Few '*Zero Level Set*'-Shot Learning of Shape Signed Distance Functions in Feature Space

Amine Ouasfi Adnane Boukhayma

Inria, Univ. Rennes, CNRS, IRISA, M2S, France

Abstract. We explore a new idea for learning based shape reconstruction from a point cloud, based on the recently popularized implicit neural shape representations. We cast the problem as a few-shot learning of implicit neural signed distance functions in feature space, that we approach using gradient based meta-learning. We use a convolutional encoder to build a feature space given the input point cloud. An implicit decoder learns to predict signed distance values given points represented in this feature space. Setting the input point cloud, i.e. samples from the target shape function's zero level set, as the support (i.e. context) in few-shot learning terms, we train the decoder such that it can adapt its weights to the underlying shape of this context with a few (5) tuning steps. We thus combine two types of implicit neural network conditioning mechanisms simultaneously for the first time, namely feature encoding and meta-learning. Our numerical and qualitative evaluation shows that in the context of implicit reconstruction from a sparse point cloud, our proposed strategy, i.e. meta-learning in feature space, outperforms existing alternatives, namely standard supervised learning in feature space, and meta-learning in euclidean space, while still providing fast inference.

1 Introduction

One of the driving motives behind the ongoing research in 3D computer vision is enabling machines to reason about and understand 3D given limited observations in the same way we humans can evidently do. This ability is in turn crucial for most downstream 3D based computer vision and machine learning tasks. A popular instance of this ability is manifested in the problem of full 3D shape reconstruction from a sparse incomplete point cloud. The prominence of this problem is additionally due to the ubiquity of such partial inputs, either as acquired through the increasingly accessible 3D scanning technologies, or being an intermediate output of numerous classical computer vision algorithms such as Structure from Motion or Multi-View Stereo. Classical solutions to this problem such as Poisson surface reconstruction [31] still offer competitive reconstruction performances from dense point sets. However, as the inputs get sparser and less complete, learning based approaches become naturally more suitable to the task by virtue of their capacity to reason about shapes more globally and inpaint missing information based on previously seen examples.

A class of the these learning based approaches that emerged recently proposes to represent shapes in the form of an implicit function whose zero level set coincides with the surface, parameterised by a neural network. Compared to their traditional alternatives, these representations offer many advantages, most notably enabling modelling shapes with variable topology unlike point clouds and meshes, and operating virtually at infinite spatial resolution unlike voxel grids. In practice, these shape functions are typically multi layer perceptrons mapping the domain to the co-domain, i.e. 3D euclidean space to occupancies or signed distances. The zero level set of the inferred field can be rendered differentiably through e.g. variants of ray marching [27] and tessellated into explicit meshes with e.g. Marching Cubes [43]. Coupling these implicit neural functions with a conditioning mechanism allows generalization across multiple shapes. For instance, combining their inputs with features generated from an additional encoder network yields single forward pass inference models that can learn to reconstruct from various input modalities. In particular, recent models [13,52] obtaining state-of-the-art performances on reconstruction from point cloud benchmarks [10] use a convolutional encoder that builds a feature embedding for euclidean points given the input point cloud. The implicit neural shape function learns to map these points from that feature space to their occupancy or signed distance values. These models are trained using dense points sampled near the surface with corresponding ground-truth singed distance or occupancy values. Our aim here is to improve the performance of such models with negligible additional test-time computational cost.

As obtaining larger training data corpora remains prohibitive in 3D, most recent advances in this avenue focus on revamping the models, e.g. their architectures [58], input representations [64,13,52], training objectives [25,38], and training procedures [18], while remaining within the standard supervised learning paradigm. Conversely, we propose here to cast the problem of surface reconstruction from a point cloud, with an encoder endowed implicit neural function, as a few-shot learning problem.

Beyond merely using the input point cloud in a single encoding forward pass for inference, we observe that we can additionally further fine-tune the conditioned shape function using the point cloud elements as training samples [25], as they naturally belong to the surface and hence can be used to further overfit the shape signed distance function with their zero target values. To ensure this fine-tuning improves the initial result and that it is initialized from optimal shape function weights, we formalize it in a more principled learning strategy that is few-shot learning [69,61,21]. Each shape is represented by a support set: the points of the input point cloud, and a query set: the dense pre-sampled training points. For a given shape, the objective is to optimize predictions on the query set, i.e. adapt the shape function to the current shape, using the support set. We implement this strategy using gradient based meta-learning, namely the MAML algorithm [21]. At every training step on a given shape, the adaptation consists in back-propagating the loss on the sparse support at the surface for a few iterations (5 steps). The main shape function's parameters are then updated by back-propagating the loss of the adapted shape function on the dense query set. Notice that by representing points in feature space during this process, we combine two types of implicit shape function conditioning through both the encoder and meta-learning for the first time.

Using standard test beds we show that our approach outperforms comparable baselines in various 3D shape reconstruction metrics, and we provide qualitative results that support this as well. Through our experiments, we show that using few-shot leaning in feature space improves on both standard supervised learning in feature space (IF-Nets [13]) and few-shot learning in euclidean space (MetaSDF [57]), both in single and multi-class shape setups for shape reconstruction from a sparse point cloud (less than 3k input points). The performance gap w.r.t. our standard supervised learning baseline increases even further with coarser inputs. We note that we follow the same experimental data setup as in our baselines IF-Nets and MetaSDF. We also point that while we use IF-Nets as our backbone model in this work, this idea could be extended to any convolutional encoder equipped implicit neural shape network.

2 Related Work

We review in this section work that we deemed most relevant to the context of our contribution.

Traditional Shape Representations Perhaps an intuitive way to categorize 3D shape representations within deep learning frameworks is into intrinsic and extrinsic representations. Intrinsic representations are efficient in that they are discretizations of the shape itself. however when represented explicitly, as in meshes [70,30] or point clouds [20], they are inherently limited to a fixed topology, which is unpractical for generating varying shape objects and classes. Other forms of intrinsic representations include combining 2D patches [26,73,17], 3D shape primitives such as cuboids [67,81], planes [39] and Gaussians [24]. However patches induce discontinuities, and the approximation quality of primitive shapes remains limited by their simplicity. Extrinsic representations on the other hand model the 3D space containing the shape of interest. The most adopted one to date has been voxel grids [76,75], being a natural extension of 2D pixels to 3D. Nonetheless, the cubic memory cost in voxel grid resolution limits the ability to represent details. Sparse voxel representations such as octrees [55,65,71] can help alleviate these memory efficiency issues albeit with complex implementations.

