Neural Vector Fields: Implicit Representation by Explicit Learning

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

Abstract

Deep neural networks (DNNs) are widely applied for nowadays 3D surface reconstruction tasks and such methods can be further divided into two categories, which respectively warp templates explicitly by moving vertices or represent 3D surfaces implicitly as signed or unsigned distance functions. Taking advantage of both advanced explicit learning process and powerful representation ability of implicit functions, we propose a novel 3D representation method, Neural Vector Fields (NVF). It not only adopts the explicit learning process to manipulate meshes directly, but also leverages the implicit representation of unsigned distance functions (UDFs) to break the barriers in resolution and topology. Specifically, our method first predicts the displacements from queries towards the surface and models the shapes as Vector Fields. Rather than relying on network differentiation to obtain direction fields as most existing UDF-based methods, the produced vector fields encode the distance and direction fields both and mitigate the ambiguity at "ridge" points, such that the calculation of direction fields is straightforward and differentiation-free. The differentiation-free characteristic enables us to further learn a shape codebook via Vector Quantization, which encodes the cross-object priors, accelerates the training procedure, and boosts model generalization on cross-category reconstruction. The extensive experiments on surface reconstruction benchmarks indicate that our method outperforms those state-of-the-art methods in different evaluation scenarios including watertight vs non-watertight shapes, category-specific vs category-agnostic reconstruction, category-unseen reconstruction, and cross-domain reconstruction. Our code will be publicly released.

1. Introduction

Reconstructing continuous surfaces from unstructured, discrete and sparse point clouds is an emergent but non-trivial task in nowadays robotics, vision and graphics applications, since the point clouds are hard to be deployed into the downstream applications without recovering to high-resolution surfaces [5, 7, 38, 42].

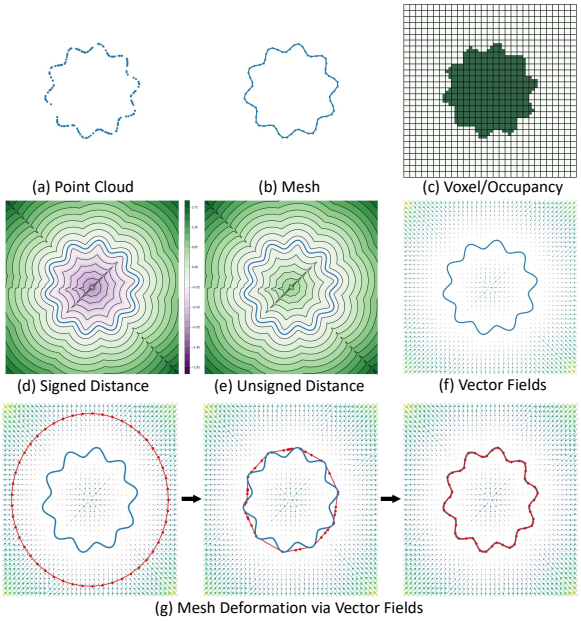


Figure 1. Common 3D representations. Explicit representations: (a) point clouds, (b) meshes, (c) voxels. Implicit representations: (c) occupancy, (d) reconstruction from the signed distance functions, and (e) reconstruction from unsigned distance functions. Our method represents continuous surfaces through (f) vector fields. (g) Vector fields can deform meshes (red) as explicit representation methods.

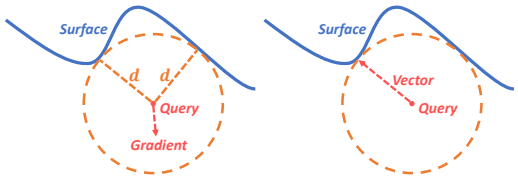
With the tremendous success of Deep Neural Networks (DNNs), a few DNN-based surface reconstruction methods have already achieved promising reconstruction performance. These methods can be roughly divided into two categories according to whether their output representations are explicit or implicit. As shown in Fig. 1, explicit representation methods including *mesh* and *voxel* based ones denote the exact location of a surface, which learn to warp templates [3, 4, 19, 26, 29, 68] or predict voxel grids [10, 30, 59]. Explicit representations are friendly to downstream applications, but they are usually limited by resolution and

topology. On the other hand, implicit representations such as *Occupancy* and *Signed Distance Functions (SDFs)* represent the surface as an isocontour of a scalar function, which receives increasing attention due to their capacity to represent surfaces with more complicated topology and at arbitrary resolution [12, 14, 22, 35, 44, 48]. However, most implicit representation methods usually require specific pre-processing to close non-watertight meshes and remove inner structures. To free from the above pre-processing requirements for implicit representation, Chibane *et al*. [15] introduced *Neural Unsigned Distance Fields (NDF)*, which employs the *Unsigned Distance Functions (UDFs)* for neural implicit functions (NIFs) and models continuous surfaces by predicting positive scalar between query locations and the target surface. Despite certain advantages, UDFs require a more complicated surface extraction process than other implicit representation methods (*e.g*., SDFs). Such a process using Ball-Pivoting Algorithm [5] or gradient-based Marching Cube [28, 82] relies on model differentiation during inference (*i.e*., differentiation-dependent). Moreover, UDFs leave gradient ambiguities at "ridge" points, where the gradients[1] used for surface extraction cannot accurately point at target points as illustrated by Fig. 2a.

In this work, we propose a novel 3D representation method, *Neural Vector Fields (NVF)*, which leverages the explicit learning process of direct manipulation on meshes and the implicit representation of UDFs to enjoy the advantages of both approaches. That is, NVF can directly manipulate meshes as those explicit representation methods as Fig. 1g, while representing the shapes with arbitrary resolution and topology as those implicit representation methods. Specifically, NVF models the 3D shapes as vector fields and computes the displacement between a point $\mathbf{q} \in \mathbb{R}^3$ and its nearest-neighbor point on the surface $\hat{\mathbf{q}} \in \mathbb{R}^3$ by using a learned function $f(\mathbf{q}) = \mathbf{\Delta q} = \hat{\mathbf{q}} - \mathbf{q}$: $\mathbb{R}^3 \Rightarrow \mathbb{R}^3$. Therefore, NVF could serve both as an implicit function and an explicit deformation function, since the displacement output of the function could be directly used to deform source meshes (*i.e*., Fig. 1g). In general, it encodes both distance and direction fields within vector fields, which can be straightforwardly obtained from the vector fields.

