Point Voxel Bi-directional Fusion Implicit Field for 3D Reconstruction

Chuanmao Fan*
cf7b6@umsystem.edu
University of Missouri
Columbia, Missouri, USA

Chenxi Zhao†
chenxiz@clemson.edu
Clemson University
Clemson, South Carolina, USA

Kevin Xue
kzx2n8@umsystem.edu
University of Missouri
Columbia, Missouri, USA

Ye Duan‡
duan@clemson.edu
Clemson University
Clemson, South Carolina, USA

ABSTRACT

3D surface reconstruction from unorganized point clouds is a fundamental task in visual computing and has numerous applications in areas such as robotics, virtual reality, augmented reality, and animation. To date, many deep learning-based surface reconstruction methods have been proposed with outstanding performance on various benchmark datasets. Among them, neural implicit field learning-based methods have been particularly popular because they can represent both complex inner structures and open surfaces in a continuous implicit distance field. Existing implicit distance field-based methods either utilize voxels with 3D convolutions or rely on point-based convolutions directly. In this paper, we propose Bifusion, a bi-directional point-voxel fusion framework that aims to seamlessly fuse point and voxel-based implicit fields. Experiments demonstrate that the proposed Bifusion can better encode local geometry details and provide a significant performance boost over existing state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies** → **Shape modeling**; **Shape representations**; **Reconstruction**.

KEYWORDS

3D surface Reconstruction, Implicit Field, Deep Learning.

ACM Reference Format:
Chuanmao Fan, Chenxi Zhao, Kevin Xue, and Ye Duan. 2018. Point Voxel Bi-directional Fusion Implicit Field for 3D Reconstruction. In *GI'24: Graphics Interface 2024, June 03–06, 2024, Halifax, Nova Scotia, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

*Co-first author.
†Co-first author.
‡Corresponding author.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

3D modeling is a fundamental task in visual computing and finds applications in robotics, virtual reality, augmented reality, animation, and various other fields. In comparison to other 3D modeling approaches such as multi-view stereo[8, 16, 39, 40, 48] active sensing approaches that utilize LIDAR scanners for 3D acquisition have gained significant traction in the 3D modeling community, due to their higher robustness and accuracy. The availability of the newer, lower cost LIDAR scanners further boosts their popularity with the increased affordability. The output from LIDAR scanners consists of unorganized point clouds, as they are typically non-uniformly sampled from a large, continuous 3D space. These point clouds lack the regular grid structures that images enjoy, which prevents a direct application of grid-based convolution, a method relied upon by most deep learning techniques. To apply deep learning on point clouds, a straightforward approach is to first voxelize point clouds into volumetric 3D grid and then apply 3D convolutions [24, 27, 46] on the voxelized 3D grid. The voxelization process however comes with limitations such as a large memory footprint, and the discretization step inevitably leads to the loss of local geometric information. To address these challenges, researchers have proposed the use of special data structures such as Octree [37, 51] to adaptively partition the 3D space to reduce the memory usage. Another approach is to employ sparse convolutions to represent high-resolution volumes [11]. Recently, researchers have also proposed approaches that can directly process point clouds. One of the seminal works in this category is PointNet, introduced by Qi *et al.*[35]. PointNet is able to process unordered point cloud inputs with permutation invariance using a sequence of multi-layer perceptrons (MLP). The subsequent work PointNet++ [36] further enhances performance by introducing a hierarchical network that encodes local neighborhood information. Inspired by [54, 60], several newer networks were proposed such as PGCNN [52] and Pointweb [60] with more advanced local feature aggregation techniques, as well as newer kernel-based convolution approaches such as PointConv[54], KPconv[44], and Spherical kernel graph CNN[21] which try to mimic standard convolution on point clouds. Comparing with voxel-based approaches, point-based methods can better preserve fine-grained local information, but are generally less efficient in neighborhood range queries. On the other hand, voxel-based approaches can more efficiently encode global, multi-scale context information. They are also more efficient in neighborhood searches.

Recently researchers have also proposed methods to integrate the voxel-based approaches with the point-based approaches. For example, Liu *et al.*. [25] introduced Point-Voxel CNN (PVCNN) that represents 3D data as points to reduce memory footprint and leverages voxel-based convolution to capture neighborhood features. Their network achieves reasonable performance for 3D semantic segmentation with lower memory usage and faster training/inference speed. Shi *et al.*.[41] proposed PV-RCNN, which leverages both 3D convolutional network and PointNet[36] based voxel set abstraction for learning point cloud features. Point clouds are voxelized and feed into a 3D convolutional network to generate high-quality proposals. The voxel-wise features are then encoded into a small set of sampled key points. PV-RCNN achieved very good performance on 3D object detection benchmark dataset.

To date, many deep learning based surface reconstruction methods have been proposed. Among them, neural implicit field learning based methods have been very successful as they can represent both complex inner structures and open surfaces in a continuous implicit distance field. Existing state-of-the-art neural implicit distance field-based methods either utilize voxels with 3D convolutions entirely or solely rely on point-based convolutions. There are works that leverage point voxel fusion for points segmentation [53, 55, 59] and 3d object detection [49], while our work explore its usage in implicit field leaning for 3d surface reconstruction. To the best of our knowledge, no one has proposed exploring a hybrid point-voxel approach for surface reconstruction from point clouds. In this paper we propose a novel framework that aims to seamlessly fuse voxel-based implicit field and point-based implicit field which combines the merits of both volume representations and point representations. Figure 1 shows an illustration of the proposed Bifusion framework. It consists of four modules: a volumetric U-Net [38] for volume representation of the implicit field, a point-based U-Net for point representation, a point voxel fusion module for exchanging information between the two modalities, and a blending module that scores volume occupancy and point occupancy predictions and blend them to produce final occupancy. The fusion module integrates point and voxel features at every corresponding layer of U-net to exchange feature information during the feature encoding phase. Given an input point cloud, 1) it passes through a point convolution network to produce point features; 2) in parallel, it is voxelized into a canonical volume grid in volume branch and then fed into a 3D convolutional network to generate multiple volume features. 3) Query points extract multi-level volumetric features from volume branch. These features are concatenated and pass through MLPs to produce an occupancy prediction. 4) In the point branch, the query points search k-nearest neighbors points. The output, i.e., feature vectors, from the point convolution will then be passed into a volumetric grid via voxelization to conduct 3D volumetric convolution. 5) The result, i.e. feature vectors, of the 3D volumetric convolution will be fused back to the point cloud via devoxelization to conduct the next round of point convolution via MLPs. Steps 1 to 3 will repeat throughout the encoding phase of the network. The decoding phase of the network follows the traditional design of a standard U-Net [38] with skip connections. We have tested the proposed Bifusion framework on multiple benchmark datasets for 3D surface reconstruction and the results show the proposed Bifusion framework can better encode local geometry details and provide significant performance boost over existing state-of-the-art methods. To summarize, the main contributions of this paper are:

1) A novel point voxel fusion framework Bifusion that can seamlessly fuse point features and voxel-based convolution, which mutually support feature learning for occupancy predictions.

