Ray-ONet: Efficient 3D Reconstruction From A Single RGB Image

Wenjing Bian
wenjing@robots.ox.ac.uk

Zirui Wang
ryan@robots.ox.ac.uk

Kejie Li
kejie.li@eng.ox.ac.uk

Victor Adrian Prisacariu
victor@robots.ox.ac.uk
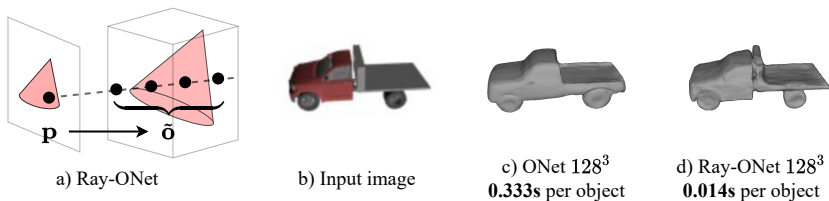
Active Vision Lab
University of Oxford
Oxford, UK

| a) Ray-ONet | b) Input image | c) ONet 128^3 **0.333s** per object | d) Ray-ONet 128^3 **0.014s** per object |

Figure 1: Ray-ONet recovers more details than ONet [15] while being $20\times$ faster, by predicting a series of occupancies $\tilde{\mathbf{o}}$ along the ray that is back-projected from a 2D point $\mathbf{p}$.

Abstract

We propose Ray-ONet to reconstruct detailed 3D models from monocular images efficiently. By predicting a series of occupancy probabilities along a ray that is back-projected from a pixel in the camera coordinate, our method Ray-ONet improves the reconstruction accuracy in comparison with Occupancy Networks (ONet), while reducing the network inference complexity to $O(N^2)$. As a result, Ray-ONet achieves state-of-the-art performance on the ShapeNet benchmark with more than $20\times$ speed-up at 128^3 resolution and maintains a similar memory footprint during inference. The project website is available at https://rayonet.active.vision.

1 Introduction

3D reconstruction from a single view is an inherently ill-posed problem. However, it is trivial for a human to hallucinate a full 3D shape of a chair from a single image, since we have seen many different types of chairs from many angles in our daily life. Therefore, recovering a plausible 3D geometry from a single view is possible when enough prior knowledge is available, and in certain situations, might be the only feasible method when multi-view constraints are inaccessible, for instance, creating a 3D chair model from a chair image on a random website for an Augmented Reality (AR) application.

Recently, the single-view object reconstruction field has been advanced significantly using deep learning. In particular, deep implicit representations, such as DeepSDF [16], Occupancy Networks [15] and their extensions [17, 21], demonstrate promising results on the single-view 3D reconstruction task, by representing 3D shapes via a continuous function that is embedded in network parameters. Although these methods enjoy the advantages of representing 3D models in any resolutions and require small memory footprint and low storage space, the inference speed is still an issue yet to address, as most existing methods require querying millions of 3D locations to recover one 3D shape, leading to an $O(N^3)$ complexity when a simple 3D sampling strategy is employed.

In this paper, we propose to predict a series of occupancies over 3D positions along a ray back-projected from a query 2D location on an image plane given local and global image features jointly. This formulation – viewer-centred ray-based occupancy prediction, taking advantage of local and global image features – improves the reconstruction accuracy while reducing the complexity to $O(N^2)$.

One issue of this viewer-centred reconstruction is the depth/scale ambiguity presented by a single view (*i.e.*, the same image can be rendered by a small object that is close to the camera or a large object that is at a distance). A common practice is to normalise all ground-truth shapes into a constant scale during training. However, this operation implicitly entangles the shape learning with the scale ambiguity. Instead, we show how one can decouple these two problems and improve the reconstruction accuracy by providing the network with a scale calibration factor, which consists of a camera focal length and a camera-object distance.

In summary, we introduce Ray-ONet with three contributions: First, our method predicts a series of occupancies over 3D positions along a ray from a 2D query point, by conditioning an occupancy prediction function to both global and local image context, reducing the complexity to $O(N^2)$. Second, we introduce a scale calibration factor, to compensate for the issue caused by different object scales. Third, we demonstrate on ShapeNet that our method improves the overall 3D reconstruction accuracy while being 20× faster than ONet.