Different from existing UDF-based methods, our NVF representation avoids the comprehensive inference process by skipping the gradient calculation during the surface extraction procedure[1], and mitigates ambiguities by directly learning displacements as illustrated by Fig. 2b. Such one-pass forward-propagation nature frees NVF from differentiation dependency, significantly reduces the inference time and memory cost, and allows our model to learn a shape codebook consisting of un-differentiable discrete shape codes in the embedded feature space. The learned shape codebook further provides cross-object priors to consequently improve the model generalization on cross-category reconstruction, and accelerates the training procedure as a regularization term during training.

[1]Learning-based methods calculate the gradients of distance fields via model differentiation. The opposite direction of gradients should point to the nearest-neighbor point on the target surface.



(a) NDF [15]. (b) NVF.

Figure 2. Gradient ambiguities. (a) NDF [15] cannot guarantee to pull points onto surfaces (*i.e*., ambiguity of gradient), while (b) our NVF address this issue by direct displacement learning.

We conduct extensive experiments on two surface reconstruction benchmark datasets: a synthetic dataset ShapeNet [8] and a real scanned dataset MGN [6]. Besides category-specific reconstruction [15, 75] as demonstrated in most reconstruction methods, we also evaluate our framework by category-agnostic reconstruction, category-unseen reconstruction, and cross-domain reconstruction tasks to exploit the model generalization. Our experimental results indicate that our NVF can significantly reduce the inference time compared with other UDF-based methods as we avoid the comprehensive surface extraction step and circumvent the requirement of gradient calculation at query locations. Also, using the shape codebook, we observe a significant performance improvement and a better model generalization across categories.

Our contributions are summarized as follows.

- We propose a 3D representation *NVF* for better 3D field representation, which bridges the explicit learning and implicit representations, and benefits from both of their advantages. Our method can obtain the displacement of a query location in a differentiation-free way, and thus it significantly reduces the inference complexity and provides more flexibility in designing network structures which may include non-differentiable components.

- Thanks to our differentiation-free design, we further propose a learned shape codebook in the feature space, which uses VQ strategy to provide cross-object priors. In this way, each query location is encoded as a composition of discrete codes in feature space and further used to learn the NVF.

- We conduct the extensive experiments to evaluate the effectiveness of our proposed method. It consistently shows promising performance on two benchmarks across different evaluation scenarios: water-tight vs

non-water-tight shapes, category-specific vs category-agnostic reconstruction, category-unseen reconstruction, and cross-domain reconstruction.

2. Related Work

2.1. Explicit 3D Representations

Early shape representation methods are built upon voxels [17, 43, 58]. As the 3D spaces are discretized into girds, so they can be processed by adapting learning-based image processing techniques [34, 37, 53, 69, 70]. However, voxels are usually with the memory footprint scales cubically with the resolution. So recent voxel-based methods introduce adaptive discretization [10, 30, 54, 59] and voxel block hashing [57] to alleviate higher computational requirements for higher resolution.

Differently, point cloud based methods produce a compact and sparser encoding of surfaces, with lower computational cost and higher accessibility. Several learning-based point cloud processing methods [9, 49, 60, 67, 81] are proposed in recent years and achieve tremendous success on 3D shape analysis [16, 32, 33, 41, 66, 78, 83] and synthesis [1, 26, 74, 77]. However, point clouds usually lack rich geometry information (*e.g.*, surfaces and topology), which results in limited applications in downstream tasks.

To complete such geometry information, polygonal meshes [36, 40, 50, 65, 73] define 3D shapes by graphs consisting of vertices and edges. However, they are usually suffered from fixed topology as most of them deform a template (*e.g.*, ellipsoid [65], sphere [36, 46, 73]) into target shapes by moving vertices. Recent works addressed these issues by deforming from charts [26], voxels [25], and pruning needless faces [46].

We observe that these methods encourage the networks to simulate vector fields within 3D space. Inspired by them, we learn the model using explicit learning with the vector fields instead of mesh-to-mesh distances to learn vector fields. In this way, we can break the boundaries of fixed topology and resolution of templates.

2.2. Implicit 3D Representations

To overcome the limited resolution and fixed mesh topology barriers from most explicit representation, implicit representation methods represent continuous surfaces through implicit functions. Such representations are usually implemented by multi-layer perceptions (MLP), which generate binary occupancy [14, 18, 24, 44, 55, 56] and SDFs [12, 35, 47] given locations as queries. On the other hand, those methods require a heavy pre-processing stage to close the shapes artificially to obtain watertight meshes due to the characteristics of occupancy and SDFs. For one thing, It is also non-trivial to define the inside and outside of open surfaces. For another, the pre-processing results in substantive geometry information loss and lack of generalization on non-watertight meshes.

To model general non-watertight shapes, Chibane *et al*. [15] introduced UDFs to learn the unsigned distance. Recently, other methods improve the performance and generalization of UDFs [62, 64, 75, 80, 82]. For example, GIFS [75] adopts an intersection classification branch and CAP-UDF [82] directly optimizes models on raw point clouds. Although such UDF-based methods can intuitively improve the generalization of implicit representation, they require non-trivial surface extraction processing such as Ball-Pivoting algorithm [15] and gradient-based Marching Cubes [28, 82].

Different than aforementioned UDF-based methods, our NVF does not rely on differentiation to obtain gradients as direction fields and can simply obtain distance and direction fields by one-time forward propagation. Such effective differentiation-free strategy can significantly reduce the inference burden.

2.3. Vector Quantization

The vector Quantization (VQ) strategy converts the extracted features into quantized and compact latent representations. Oord *et al*. first introduced VQ-VAE [61] to combining VQ strategy with Variational AutoEncoder for images and speech generation. Then, Yu *et al*. introduce VQ-GAN [76] to cooperate codebooks with adversarial learning for HQ image synthesis. Overall, several methods employ this strategy for image generation [21, 27, 39, 51], speech synthesis [20, 72], video [23, 31, 52, 63], shape generation [13, 45, 79] and compression [2, 11]. In this work, we also employ VQ and adopt a multi-head codebook in feature space to denoise, accelerate training and improve generalization on shape reconstruction.

