ARO-Net: Learning Implicit Fields from Anchored Radial Observations

Yizhi Wang\textsuperscript{1,2}\textsuperscript{*}, Zeyu Huang\textsuperscript{1}, Ariel Shamir\textsuperscript{3}, Hui Huang\textsuperscript{1}, Hao Zhang\textsuperscript{2}, Ruizhen Hu\textsuperscript{1}\textsuperscript{†}

\textsuperscript{1}Shenzhen University \textsuperscript{2}Simon Fraser University \textsuperscript{3}Reichman University

Abstract

We introduce anchored radial observations (ARO), a novel shape encoding for learning implicit field representation of 3D shapes that is category-agnostic and generalizable amid significant shape variations. The main idea behind our work is to reason about shapes through partial observations from a set of viewpoints, called anchors. We develop a general and unified shape representation by employing a fixed set of anchors, via Fibonacci sampling, and designing a coordinate-based deep neural network to predict the occupancy value of a query point in space. Differently from prior neural implicit models that use global shape feature, our shape encoder operates on contextual, query-specific features. To predict point occupancy, locally observed shape information from the perspective of the anchors surrounding the input query point are encoded and aggregated through an attention module, before implicit decoding is performed. We demonstrate the quality and generality of our network, coined ARO-Net, on surface reconstruction from sparse point clouds, with tests on novel and unseen object categories, “one-shape” training, and comparisons to state-of-the-art neural and classical methods for reconstruction and tessellation.

1. Introduction

Despite the substantial progress made in deep learning in recent years, transferability and generalizability issues still persist due to domain shifts and the need to handle diverse, out-of-distribution test cases. For 3D shape representation learning, a reoccurring challenge has been the inability of trained neural models to generalize to unseen object categories and diverse shape structures, as these models often overfit to or “memorize” the training data.

In this paper, we introduce a novel shape encoding for learning an implicit field [34] representation of 3D shapes that is category-agnostic and generalizable amid significant shape variations. In Figure 1 (top), we show 3D reconstructions from sparse point clouds obtained by our approach that is trained on chairs but tested on airplanes, rifles, and animals. Our model can even reconstruct a variety of shapes when the training data consists of only one shape, augmented with rotation and scaling; see Figure 1 (bottom).

The main idea behind our work is to reason about shapes from partial observations at a set of viewpoints, called anchors, and apply this reasoning to learn implicit fields. We develop a general and unified shape representation by designating a fixed set of anchors and designing a coordinate-based neural network to predict the occupancy at a query point. In contrast to classical neural implicit models such as IM-Net [8], OccNet [23], and DeepSDF [24], which learn global shape features for occupancy/distance prediction, our novel encoding scheme operates on local shape features obtained by viewing the query point from the anchors.

Specifically, for a given query point \( x \), we collect locally observable shape information surrounding \( x \), as well as directional and distance information toward \( x \), from the perspective of the set of anchors, and encode such information using...
while the PointNet, attention module, and implicit decoder are fixed during inference – their weights were determined during training.

**Radial Observations**

The results from OccNet [23] and ConvONet show more signs of overfitting to the training sphere than ARO-Net (d).

In our current implementation of ARO-Net, we adopt Fibonacci sampling [19] to obtain the fixed set of anchors. We demonstrate the quality and generalizability of our method on surface reconstruction from sparse point clouds, with testing on novel and unseen object categories, as well as “one-shape” training (see bottom of Figure 1). We report extensive quantitative and qualitative comparison results to state-of-the-art methods including both neural models for reconstruction [8, 11, 23, 25, 26] and tessellation [7], as well as classical schemes such as screen Poisson reconstruction [18]. Finally, we conduct ablation studies to evaluate our choices for the number of anchors, the selection strategies, and the decoder architecture: MLP vs. attention modules.

**2. Related work**

Learning neural fields is an emerging and one of the most intensely studied subjects in the vision and machine learning communities in recent years, while surface reconstruction is one of the most classical problems in computer vision and graphics. In this section, we shall only focus on the most closely related prior works. For a more comprehensive coverage on the relevant topics, we refer the readers to the surveys on surface reconstruction from point clouds [1] and neural fields [34], as well as the excellent summary on neural implicit representations by Vincent Sitzmann.
Surface reconstruction and local priors. Classical reconstruction schemes such as Screened Poisson (SPR) [18] often produce artifacts amid noise and under sampling. One remedy is to rely on local shape priors, which are reoccurring patch templates that can be matched to help reconstruction over sparse input. Gal et al. [12] develop such an example-based surface reconstruction method, relying on a pre-built set of enriched patches sampled from a database of 3D models that include surface normals and point feature classification (e.g., edge or corner) to facilitate reconstruction. These patches essentially serve as a training set.