2) A blending module that relies on the properties of neighbor points to learn scores and blend the volume occupancy and point occupancy based on the confidence scores.

3) Experimental results demonstrate that the proposed Bifusion framework can better leverage the advantage of both point and volume representations and can provide significant performance boost over the existing state-of-the-art methods.

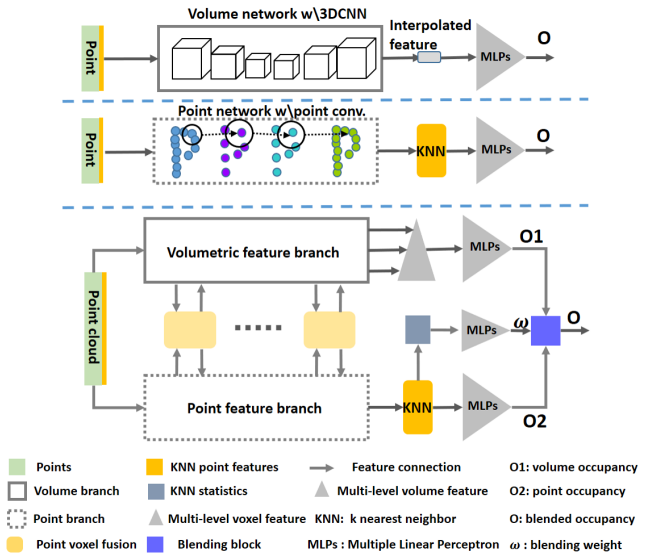

Figure 1: There exists two broadly used designs: one category use volume features, while the other category focus on point features for implicit field learning. Ours Bifusion network design fuses the two types of network together. The volume branch voxelizes the points to an occupancy volume, then feeds the volume to a volumetric feature network. The point branch infers point features. The fusion model between the point and volume branch feeds point feature to volume and volume feature to point which deeply fuse two types of features. For any query point, multi-level volumetric features are extracted from volume features and point features are aggregated from k-nearest neighbors (KNN) points. Then multi-level volumetric features are fed to a series of multiple linear perceptrons (MLPs) to predict volume occupancy. And KNN features are feed to a second series of MLPs to predict point occupancy. KNN points statistics and features are fed to a score network (MLPs) to predict the weight score which is used to blend the volume occupancy and point occupancy.

2 RELATED WORK

2.1 Neural implicit 3D modeling

Deep learning based methods have been very successful in surface reconstruction on many benchmark datasets. Among these methods, neural implicit field methods have been the most dominant approach as they can represent both complex inner structures and open surface in a continuous implicit distance field. AtlasNet [13], DeepSDF [31], Occupancy Network[28] stand out as some of the earliest endeavors to utilize deep learning for learning neural implicit representations in 3D modeling. These works usually encode the input into a latent code and concatenate it with the geometric coordinates of a given query point before they being fed together into a neural network. This network is often composed of several layers of multi-layer perceptron (MLPs), and outputs either the signed distance value or the occupancy probability for the query point. Specifically, AtlasNet [13] samples query points from a parametric representation of a surface patch, which is simple to implement, but requires handling the complexity of ensuring that overlapping patches have consistent output. DeepSDF [31] and Occupancy Network[28] share similar network designs. DeepSDF [31] estimates the signed distance value of a query point, and Occupancy Network[28] predicts the occupancy of a query point, i.e. whether the query point lies inside or outside the 3D surface, essentially treating it as a binary classification problem. Follow up works such as [1, 12, 43] further extends [5, 28, 31] with various modifications such as period activation functions [43], regularizing neural implicit functions by applying Eikonal equation and normal constraints[12], and training with level sets [1]. Another line of research build implicit surface field learning on Nerf [29] related theory. [50] et. al. relate the signed distance field to density of radiance field. thus once the radiance field optimization finishes, one can extract the underlying signed distance of the query point, which could be applied to extract surface. [30] et al. relate the density of readiance field to occupancy field instead, which could be used to extract sharper surface once optimized as density computed from occupancy is either small (transparent) from empty space with zero occupancy or large(totally opaque) from occupied space with one occupancy. There are following up works [23, 58] built on similar concept. However all these works requires multi-view posed images, and requires per-scene optimization, which limit their generalizability. Yet another branch of works utilize the current diffusion models [14] to generate 3d field or points for 3d surface reconstruction. LION [45] use point voxel network to infer a latent code from input, and apply diffusion on latent space, once trained, it scan be used to generate dense 3d points by manipulating the latent code. Surface reconstruction is carried out by differentiable Poisson solver: SAP [33]. Community [7, 42] also fuse the SDF field learning using diffusion model, which similarly can generate conditional implicit field. Generative 3d modeling can generate pleasant surface, however it is hard to geometrically align to the real scans. Thus it is usually targeted to artist.

2.2 Point based implicit field for 3D modeling

With the advent of methods that extract features directly from points such as PointNet, surface reconstruction from points has become feasible. Compared to the loss of details and artifacts caused by voxelization, point-based method could predict more realistic surfaces. By predicting occupancies or distances to the surface for each query point, marching cubes[26] or similar methods are applied to extract surfaces. Points2surf[10] presents a network learns the absolute distance of sign distance field (SDF) and the logit of the sign probability simultaneously. Given a point cloud and query point, some patches with difference distance to the query point are encoded into feature vectors and decoded into SDF. POCO[2] presents an attention-based decoder architecture for any point cloud segmentation backbone and predicts the occupancy for any query point. [15] proposed a feature aggregation module for local features of neighboring input point clouds. Other methods like [4], [9] are also point-based implicit fields methods. But they have restrictions on specific data category.

2.3 Volume based implicit field for 3D modeling

The voxel representation of point cloud is the earliest adapted representation in the field of deep learning due to its simplicity and the accessibility of common convolution operations. A pre-defined volume grid is adopted to cover the point cloud usually. Occupied voxels are filled with non-zeros vectors, and empty voxels are filled with zero vectors. VoxelNet[62] is the first work to voxelize points to volume representation for object detection. Pointgrid[20] is a follow up work which randomly pick a pre-defined number of points for a occupied voxel. Occupancy Network [28] is among the first work to use volume representation to learn features for implicit field for 3d reconstruction. Points are voxelized to occupancy volumes where occupied voxels are filled with ones, and empty voxels are filled with zeros. Convolution is applied to volume to produce volume features. IFNet [6] utilizes multi-level learning to obtain a rich encoding of the data. Convolutional occupancy Networks[34] similarly utilize voxelization as a pre-processing method to handle point clouds, representing another approach for transforming unstructured data into structured data. All though these voxel based methods could generate smooth surface after applying marching cubes or other similar mesh generation algorithms, directly reforming points into voxels may lose the details inevitably.