3. Methodology

Given a 3D query location $\mathbf{q}$ from a query set $Q \in \mathbb{R}^{M\times 3}$ and a sparse point cloud $P \in \mathbb{R}^{N\times 3}$, our NVF framework predicts the displacement vector $\Delta\mathbf{q} \in \mathbb{R}^3$ that moves $\mathbf{q}$ to the underlying surface of P. This is achieved through three main modules, *i.e.*, Feature Extraction, Multi-head Codebook, and Field Prediction, as illustrated in Fig. 3. Once the predicted vector field is obtained, we adopt Marching Cubes to generate the surface mesh. The details of each module are elaborated as follows.

3.1. Neural Vector Fields

As mentioned above, our NVF is a neural network function, which models shapes $\mathcal{S}$ by predicting the field of displacements $\Delta\mathbf{q}$ for a given query point $\mathbf{q} \in Q$ to its nearest point $\hat{\mathbf{q}} \in \mathcal{S}$ on the surface. It is formulated by

$$NVF(\mathbf{q}) = \Delta\mathbf{q} = \min_{\hat{q}\in\mathcal{S}} \hat{\mathbf{q}} - \mathbf{q}. \quad (1)$$

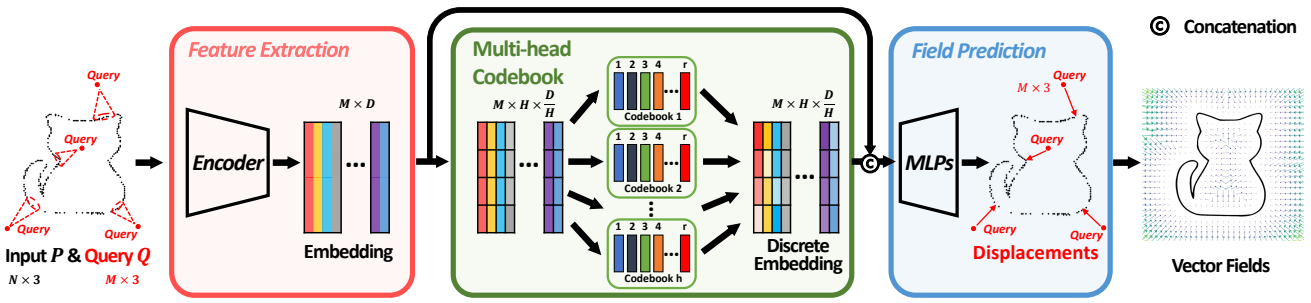


Figure 3. Overview of our NVF framework. It encodes the input point cloud (black dots), samples features for each query point (red dots) and discretizes query embeddings via Vector Quantization to predict the displacements of query location to the target surface (red arrow).

Please note that NVFs are directional distance fields, which encode both distance fields $d = ||\hat{\mathbf{q}} - \mathbf{q}||_2$ and direction fields $\mathbf{g} = (\hat{\mathbf{q}} - \mathbf{q})/||\hat{\mathbf{q}} - \mathbf{q}||_2$ in a straightforward way. Therefore, unlike distance fields, our NVF does not require the differentiation of distance fields for direction information. This property is referred to as "differentiation-free" in this paper, which significantly reduces the inference time. On the other hand, our NVF enjoys the merits of UDFs and can represent much wider classes of surfaces and manifolds even not necessarily closed.

3.2. Feature Extraction

The input point cloud P is first sent to a feature encoder to extract 3D shape features for each point, which leads to the output feature maps $F \in \mathbb{R}^{N\times C}$. Multiple point cloud networks [9, 49, 81] could be used as the feature encoder, and an ablation study about the backbones is given in our experiment. For the query point $\mathbf{q}$, we find its K nearest points $\mathbf{p}_i (i = 1, \cdots, k)$ in $P \in \mathbb{R}^{N\times 3}$ based on a search using Euclidean distance, and form the feature embedding z_q of $\mathbf{q}$ by concatenating the signatures z_{qi} of its nearest points $\mathbf{p}_i$. Each signature z_{qi} is a non-linear mapping of the position features (*i.e.*, the positions of the query point $\mathbf{q}$, the nearest point $\mathbf{p}_i$, and the relative position $\mathbf{q} - \mathbf{p_i}$) and the shape features (i.e., $\mathbf{p_i}$'s corresponding feature $f_i \in F$), which is realized by MLP. Mathematically, the query embedding $z_q \in \mathbb{R}^D$ is obtained by

$$\begin{aligned} z_q &= z_{q1} \oplus z_{q2} \oplus ...z_{qi}... \oplus z_{qk}, \\ z_{qi} &= MLP(\mathbf{q}, \mathbf{p_i}, \mathbf{p_i} - \mathbf{q}, f_i), \quad i = 1, 2, ..., k. \end{aligned} \tag{2}$$

The symbol $\oplus$ indicates the concatenation operation. As can be seen, the query embedding z_q carries the information of both the query point $\mathbf{q}$ and its K nearest points on P.

3.3. Multi-head Codebook

After obtaining the embedding z_q whose dimension is D, we learn a multi-head codebook $\mathcal{C} \in \mathbb{R}^{H\times R\times \frac{D}{H}}$ to produce cross-object priors. Our codebook contains H sub-codebooks (i.e., heads). Each sub-codebook $\mathcal{C}^h$ $(h = 1, \cdots, H)$ contains R discrete codes c_r^h $(r = 1, \cdots, R)$ and each code has a dimension of $\frac{D}{H}$. The codebook is randomly initialized.