Implicit Neural Shape Representations Recent years have seen a surge in extrinsic implicit neural shape representations for modelling 3D objects and scenes. Thanks to their ability to continuously represent detailed shapes with arbitrary topologies in a memory-efficient way, these representations remedy many of the shortcomings of the aforementioned traditional alternatives, and are currently drawing increasing attention both in 3D shape and appearance modelling (e.g. [46,32,78]). Implicit neural shape models are typically parameterized with MLPs that map 3D space to occupancy [45], signed [50] or unsigned distances

[14] relative to the shape. Different forms of training supervision have been proposed, the most common one being 3D points pre-sampled around the surface, with weaker forms of supervision such as 2D segmentation masks through a 2D based SDF lower bound [37], or color and depth images [48,32,78] through differentiable rendering [29,42,29]. Recent contributions in this area include learning octree scafolded implicit mooving least squares [41], representing shapes as an implicit template and an implicit warp [80], and implicit/explicit hybrid representations [11,16,79] based on differentiable space partitioning.

Conditioning Implicit Neural Shape Models Implicit shape models require conditioning mechanisms to represent more than a single shape. The mechanisms explored so far include concatenation, batch normalization, hypernetworks [58,60,59] and meta-learning [57]. Concatenation like conditioning was first introduced through a single latent code [45,12,50], and subsequently improved through the use of local features [23,66,62,52,13,28].

Current methods that meta-learn implicit 3D neural representations use gradient based meta-learning (e.g. MAML [21], Reptile [47]) to learn a meta-radiance field that can be adapted from images [63,22], or a meta-SDF that can be adapted from both zero level set and random domain samples [57]. In contrast to these methods, we propose here to combine encoder-based local feature concatenation conditioning and meta-learning conditioning in the same model, performing implicit reconstruction from a sparse point cloud. We note also that in the work by Sitzmann et al., the meta-learning conditioning requires many surface samples (10k new points sampled at each of the 5 MAML iterations = 50k pts). Differently, we extend this idea to a true few-shot reconstruction setup (300 or 3k fixed input points) and multi shape class for the first time, and show that it can only scale thusly in feature space.

Reconstruction From a Point Cloud Among classical solutions to this task, combinatorial approaches define shapes with a space partitioning based on the input points, using e.g. alpha shapes [4] Voronoi diagrams [1] or triangulation [8,40,54]. Alternatively, implicit function based approaches use the point samples to define a function whose zero level set approximates the surface, through fitting e.g. radial basis functions [7], Gaussian kernels [56], piece-wise polynomials [49], moving least-squares [34,41], or by solving a Poisson equation [31]. Closer to our scope, recent work proposes to obtain these implicit functions through deep learning. These include two families of work: supervised and unsupervised ones.

For the latter, a neural network is fitted to the raw input point cloud without any further supervision. Among contributions in this area, Gropp et al. [25] introduces a regularization on the function's spatial gradient based on the Eikonal equation. Atzmon et al. learns a signed distance function from unsigned distance supervision [2], and further supervises the spatial gradient of the function using point normals [3]. Ma et al. [44] supervises the training through expressing the nearest neighbor on the surface as a function of the neural signed distance and its spatial gradient. All of the aforementioned methods benefit from efficient gradient computation through back-propagation in the implicit neural function.

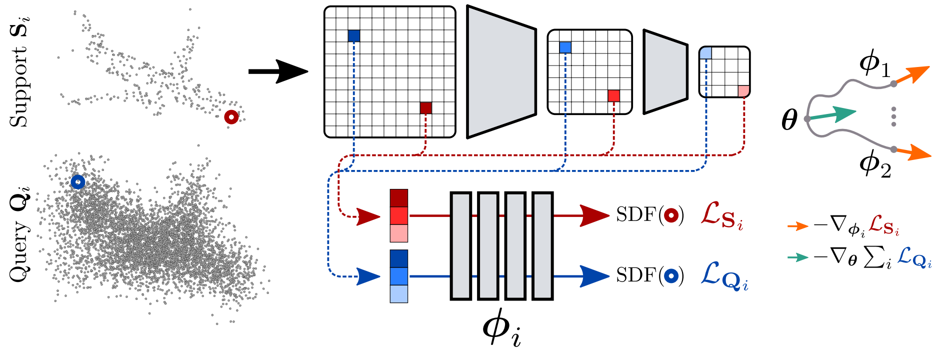
Support $\mathbf{S}_i$
Query $\mathbf{Q}_i$
ϕ_1
$\boldsymbol{\theta}$
ϕ_2
SDF(●) $\mathcal{L}_{\mathbf{S}_i}$
SDF(●) $\mathcal{L}_{\mathbf{Q}_i}$
$-\nabla_{\phi_i}\mathcal{L}_{\mathbf{S}_i}$
$-\nabla_{\boldsymbol{\theta}}\sum_i \mathcal{L}_{\mathbf{Q}_i}$
ϕ_i

Fig. 1: Overview of our method. Our input is a sparse point cloud (Support $\mathbf{S}_i$) and our output is an implicit neural SDF f. f is a neural network comprised of a convolutional encoder (top in gray) and an MLP decoder (bottom in gray). The decoder predicts SDF values for 3D points (red/blue circles) through their spatially sampled features (squares in shades of red/blue) from the encoder's activation maps. Following a gradient-based few-shot learning algorithm (MAML [21]), we learn a meta-decoder in encoder feature space, parameterized with $\boldsymbol{\theta}$, that can quickly adapt to a new shape, i.e. new parameters $\boldsymbol{\phi_i}$, given its support. This is achieved by iterating per-shape 5-step adaptation gradient descent (orange arrow) using the support loss $\mathcal{L}_{\mathbf{S}_i}$, and one-step meta gradient-descent (green arrow) by back-propagating the Query set ($\mathbf{Q}_i$) loss $\mathcal{L}_{\mathbf{Q}_i}$ evaluated with the specialized parameters $\boldsymbol{\phi_i}$ w.r.t. the meta-parameters $\boldsymbol{\theta}$. At test time, 5 fine-tuning iterations are performed similarly starting from the converged meta-model to evaluate f.