3 METHODS

3.1 Overview

In a common network design of neural implicit field, a point cloud P is often voxelized into a volume V of resolution such as 32^3, 64^3, 128^3. The field prediction from volume features is robust due to the stable feature propagation spatially. However, voxelization lacks geometry details, and the point-to-volume discretization error limits the accuracy of 3D modeling. While point-only implicit fields utilize original shape feature, which have the potential to generate accurate prediction. However, query point's k-nearest neighbors features are often used for subsequent field inference. The neighborhood geometric support range depends on relative distance of query points to input points; neighbor points of a query point cover different regions of input, which could lead to ambiguous prediction. By considering these issues, we propose to fuse points feature and volume features bi-directionally to complement each other. Section 3.2 explains the details of bi-direction fusion block. Section 3.3 explains the blending based implicit field decoder. As we have

two occupancy predictions from volume branch and point branch respectively. The two are blended based on a learned weight to produce final result. In section 3.4, we present details about data preparation, training and inference.

3.2 Bi-directional feature fusion block

The network contains two main parts as shown in figure 1. The first part is the feature encoder which is based on a bi-directional point and voxel feature fusion block. While the second part is a coordinate based network that combine query points' coordinates and corresponding features to infer occupancy. The feature encoder includes three sub-parts: volumetric feature encoding, point feature encoding, and a layered point-volume bi-directional fusion. The uniqueness of the feature encoding network lies in its layered bi-directional fusion modules, which feed volume feature of volume branch to point branch and vice versa. Please refer to figure 2 for network design details.

The volume feature encoding network design follows U-net structures [38]. Given a point cloud of N points $P\{p_1, p_2, ..., p_n\} \in \mathbb{R}^3$, it is first normalized into range [-0.5, 0.5]. The normalized points are then voxelized into volume of resolution L. Points within each voxel are averaged and centered around the voxel, while empty voxels are filled with zeros. The voxelized points are then fed into the volumetric U-net. The network begins with volume resolution of L in each dimension. Through each layer's operation, it gradually decreases to $\frac{L}{2}$ ..., $\frac{L}{2^n}$ and then up backs to L symmetrically. Skip connections are applied between the symmetric layers. The features of each level are represented as$[F'_{v0}, ..F'_{vn}, F_{vn}...F_{v0}]$, whose length of features gradually increases from contraction to bottleneck as it encode larger scale context gradually, then decreases gradually from bottleneck to expansion layers. Given a query point X, multi-scale features are extracted from expansion layers by applying trilinear interpolations. We denote the volume features as:

$$G_v(X) = F_{v1}(X), F_{v2}(X), ...F_{vn}(X) \tag{1}$$

The point feature encoding module also follows a U-shaped structure. We construct the network by following PointNet++'s[36] feature abstraction and propagation blocks which contains three modules for each point encoding block: sampling, grouping and feature extraction. A fixed number of centroids of local regions which are distant enough from each other are chosen through the furthest point sampling(FPS); K-nearest neighbors(KNN) search method groups a certain number of neighboring points; the features of these local regions are extracted through convolutional layers. In the propagation module the features of all the points are obtained by interpolation based on distance weight. It should be noted that any point cloud feature learning network could be utilized in the point branch. We choose PointNet++ considering its usability and accessibility. Given an arbitrary query point X(x,y,z),its K-nearest neighbors points' features are inquired from the last layer. And We denote these features as:

$$G_p(X) = F_{p1}(X), F_{p2}(X), ...F_{pk}(X) \tag{2}$$

The layered point voxel feature fusion module is to bring the merits of both points and volume representations to each other. Figure 3 illustrates the design of the fusion block, which can be understood as a multiple-in-multiple-out(MIMO) block. The volume encoder and points encoder are mirror symmetric, and the point voxel fusion block appears in each layer pair. For each layer pair, points at i_{th} layer with features F_{pi} is voxelized into volume of corresponding volume layer of resolution $\frac{L}{2^i}$ by averaging point features within a voxel. It is represented as F_{pvi}. Correspondingly, i_{th} layer volume features are propagated to paired point layer by extracting feature via trilinear interpolation, represented as F_{vpi}. F_{pvi} and F_{vpi} are fused into i_{th} volume feature and point feature respectively.

$$F'_{vi} = F_{vi} + F_{pvi} \tag{3}$$

$$F'_{pi} = F_{pi} + F_{vpi} \tag{4}$$

It should be noted that the fusion module design is open to other forms. As long the design follows the MIMO, it can be adapted to the network seamlessly.

3.3 Blending based implicit field decoder

A query point with features from the encoder can be fed into the coordinate-based field decoder network to produce occupancy, typically using Multiple Linear Perceptrons (MLPs) as the decoder network. Our network has both volume and points branches, each with its own decoder.

In the volume branch, We extract the query point's multi-level features $G_v(X)$ from volumetric features as stated in equation 1. The features are concatenated and fed to an MLP f_v to predict occupancy.

$$O_v(X) = f_v(X, F_{v1}(X), F_{v2}(X), ...F_{vn}(X)) \tag{5}$$

In the point branch, we search for the K-nearest neighbors (KNN) points of query point X at the last point layer and collect neighbor features $G_p(X)$ correspondingly, as stated in Equation 2. We concatenate the relative points difference $X - P_{in}$, where i = 1,...K among the query and K neighbor points, and $F_{pi}(X)$, where i = 1,...K. We then feed this concatenated information to the point-side field decoder MLPs f_p:

$$O_p(X) = f_p(X - P_{1n}, F_{p1}(X), ...X - P_{kn}, F_{pk}(X)) \tag{6}$$

The same KNN features of Equation 6 are also fed into a third weighting network consisting of MLPs. However, the last layer applies a sigmoid activation function to produce the blending weight $\omega_X = f_w(X - P_{in}, F_{pi}(X)) \in [0, 1]$. This design is based on the consideration that query point and neighbor properties such as relative distance, neighbor points distributions, geometry, etc., of both the query point and its neighbors contain uncertainty information about existing occupancy inference. The weight blends the two occupancies of point and volume branch predictions as follows:

$$O(X) = \omega O_p(X) + (1 - \omega)O_v(X) \tag{7}$$

3.4 Data preparation, training and inference

To train our network, we prepare data with watertight meshes. The data include input surface points, query points, and ground truth occupancy of query points. In order to get data, the meshes are first normalized to a unit sphere. Then we sample N points on surface. N could be 300, 3,000, 10,000 in our preparation. For query points, we sample 50,000 points on mesh surface, then add Gaussian noise offset with standard deviation σ = 0.003, 0.05, 0.5 respectively