Accordingly, we split the continuous embedding z_q into H segments along the channel dimension. For each embedding segment z_q^h, we search its closest code from the corresponding sub-codebook $\mathcal{C}^h$ according to their Euclidean distance. In this way, we could discretize the continuous embedding z_q to $\hat{z}_q$ that is composed of discrete segment codes. This process can be formulated as follows,

$$\begin{aligned} z_q &= z_q^1 \oplus z_q^2 \oplus ... \oplus z_q^h ... \oplus z_q^H, \\ \hat{z}_q &= c_{r_1^*}^1 \oplus c_{r_2^*}^2 \oplus ... \oplus c_{r_h^*}^h ... \oplus c_{r_H^*}^H, \\ r_h^* &= \underset{r\in\{1,...,R\}}{\arg\min} ||z_q^h - c_r^h||_2. \end{aligned} \tag{3}$$

Compared with using only one codebook $\mathcal{C} \in \mathbb{R}^{(H*R)\times D}$, the multi-head codebook enables R^H permutations, which extends the feature space from $\mathbb{R}^{H*R}$ to $\mathbb{R}^{R^H}$, and enhances the codebook representation capacity. Note that the nearest search stops the gradient here.

The multi-head codebook can be jointly learned during training by minimizing the distance between the continuous embedding z and its discretization $\hat{z}_q$ as follows,

$$\mathcal{L}_{code} = \sum_{h\in\{1,...,H\}} ||sg(c_{r_h^*}^h) - z_q^h||_2^2 + \beta||sg(z_q^h) - c_{r_h^*}^h||_2^2, \tag{4}$$

where sg stands for *stop gradient* operation and β is the weight to the commitment loss in the second term. Alternatively, we can update the discretized embedding $\hat{z}_q$ and therefore its corresponding codes $(\hat{z}_q^h \equiv c_{r_h^*}^h)$ in the codebook via *Exponential Moving Average* [61],

$$c_{r_h^*}^h := \gamma c_{r_h^*}^h + (1 - \gamma) z_q^h, \tag{5}$$

with γ is a value between 0 and 1.

3.4. Field Prediction

Last, we take the continuous embedding z_q and its discretization $\hat{z}_q$ to predict a vector as the displacement for

the query point $\mathbf{q}$ to move to the ground truth surface $\mathcal{S}$. By combining the embeddings z_q and $\hat{z}_q$, the final vector field is predicted using MLPs, i.e., $NVF(\mathbf{q}) \approx \Delta\mathbf{q} = MLP(z_q, \hat{z}_q)$.

3.5. Optimization

We optimize NVF by minimizing the difference between the predicted displacement $\Delta\mathbf{q}$ and the ground-truth displacement $\hat{\mathbf{q}} - \mathbf{q}$, as well as the error of discretizing the continuous embedding z_q into $\hat{z}_q$. We use stop-gradient operation (*i.e.*, *sg*) to cut the gradient after vector quantization. In sum, our overall objective function is,

$$\mathcal{L} = \sum_{\mathbf{q}\in Q} ||\hat{\mathbf{q}} - \mathbf{q} - \Delta\mathbf{q}||_1 + \lambda \sum ||z_q - sg(\hat{z}_q)||_2^2. \quad (6)$$

The hyperparameter λ balances the two loss terms. The commitment loss term $||sg(z) - z^q||_2^2$ used in Eq. 4 is replaced by the *Exponential Moving Average* algorithm to update corresponding codes in the multi-head codebook.

3.6. Surface Extraction

Both signed and unsigned distance functions define the surfaces as a 0-level set. SDFs extract surfaces through Marching Cubes [42] faithfully due to the intersection being easily captured via sign flipping. Recently, MeshUDF [28] and CAP-UDF [82] propose to detect the opposite gradient directions to replace the sign in SDFs and successfully apply the Marching Cubes [42] on UDFs. Note that these two methods learn distance fields and they need to differentiate the distance field to obtain the gradient direction. In contrast, our NVF can similarly extract surfaces using Marching Cubes while avoiding the differentiation of the distance field. This is because our NVF directly encodes both distance and direction fields. The surface extraction algorithm divides the space into grids and decides whether the adjacent corners locate on the same side or two sides. NVF predicts the vectors $\Delta\mathbf{q}$ of all lattices and normalizes them into normal vectors $\mathbf{g_i} = \Delta\mathbf{q_i}/||\Delta\mathbf{q_i}||_2$. Given a lattice gradient $\mathbf{g_i}$ and its adjacent lattices' gradients $\mathbf{g_j}$, the algorithm checks if their gradients have opposite orientations, and assigns the pseudo sign s_i to the lattice. For more details, please refer to MeshUDF [28].

4. Experiments

4.1. Experimental Protocol

Tasks. We evaluate the effectiveness of our framework on four tasks: 1) category-specific, 2) category-agnostic, 3) category-unseen and 4) cross-domain reconstruction. We first demonstrate the ability of our NVF to reconstruct non-watertight meshes by category-specific reconstruction in Sec. 4.2. Next, to evaluate the generalization ability, we compare our NVF with existing methods on category-agnostic and category-unseen reconstruction in Sec. 4.3. We also test cross-domain reconstruction by reconstructing real scanned data without training or fine-tuning in Sec. 4.4.

Methods	CD↓	EMD↓	$F1_{1\times10^{-5}}$	$F1_{2\times10^{-5}}$
Input	0.363	0.707	23.735	41.588
NDF [15]	0.197	1.248	64.116	84.902
GIFS [75]	0.146	0.970	54.867	79.722
Ours	**0.114**	**0.945**	**64.261**	**85.290**

Table 1. Quantitative evaluation on ShapeNet Cars. We train and evaluate our method on the raw data of the ShapeNet "Car" category. Our method achieves better performance than the state-of-the-art UDF-based methods.

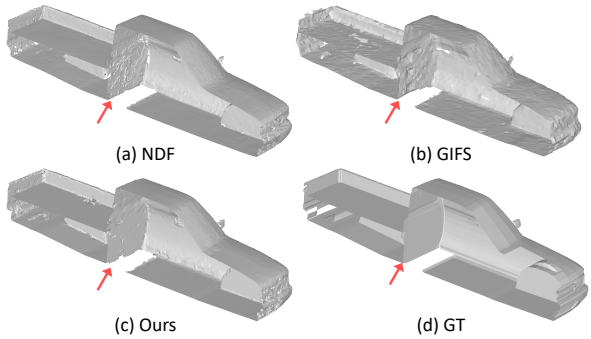


Figure 4. Qualitative visualization of Category-specific reconstruction on ShapeNet Cars. We cut parts of the shapes to visualize inner structures better.