[58] introduces periodic activations. [74] proposes to learn infinitely wide shallow ReLU networks as random feature kernels. Lipman [38] formalizes a loss ensuring the function converges to occupancy while its log transform converges to a distance function.

Supervised methods on the other hand assume a training dataset of shapes with ground-truth signed distance or occupancy values for dense space samples i.e. points. Auto-decoding based methods e.g. [50,66,28,9] require test time optimization to fit the implicit function's features to the observed point cloud, which can take several seconds for a simple object. Conversely, encoder-decoder based approaches enable faster single forward pass inference and superior generalization. For these approaches, pooling-based set encoders (e.g. PointNet [53]) were first proposed [45,12,23,19], but they have been shown to underfit for large and detailed inputs. More recently, convolutional encoders [13,52,41] enable access to more expressive local point features and incorporate inductive biases such as translational equivariance, thus enabling fine-grained implicit reconstruction. We propose here to extend such supervised convolutional encoder-decoder models to a few-shot setting to further improve their reconstruction abilities from a sparse point cloud input, while still offering fast inference unlike auto-decoding.

3 Method

The task at hand is to recover a continuous shape surface $\mathcal{S}$ given an input point could $\mathbf{X} \subset \mathbb{R}^{3\times N_p}$ representing that underlying shape i.e. $\mathbf{X} = \{x_i \sim \mathcal{S}\}_{i=1}^{N_p}$. To this end, we train a deep neural network f to approximate the signed distance function of the target shape given point cloud $\mathbf{X}$. The inferred shape can then be obtained as the zero level set of f:

$$\hat{\mathcal{S}} = \{x \in \mathbb{R}^3 \mid f(\mathbf{X}, x) = 0\}. \tag{1}$$

We can reconstruct an explicit triangle mesh for shape $\hat{\mathcal{S}}$ using e.g. Marching Cubes [43]. We assume shapes to be watertight manifolds, and that they are all normalized into a domain $\Omega \subset \mathbb{R}^3$.

3.1 The Base Model

For our neural network f, we use an encoder-decoder architecture that follows the model introduced by Chibane et al. [13]. Such models (e.g. [13,52]) combining local features extracted with convolutional encoders with implicit decoders have been shown to yield superior performances in the class of single forward pass prediction methods for surface reconstruction. Differently from [13], we note that we learn signed distance instead of occupancy functions.

As illustrated in Fig.1, The encoder takes as input point cloud $\mathbf{X}$ and produces spatial feature maps. In order to apply this 3D convolutional network to the point cloud, the latter is first voxelized into a discrete 3D grid in $\mathbb{R}^{N\times N\times N}$ (cf. Fig.2), N being the input spatial resolution. It then passes through successive convolutional down-sampling blocks resulting in n multi-scale deep feature grids $\mathbf{F}_1, \dots, \mathbf{F}_n$, where $\mathbf{F}_k \in \mathbb{R}^{C_k\times N_k\times N_k\times N_k}$. The feature map channels C_k increase with the encoder's depth while their resolution decreases $N_k = N/2^{k-1}$. The shallow features represent local details while the deeper ones account for more global shape variation. Given a 3D point $x \in \Omega$, we can extract its encoder generated features using trilinear interpolation. We define this process with a neural function $\Psi_{\mathbf{X}} : \Omega \to \mathbb{R}^{C_1} \times \dots \times \mathbb{R}^{C_n}$ such that:

$$\Psi_{\mathbf{X}}(x) = (\mathbf{F}_1(x), \dots, \mathbf{F}_n(x)). \tag{2}$$

The decoder is tasked with predicting the signed distance to the ground-truth shape $\mathcal{S}$ for a given 3D point x. It uses the features obtained with the encoder as input point representation. It consists of a MLP with ReLU non-linearities and a final Tanh activation, and we denote it as $\Phi : \mathbb{R}^{C_1} \times \dots \times \mathbb{R}^{C_n} \to \mathbb{R}$. Hence we can express the approximated signed distance function given a point cloud $\mathbf{X}$ as follows:

$$f(\mathbf{X}, x) = \Phi \circ \Psi_{\mathbf{X}}(x). \tag{3}$$

In standard supervised learning, this network is trained by back-propagating the prediction loss over a set of training points $\mathbf{Y} \subset \Omega$ per training shape using their respective pre-computed ground-truth signed distance values. These dense point sets are typically built by sampling near the ground-truth surface $\mathcal{S}$, i.e. sampling points on the surface and offsetting them with normally distributed displacements:

$$\mathbf{Y} = \{x + n : x \sim \mathcal{S}, n \sim \mathcal{N}(0, \mathbf{\Sigma})\}, \tag{4}$$

where $\mathbf{\Sigma} = \text{diag}(\sigma) \in \mathbb{R}^{3\times3}$ is a diagonal covariance matrix. An illustrative example of such a set can be seen in the bottom left of Fig.1.

3.2 Few-shot Learning in Feature Space

We would like to build a model f that can learn to adapt to a new shape $\mathcal{S}$ given limited observations, namely the input point cloud $\mathbf{X}$. While the network is already conditioned to the input $\mathbf{X}$ through the encoder in the standard supervised learning regime e.g. [13,52], we seek here to adapt it even further to that input through meta-learning. Let us recall that for each training shape $\mathcal{S}_i$ we have two sets of points available: $\mathbf{X}_i$ the sparse input point cloud at the surface, and $\mathbf{Y}_i$ the dense point set sampled near the ground-truth surface. Corresponding ground-truth signed distances are available for both of these sets as well. However at test time, only $\mathbf{X}_i$ is available.