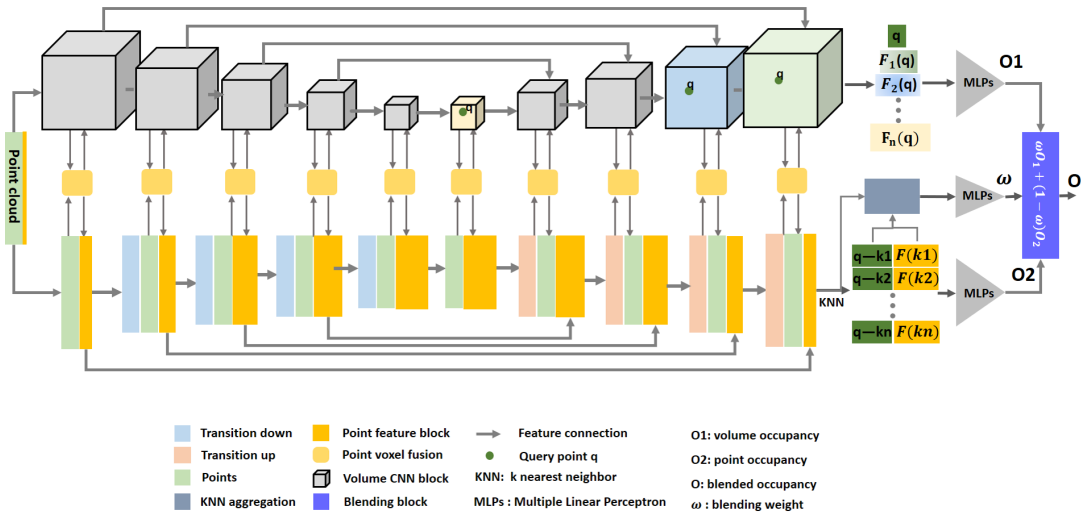

Figure 2: The network architecture: Input points attached with ones (input features) are fed into the volume U-net branch and the point U-net branch respectively. In volume branch, the input is voxelized based on point locations. Points within a voxel are voxel-centralized and averaged, and then undergo a series of 3D convolutional neural layers, which generate multiple level of volume features. In the point convolution branch, inputs traverse through a U-net shaped point neural network. Blocks of the point neural net include transition down and transition up which is to down-sample and up-sample points, and pool-down and interpolate-up features correspondingly among layers. The blocks also include point feature grouping and aggregations to infer high level features. Point U-net is mirror-symmetric with volume U-net; the mirror-paired layers' features are fused by a point-voxel fusion block. In the volume branch, a query point extract multi-level features from volume features. The features and query point are concatenated and passed to an implicit field decoder consisting of a sequence of Multiple Linear Perceptrons(MLPs) to produce occupancy O1 for the query point. In point branch, the K-nearest neighbors of query point are aggregated and passed to another set of MLPs to infer occupancy O2. Same features also produce blending weight ω via a third set of MLPs, which blend O1 and O2 to generate the final occupancy prediction.

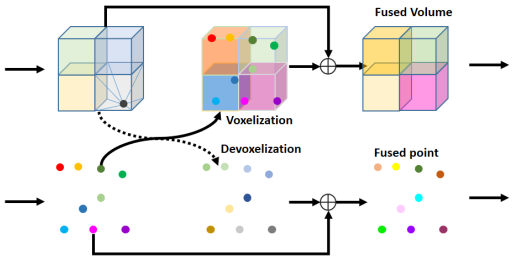

Figure 3: The structure of bi-directional point voxel feature fusion block, Point voxelization averages points' features within a voxel. Devoxelization uses trilinear feature interpolation to extract feature from voxel to point.

to get three groups of query points for each mesh. Ground truth occupancy of query points are computed by computing the sign of distance from query points to mesh, occupancy is assigned as -1 if sign is negative, otherwise 1.

All the models are implemented with Pytorch [32]. For training, we use Adam optimizer [18] with parameters $\beta 2 = 0.9$, $\beta 2 = 0.999$, and an initial learning rate of 1.0 10^{-4}. The learning rate reduces by 0.1 × with step scheduler setting at 50 and 100 epochs, respectively.

All the training and testing are conducted with a desktop computer with an Nvidia RTX 3090 GPU with 24 GB memory. In the training process, a batch of N input surface points, 50,000 query points with ratio 15:35:50 sampled from the three groups and corresponding ground truth occupancy are fed to the network. The optimizer will minimize the following binary cross entropy (BCE) loss to optimize the network:

$$Loss = -\frac{1}{BN}\sum_{b=1}^{B}\sum_{i=1}^{N} BCE(O_{bi}, O_{bi,gt}) \tag{8}$$

Where O_{bi} is the predicted occupancy and $O'_{bi,gt}$ is the ground truth occupancy.

For inference, we create a grid of query points and evaluate occupancy values on these grid points using the network. We can define any resolution of grid, but considering fair comparison with existing methods and processing time, we specifically define a mesh grid of resolution 256^3 within the range [-0.5, 0.5]. Then, occupancy values for all grid points are evaluated, which generates an explicit occupancy field. Subsequently, marching cubes [26] with a threshold of 0 is applied to the occupancy field to extract a triangular mesh surface.

For evaluation, we utilize intersection over union (IOU) between the ground truth mesh and the prediction mesh. Additionally, Chamfer distance (CD), F-score, and Normal completeness (NC) are used

Table 1: ABC, Famous, Thingi10k. Training on ABC shapes with 10 scans, variable Gaussian noise (uniformly picked in [0, 0.05L], L largest box length). Chamfer distance × 100 on ABC, Famous and Thingi10k test sets, as prepared by point2surf: 'no-n.' (no noise), 'var-n.' (variable noise, as training), 'max-n.' (= 0.05L), 'med-n.' (= 0.01L), 'sparse' (5 scans), 'dense' (30 scans). Only SPR uses normals.

	ABC(100 shapes)			Famous (22 shapes)					Thingi10k(100 shapes)				
Method	no-n.	var-n.	max-n.	no-n.	med-n.	max-n.	sparse	dense	no-n.	med-n.	max-n.	sparse	dense
DeepSDF[31]	8.41	12.51	11.34	10.08	9.89	13.17	10.41	9.49	9.16	8.83	12.28	9.56	8.35
AtlasNet[13]	4.69	4.04	4.47	4.69	4.54	4.14	4.91	4.35	5.29	5.19	4.90	5.64	5.02
SPR[17]	2.49	3.29	3.89	1.67	1.80	3.41	2.17	1.60	1.78	1.81	3.23	2.35	1.57
Points2Surf[10]	1.80	2.14	2.76	1.41	1.51	2.52	1.93	1.33	1.41	1.47	2.62	2.11	1.35
POCO[2] 3k	1.87	2.26	2.90	1.56	1.75	2.99	1.99	1.70	1.47	1.64	3.21	2.00	1.55
POCO 10k	1.72	2.15	2.72	1.57	1.61	3.04	1.92	1.57	1.50	1.57	2.82	2.08	1.51
Bifusion 3k	0.63	1.96	3.06	0.52	0.93	3.16	1.03	1.19	0.36	1.02	2.91	1.11	1.16
Bifusion 10k	**0.58**	**1.50**	**2.43**	**0.50**	**0.86**	**2.19**	**0.87**	**0.96**	**0.33**	**0.83**	**2.26**	**0.96**	**1.09**

to evaluate the network performance. Further details about the evaluations are explained in the supplemental materials.

