gargoyle's wings, and preserves accurate geometry even in areas with corruptions in the original scan, demonstrating robust detail retention and reconstruction fidelity.

4.4. Ablation Studies

Figure 4. Visual results of surface reconstruction using different tri-plane resolutions.

4.4.1. Effect of Tri-plane Resolution

Considering that tri-plane resolution impacts surface reconstruction quality, we empirically evaluate the performance of NumGrad-Pull using various resolutions, 8, 16, 32, and 64, on the FAMOUS dataset. Visual results are shown in Fig. 4, and quantitative results are provided in Tab. 4. Our findings indicate that as tri-plane resolution increases, surface reconstruction quality (measured by CD_{ℓ_2}) improves. The most substantial gain is observed when increasing the resolution from 8 to 16, while increasing to 32 and 64 resolution yields marginal improvements. The significant quality boost from 8 to 16 suggests that an 8×8 tri-plane lacks sufficient capacity to store the necessary information for accurate surface reconstruction, as shown in Fig. 4(b). Higher resolutions enhance the network's ability to capture fine structures; however, with a 64×64 tri-plane, while the metric show slight improvements, the visual results reveal unwanted surface unevenness, indicating that excessive high-frequency noise is being captured. Overall, to balance reconstruction quality with robustness against noise, we apply a 32×32 tri-plane for our main experiments.

We also present efficiency comparisons between recent methods with our NumGrad-Pull using different tri-plane resolutions in Tab. 4. Since our proposed tri-plane-based neural SDF can significantly improve query speed, we achieve a 1.8× speedup over Neural-Pull[1] and a 5.4× speedup over IF[2], while retaining superior surface reconstruction quality. Furthermore, the tri-plane representation exhibits excellent scalability with respect to resolution: as the tri-plane resolution increases, the query speed remains stable. This stability is attributed to the feature interpolation process being an $O(1)$ operation, ensuring that query speed is unaffected by resolution changes.

4.4.2. Effect of Each Module

We conduct a detailed ablation study to validate the effectiveness of each proposed module, including the following model variants: A) full model without our data sampling strategy; B) full model without the tri-plane structure; C) full model without numerical gradients; and D) full model without progressive tri-plane expansion. These variants are compared with a baseline model Neural-Pull[1] and our complete model incorporating all proposed modules. We evaluate the performances on the FAMOUS dataset, provide quantitative results in Tab. 5, and present a visualization of the *hand* sample in Fig. 5.

Our full model achieves the best visual and quantitative results, capturing all essential details without artifacts. In contrast, each variant shows varying degrees of performance degradation. **Model A** lacks a comprehensive data sampling strategy, sampling query points only near the object surface, similar to Neural-Pull. This model suffers from insufficient supervision for regions far from the surface, leading to poorly trained tri-plane features at distant positions and causing random predictions and artifacts. **Model B** replaces the tri-plane with an 8-layer fully connected network, achieving results close to our full model but with reduced reconstruction detail and training stability. While it retains more structural detail than the baseline by employing numerical gradients, it is approximately 3.2 times slower due to the slower query speed of the fully connected network compared to the tri-plane structure. **Model C** fails to train effectively due to the adoption of analytical gradients, which only propagate supervision to local grids and causes instability and corrupted outputs. **Model D** struggles without using our proposed progressive expansion strategy, which is crucial for preventing the network from getting stuck in local optima. This experimental comparison highlights the necessity of our strategy for the effective training of the tri-plane structure. Overall, these results demonstrate that each of the proposed modules is essential, contributing to the superior performance of the full model.
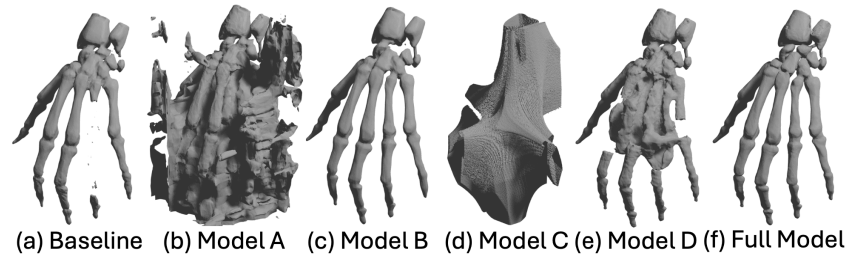

(a) Baseline   (b) Model A   (c) Model B   (d) Model C   (e) Model D   (f) Full Model

Figure 5. Visual results of using different model variants.

Model	w/o Data	w/o Tri-plane	w/o NumGrad	w/o Progressive	CD_{L2}
baseline	✗	✗	✗	✗	11.35
A	✗	✓	✓	✓	50.21
B	✓	✗	✓	✓	0.41
C	✓	✓	✗	✓	331.84
D	✓	✓	✓	✗	0.45
Full	✓	✓	✓	✓	**0.39**

Table 5. Effect of each proposed module of our method on sur- face reconstruction for the FAMOUS dataset, measured by CD_{l2}, scaled by 10^4.