Support and Query Sets We position ourselves in a meta-learning based few-shot learning setup [69,61,21]. Traditionally, a network is trained to adapt to a new task given limited training samples in this setup. A task is defined with a loss, a support (or context) set and a query set. These sets are input-target pairs for the given task. The model is trained to perform tasks on their query sets, after being adapted to them through e.g. metric learning [69] or gradient descent [21] on their respective limited support sets. We adopt the same strategy, where a task consists in learning the signed distance function f for a given shape $\mathcal{S}_i$. We define the support set as the pairs made of the points of the input point cloud and their corresponding ground-truth signed distances, as such a set is available at test time:

$$\mathbf{S}_i = \{(x, s) : x \in \mathbf{X}_i, s := \text{SDF}(x)\}. \tag{5}$$

$$= \{(x, 0) : x \in \mathbf{X}_i\}. \tag{6}$$

Since $\mathbf{X}_i$ contains exclusively points from the surface, i.e. the zero level set of the shape function, all ground-truth singed distances are null. We define the query set as the pairs made of the dense points pre-sampled around the surface and their corresponding ground-truth singed distance values:

$$\mathbf{Q}_i = \{(x, s) : x \in \mathbf{Y}_i, s := \text{SDF}(x)\}. \tag{7}$$

Meta-Learning in Feature Space We apply gradient-based meta-learning to our supervised few-shot shape function learning, in particular the MAML algorithm by Finn et al. [21]. For a given shape $\mathcal{S}_i$, and assuming a pre-trained

Algorithm 1 The training procedure of our model.

```
Input: Dataset fo shapes $\mathcal{S}_i$, pre-trained encoder $\Psi$, meta-decoder learning rate $\beta$
Output: meta-decoder weights $\boldsymbol{\theta}$, decoder learning rates $\boldsymbol{\alpha}$
  initialize $\boldsymbol{\theta}$, $\boldsymbol{\alpha}$
  while not done do
      sample batch of shapes $\{\mathcal{S}_i\} := \{(\mathbf{X}_i, \mathbf{Q}_i)\}$
      initialize $\mathcal{L}_{\mathbf{Q}} \leftarrow 0$
      for $\mathcal{S}_i$ in $\{\mathcal{S}_i\}$ do
          initialize $\boldsymbol{\phi}_i \leftarrow \boldsymbol{\theta}$
          for $K$ times do
              $\mathcal{L}_{\mathbf{S}_i} = \sum_{x \in \mathbf{X}_i} |\Phi_{\boldsymbol{\phi}_i} \circ \Psi_{\mathbf{X}_i}(x)|$
              $\boldsymbol{\phi}_i \leftarrow \boldsymbol{\phi}_i - \boldsymbol{\alpha} \odot \nabla_{\boldsymbol{\phi}_i} \mathcal{L}_{\mathbf{S}_i}$
          end for
          $\mathcal{L}_{\mathbf{Q}} \leftarrow \mathcal{L}_{\mathbf{Q}} + \sum_{(x,s) \in \mathbf{Q}_i} |\Phi_{\boldsymbol{\phi}_i} \circ \Psi_{\mathbf{X}_i}(x) - s|$
      end for
      $(\boldsymbol{\theta}, \boldsymbol{\alpha}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{\alpha}) - \beta \nabla_{\boldsymbol{\theta}, \boldsymbol{\alpha}} \mathcal{L}_Q$
  end while
```

encoder Ψ, the signed distance function f is obtained through a specialization denoted $\boldsymbol{\phi}_i$ of the parameters $\boldsymbol{\theta}$ of an underlying meta-decoder $\Phi_{\boldsymbol{\theta}}$ operating in feature space $\Psi_{\mathbf{X}_i}(\Omega)$:

$$f(\mathbf{X}_i, x) = \Phi_{\boldsymbol{\phi}_i} \circ \Psi_{\mathbf{X}_i}(x). \tag{8}$$

For lower computational and memory costs and a less noisy meta-learning loss (cf. Section 4.4), we fix the convolutional encoder Ψ after pre-training it. This encoder is pre-trained by training the base model f in the standard supervised learning regime using the training dataset's query sets $\{\mathbf{Q}_i\}$ for supervision (i.e. standard supervised learning). As such, the meta-learning of model f consists in training the meta-decoder $\Phi_{\boldsymbol{\theta}}$ in feature space. Each training step in this process is two fold: First, a fixed number of inner training steps, i.e. adaptation of the meta-decoder $\Phi_{\boldsymbol{\theta}}$ into $\Phi_{\boldsymbol{\phi}_i}$, followed by an outer training step, i.e. update of the meta-decoder $\Phi_{\boldsymbol{\theta}}$. Similarly to Sitzmann et al. [57], we build on the Meta-SGD [35] MAML [21] variant proposed by Li et al., which advocates the use of per-parameter learning rates in the adaptation stage for improved flexibility.

Given a batch of training shapes $\{\mathcal{S}_i\}$, the inner training step of the decoder is performed for each shape $\mathcal{S}_i$ independently. The L_1 loss $\mathcal{L}_{\mathbf{S}_i}$ is computed using the current specialized decoder $\Phi_{\boldsymbol{\phi}_i}$ over the support set $\mathbf{S}_i$ (i.e. $\mathbf{X}_i$), and is back-propagated w.r.t. $\boldsymbol{\phi}_i$:

$$\mathcal{L}_{\mathbf{S}_i} = \sum_{x \in \mathbf{X}_i} |\Phi_{\boldsymbol{\phi}_i} \circ \Psi_{\mathbf{X}_i}(x)|, \tag{9}$$

$$\boldsymbol{\phi}_i \leftarrow \boldsymbol{\phi}_i - \boldsymbol{\alpha} \odot \nabla_{\boldsymbol{\phi}_i} \mathcal{L}_{\mathbf{S}_i}, \tag{10}$$

where weights $\boldsymbol{\phi}_i$ are initialized with the current meta-decoder weights $\boldsymbol{\theta}$ for all the batch shapes. $\boldsymbol{\alpha}$ contains the per parameter learning rates, which are learned as part of the outer training loop. $\odot$ symbolizes element-wise product. We note that while the support loss $\mathcal{L}_{\mathbf{S}_i}$ could include additional regularisation such as the Eikonal constraint [25], we keep it simple to limit the computational footprint of the meta-learning.