4 EXPERIMENTS

We conducted an extensive study of our network using different datasets, fusion regimes, and blend methods. We followed state-of-the-art methods for experimental configurations to ensure a fair comparison. The results demonstrated that the bi-directional fusion network design produced improved results.

4.1 Datasets, metrics and baselines

ABC[19] is a collection of one million Computer-Aided Design (CAD) models for research of geometric deep learning methods and applications. We pick 3182 watertight shapes for training, 796 for validation, and 100 for testing.

Famous22[10] is a dataset contains 22 shapes from various origins. We use all the data prepared by [10] for testing only.

Thingi10k[61] contains 100 shapes prepared by[10]. We use all the data for testing only.

ShapeNet[3] contains shapes from 13 categories. Due to the large size of the dataset we use one category of the dataset, car for training and testing.

THuman 2.0[57] contains 500 high-quality human scans captured by a dense DLSR rig. We select 369 shapes for training, 52 for validation, and 105 for testing.

The **metrics** include volumetric IOU, CD $l1 \times 10^{-2}$ and CD $l2 \times 10^{-4}$, NC and F-score with 1% threshold.

We choose POCO [2] and IFNet [6] as our **baselines**, considering that they demonstrated state-of-the-art results on occupancy field. Besides, we also did ablation studies with volume-only and points-only implicit field networks for demonstrating advantage of bi-directional fusion.

4.2 Reconstruction

Table 1 shows the quantitative results on ABC, famous and Thingi10k from various baselines. Some of the numbers are directly referred from [2]. The training process was on our selected ABC dataset and testing datasets are all from Point2Surf [10]. Our method outperform all the baselines on these evaluations. Figure 4 illustrates the reconstruction results with 3K and 10K input points. Corresponding POCO [2] results are generated with pre-trained model obtained from official web page. Compared with POCO [2], it could be clearly seen that ours could reconstruct more details. For comparison with a latest work [22], we process ShapeNet cars 3000 input points using its pre-trained model. Table 4 shows the comparison. From left to right, these metrics are mean IOU, mean NC, mean F-Score, mean Chamfer distance l_1 scores $(\times 10^{-3})$, mean Chamfer distance l_2 scores $(\times 10^{-4})$. It indicates that IOU of GridFormer is slightly better than ours, Bifusion outperforms all other items.

It is worth mentioning that some of our metrics from datasets with fewer input points still show better performance compared to results obtained by other baselines with more input points.

4.3 Ablation studies

To validate our network design, we conducted two ablation studies.

In the first ablation study, we investigated the fusion designs and blending designs. Table 5 includes two network design besides Bifusion (v1). Bifusion base (v0) network use average of volume branch occupancy output O1 and point branch occupancy O2 instead of learnable blending. It turns out that learned blending weighing based on local neighbor geometry properties tends to produce significantly better results, which validates our conjecture. Bifusion v2 incorporates a learnable bi-directional fusion block, where the point and volume features are updated with an attention module before being sent out MIMO. Bifusion v2 has a more complex design than v1 but shows similar performance. Please refer to the supplemental materials for detailed structure design.

The second ablation study aimed to verify whether bi-directional fusion improves field learning. This study involved comparing two base designs: one with a volume-only network, Volume Base; and another with a point-only network. Point Base. These designs were derived from the volume branch and point branch of the Bifusion network, respectively. In Table 3, the qualitative results for the Point Base and Volume Base networks are presented. The study was conducted using THuman2.0. We conclude that the Point Base network has the worst performance, while the Volume Base network is significantly better than the Point Base one. Furthermore, bi-directional fusion boosts the performance by a large margin. We