Taking local templates to the extreme gives us classical tessellation methods such as Marching Cubes [22] and Dual Contouring [16], whose “neural” versions have been developed recently as neural Marching Cubes [9] and neural Dual Contouring (NDC) [7]. In particular, the unsigned version of NDC, or UNDC, can achieve state-of-the-art results on reconstruction from un-oriented point clouds with sufficient sampling. Indeed, what is common about all of these purely local reconstruction schemes is their ability to generalize across object categories, while disregarding any contextual shape information and global reconstruction priors.

Neural implicit models. Since OccNet [23], IM-Net [8], and DeepSDF [24] concurrently introduced the coordinate-based implicit neural representation in 2019, research and development on neural implicit fields [34] have flourished. In these early works, the shape encoder [8, 23] and latent code [24] employed for occupancy/distance predictions only model global shape features. The resulting learned representations were unable to reconstruct fine geometric details.

The next wave of developments has focused on conditioning implicit neural representations on local features stored in voxel [3, 10, 15, 25] or image grids [21, 28, 35] to more effectively recover geometric or topological details and to scale to scene reconstruction, or at the patch level [13, 32] to improve generalizability across object categories.

Some of these locally conditioned neural implicit models essentially learn local shape priors [12] from training shapes, where the learned priors are parameterized in different ways, e.g., voxel-level latent codes in Deep Local Shape [3], embeddings of ShapeNet [5] parts in local implicit grid representations [15], structured Gaussians in Local Deep Implicit Functions (LDIF) [13], and canonical patches in PatchNets [32]. Other works [10, 11, 25], like ARO-Net, also utilize query-specific shape features.

Implicit field from query-specific features. For occupancy prediction at a query point \( p \), ConvONet [25] extracts convolutional features at \( p \) by interpolating over a feature field defined for an input shape. IF-Net [10] also extracts a feature at \( p \) but its associated feature grid encodes multi-scale features representing both global and local shape information. Given an input point cloud, Points2Surf [11] samples a local point patch close to the query \( p \), as well as a global point sub-sampling emanating from \( p \), and encodes them into two features vectors for implicit decoding. More recently, AIRNet [14], POCO [2], 3DLIG [36], and SA-ConvONet [31], improve over ConvONet with irregular grids, attention mechanisms, and sign-agnostic optimizations.

Compared to all these query-specific encodings, ARO captures shape information around the query \( p \) in a more contextual and structure-aware manner. In particular, while Points2Surf provides a “uni-directional” observation from \( p \), the local observations by ARO are radially distributed around \( p \), providing a more global context. In addition, the inclusion of distance and directional information in ARO reveals a direction connection to occupancy prediction, as explained in Section 3.1. Owing to these distinctive properties of ARO, ARO-Net consistently produces higher-quality and more faithful reconstructions, as demonstrated in Section 4.

Basis point set. Our view-based ARO encoding is related to basis point set (BPS) [26] for point cloud encoding. BPS is a fixed-length feature vector computed as the minimal distances from a fixed set of anchors to a given point cloud. BPS can serve as a global shape feature, e.g., in place of ARO, for neural implicit reconstruction from point clouds.

3. Method

In this section, we describe and analyze our novel query-specific shape encoding, ARO, in detail. We first show in Section 3.1 that when a given set of anchors can provide a full coverage of a shape, i.e., it observe the entire shape surface, then point occupancies around the shape surface can be fully determined from the corresponding ARO representation. More generally however, the full coverage condition may not be satisfied or in the case of 3D reconstruction from a noisy and sparse point cloud, the partial observations at the anchors may be inaccurate. In such cases, we train a neural network to predict the point occupancies. This network learns a mapping from the anchored observations to the occupancy value at any spatial query point. We present this network, ARO-Net, in Section 3.2.