After K such shape specific adaptation steps, one outer training step is performed for the entire batch of shapes. The L_1 losses $\{\mathcal{L}_{\mathbf{Q}_i}\}$ are computed using

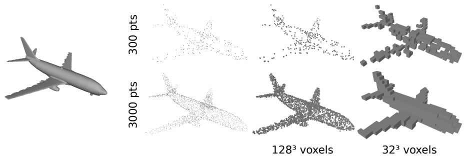

Fig. 2: Visualization of voxelizations at resolutions 32^3 and 128^3 of input point clouds with 300 and 3000 points.

the specialized decoders $\{\Phi_{\boldsymbol{\phi}_i}\}$ over their respective query sets $\{\mathbf{Q}_i\}$, and their average is back-propagated w.r.t. the meta-parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ accordingly:

$$\mathcal{L}_{\mathbf{Q}_i} = \sum_{(x,s)\in\mathbf{Q}_i} |\Phi_{\boldsymbol{\phi}_i} \circ \Psi_{\mathbf{X}_i}(x) - s|, \tag{11}$$

$$(\boldsymbol{\theta}, \boldsymbol{\alpha}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{\alpha}) - \beta \nabla_{\boldsymbol{\theta},\boldsymbol{\alpha}} \sum_i \mathcal{L}_{\mathbf{Q}_i}, \tag{12}$$

where β is a scalar learning rate. For ease of understanding, Algorithm 1 provides a summary of this training procedure.

At test time, given an input $\mathbf{X}$, the inference consists in a forward pass of the model f after a K-step adaptation of the converged meta-decoder $\Phi_{\boldsymbol{\theta}}$. To produce mesh reconstructions, we use the model to predict signed distance values of a grid of points at a desired resolution, and then apply the Marching Cubes [43] algorithm on the inferred signed distance grid.

4 Results

We present in this section our experimental setup and showcase our results. We evaluate our method on both multi-class and single class setups in ShapeNet [10] on reconstruction from a sparse point cloud, and we also show results on the FAUST [5] dataset. We follow the noise-free benchmark in our baselines IF-Nets [13] and MetaSDF [57]. We experiment with two sizes of input point clouds $N_p = 3000$ and $N_p = 300$ similarly to [13], and two voxelization resolutions $N = 128$ and $N = 32$ (Fig.2) of these point sets. Let us recall that inputs require voxelization due to the 3D convolutional encoder of our base model [13]. We evaluate numerically in terms of Intersection over Union (IoU), L1 ($\mathrm{CD}_1 \times 10^{-1}$) and L2 ($\mathrm{CD}_2 \times 10^{-3}$) Chamfer distance. We detail the expressions of these metrics in the supplementary material. In summary, our proposed approach outperforms all baselines including the same base model trained in standard supervised learning (IF-Nets [13]), and the decoder of the base model trained through meta-learning (MetaSDF [57]). Results show additionally that our approach is more resilient to coarser inputs compared to IF-Nets. Besides, the performance increase brought by our approach comes with minimal additional computational cost, as inference takes 150 ms for our model, and 60 ms for IF-Nets, on a RTX A4000.

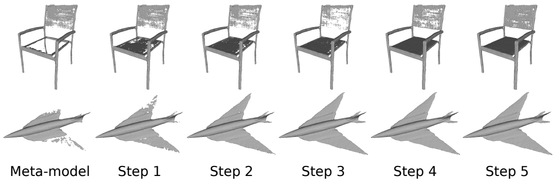

Fig. 3: Visualization of reconstructions from 3000 points throughout the $K = 5$ testing inner-loop iterations in the ShapeNet multi-class setup.

4.1 Implementation Details

The base model follows the architecture in [13], hence we use $n = 6$ feature maps with feature dimensions $C_1 = 1$, $C_2 = 16$, $C_3 = 32$, $C_4 = 64$, $C_5 = 128$ and $C_6 = 128$. Regarding the meta-learning, we use $K = 5$ steps in the inner training loop and we initialize the per-parameter learning rates α with 10^{-6}. In the outer loop, we set the meta-decoder learning rate to $\beta = 10^{-6}$. We train for 100 epochs with batches of 4 shapes, leveraging the N_p training points in the inner loop, and 50k training points in the outer loop per shape. To train the base model in the standard supervised learning mode, we perform a maximum of 50 epochs with a learning rate of 10^{-5}, using batches of 8 shapes with 50k training points per shape. All trainings use the Adam[33] solver on a RTX A4000 in the PyTorch [51] framework. All Marching Cubes reconstructions are done with a 256^3 sized grid.

4.2 Datasets

Similar to prior work we evaluate our method using the ShapeNet benchmark [10] which consists of various instances of 13 different object classes. Similarly to [13], we use the pre-processing by [77] to obtain watertight meshes which enables computing ground-truth signed distances. All meshes are subsequently normalized using their bounding boxes thus fitting inside the domain $\Omega = [-1, 1]^3$. We use the train/test split provided by [13], which is based on the original split of Choy et al. [15] minus 508 distorted shapes due to pre-processing failures. To create the input point cloud $\mathbf{X}$ for a given shape, N_p sized sets of points are randomly pre-sampled from the processed mesh. For the training points with ground-truth signed distances, we pre-sample 100k points near the surface with $\sigma = 0.1$ and $\sigma = 0.01$ (cf. Equ. 4). At training, 50k points are sampled equally from these pre-made two sets to make the per shape training points batch $\mathbf{Y}$. We also use the FAUST dataset [5] for testing. It consists of 100 registered meshes of 10 human body identities in 10 different poses.

4.3 Multiple Shape Class Evaluation

We evaluate here our work and the competition using the entire ShapeNet dataset, which counts 26834 training shapes and 7148 testing ones. For the input