5. Future Work

While our proposed NumGrad-Pull shows promising results, there are limitations to address in future work. In our ablation study on tri-plane resolution, our method captures high-frequency noise at higher resolutions, limiting its effectiveness in finer-grained modeling and highlighting a potential improvement area. Additionally, we plan to explore various regularization techniques to improve the robustness of surface reconstruction against noisy and sparse point clouds. Moreover, although our method is primarily introduced for object-level surface reconstruction, the tri-plane structure holds

strong potential for scene-level modeling. We intend to conduct a more comprehensive evaluation using scene-level reconstruction data to further validate and extend the applicability of our approach.

6. Conclusion

In this paper, we introduce NumGrad-Pull, a novel approach for high-fidelity surface reconstruction from unoriented point clouds. Our method employs a hybrid explicit–implicit tri-plane representation to improve query speed and enhance reconstruction fidelity. Furthermore, we propose a numerical gradients-based technique to stabilize the training process and a progressive expansion strategy to fully leverage the capabilities of the tri-plane structure. Extensive experiments demonstrate that our NumGrad-Pull consistently outperforms state-of-the-art methods across various challenging scenarios, showcasing significant effectiveness in surface reconstruction problems.

Broader Impact: We believe NumGrad-Pull has strong potential to improve 3D content creation and streamline the conversion of raw 3D data into production-ready formats. While we do not foresee any immediate harmful uses, we emphasize the importance of users exercising caution to minimize potential risks, given that surface reconstruction tools could also be leveraged for malicious purposes.

References

1. a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t *Ma B, Han Z, Liu YS, Zwicker M. Neural-pull: Learning signed distance function from point clouds by learning to pull space onto surface. In: Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. PMLR; 2021. p. 7246-7257.*

2. a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r *Li S, Gao G, Liu Y, Gu M, Liu YS. "Implicit filtering for learning neural signed distance functions from 3d point clouds." In: Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VI. Springer; 2024. p. 234-251.*

3. a, b, c, d, e, f, g, h *Chen C, Liu YS, Han Z (2023). "Gridpull: Towards scalability in learning implicit representations from 3d point clouds". Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18322–18334.*

4. a, b, c, d, e, f *Ben-Shabat Y, Hewa Koneputugodage C, Gould S (2022). "Digs: Divergence guided shape implicit neural representation for unoriented point clouds". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19323–19332.*

5. ^ *Zhu R, Kang D, Hui KH, Qian Y, Qiu S, Dong Z, Bao L, Heng PA, Fu CW. "SSP: Semi-signed prioritized neural fitting for surface reconstruction from unoriented point clouds." In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024. p. 3769–3778.*

6. ^ *Cui R, Qiu S, Anwar S, Zhang J, Barnes N (2022). "Energy-based residual latent transport for unsupervised point cloud completion". arXiv preprint arXiv:2211.06820. Available from: https://arxiv.org/abs/2211.06820.*

7. ^ *Shi Qiu, Anwar S, Barnes N (2021). "Pnp-3d: A plug-and-play for 3d point clouds". IEEE Transactions on Pattern Analysis and Machine Intelligence. 45 (1): 1312–1319.*

8. ^ *Shi Qiu, Saeed Anwar, Nick Barnes. "Pu-transformer: Point cloud upsampling transformer." In Proceedings of the Asian conference on computer vision, pages 2475–2493, 2022.*

9. ^ *Cui R, Qiu S, Anwar S, Liu J, Xing C, Zhang J, Barnes N (2023). "P2c: Self-supervised point cloud completion from single partial clouds". Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14351–14360.*

10. a, b, c, d, e *Chamin~Hewa Koneputugodage, Yizhak Ben-Shabat, Dylan Campbell, Stephen Gould. "Small steps and level sets: Fitting neural surface models with point guidance." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 21456–21465, 2024.*

11. ^ *Chamin~Hewa Koneputugodage, Yizhak Ben-Shabat, Gould S. Octree guided unoriented surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023:16717-16726.*

12. a, b, c, d *Ma B, Liu YS, Zwicker M, Han Z (2022). "Surface reconstruction from point clouds by learning predictive context priors". In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. IEEE; 2022. p. 6316-6327.*

13. a, b, c *Chan ER, Lin CZ, Chan MA, Nagano K, Pan B, De Mello S, Gallo O, Guibas LJ, Tremblay J, Khamis S, et al. Efficient geometry-aware 3D generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022:16123-16133.*

14. ^ *Boissonnat JD (1984). "Geometric structures for three-dimensional shape representation". ACM Transactions on Graphics (TOG). 3 (4): 266–286.*

15. a, b, c, d, e *Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing, 2006.*

16. ^ *Levin D (1998). "The approximation power of moving least-squares". Mathematics of computation. 67 (224): 1517–1531.*

17. a, b *Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, Evans TR (2001). "Reconstruction and representation of 3D objects with radial basis functions". In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 67–76.*

18. ^ *Amenta N, Bern M. Surface reconstruction by voronoi filtering. In: Proceedings of the fourteenth annual symposium on Computational geometry. 1998. p. 39-4 8.*