3.1. Anchor visibility and point occupancy

Given an anchor point \( a \), we can cast rays from \( a \) in all directions to obtain its visible region relative to a given watertight surface \( S \). A ray that does not intersect \( S \) can be clipped by the bounding box of the \( S \). In previous works, such visible regions have been adopted for points inside the shape to assist in shape partition, i.e., the Shape Diameter Function [30], as well as for points outside the shape for reconstruction [29], or to guide relationship optimization between two shapes [37]. These works have shown that the visible regions can not only characterize local shape regions, but also provide a global and contextual view of the shape.
structure. Figure 4 (a) shows the visible regions of one anchor inside the shape (blue) and the other anchor outside the shape (green).

Our key observation to guide the design of our ARO representation is that if we have a set of well-distributed anchors, then combining their visible regions together can fully characterize the entire shape. We consider a set of anchors to be well-distributed if any surface point of the shape can be viewed by at least one anchor. In the following, we discuss different cases of well-distributed anchors and how they define the shape; see illustrations in Figure 4(b-d).

When the set of well-distributed anchors is all inside the shape as shown in Figure 4(b), then the union of all the visible regions of the anchors is identical to the interior region of the shape. Thus, a query point, the red star in Figure 4(b), is inside the shape if and only if it is covered by at least one radial observation, which we call a covering anchor, i.e., the red point in Figure 4(b). This is true if the distance between the query point and the covering anchor is smaller than the radial distance of this anchor along the ray direction pointing to the query point.

When the set of well-distributed anchors is all outside the shape as shown in Figure 4(c), the conclusion is similar for the points outside the shape. Thus, the union of all the visible regions of the anchors is identical to the region exterior to the shape. Therefore, a query point, i.e., the red star in Figure 4(c), is inside the shape if and only if it is not covered by any of the visible regions. That is, its distance to each anchor is always larger than the radial distance of this anchor along the ray direction pointing to the query point. Thus, all the anchors together determine the occupancy of the query point.

When the anchor locations are mixed, with some inside and some outside the shape as shown in Figure 4(d), the uncovered spatial regions, $R_1$ and $R_2$, could be either inside or outside the shape. If the region is bounded by the shape boundary and the visible region boundaries for the anchors, such as $R_1$, then the entire region is inside the shape. Locally, this is similar to cases shown in (c) with those red anchors fully covering its boundary. As for $R_2$, the region is partially bounded by the virtual bounding box, i.e., for the image/volume capture, then it is totally outside the shape.

To summarize, if we have a set of anchors that together observe the entire surface of the shape, then their visible regions together can fully determine the inside and outside regions of the shape. Moreover, one important property is that it converts the inside/outside judgment of a spatial point relative to the shape, to the point’s relationship to the radial observations of the anchors, which leads to our definition of query-specific ARO. This property allows to design a network that predicts the occupancy of a spatial point based on local features captured by ARO when the anchors are not well-distributed and the shape is only partially observed.

### 3.2. ARO-Net for occupancy prediction

Given a point cloud $P$, our goal is to reconstruct the corresponding watertight surface $S$ based on our ARO representation for point clouds. We choose to use an implicit representation of the surface, which is essentially a function determining whether a given query point $x$ in space is inside or outside the surface $S$. Unlike previous methods that encode the entire point cloud as a single global latent code [8, 23], we utilize the set of anchors to identify local surface regions that are essential to classify occupancy. This allows us to implement a category-agnostic method to reconstruct any shape. Figure 2 shows how query-specific ARO is extracted and utilized by ARO-Net, which outputs the
occupancy for any given query point in space.

The input of our method consists of a point cloud \( P = \{p_1, \ldots, p_n\} \) and an anchor set \( A = \{a_1, \ldots, a_m\} \) as shown in Figure 2 (a), where the point cloud are drawn in gray and anchors are drawn with different colors. Given a query point \( x \) shown as the red dot in Figure 2 (b), for each anchor \( a_i \), we can compute the relative position of \( x \) to \( a_i \) as \( r_i = x - a_i \). Next, we create a cone with apex at \( a_i \) and \( r_i \) as the axis to get the nearest \( k \) points \( O_i = \{p_i^j\}_{j=1, \ldots, k} \) from \( P \), which is defined as the radial observation of \( a_i \) relative to the query point \( x \) and referred as ARO. Each ARO \( O_i \) is then passed to the PointNet encoder to get the corresponding embedding \( f_i \). The relative location \( r_i \) and its norm \( ||r_i|| \) are further concatenated to get a local feature representative of query \( x \) relative to anchor \( a_i \). All the local features relative to all the anchors are then passed to an attention module and finally decoded into an occupancy value.