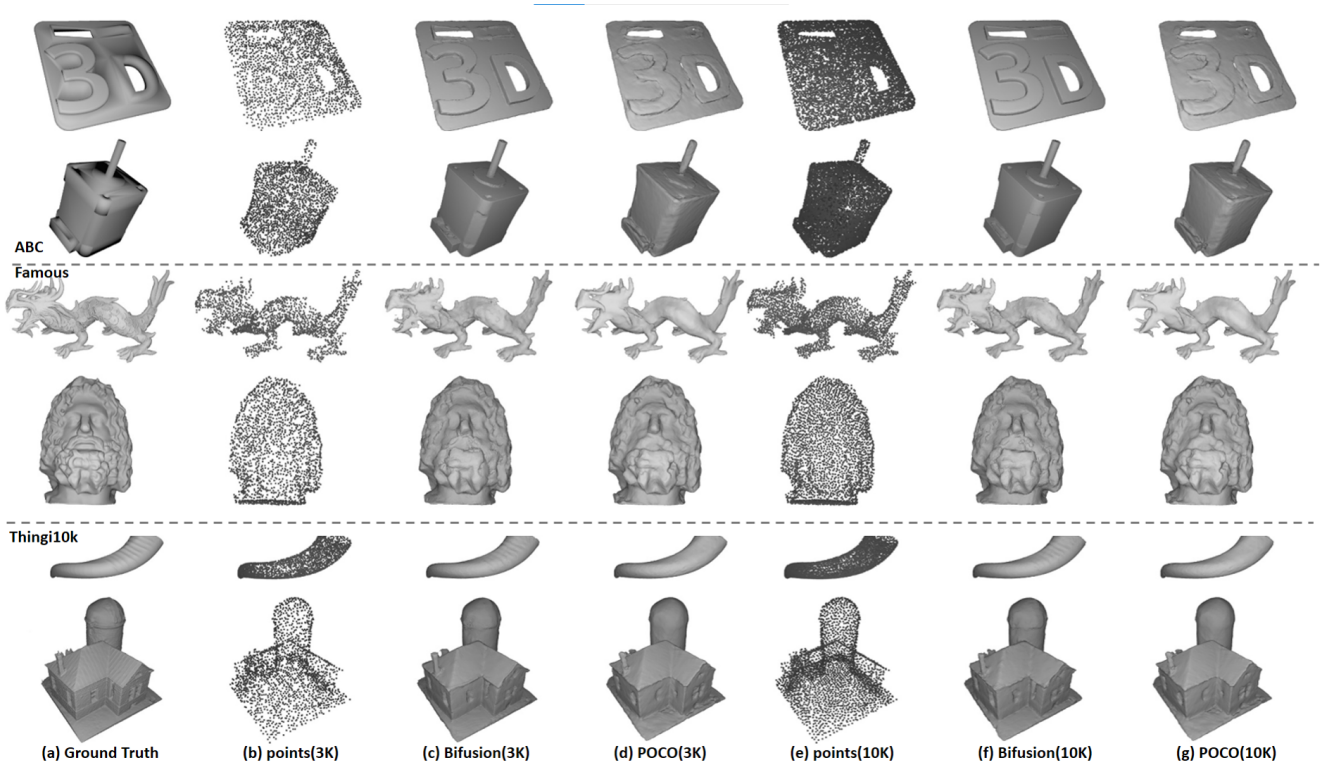

Figure 4: Selected ABC, Famous and Thingi10K reconstruction results with 3K point and 10K point respectively. corresponding POCO [2] results are generated with official pre-trained models.

Table 2: Ablation study with different fusion design architecture quantitative comparisons on the watertight ShapeNet cars dataset with both 3000 and 300 input points. Chamfer distance l_2 scores $(\times 10^{-5})$ of the mean (left column) and median (right column) are shown. Bifusion v0 average O1 and O2 of the two occupancy outputs instead 1. Bifusion v1 is with learnable bi-directional fusion block, where the point and volume features are updated before sending out MIMO. Bifusion v2 is our demonstrated design. For a more complete comparison, we also tested Bifusion(MLPs), which instead of using the point convolution branch of Bifusion v2, use plain MLPs like PVCNN [41]

	IOU		NC		F-score		CD l_1 $\times 10^{-3}$		CD l_2 $\times 10^{-5}$	
input points	3000	300	3000	300	3000	300	3000	300	3000	300
Bifusion v0	0.92	0.823	0.940	0.866	0.980	0.864	2.77	5.57	2.91	9.02
Bifusion v1	0.91	0.829	0.944	0.872	0.980	0.871	2.84	5.36	3.01	7.94
Bifusion v2	0.92	0.850	0.942	0.891	0.982	0.895	2.67	4.81	1.91	6.73
Bifusion v2 (w/MLPs)	0.85	0.840	0.909	0.888	0.969	0.879	3.98	5.52	19.25	8.05

attribute the poor performance of the Point Base network to the unstable nearest neighborhood of the query point. Due to the stable feature propagation spatially in volume features, volume-based methods like the Volume Base and IFNet [43] tend to yield better results. Figure 6 demonstrates the experimental results, supporting our conclusion.

We also investigated the runtime overhead of different designs as shown in Table 5. We computed the encoding time, decoding time and meshing time (using the marching cubes method) for Point Base network, Volume Base network, and Bifusion. The results for different resolutions—64, 128, and 256—are presented in the table. The decoder is the most time-consuming component. For the other processes, the runtime for all three methods is almost less than one second. At resolutions 64 and 128, the difference in runtime between the two base networks and Bifusion is negligible.

4.4 Conclusion and discussion

We proposed a bi-directional point-voxel fusion framework aimed at seamlessly fusing point and voxel-based implicit fields. Our method can generate high-quality 3D meshes and outperforms other recent methods by producing smooth surfaces with rich details. The performance quality is however still restricted by resolution. High resolution may result in long running times and require large GPU

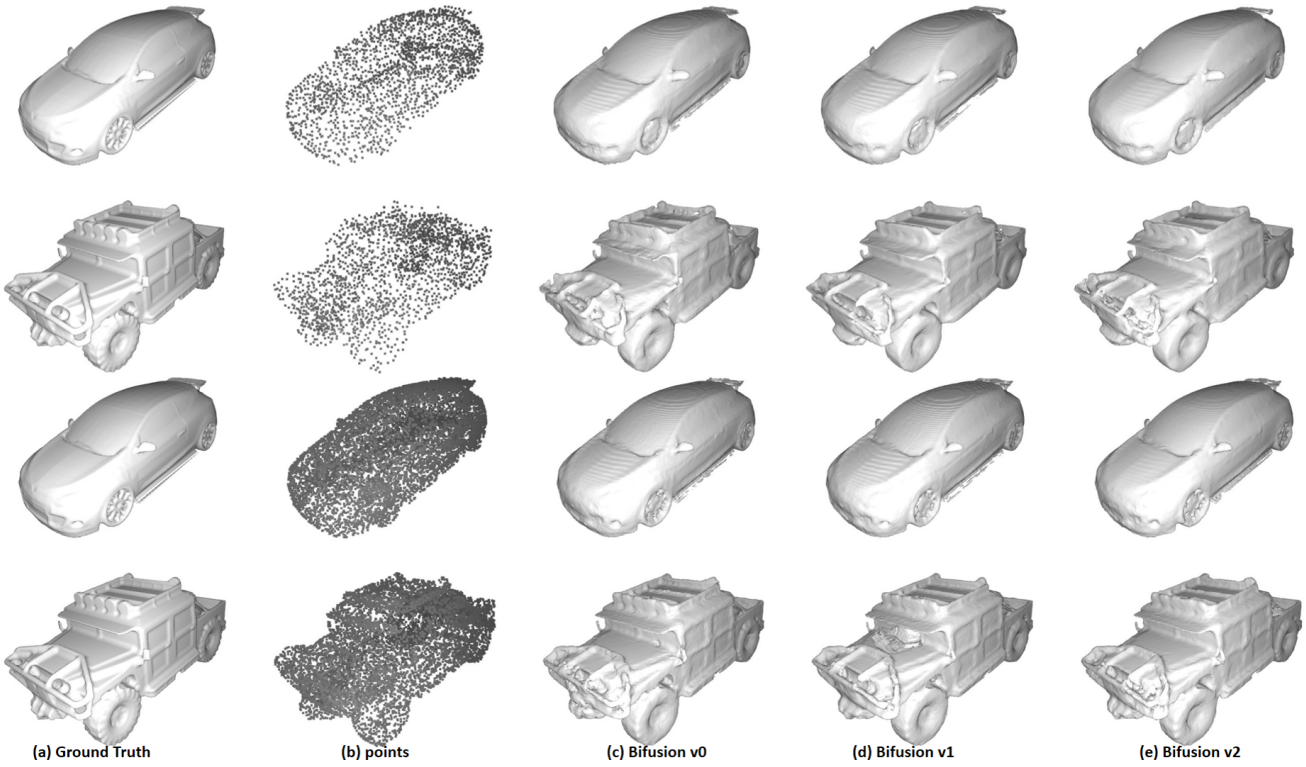

Figure 5: Ablation study with various network designs, Bifusion v0 use simple average of occupancy O1 of volume branch and occupancy O2 of point branch. i.e. *omega* = 0.5. Bifusion v1 use complex fusion module instead of the version shown in 3. Please refer to the supplemental materials for detailed structure design. Bifusion v2 refers to the main network demonstrated in this manuscript of figure 1 which uses fusion modules of figure 3. We basically can see their visual qualities are close with using 3K input points, Bifusion v2 shows robust results with 300 input points.