19. ^ *Huang H, Li D, Zhang H, Ascher U, Cohen-Or D (2009). "Consolidation of unorganized point clouds for surface reconstruction". ACM Transactions on Graphics (TOG). 28 (5): 1–7.*

20. a, b *Mescheder L, Oechsle M, Niemeyer M, Nowozin S, Geiger A (2019). "Occupancy networks: Learning 3d reconstruction in function space". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pages 4460–4470.*

21. ^ *Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S (2019). "Deepsdf: Learning continuous signed distance functions for shape representation". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pages 165–174.*

22. a, b, c, d, e, f *Erler P, Guerrero P, Ohrhallinger S, Mitra NJ, Wimmer M. Points2surf learning implicit surfaces from point clouds. In: Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V. Springer; 2020. p. 108-124.*

23. ^ *Lin CH, Gao J, Tang L, Takikawa T, Zeng X, Huang X, Kreis K, Fidler S, Liu MY, Lin TY. "Magic3D: High-resolution text-to-3D content creation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 300-309.*

24. a, b *Cui R, Song X, Sun W, Wang S, Liu W, Chen S, Shang T, Li Y, Barnes N, Li H, et al. Lam3d: Large image-point-cloud alignment model for 3d reconstruction from single image. arXiv preprint arXiv:2405.15622. 2024.*

25. ^ *Müller T, Evans A, Schied C, Keller A (2022). "Instant neural graphics primitives with a multiresolution hash encoding". ACM Transactions on Graphics (TOG). 41 (4): 1–15.*

26. ^ *Mildenhall B, Srinivasan P, Tancik M, Barron J, Ramamoorthi R, Ng R (2021). "Nerf: Representing scenes as neural radiance fields for view synthesis". Communications of the ACM. 65 (1): 99–106.*

27. ^ *Li Z, M\uoofcller T, Evans A, Taylor RH, Unberath M, Liu MY, Lin CH (2023). "Neuralangelo: High-fidelity neural surface reconstruction". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8456-8465.*

28. ^ *Cui R, Liu W, Sun W, Wang S, Shang T, Li Y, Song X, Yan H, Wu Z, Chen S, et al. Neusdfusion: A spatial-aware generative model for 3d shape completion, reconstruction, and generation. arXiv preprint arXiv:2403.18241. 2024.*

29. ^ *Wu Z, Li Y, Yan H, Shang T, Sun W, Wang S, Cui R, Liu W, Sato H, Li H, et al. Blockfusion: Expandable 3D scene generation using latent tri-plane extrapolation. ACM Transactions on Graphics (TOG). 43 (4): 1–17, 2024.*

30. ^ *Lorensen WE, Cline HE. Marching cubes: A high resolution 3d surface construction algorithm. In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987. ACM; 1987. p. 163-169.*

31. a, b *Gropp A, Yariv L, Haim N, Atzmon M, Lipman Y (2020). "Implicit geometric regularization for learning shapes". arXiv preprint arXiv:2002.10099. Available from: https://arxiv.org/abs/2002.10099.*

32. ^ *Kingma DP, Ba J (2015). "Adam: A method for stochastic optimization." In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

33. a, b, c *Sitzmann V, Martel JNP, Bergman AW, Lindell DB, Wetzstein G (2020). "Implicit neural representations with periodic activation functions". In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2 020.*

34. a, b, c *Koch S, Matveev A, Jiang Z, Williams F, Artemov A, Burnaev E, Alexa M, Zorin D, Panozzo D. "ABC: A big CAD model dataset for geometric deep learning." In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE; 2019. p. 9601-9611.*

35. a, b *Chen C, Liu YS, Han Z (2022). "Latent partition implicit with surface codes for 3D representation". In: European Conference on Computer Vision. Springer. pp. 322–343.*

36. a, b, c *Chang AX, Funkhouser TA, Guibas LJ, Hanrahan P, Huang QX, Li Z, Savarese S, Savva M, Song S, Su H, Xiao J, Yi L, Yu F (2015). "Shapenet: An information-rich 3d model repository". CoRR. abs/1512.03012. arXiv:1512.03012.*

37. a, b *Pumarola A, Sanakoyeu A, Yariv L, Thabet A, Lipman Y (2022). "Visco grids: Surface reconstruction with viscosity and coarea grids". Advances in Neural Information Processing Systems. 35: 18060–18071.*

38. ^ *Peng S, Jiang C, Liao Y, Niemeyer M, Pollefeys M, Geiger A (2021). "Shape as points: A differentiable poisson solver". Advances in Neural Information Processing Systems. 34: 13032–13044.*

39. a, b, c *Williams F, Schneider T, Silva C, Zorin D, Bruna J, Panozzo D. Deep geometric prior for surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019:10130-10139.*

40. ^ *Liu S, Saito S, Chen W, Li H (2019). "Learning to infer implicit surfaces without 3D supervision". Advances in Neural Information Processing Systems. 32.*

Declarations