**Anchors placement.** A good set of anchor points \( A = \{a_1, a_2, \ldots, a_m\} \) must be located in different positions in the space bounding the shape, so that they will observe more of the shape’s surface with high probability. Hence, to build our set, we sample a set of anchors on three spheres with different radii around the shape. In more detail, assuming that the input point cloud is normalized inside a unit sphere, we first uniformly sample \( m \) points on the surface of the unit sphere with radius \( r = 1/2 \) using Fibonacci Sampling [19], then for the points with index \( i \% 3 = 1 \) we move the point towards the center to position them on the sphere with radius \( r = 1/4 \), and for the points with index \( i \% 3 = 2 \) we further move the point to the sphere with radius \( r = 1/8 \), as shown in Figure 5. This creates a set of anchors distributed in space that can well observe any shape within the sphere.

**4. Results and evaluation**

**4.1. Experiment setting**

**Implementation details.** We employ a binary cross entropy loss for predicted occupancies and an Adam optimizer whose initial learning rate is \( 3e-4 \), decaying by 0.5 every 100 epochs. The conical angle is set to be 24\(^\circ\) and \( k = 16 \) in all our experiments. The attention module is a Transformer [33] encoder consisting of 6 encoding layers and 8 attention heads. The implicit decoder is a fully connected layer which outputs the occupancy value. We use the sample PointNet architecture as Occ-Net [23]. ARO-Net receives 2,048 points as input when training, and can handle arbitrary point numbers when testing. We tested 3 sets of point numbers \( n = \{512, 1024, 2048\} \).

**Dataset.** Our experiments are conducted on the “Big CAD dataset”, ABC [20], and ShapeNet V1 [4]. For ABC, we use the same train/test split as NDC [7], with 4,011 shapes for training and 982 shapes for testing. For ShapeNet V1, we train ARO-Net using 4K chairs and test 100 shapes per category. Note that ABC is an especially challenging dataset for 3D representation learning since there is no clear notion of object category (no category labels at all), while the shapes exhibit significant geometry and topology variations.

**Evaluation metrics.** We adopt Chamfer distance (CD), Hausdorff distance (HD), earth mover’s distance (EMD), occupancy intersection over union (IOU) proposed in [23], and light filed distance (LFD) [6] as the evaluation metrics for reconstructed meshes. Note that since the IOU computation method provided by [23] requires watertight shapes, we report IOU only on the ABC dataset.

**4.2. Comparisons to state-of-the-art methods**

**Reconstruction on ABC.** Quantitative comparisons to state-of-the-art neural 3D reconstruction models on ABC are shown in Table. 1. We can see that those methods using only global shape features, i.e., Occ-Net [23], IM-Net [8], and BPS [26], are consistently outperformed by UNDC, which learns local tessellation, and ConvONet, Points2Surf, and ARO-Net, which all encode local and query-specific features. Among the last four methods, ARO-Net exhibits a clear advantage in Hausdorff distance and OCC-IOU, and produces the overall best performance by ranking the first in three out of five metrics (IOU, EMD and HD) and rank-
ing the second in one metric (LFD). For the remaining CD metric, ARO-Net is still a close runner-up.

Some visual comparisons are shown in Figure 6. As shape variations in ABC is large, Occ-Net [23] and IM-Net [8] cannot faithfully reconstruct global shape structures. With the anchored global features, BPS [26] better captures the overall shapes but fails to reconstruct local details. ConvONet [25] yields smoother local geometry but is unable to reconstruct sharp features. Points2Surf [11] tends to produce random noises near reconstructed surfaces. While showing high performance in quantitative evaluations, visual results from UNDC [7] often contain holes with sparse point cloud inputs. Compared to all those methods, ARO-Net is able to reconstruct both global structure and local detail well, including sharp features as UNDC, and its results exhibit the least amount of artifacts in terms of holes and noise.

ARO-Net also exhibits superior robustness to sparsity of input point clouds. Once trained with 2,048-point input, ARO-Net can produce quality results with sparser inputs without re-training. As shown in Figure 7, when the number of points decreases from 2,048 to 512, the reconstruction quality of both UNDC [7] and Points2Surf [11] degrades dramatically, as also reflected in the CD and IOU metrics. Specifically, Points2Surf [11] tends to generate more holes and UNDC [7] produces rougher, oscillatory surfaces. ConvONet [25] is relatively stable but holes also appear with 512-point inputs. By contrast, ARO-Net yields the highest level of robustness both visually and quantitatively.