Table 3: THuman2.0 human body shape reconstruction. The results are tested with 3000 input points. IFNet is trained by us. POCO result is tested with pre-trained model using ShapeNet. From left to right, these metrics are mean IOU, mean NC, mean F-Score, mean Chamfer distance l_1 scores $(\times 10^{-3})$, mean Chamfer distance l_2 scores $(\times 10^{-4})$. Point base is our point branch of Bifusion, volume base is the volume branch network of Bifusion.

methods	IOU	NC	F Score	CD l_1	CD l_2
IFNet	0.951	0.920	0.998	2.07	0.736
POCO	0.919	0.893	0.989	2.90	1.31
Point base	0.943	0.889	0.958	3.65	12.5
Volume base	0.951	0.924	0.998	2.12	1.12
Bifusion	**0.977**	**0.941**	**0.999**	**1.63**	**0.65**

Table 4: Comparision with GridFormer on ShapeNet cars dataset with 3000 input points.

	IOU	NC	F-score	CD l_1	CD l_2
Gridformer	**0.928**	0.922	0.945	4.14	5.944
Bifusion v2	0.920	**0.942**	**0.982**	**2.67**	**1.91**

memory. To the authors' knowledge, this remains a major challenge for 3D reconstruction methods. Additionally, our method may fail when there are hollows on the object's surface, such as shutters. These hollows can cause issues with the sampling point cloud on the surface, leading to unreliable obtained features. In the future, we would like to explore attention-based decoders for the implicit field to further enhance the performance.

Besides the 3D reconstruction tasks, we also see many potentials of the pipeline utilized in other downstream tasks such as joint 3d Reconstruction and point segmentation, or motion field [47], where volume branch is used for 3d reconstruction task and point branch can be used for segmentation or motion field prediction. considering the speed of inference, it has the potential to apply to dynamic 3D reconstruction [47, 56].

ACKNOWLEDGMENTS

This research was partially supported by the National Science Foundation under award CNS-2018850, the National Institute of Health under awards NIBIB-R01-EB02943, and U.S. Army Research Laboratory W911NF2120275. Any opinions, findings, and conclusions

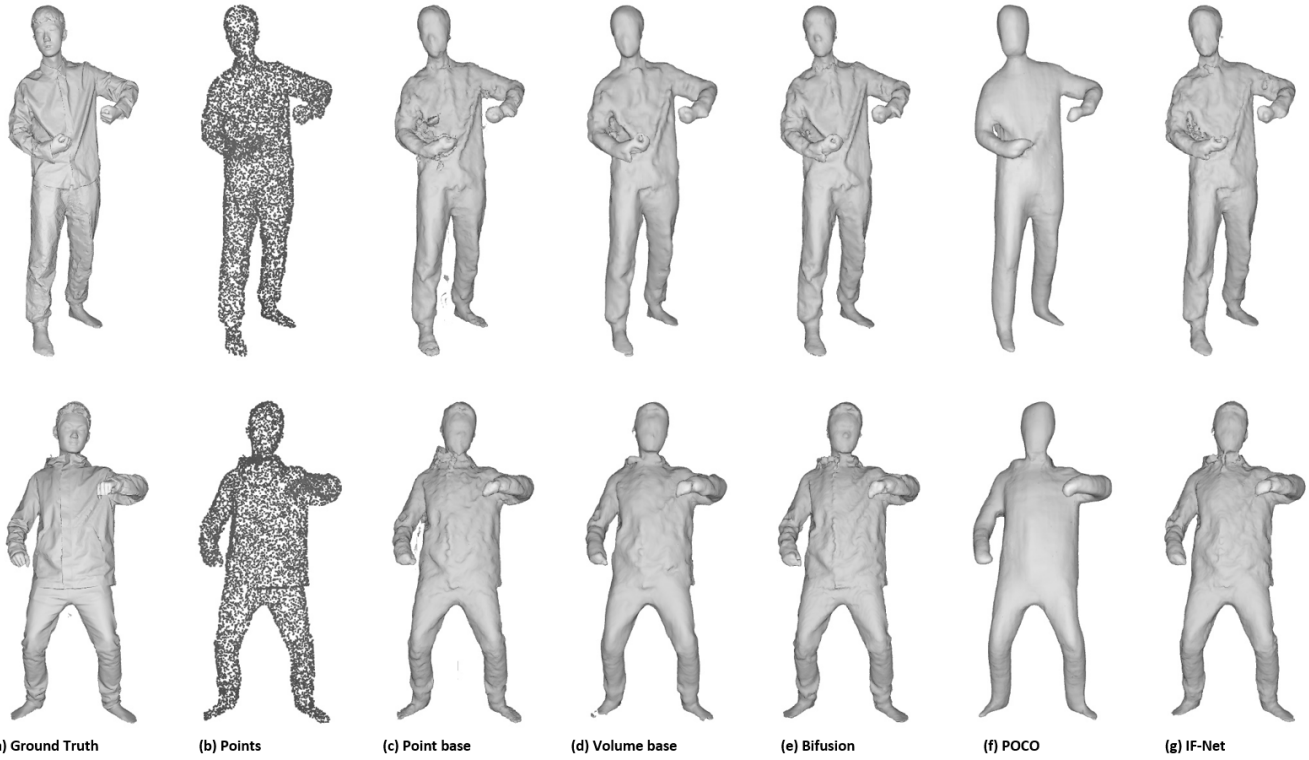

Figure 6: THuman 2.0 reconstruction results with 10K input points. IFNet model[43] is trained by us. POCO [2] results are generated by official provided pre-trained model on ShapeNet.

Table 5: Mean run time (in seconds) comparison among point base network, volume base network, Bifusion v2 and Bifusion v2 using MLPs instead in points branch. The tests are done with 3000 input points, and reconstruction volume grid resolution 64^3, 128^3 and 256^3 respectively. The running stages for the test include feature encoding, field decoding, and meshing with marching cube algorithm. The run time is evaluated with 100 samples.

query grid resolution	64^3			128^3			256^3		
test stage	encoding	decoding	meshing	encoding	decoding	meshing	encoding	decoding	meshing
point base	0.014	0.446	0.036	0.015	0.341	0.224	0.014	2.635	1.518
volume base	0.024	0.052	0.027	0.023	0.204	0.150	0.023	1.354	1.075
Bifusion v2	0.029	0.174	0.013	0.030	1.220	0.100	0.030	9.554	0.883
Bifusion v2(wMLPs)	0.032	0.235	0.010	0.032	1.500	0.053	0.032	11.051	0.403

or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U. S. Government or agency thereof.