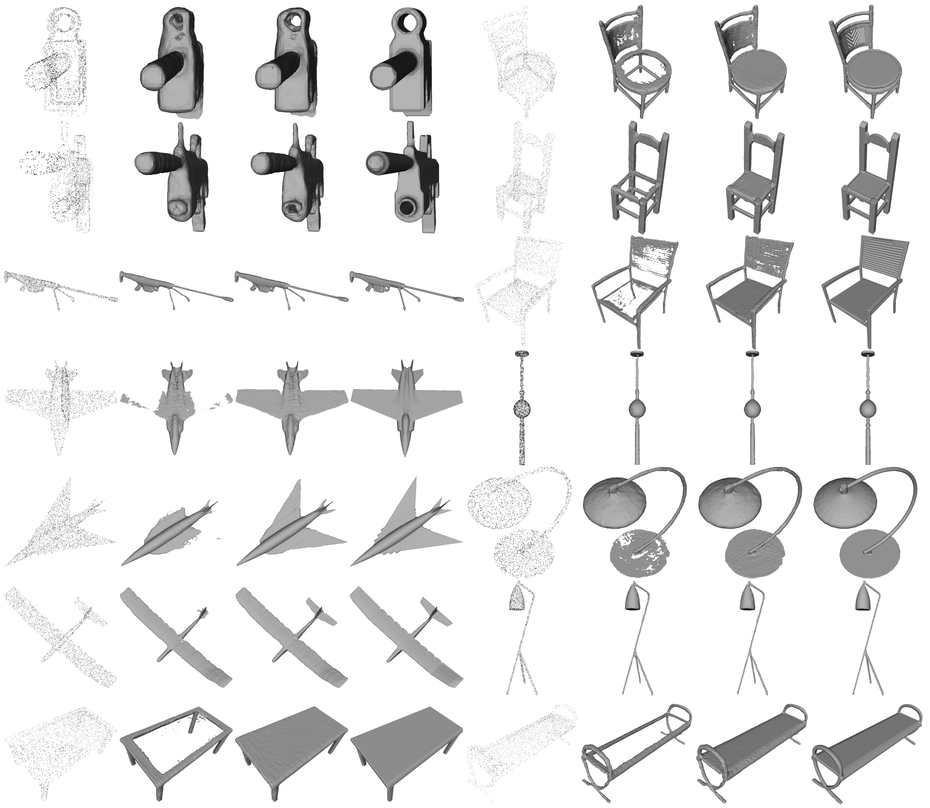

Fig. 4: Qualitative comparison of reconstructions from 3000 points on ShapeNet with our main baseline, i.e. same base model in standard supervised learning. (Input / IF-Nets[13] / **Ours** / Ground-truth).

point cloud size and voxelization resolution of our method, we consider here the two extreme cases for brevity: i.e. 3000 points at 128^3 resolution (Tab.1a), and 300 points at 32^3 resolution (Tab.1b). We refer the reader to a more detailed analysis in the ablative single class evaluation (Section 4.4). We relay the performances of OccNet [45], DMC [36], PSGN [20] as they were reported in [13]. We reproduced the performance of ConvOccNet [52] with 3000 input points in the noise-free benchmark of IF-Nets [13] and we obtained an IoU of 0.86, which is also confirmed by the results of other contemporary work [72] (see fig. 9 in that paper). The authors of ConvOccNet report higher numbers (0.88), where noise is added to the input point cloud. For fairness, we report ConvOccNet's higher numbers (i.e. 0.88 IoU). We train our own IF-Nets [13] model on signed distances and reproduce the same results in the main paper. MetaSDF here refers to our implementation of the work in [57] with 3000 input surface points, i.e. our model without an encoder trained for hundreds of epochs, from which we perform nu-

	IoU↑	CD$_1$↓	CD$_2$↓
OccNet	0.72	–	0.4
DMC	0.65	–	0.1
PSGN	–	–	0.4
MetaSDF	0.63	0.123	0.458
ConvOccNet	0.88	0.044	–
IF-Nets	0.88	0.032	0.032
Ours	**0.91**	**0.028**	**0.026**

(a)

	IoU↑	CD$_1$↓	CD$_2$↓
OccNet	0.73	–	0.3
DMC	0.58	–	0.3
PSGN	–	–	0.4
MetaSDF (3k pts)	0.63	0.123	0.458
IF-Nets	0.67	0.091	0.232
Ours	**0.74**	**0.070**	**0.209**

(b)

Table 1: Reconstruction on ShapeNet from (a) 3000 points voxelized at resolution 128^3, and (b) 300 voxelized at 32^3.

	IoU↑	CD$_1$↓	CD$_2$↓
IF-Nets	0.82	0.037	0.060
Ours	**0.84**	**0.035**	**0.051**

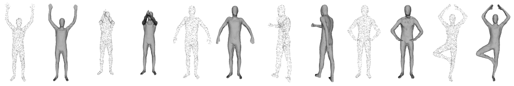

Table 2: Reconstruction on FAUST with models trained on ShapeNet, from 3000 points voxelized at resolution 128^3. Qualitative results from our model.

merous evaluations and pick the best one. We note that original paper [57] only showed results for 10k input points. This same number is reported in table 1b under the name MetaSDF (3k pts).

Tables 1a and 1b report the average reconstruction performance from 3000 and 300 input points respectively on the entire multi-class testing set. We additionally provide the per-class numbers in the supplementary material. PSGN generates point sets with competitive distances to the ground-truth but does not provide any connectivity (Hence the absence of IoU). DMC's performance is limited by its voxel grid resolution. OccNet performs strongly and almost similarly in the 300 and 3000 input cases, which suggests that pooling set encoders underfit the context. For both input situations and across all metrics, our method outperforms the competition, including convolutional encoder equipped implicit shape models (IF-Nets and ConvOccNet), and our encoder-free meta-learning baseline MetaSDF. We find the performance of the latter particularly underwhelming, which suggests that despite the encouraging single class reconstruction results in [57] from 10k input surface points, such strategy struggles to scale to more challenging settings with multiple classes of shape, sparser point clouds, and under relatively limited training time. When decreasing the input size and voxel resolution in Table 1b, both the encoders of our method and IF-Nets are exposed to very poor inputs (cf. Fig 2). While the performance of IF-Nets is heavily affected by these coarser inputs, our method is more resilient thanks to the meta-learning addition. In fact, our IoU drops by 18% compared to 23% for the standard supervisedly learned baseline. We additionally show reconstruction results on the FAUST dataset from models trained on ShapeNet in Table 2, where out method outperforms IF-Nets. Note that neither models have seen human shapes nor articulated shapes for that matter in training.

The numerical superiority of our method w.r.t. our closest baseline (IF-Nets) is supported with qualitative comparisons in Fig.4. We notice that our method manages to recover more thin structures and fine topological features, such as cylindrical holes in rifles, wings in planes, thin flat surfaces in tables, benches and chairs. In addition, we provide examples in Fig.3 showing the evolution of the reconstruction at various iterations of our inference.

4.4 Single Shape Class Ablation

We show further quantitative evaluations in a single shape class setup on the largest class of ShapeNet, i.e. table, for ablative purposes and also for a more fair comparison to MetaSDF [57]. The table class counts 5364 training shapes and 1679 testing ones. We report numbers for MetaSDF and a pooling set encoder based implicit model (PoinNet enc.) from their paper [57] using 10k input points. We compare multiple variants of our approach. For our method (Ours), the meta-decoder is initialized from the weights of a pre-trained base model. For Ours w/o dec. pret., it is initiated with the standard PyTorch initialization instead. Finally Ours w/o meta learn. (i.e. IF-Nets) is again our base model trained in standard supervised learning.