**Reconstruction on ShapeNet V1.** For our experiments on the ShapeNet V1 dataset, we train ARO-Net on 4K chairs and test it on other object categories: airplanes and rifle in ShapeNet V1 and animals in PSB [17]. Quantitative comparisons are shown in Table 2, with some visual results in Figure 8. It is clear that implicit models employing global features do not generalize well. Compared to UNDC [7], ConvONet [25], and Points2Surf [11], ARO-Net performs the best overall, quantitatively, especially with the test on airplanes, showing its strong generalizability. Visually, ARO-Net produces the most complete, artifact-free results with faithful reconstruction of fine details. In the airplane results in second row, it is worth observing that ARO-Net managed to reconstruct the left-side turbine from extremely sparse points, while other methods all completely missed it.

**One-shape training.** As a stress test, we train the networks using only one 3D shape, Fertility (see Figure 1), with rotation and scaling. From the comparisons in Figure 9, we see that SPR suffers from noise and under-sampling (e.g., chair and human hand), as expected. OccNet clearly over-fits to the training shape. ConvONet generalizes better but produces overly smoothed results. In contrast, ARO-Net re-
constructs both global structures and fine details better than the alternatives, with consistence over the shape variety.

4.3. Ablation study

Anchor visibility vs. point occupancy. To verify whether ARO-Net can learn the close correlation between anchor visibility and point occupancy with the help of a visualization, we implemented a 2D vision of ARO-Net with implementation details provided in the supplementary material. We train ARO-Net using the letter “G” with anchors distributed as shown in Figure 4(c). Figure 10 visualizes the activation of each anchor by masking out the other anchors’ radial observations. The output with all seven anchors considered are shown in the top-left corner. The probability of each point being inside the shape is drawn using the blue to yellow colormap, and the anchors are shown in white dots. We can see that the decision boundary of each anchor closely follows its visible region, without having that information specifically passed to ARO-Net. Another noteworthy observation is that for regions invisible to the anchor, points closer to the visibility boundary are assigned higher probability of

Table 2. Quantitative comparisons to state-of-the-art methods on the ShapeNet V1 dataset. Number in bold means ranking the first and the one with underline means ranking the second in each column.

<table>
<thead>
<tr>
<th>Trained on chairs, tested on chairs</th>
<th>Method</th>
<th>LFD↓</th>
<th>HD↓</th>
<th>CD↓</th>
<th>EMD↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>IM-Net [8]</td>
<td>3.18</td>
<td>9.04</td>
<td>13.84</td>
<td>15.40</td>
<td></td>
</tr>
<tr>
<td>OccNet [23]</td>
<td>2.59</td>
<td>7.77</td>
<td>12.36</td>
<td>13.47</td>
<td></td>
</tr>
<tr>
<td>BPS [26]</td>
<td>4.31</td>
<td>12.28</td>
<td>20.51</td>
<td>23.30</td>
<td></td>
</tr>
<tr>
<td>ConvONet [25]</td>
<td>2.12</td>
<td>6.22</td>
<td>11.20</td>
<td>12.30</td>
<td></td>
</tr>
<tr>
<td>Points2Surf [11]</td>
<td>2.51</td>
<td>6.46</td>
<td>8.60</td>
<td>7.34</td>
<td></td>
</tr>
<tr>
<td>UNDC [7]</td>
<td>2.19</td>
<td>5.60</td>
<td><strong>6.06</strong></td>
<td><strong>6.80</strong></td>
<td></td>
</tr>
<tr>
<td>ARO-Net</td>
<td><strong>1.92</strong></td>
<td><strong>5.33</strong></td>
<td>7.14</td>
<td>7.70</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trained on chairs, tested on airplanes</th>
<th>Method</th>
<th>LFD↓</th>
<th>HD↓</th>
<th>CD↓</th>
<th>EMD↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>IM-Net [8]</td>
<td>14.30</td>
<td>19.50</td>
<td>44.84</td>
<td>41.38</td>
<td></td>
</tr>
<tr>
<td>OccNet [23]</td>
<td>12.31</td>
<td>18.86</td>
<td>39.75</td>
<td>38.57</td>
<td></td>
</tr>
<tr>
<td>BPS [26]</td>
<td>13.73</td>
<td>19.48</td>
<td>38.10</td>
<td>36.61</td>
<td></td>
</tr>
<tr>
<td>ConvONet [25]</td>
<td>5.69</td>
<td>15.98</td>
<td>13.50</td>
<td>10.75</td>
<td></td>
</tr>
<tr>
<td>Points2Surf [11]</td>
<td>5.48</td>
<td>4.70</td>
<td>4.86</td>
<td>5.76</td>
<td></td>
</tr>
<tr>
<td>UNDC [7]</td>
<td>3.62</td>
<td>4.40</td>
<td><strong>3.98</strong></td>
<td><strong>3.98</strong></td>
<td></td>
</tr>
<tr>
<td>ARO-Net</td>
<td><strong>3.56</strong></td>
<td><strong>4.32</strong></td>
<td>4.61</td>
<td>4.73</td>
<td></td>
</tr>
</tbody>
</table>