Implementations. We employ PointTransformer [81] as our feature encoder and set $k = 16$ for the nearest points. The multi-head codebook consists of 4 sub-codebooks, each containing 128 codes and 64 dimension channels. We set the grid resolution for surface extraction as 256. To train our network, the learning rate is set as $l_r = 0.001$ and decays with the factor 0.3 for the subsequent 30, 70, and 120 steps. The loss weight λ is set to be 0.001.

Datasets. Most experiments are conducted on a large-scale synthetic dataset ShapeNet [8] and a real scanned dataset MGN [6]. We conduct our category-specific, category-agnostic, and category-unseen reconstruction experiments on ShapeNet [8] while conducting cross-domain reconstruction experiments on MGN [6] to demonstrate our model generalization in the wild. All meshes are normalized to a unit cube with a range $[-0.5, 0.5]$.

ShapeNet is a synthetic dataset with 55 object categories. We choose *cars* for category-specific reconstruction following UDF literature, and *cars, chairs, planes, and tables* for category-agnostic reconstruction. For category-unseen reconstruction, we take *speakers, bench, lamps, and watercraft* as the testing set. The above 4 categories for category-agnostic reconstruction are referred to as *base classes* and the ones for category-unseen reconstruction are referred to

Methods	Base				Novel			
	CD↓	EMD↓	$F1_{2.5\times10^{-5}}$ ↑	$F1_{1\times10^{-4}}$ ↑	CD↓	EMD↓	$F1_{1\times10^{-5}}$ ↑	$F1_{2\times10^{-5}}$ ↑
Input	0.840	1.045	14.148	25.111	0.800	1.024	17.576	29.815
OccNet [44]	2.766	1.694	30.877	46.644	44.762	4.013	15.943	24.433
IF-Net [14]	0.190	1.120	65.975	85.421	0.596	1.608	61.670	81.106
NDF [15]	0.169	1.538	66.802	84.809	0.169	1.741	65.622	84.069
GIFS [75]	0.179	1.280	56.188	78.458	0.194	1.534	56.644	78.016
Ours	**0.091**	**1.079**	**78.503**	**91.408**	**0.144**	**1.145**	**80.883**	**91.836**

Table 2. Quantitative results of category-agnostic and category-unseen reconstructions on watertight shapes of ShapeNet. We train all models on the base classes, and evaluate them on the base and the novel classes, respectively.

Methods	Base				Novel			
	CD↓	EMD↓	$F1_{2.5\times10^{-5}}$ ↑	$F1_{1\times10^{-4}}$ ↑	CD↓	EMD↓	$F1_{1\times10^{-5}}$ ↑	$F1_{2\times10^{-5}}$ ↑
Input	0.317	0.867	32.875	51.105	0.289	0.843	39.902	58.092
NDF [15]	0.099	1.372	72.425	88.754	0.093	1.532	76.162	89.977
GIFS [75]	0.118	1.260	64.915	85.115	0.296	1.499	69.252	86.518
Ours	**0.085**	**1.197**	**75.372**	**90.266**	**0.078**	**1.340**	**79.723**	**91.576**

Table 3. Quantitative results of category-agnostic and category-unseen reconstructions on non-watertight shapes of ShapeNet. We train all models on the base classes and evaluate them on the base and the novel classes, respectively.

Methods	CD↓	EMD↓	$F1_{1\times10^{-5}}$	$F1_{2\times10^{-5}}$
Input	0.124	0.157	52.189	72.969
NDF [15]	0.025	0.216	96.338	98.687
GIFS [75]	0.039	0.192	93.330	97.295
Ours	**0.014**	**0.184**	**98.499**	**99.498**

Table 4. Quantitative results of cross-domain reconstruction on MGN [6]. We train our models based on ShapeNet with the base classes and evaluate them on the raw data from MGN.

as *novel classes* in Tab. 2 and Tab. 3.

MGN is a real scanned dataset containing 5 garment categories, (*i.e.*, long coat, pants, shirt, short pants, and T-shirt), which are all open surfaces. We evaluate our model trained by ShapeNet on MGN to demonstrate the reconstruction ability in the wild. The results are in Sec. 4.4.

Metrics. We use Chamfer Distance (CD), Earth Mover Distance (EMD) and F-score as our evaluation metrics. Specifically, we sample 100k points from the reconstructed surfaces for the Chamfer Distance (CD $\times 10^{-4}$), 2048 points for the Earth Mover Distance (EMD $\times 10^{-2}$), and 100k points for F-score ($\times 10^{-2}$) with thresholds 1×10^{-5} and 2×10^{-5}.

4.2. Category-specific Cases

We first report the results of *category-specific reconstruction* on ShapeNet cars to demonstrate the representation ability to one category of our NVF. We perform the same training/testing split and sample 10k points as the input as in NDF [15] and GIFS [75]. The quantitative comparison in Tab. 1 shows that our method achieves better results with previous state-of-the-art methods including NDF and GIFS. Specifically, on CD and EMD, our NVF outperforms the second-best GIFS; and on F-scores, ours outperforms the second-best NDF. Moreover, we provide the qualitative comparison in Fig. 4. The visualization indicates that although NDF, GIFS, and our NVF are able to recover inner structures, our NVF could yield smoother surfaces and fewer holes in the finer parts (*e.g.*, the back of driving seats).

4.3. Category-agnostic and Category-unseen Cases

Base vs Novel categories on watertight shapes. We conduct category-agnostic and category-unseen reconstruction experiments to explore the generalization ability of our NVF. For a fair comparison with the previous methods (*e.g.*, OccNet [44] and IF-Net [14]) which cannot handle non-watertight shapes as the input, we report the quantitative experimental results on watertight shapes pre-processed by DISN [71] in Tab. 2 and give qualitative visualization in Fig. 5. We use the same training/testing split as in IF-Net [14]. The input of all methods is 3k points in this experiment.