Tables 3 and 4 show reconstruction results from 3000/300 points at 128^3 and 32^3 input voxel resolution respectively. Even under 10k input points, MetaSDF [57] can yet barely reach our performance on just 300 input points. Most notably, our method improves on the standard supervised learning baseline (Ours w/o meta learn. (i.e. IF-Nets)) across all input sizes and input voxel discretizations for all metrics. While the input point cloud size affects both our method and IF-Nets almost equally, decreasing the encoder's input resolution hinders IF-Nets's performance substantially more severely. In fact, when going from 128^3 to 32^3 resolution inputs, our IoU drops by roughly 14% vs. 20% for IF-Nets for 3000 points, and by 13% vs. 19% for IF-Nets when using 300 points.

Decoder pre-training As witnessed by tables 3 and 4, while initializing the meta-decoder can improve the performance slightly, it is not crucial for obtaining satisfactory results, which suggests that we learn a proper meta-decoder and not just a fine-tuned base-model.

Encoder pre-training and tuning As meta-learning both the encoder and decoder is computationally and memory expensive, we only meta-learn the decoder. We found that tuning the encoder during this meta-learning leads to noisy losses, without a clear improvement in the results. In fact, for reconstruction from 3000 points at resolution 128^3 in class table, whilst fixing the encoder yields an IoU of 0.87, tuning it gives a comparable performance (0.86) while requiring more time and memory for training. We found the resulting noisier loss makes it also harder to decide the convergence epoch in this case. Thus we fix the encoder after pre-training it. We pre-train the encoder by training the encoder-decoder in the standard supervised learning setup.

	IoU↑		CD$_1$↓		CD$_2$↓	
PointNet enc. (10k pts)	0.66		–		0.69	
MetaSDF (10k pts)	0.75		–		0.32	
Ours w/o meta learn. (i.e. IF-Nets)	0.82	0.72	0.040	0.057	0.062	0.097
Ours w/o dec. pret.	0.86	0.74	0.035	0.057	0.030	0.203
Ours	**0.87**	**0.76**	**0.033**	**0.051**	**0.030**	**0.082**

Table 3: Reconstruction on class table of ShapeNet from 3000 (left) and 300 (right) points voxelized at resolution 128^3.

	IoU↑		CD$_1$↓		CD$_2$↓	
Ours w/o meta learn. (i.e. IF-Nets)	0.65	0.58	0.071	0.092	0.089	0.169
Ours w/o dec. pret.	0.73	0.61	0.057	0.083	0.082	0.169
Ours	**0.74**	**0.66**	**0.052**	**0.076**	**0.068**	**0.142**

Table 4: Reconstruction on class table of ShapeNet from 3000 (left) and 300 (right) points voxelized at resolution 32^3.

5 Limitations

As SDFs can only represent closed surfaces, we will experiment next with other representations such as points [68,6] and unsigned distances [14]. Point cloud voxelization (cf. Fig.2) hinders the expressiveness of the input, thus we will be considering different convolutional encoders subsequently. Furthermore, the MAML algorithm [21] requires computing second-order gradients which raises the memory complexity in training. Finally, we follow here the noise-free benchmarks in our baselines IF-Nets [13] and MetaSDF [57]. Considering noisy and real inputs (e.g. 2.5D, SFM, etc.) is part of our future work.

6 Conclusion

We proposed to perform 3D shape reconstruction from a sparse point cloud using a implicit neural model conditioned with both encoder generated local features and meta-learning simultaneously. Our results demonstrate numerically and qualitatively that this approach improves on its standard supervised learning counterpart with minimal additional test time computational cost, and this performance gap increases for coarser inputs. Future avenues of improvement include tackling more real world downstream tasks such as partial shape reconstruction, making use of normals, meta-learning of reconstruction from images and depth maps through differentiable rendering, and exploring other meta-learning techniques.

References

1. Amenta, N., Choi, S., Kolluri, R.K.: The power crust, unions of balls, and the medial axis transform. CG (2001) 4

2. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: CVPR (2020) 4
3. Atzmon, M., Lipman, Y.: Sald: Sign agnostic learning with derivatives. In: ICML (2020) 4
4. Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G.: The ball-pivoting algorithm for surface reconstruction. TVCG (1999) 4
5. Bogo, F., Romero, J., Loper, M., Black, M.J.: FAUST: Dataset and evaluation for 3D mesh registration. In: CVPR (2014) 9, 10
6. Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N., Hariharan, B.: Learning gradient fields for shape generation. In: ECCV (2020) 14
7. Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3d objects with radial basis functions. In: SIGGRAPH (2001) 4
8. Cazals, F., Giesen, J.: Effective Computational Geometry for Curves and Surfaces (2006) 4
9. Chabra, R., Lenssen, J.E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., Newcombe, R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In: ECCV (2020) 5
10. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015) 2, 9, 10
11. Chen, Z., Tagliasacchi, A., Zhang, H.: Bsp-net: Generating compact meshes via binary space partitioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020) 4
12. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: CVPR (2019) 4, 5
13. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: CVPR (2020) 2, 3, 4, 5, 6, 7, 9, 10, 11, 14
14. Chibane, J., Mir, A., Pons-Moll, G.: Neural unsigned distance fields for implicit function learning. In: NeurIPS (2020) 4, 14
15. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: ECCV (2016) 10
16. Deng, B., Genova, K., Yazdani, S., Bouaziz, S., Hinton, G., Tagliasacchi, A.: Cvxnet: Learnable convex decomposition. In: CVPR (2020) 4
17. Deprelle, T., Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: Learning elementary structures for 3d shape generation and matching. In: NeurIPS (2019) 3
18. Duan, Y., Zhu, H., Wang, H., Yi, L., Nevatia, R., Guibas, L.J.: Curriculum deepsdf. In: ECCV (2020) 2
19. Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N.J., Wimmer, M.: Points2surf learning implicit surfaces from point clouds. In: ECCV (2020) 5
20. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: CVPR (2017) 3, 11
21. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017) 2, 4, 5, 7, 8, 14
22. Gao, C., Shih, Y., Lai, W.S., Liang, C.K., Huang, J.B.: Portrait neural radiance fields from a single image. arXiv preprint arXiv:2012.05903 (2020) 4
23. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local deep implicit functions for 3d shape. In: CVPR (2020) 4, 5

24. Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. In: ICCV (2019) 3
25. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. In: ICML (2020) 2, 4, 8
26. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: CVPR (2018) 3
27. Hart, J.C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. The Visual Computer (1996) 2
28. Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T., et al.: Local implicit grid representations for 3d scenes. In: CVPR (2020) 4, 5
29. Jiang, Y., Ji, D., Han, Z., Zwicker, M.: Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In: CVPR (2020) 4
30. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: CVPR (2018) 3
31. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. TOG (2013) 1, 4
32. Kellnhofer, P., Jebe, L.C., Jones, A., Spicer, R., Pulli, K., Wetzstein, G.: Neural lumigraph rendering. In: CVPR (2021) 3, 4
33. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 10
34. Kolluri, R.: Provably good moving least squares. TALG (2008) 4
35. Li, Z., Zhou, F., Chen, F., Li, H.: Meta-sgd: Learning to learn quickly for few-shot learning. In: NeurIPS (2017) 8
36. Liao, Y., Donne, S., Geiger, A.: Deep marching cubes: Learning explicit surface representations. In: CVPR (2018) 11
37. Lin, C.H., Wang, C., Lucey, S.: Sdf-srn: Learning signed distance 3d object reconstruction from static images. In: NeurIPS (2020) 4
38. Lipman, Y.: Phase transitions, distance functions, and implicit neural representations. In: ICML (2021) 2, 5
39. Liu, C., Yang, J., Ceylan, D., Yumer, E., Furukawa, Y.: Planenet: Piece-wise planar reconstruction from a single rgb image. In: CVPR (2018) 3
40. Liu, M., Zhang, X., Su, H.: Meshing point clouds with predicted intrinsic-extrinsic ratio guidance. In: ECCV (2020) 4
41. Liu, S.L., Guo, H.X., Pan, H., Wang, P.S., Tong, X., Liu, Y.: Deep implicit moving least-squares functions for 3d reconstruction. In: CVPR (2021) 4, 5
42. Liu, S., Saito, S., Chen, W., Li, H.: Learning to infer implicit surfaces without 3d supervision. In: NeurIPS (2019) 4
43. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: SIGGRAPH (1987) 2, 6, 9
44. Ma, B., Han, Z., Liu, Y.S., Zwicker, M.: Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. In: ICML (2021) 4
45. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: CVPR (2019) 3, 4, 5, 11
46. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 3
47. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999 (2018) 4

48. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: CVPR (2020) 4
49. Ohtake, Y., Belyaev, A., Alexa, M.: Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing. In: SGP (2005) 4
50. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: CVPR (2019) 3, 4, 5
51. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. NeurIPS (2019) 10
52. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: ECCV (2020) 2, 4, 5, 6, 7, 11
53. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2017) 5
54. Rakotosaona, M.J., Aigerman, N., Mitra, N., Ovsjanikov, M., Guerrero, P.: Differentiable surface triangulation. In: SIGGRAPH Asia (2021) 4
55. Riegler, G., Osman Ulusoy, A., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: CVPR (2017) 3
56. Schölkopf, B., Giesen, J., Spalinger, S.: Kernel methods for implicit surface modeling. In: NeurIPS (2004) 4
57. Sitzmann, V., Chan, E.R., Tucker, R., Snavely, N., Wetzstein, G.: Metasdf: Meta-learning signed distance functions. In: NeurIPS (2020) 3, 4, 8, 9, 11, 12, 13, 14
58. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: NeurIPS (2020) 2, 4, 5
59. Sitzmann, V., Rezchikov, S., Freeman, W.T., Tenenbaum, J.B., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. In: NeurIPS (2021) 4
60. Sitzmann, V., Zollhoefer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In: NeurIPS (2019) 4
61. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: NeurIPS (2017) 2, 7
62. Takikawa, T., Litalien, J., Yin, K., Kreis, K., Loop, C., Nowrouzezahrai, D., Jacobson, A., McGuire, M., Fidler, S.: Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In: CVPR (2021) 4
63. Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P.P., Barron, J.T., Ng, R.: Learned initializations for optimizing coordinate-based neural representations. In: CVPR (2021) 4
64. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. In: NeurIPS (2020) 2
65. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: ICCV (2017) 3
66. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Stoll, C., Theobalt, C.: Patchnets: Patch-based generalizable deep implicit 3d shape representations. In: ECCV (2020) 4, 5
67. Tulsiani, S., Su, H., Guibas, L.J., Efros, A.A., Malik, J.: Learning shape abstractions by assembling volumetric primitives. In: CVPR (2017) 3
68. Venkatesh, R., Karmali, T., Sharma, S., Ghosh, A., Babu, R.V., Jeni, L.A., Singh, M.: Deep implicit surface point prediction networks. In: CVPR (2021) 14

69. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: NeurIPS (2016) 2, 7
70. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: ECCV (2018) 3
71. Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X.: O-cnn: Octree-based convolutional neural networks for 3d shape analysis. TOG (2017) 3
72. Williams, F., Gojcic, Z., Khamis, S., Zorin, D., Bruna, J., Fidler, S., Litany, O.: Neural fields as learnable kernels for 3d reconstruction. In: CVPR (2022) 11
73. Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., Panozzo, D.: Deep geometric prior for surface reconstruction. In: CVPR (2019) 3
74. Williams, F., Trager, M., Bruna, J., Zorin, D.: Neural splines: Fitting 3d surfaces with infinitely-wide neural networks. In: CVPR (2021) 5
75. Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: NeurIPS (2016) 3
76. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: CVPR (2015) 3
77. Xu, Q., Wang, W., Ceylan, D., Mech, R., Neumann, U.: Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In: NeurIPS (2019) 10
78. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. In: NeurIPS (2020) 3, 4
79. Yavartanoo, M., Chung, J., Neshatavar, R., Lee, K.M.: 3dias: 3d shape reconstruction with implicit algebraic surfaces. In: ICCV (2021) 4
80. Zheng, Z., Yu, T., Dai, Q., Liu, Y.: Deep implicit templates for 3d shape representation. In: CVPR (2021) 4
81. Zou, C., Yumer, E., Yang, J., Ceylan, D., Hoiem, D.: 3d-prnn: Generating shape primitives with recurrent neural networks. In: CVPR (2017) 3