Figure 8. Visual comparisons on ShapeNet V1 and PSB datasets to show reconstruction quality and generalizability of different methods. All methods (except SPR) were trained on chairs in ShapeNet and tested on chairs, airplanes, and rifles in ShapeNet V1 and animals in PSB. Zoom-ins highlight contrasts in details and artifacts in the reconstructions. More results can be found in supplementary material.

Figure 9. Visual comparison on the results obtained by training only on the “fertility” model with rotation and scaling.
being inside the shape. This offers evidence that ARO-Net does implicitly learn to predict the point occupancy based on anchor visibility.

**Anchor placement strategy.** Other than our layered Fibonacci sampling, we also tried two different sampling strategies: uniform and grid-based sampling. In uniform sampling, we randomly sample \( m \) points in a unit sphere as anchors. In grid-based sampling, we randomly sample \( m \) grid points of a unit cube as anchors. We found that our layered Fibonacci sampling achieves the best performance. More detailed comparison can be found in the supplementary material.

**Anchor count.** There is a trade-off between reconstruction performance and computational cost when choosing, \( m \), the number of anchors. We tested \( m = \{24, 48, 96\} \) and report results in the top half of Table 3. We found that the performance gain by \( m = 96 \) is marginal, while the computation cost doubles. With \( m = 24 \), ARO’s performance drops significantly due to lack of sufficient shape coverage.

**Decoder architecture.** The last two rows of Table 3 compare the current ARO-Net with a version by replacing our attention module and implicit decoder with the much simpler MLPs by IM-Net [8]. The similar performance numbers indicate that the superior results by ARO-Net are predominantly owing to ARO rather than the decoder architecture.

**Fixed vs. adaptive anchors.** When computing the radial observations, we perform top-\( k \) selection (pooling) on the cosine and Euclidean distances between the anchors and input points. Therefore, our network loss function is differentiable with respect to anchor coordinates: the gradients are passed through the indices that form the top \( k \) values, and then passed to anchor coordinates. While this does offer the possibility of fine-tuning the anchor positions during training and testing, we have found that optimizing anchor positions along with the network parameters can make the training process unstable and lead to worse results. Overall, we believe that with the anchors fixed, it is easier for the network to learn the mapping from ARO to point occupancy.

During inference, it is possible to adjust the anchor locations to obtain a larger coverage of the shape surface to improve reconstruction. However, we found the improvements to be marginal; see supplementary material.

**Distance and direction information in ARO.** At last, we have confirmation that the \( r \) and \( ||r|| \) play an important role as input to ARO-net. The reconstruction performance by the network drops significantly when we remove such information, with only the PointNet features. Again, more details can be found in the supplementary material.

## 5. Conclusion, limitation, and future work

We introduce a novel neural implicit representation for 3D shapes that is conditioned on Anchored Radial Observations (ARO). We use ARO to train our reconstruction network, ARO-Net, that outputs an occupancy at any query point in space. ARO is query-specific, defined by a set of local descriptors from the perspective of a set of fixed anchors to provide a global context. ARO-Net is category-agnostic and provides superior reconstruction results to alternative methods. It also has strong generalization capabilities and can even be used to learn from a single shape.

Some current limitations and future works are as follows. The quality and efficiency of the resulting representation are dependant on the number and placement of the anchors. We tested a number of configurations and settled on a fixed set of 48 anchors via Fibonacci sampling. However, further research into anchor selection, even learning, is needed. In addition, the ARO-Net representation still requires storing the original input point set and therefore is less memory efficient than pure network-based representations. Lastly, both training and inference of ARO-Net are slower than prior implicit models based on simpler and less contextual shape encodings (e.g. ConvONet). This can be alleviated by adaptive sampling of the anchors and using more efficient spatial search data structures to find closest points.
References