Table 2 indicates that the surface reconstruction results produced by UDF-based methods are much better than occupancy representation, especially for novel classes. Among them, our NVF outperforms others by a large margin, no matter in base classes or novel classes. The qualitative comparison in Fig. 5 shows that OccNet and IF-Net fail to construct a lot of details. NDF and GIFS are better than them, but they still fail to capture some details, *e.g.*, undercarriages of planes, or little bumps from watercraft. In contrast, our method can reconstruct flat and faithful surfaces, and thereby achieves a better visual effect.

Base vs Novel categories on non-watertight shapes. We also test on non-watertight shape reconstruction with the same setting above and the results are in Tab. 3 and Fig. 6.

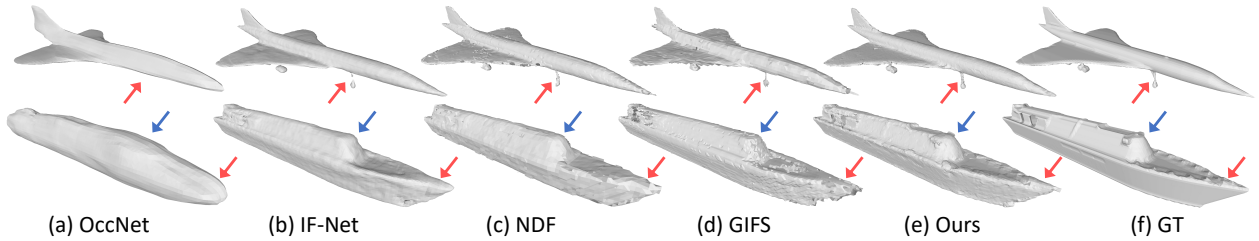


Figure 5. Visualization of category-agnostic and category-unseen reconstructions on watertight shapes from the ShapeNet dataset. The 1^{st} row, planes, is from the base classes. The 2^{rd} row, watercraft, is from the novel classes.

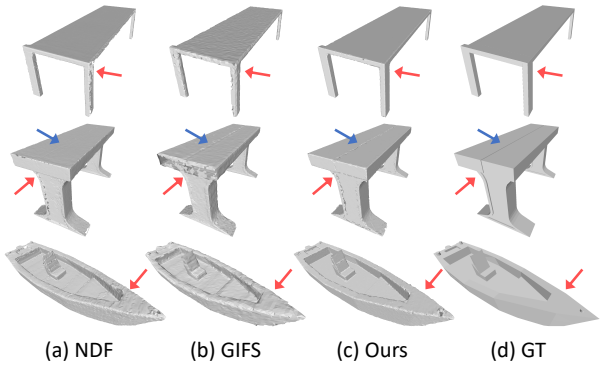


Figure 6. Visualization of category-agnostic and category-unseen reconstructions on non-watertight shapes from the ShapeNet dataset. The 1^{st} row, tables, is from the base classes. The 2^{nd} and 3^{rd} rows, benches and watercraft, are from the novel classes.

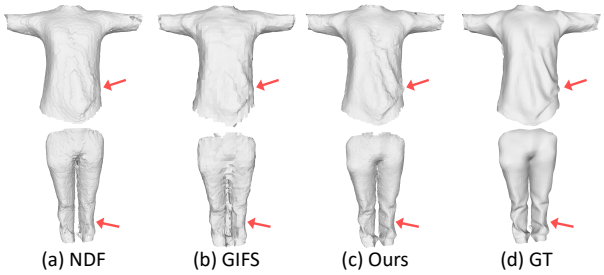


Figure 7. Visualization of cross-domain reconstruction on the MGN dataset. All models are trained based on ShapeNet with the base classes and evaluated directly on MGN.

For the training/testing split, we randomly sample 3000 shapes from each bases class as the training set and 200 shapes from each base and novel class as the validation and testing set. The input is 10k points sampled from the raw data of these shapes.

The quantitative comparison in Tab. 3 shows that our NVF outperforms the NDF and GIFS on either the category-agnostic or the category-unseen reconstruction. For example, we decrease $\sim 10\%$ CD on base classes and $\sim 15\%$ CD on novel classes from NDF. The qualitative visualization in Fig. 6 further supports the results, in which NDF and GIFS miss some important details (*e.g.*, seams on benches)

	K	Codebook	CD↓	EMD↓	$F1_{1\times10^{-5}}$	$F1_{2\times10^{-5}}$
Base	k=8	✗	0.089	1.195	74.139	89.392
	k=8	✓	0.087	**1.172**	75.104	90.057
	k=16	✗	0.121	1.212	73.642	88.962
	k=16	✓	**0.085**	1.197	**75.372**	**90.266**
Novel	k=8	✗	0.080	1.334	78.701	90.972
	k=8	✓	0.083	1.354	79.522	91.434
	k=16	✗	0.081	**1.329**	78.800	91.084
	k=16	✓	**0.078**	1.340	**79.723**	**91.576**

Table 5. Effect of feature number K and multi-head codebook. The multi-head codebook improves the performance and achieves the best for $K = 16$.

and leave a number of holes (*e.g.*, legs of tables, sides of benches), and their surfaces are not as smooth as ours (*e.g.*, the board of watercraft).

4.4. Cross-domain Cases

Rather than only concentrating on intra-domain performance as most previous reconstruction methods, we also explore the cross-domain ability of our method, in which we design a cross-domain reconstruction experiment and demonstrate the quantitative results and qualitative visualization respectively in Tab. 4 and Fig. 7. Specifically, we directly deploy our category-agnostic models trained based on ShapeNet base classes to MGN [6] to evaluate its performance. We randomly sample 20 shapes from the 5 categories in MGN and 3,000 points from the raw data as the input.

Tab. 4 indicates that our NVF outperforms other methods by a large margin under all metrics. For example, our method reduces almost 40% and 60% CD respectively from NDF and GIFS. The qualitative visualization in Fig. 7 also supports this observation, where only our NVF can reconstruct the crinkles of the shirts and pants faithfully.