REFERENCES

[1] Matan Atzmon, Niv Haim, Lior Yariv, Ofer Israelov, Haggai Maron, and Yaron Lipman. 2019. Controlling neural level sets. *Advances in Neural Information Processing Systems* 32 (2019).

[2] Alexandre Boulch and Renaud Marlet. 2022. Poco: Point convolution for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6302–6314.

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

[4] Zhaiyu Chen, Hugo Ledoux, Seyran Khademi, and Liangliang Nan. 2022. Reconstructing compact building models from point clouds using deep implicit fields. *ISPRS Journal of Photogrammetry and Remote Sensing* 194 (2022), 58–73.

[5] Zhiqin Chen and Hao Zhang. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5939–5948.

[6] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. 2020. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6970–6981.

[7] Gene Chou, Yuval Bahat, and Felix Heide. 2023. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2262–2272.

[8] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 2016. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer, 628–644.

[9] Shivam Duggal, Zihao Wang, Wei-Chiu Ma, Sivabalan Manivasagam, Justin Liang, Shenlong Wang, and Raquel Urtasun. 2022. Mending Neural Implicit Modeling for 3D Vehicle Reconstruction in the Wild. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 1900–1909.

[10] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. 2020. 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, 108–124.

[11] Ben Graham. 2015. Sparse 3D convolutional neural networks. *arXiv preprint arXiv:1505.02890* (2015).

[12] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. 2020. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099* (2020).

[13] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 2018. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 216–224.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[15] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. 2020. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6001–6010.

[16] Abhishek Kar, Christian Häne, and Jitendra Malik. 2017. Learning a multi-view stereo machine. *Advances in neural information processing systems* 30 (2017).

[17] Michael Kazhdan and Hugues Hoppe. 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)* 32, 3 (2013), 1–13.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. 2019. ABC: A Big CAD Model Dataset For Geometric Deep Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[20] Truc Le and Ye Duan. 2018. Pointgrid: A deep network for 3d shape understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 9204–9214.

[21] Huan Lei, Naveed Akhtar, and Ajmal Mian. 2020. Spherical kernel for efficient graph convolution on 3d point clouds. *IEEE transactions on pattern analysis and machine intelligence* 43, 10 (2020), 3664–3680.

[22] Shengtao Li, Ge Gao, Yudong Liu, Yu-Shen Liu, and Ming Gu. 2024. GridFormer: Point-Grid Transformer for Surface Reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[23] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. 2023. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8456–8465.

[24] Yiyi Liao, Simon Donne, and Andreas Geiger. 2018. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2916–2925.

[25] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. 2019. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems* 32 (2019).

[26] William E Lorensen and Harvey E Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics* 21, 4 (1987), 163–169.

[27] Daniel Maturana and Sebastian Scherer. 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 922–928.

[28] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4460–4470.

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[30] Michael Oechsle, Songyou Peng, and Andreas Geiger. 2021. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5589–5599.

[31] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 165–174.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[33] Songyou Peng, Chiyu "Max" Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. 2021. Shape As Points: A Differentiable Poisson Solver. (Jun 2021). http://arxiv.org/abs/2106.03452v2

[34] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. 2020. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 523–540.

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

[36] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* 30 (2017).

[37] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. 2017. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3577–3586.

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

[39] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2304–2314.

[40] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. 2020. PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization. In *CVPR*.

[41] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10529–10538.

[42] Jaehyeok Shim, Changwoo Kang, and Kyungdon Joo. 2023. Diffusion-based signed distance fields for 3d shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20887–20897.

[43] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. 2020. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems* 33 (2020), 7462–7473.

[44] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6411–6420.

[45] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. 2022. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems* 35 (2022), 10021–10039.

[46] Rahul Venkatesh, Sarthak Sharma, Aurobrata Ghosh, Laszlo Jeni, and Maneesh Singh. 2020. Dude: Deep unsigned distance embeddings for hi-fidelity representation of complex 3d surfaces. *arXiv preprint arXiv:2011.02570* (2020).

[47] Tuan-Anh Vu, Duc Thanh Nguyen, Binh-Son Hua, Quang-Hieu Pham, and Sai-Kit Yeung. 2022. Rfnet-4D: joint object reconstruction and flow estimation from 4D point clouds. In *European Conference on Computer Vision*. Springer, 36–52.

[48] Dan Wang, Xinrui Cui, Xun Chen, Zhengxia Zou, Tianyang Shi, Septimiu Salcudean, Z Jane Wang, and Rabab Ward. 2021. Multi-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5722–5731.

[49] Ke Wang and Zhichuang Zhang. 2022. Point-Voxel Fusion for Multimodal 3D Detection. In *2022 IEEE Intelligent Vehicles Symposium (IV)*. 1716–1719. https://doi.org/10.1109/IV51971.2022.9827226

[50] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689* (2021).

[51] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. 2017. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)* 36, 4 (2017), 1–11.

[52] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)* 38, 5 (2019), 1–12.

[53] Zongshun Wang, Ce Li, Jialin Ma, Zhiqiang Feng, and Limei Xiao. 2024. PVI-Net: Point–Voxel–Image Fusion for Semantic Segmentation of Point Clouds in Large-Scale Autonomous Driving Scenarios. *Information* 15, 3 (2024). https://doi.org/10.3390/info15030148

[54] Wenxuan Wu, Zhongang Qi, and Li Fuxin. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 9621–9630.

[55] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. 2021. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16024–16033.

[56] Yuxin Yao, Siyu Ren, Junhui Hou, Zhi Deng, Juyong Zhang, and Wenping Wang. 2024. DynoSurf: Neural Deformation-based Temporally Consistent Dynamic Surface Reconstruction. *arXiv preprint arXiv:2403.11586* (2024).

[57] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. 2021. Function4D: Real-time Human Volumetric Capture from Very Sparse Consumer RGBD Sensors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2021)*.

[58] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. 2022. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems* 35 (2022), 25018–25032.

[59] Feihu Zhang, Jin Fang, Benjamin Wah, and Philip Torr. 2020. Deep FusionNet for Point Cloud Semantic Segmentation. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 644–663.

[60] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. 2019. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5565–5573.

[61] Qingnan Zhou and Alec Jacobson. 2016. Thingi10K: A Dataset of 10, 000 3D-Printing Models. *ArXiv* abs/1605.04797 (2016). https://api.semanticscholar.org/CorpusID:39867743

[62] Yin Zhou and Oncel Tuzel. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4490–4499.