5. Ablation Study

We conduct comprehensive ablation studies on non-watertight category-agnostic and category-unseen reconstruction (same settings as Sec. 4.3.) to evaluate the effectiveness of the proposed modules including multi-head codebook, and the influence of hyper-parameters like the

	Backbone	CD↓	EMD↓	$F1_{1\times10^{-5}}$	$F1_{2\times10^{-5}}$
Base	NDF [15]	0.099	1.372	72.425	88.754
	3D Conv	0.092	**1.189**	72.415	89.365
	Pointnet++	0.088	1.198	74.156	89.418
	PointTransformer	**0.085**	1.197	**75.372**	**90.266**
Novel	NDF [15]	0.093	1.532	76.162	89.977
	3D Conv	0.090	1.342	76.155	90.462
	Pointnet++	0.080	**1.339**	78.444	90.850
	PointTransformer	**0.078**	1.340	**79.723**	**91.576**

Table 6. Comparison of different backbones on non-watertight ShapeNet. The point cloud based backbones outperform the 3D convolution backbone. PointTransformer performs the best.

Methods	Backbone	Codebook	Runtime	Memory
NDF [15]	3D Conv	✗	0.75s	13.44G
Ours	3D Conv	✗	0.27s	9.21G
	3D Conv	✓	0.34s	9.21G
	PointTransformer	✗	0.29s	9.28G
	PointTransformer	✓	0.35s	9.28G

Table 7. Inference analysis. The runtime and memory are time cost and peak memory during the inference of 200k queries. NVF is more efficient on inference runtime and memory.

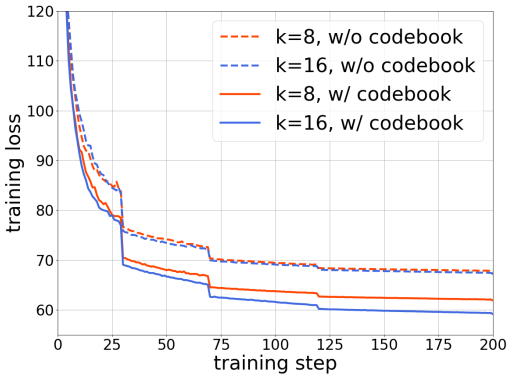


Figure 8. Training curves of models w/ and w/o codebook. The models w/ codebooks converge faster than those w/o codebooks.

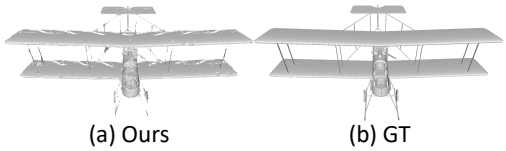


Figure 9. Visualization of one failure case of our model when reconstructing a very thin and complex 3D structure.

feature number K (i.e., the number of the nearest points in Sec. 3.2) on the results. Complexity analysis is also provided to demonstrate the efficiency of our model.

Feature number. The number of features for query points (determined by the number of nearest points on the point cloud P) is an important factor of the model. We report the performance of our network using $k = 8, 16$ in Tab. 5, and observe that a larger feature number k yields better results but at the cost of memory.

Codebook. We also demonstrate the effectiveness of the introduced codebook mechanism in Tab. 5. The results indicate that using codebook can generally improve the overall performance regarding all metrics.

Training Effectiveness. We would like to highlight that the introduced multi-head codebook could also work as regularization to reduce the training time. As shown in Fig. 8, the loss curves of the models with (w/) codebooks (*i.e.*, solid lines) generally converge faster than their corresponding models without (w/o) codebooks (*i.e.*, dashed lines).

3D Feature Extraction. We also report our methods with different 3D feature extraction backbones (*i.e.*, 3D convolution [14, 15, 75] , Pointnet++ [49], and PointTransformer [81]) in Tab. 6. The results show that our NVF can already achieve comparable (if not better) results with NDF using the same extraction backbone (*e.g.*, 3D convolution), while ours is more efficient on runtime and memory cost. Using PointTransformer can consistently improve the overall performance.

Complexity. We report the inference time and peak GPU memory on the machine with one RTX 3090Ti in Tab. 7 for obtaining the distance and direction of 200k query points. The results show that our method only requires 50% inference time and 70% memory compared with NDF when using the same feature extraction backbone. This is due to the cost from the differentiation of distance field in NDF, while our method avoids this issue and introducing multi-head codebook only brings negligible overhead of memory and time as shown in the Tab. 7.

Limitations. Our method can handle watertight or non-watertight shapes with arbitrary resolution and topology. However, the reconstruction for very thin or complex structures is still far from perfect as visualized in Fig. 9.

6. Conclusion

In this work, we introduce a novel surface reconstruction representation *NVF*, which leverages the advantages of both the implicit representations and the explicit learning process. It allows the reconstruction of high-quality general object shapes including watertight, non-watertight, and multi-layer shapes. Thanks to our differentiation-free design, we also introduce vector quantization and build a multi-head codebook to improve the generalization of cross-category reconstruction. Experiments demonstrate that our method not only achieves state-of-the-art reconstruction performance for varied topology, but also improves the training and inference efficiency.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Representation learning and adversarial

generation of 3d point clouds. *ArXiv*, abs/1707.02392, 2017. 3

[2] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc V Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[3] Abhishek Badki, Orazio Gallo, Jan Kautz, and Pradeep Sen. Meshlet priors for 3d mesh reconstruction. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. 1

[4] Jan Bednarik, Shaifali Parashar, Erhan Gundogdu, Mathieu Salzmann, and Pascal Fua. Shape reconstruction by learning differentiable surface representations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. 1

[5] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999. 1, 2

[6] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct 2019. 2, 5, 6, 7

[7] Edwin Catmull and James Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-aided design*, 10(6):350–355, 1978. 1

[8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 5

[9] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 3, 4

[10] Jiawen Chen, Dennis Bautembach, and Shahram Izadi. Scalable real-time volumetric surface reconstruction. *ACM Trans. Graph.*, 32(4), jul 2013. 1, 3

[11] Ting Chen and Yizhou Sun. Differentiable product quantization for end-to-end embedding compression. In *ICML*, 2020. 3

[12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 2, 3

[13] An-Chieh Cheng, Xueting Li, Sifei Liu, Min Sun, and Ming-Hsuan Yang. Autoregressive 3d shape generation via canonical mapping. In *ECCV*, 2022. 3

[14] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. 2, 3, 6, 8

[15] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2020. 2, 3, 5, 6, 8

[16] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3070–3079, 2019. 3

[17] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 3

[18] Boyang Deng, J. P. Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. *Lecture Notes in Computer Science*, page 612–628, 2020. 3

[19] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. In *Advances in Neural Information Processing Systems*, pages 7433–7443, 2019. 1

[20] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020. 3

[21] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In *NeurIPS*, 2021. 3

[22] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J. Mitra, and Michael Wimmer. Points2Surf: Learning implicit surfaces from point clouds. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 108–124, Cham, 2020. Springer International Publishing. 2

[23] Songwei Ge, Thomas Hayes, Harry Yang, Xiaoyue Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *ECCV*, 2022. 3

[24] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *20202 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[25] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9785–9795, 2019. 3

[26] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 1, 3

[27] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10686–10696, 2022. 3

[28] Benoit Guillard, Federico Stella, and Pascal Fua. Meshudf: Fast and differentiable meshing of unsigned distance field networks. *arXiv preprint arXiv:2111.14549*, 2021. 2, 3, 5

[29] Kunal Gupta and Manmohan Chandraker. Neural mesh flow: 3d manifold mesh generationvia diffeomorphic flows. In *Advances in Neural Information Processing Systems*, volume abs/2007.10973, 2020. 1

[30] Christian Hane, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. *2017 International Conference on 3D Vision (3DV)*, Oct 2017. 1, 3

[31] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers, 2022. 3

[32] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. 3

[33] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 984–993, 2018. 3

[34] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2307–2315, 2017. 3

[35] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias NieBner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. 2, 3

[36] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. *Lecture Notes in Computer Science*, page 386–402, 2018. 3

[37] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NeurIPS*. 2017. 3

[38] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 1

[39] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2022. 3

[40] Chen-Hsuan Lin, Oliver Wang, Bryan C. Russell, Eli Shechtman, Vladimir G. Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 3

[41] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 3

[42] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, aug 1987. 1, 5

[43] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Ieee/rsj International Conference on Intelligent Robots and Systems*, pages 922–928, 2015. 3

[44] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 6

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

[46] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. 3

[47] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 3

[48] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[49] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3, 4, 8

[50] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3d faces using convolutional mesh autoencoders. *Lecture Notes in Computer Science*, page 725–741, 2018. 3

[51] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. *Generating Diverse High-Fidelity Images with VQ-VAE-2*. Curran Associates Inc., Red Hook, NY, USA, 2019. 3

[52] Jingjing Ren, Qingqing Zheng, Yuan yu Zhao, Xuemiao Xu, and Chen Li. Dlformer: Discrete latent transformer for video inpainting. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3501–3510, 2022. 3

[53] Danilo Jimenez Rezende, S. M. Ali Eslami, Shakir Mohamed, Peter W. Battaglia, Max Jaderberg, and Nicolas Manfred Otto Heess. Unsupervised learning of 3d structure from images. In *NIPS*, 2016. 3

[54] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 3

[55] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Hao Li, and Angjoo Kanazawa. Pifu: Pixel-aligned implicit function for high-resolution clothed human digiti-

zation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. 3

[56] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 3

[57] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Volumetric 3d mapping in real-time on a cpu. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2021–2028, 2014. 3

[58] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[59] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 3

[60] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas Guibas. Kpconv: Flexible and deformable convolution for point clouds. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. 3

[61] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NIPS*, 2017. 3, 4

[62] Rahul Venkatesh, Tejan Karmali, Sarthak Sharma, Aurobrata Ghosh, R. Venkatesh Babu, Laszlo A. Jeni, and Maneesh Singh. Deep implicit surface point prediction networks. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021. 3

[63] Jacob Walker, Ali Razavi, and Aäron van den Oord. Predicting video with vqvae. *ArXiv*, abs/2103.01950, 2021. 3

[64] Bing Wang, Zhengdi Yu, Bo Yang, Jie Qin, Toby Breckon, Ling Shao, Niki Trigoni, and Andrew Markham. Rangeudf: Semantic surface reconstruction from 3d point clouds, 2022. 3

[65] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 3

[66] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. 3

[67] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, Nov 2019. 3

[68] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 1

[69] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, William T Freeman, and Joshua B Tenenbaum. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *Advances In Neural Information Processing Systems*, 2017. 3

[70] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Joshua B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, 2016. 3

[71] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 492–502. Curran Associates, Inc., 2019. 6

[72] Karren Yang, Dejan Markovic, Steven Krenn, Vasu Agrawal, and Alexander Richard. Audio-visual speech codecs: Rethinking audio-visual speech enhancement by re-synthesis. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2022. 3

[73] Xianghui Yang, Guosheng Lin, and Luping Zhou. Zeromesh: Zero-shot single-view 3d mesh reconstruction, 2022. 3

[74] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Interpretable unsupervised learning on 3d point clouds. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, abs/1712.07262, 2017. 3

[75] Jianglong Ye, Yuntao Chen, Naiyan Wang, and Xiaolong Wang. Gifs: Neural implicit function for general shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3, 5, 6, 8

[76] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *ICLR*, abs/2110.04627, 2022. 3

[77] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. *2018 International Conference on 3D Vision (3DV)*, Sep 2018. 3

[78] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alex Smola. Deep sets. *Advances in neural information processing systems*, abs/1703.06114, 2017. 3

[79] Biao Zhang, Matthias Nießner, and Peter Wonka. 3dilg: Irregular latent grids for 3d generative modeling. *NIPS*, abs/2205.13914, 2022. 3

[80] Fang Zhao, Wenhao Wang, Shengcai Liao, and Ling Shao. Learning anchored unsigned distance functions with gradient direction alignment for single-view garment reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12674–12683, 2021. 3

[81] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021. 3, 4, 5, 8

[82] Junsheng Zhou, Baorui Ma, Liu Yu-Shen, Fang Yi, and Han Zhizhong. Learning consistency-aware unsigned distance

functions progressively from raw point clouds. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3, 5

[83] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. 3