2   Related Work

Shape Representation. Computation efficiency and reconstruction quality are both effected by the choice of representation for the model output. Researchers resort to volumetric representations [4, 22] when deep networks are used for shape generation because 3D voxel grids naturally fit into the convolutional operation. However, the resolution of voxels is often limited by memory, which grows exponentially for high-resolution outputs. Alternatively, point clouds have been adopted [6], but they cannot represent surfaces directly. Meshes [25], being compact and surface aware, assume a pre-defined topology.

In recent years, implicit functions [3, 15, 16] are widely used for representing continuous 3D shapes. Networks using these implicit representations are trained such that the decision boundary represents the iso-surface of an object shape. These models are able to reconstruct high-resolution 3D shapes with low memory cost, but the inference time increases cubically with sampling resolution. Hybrid models [19, 20] which combine two representations have also been developed recently. They couple explicit and implicit representations with 2 branches, where the explicit branch accelerates surface reconstruction while the implicit branch reconstructs fine details of object shapes, but the network size is inevitably increased dramatically. In contrast, Ray-ONet is able to speed up inference without increasing the

memory footprint by predicting occupancy probability along a ray in a single network forward.

Single-View 3D Reconstruction. Reconstructing from a single view is a severely ill-posed problem, which was traditionally solved by "shape-from-X" techniques [11, 30] (X could be silhouette, shading etc.) using geometric or photometric information. The resurgence of deep learning in recognition tasks has encouraged researchers to solve single-view 3D reconstruction in a data-driven manner. PSGN [6] and 3D-R2N2 [4] pioneer using a deep network to learn a mapping from an RGB image to an object shape represented by voxels or point clouds. Following works are able to achieve better performance by using more sophisticated representations (e.g. Octree [23], dense point clouds [13], or implicit representations [15]) or building shape prior [26].

More recently, CoReNet [18] predicts translation equivariant voxel grid using a variable grid offset. It enables fast mesh extraction but still requires significantly more memory than implicit models. The closest to our work is SeqXY2SeqZ [8], which also tries to reduce the time complexity of inference in implicit representations. To this end, they design a hybrid representation, which interpret a 3D voxel grid as a set of tubes along a certain axis. However, its reconstruction resolution is fixed in all 3 dimensions, because of the built-in location embedding matrices.

Our work is also a hybrid approach, which explicitly predicts occupancy probabilities along a certain ray that passes through a point on the input image. Unlike Han et al. [8], we predict occupancies along a ray in a single network forward, which results in further reduction in time complexity. In addition, the coordinate system of Ray-ONet is aligned with the input image to take advantage of local features. As a result, we can predict 3D shapes with fine details.

Coordinate System. Two types of coordinate systems are used in single-view object reconstruction: i) viewer-centred aligned to the input image or; ii) object-centred aligned to a canonical position in the shape database. As discussed in [22, 24], an object-centred coordinate system with aligned objects of the same category encourages learning category information, which provides a robust prior for learnt categories, whereas predictions made in the view space rely more on geometric information from the input image, and can generalise better to unseen categories. Furthermore, recent works [18] using the viewer-centred coordinate system demonstrate reconstruction with fine details by using image local feature. However, they often assume 3D shapes are normalised to a constant scale. Instead, we introduce a scale calibration factor to facilitate training given GT shapes in different scales.

3 Preliminary

Occupancy network (ONet) [15] formulates shape reconstruction as a continuous function mapping problem, where the goal is to learn a function $\mathcal{F}$ that maps a 3D position $\mathbf{x} \in \mathbb{R}^3$ to an occupancy probability $\tilde{o}$, where $\{\tilde{o} \in \mathbb{R} : 0 < \tilde{o} < 1\}$. The object shape is implicitly represented by the decision boundary of the learned function $\mathcal{F}$. To capture shape variations, the occupancy function $\mathcal{F}$ above is often conditioned on observation, for example, an input image. Mathematically, the occupancy prediction function $\mathcal{F}$ can be defined as:

$$\mathcal{F}(\mathbf{x}, \mathbf{z}) = \tilde{o}, \qquad (1)$$

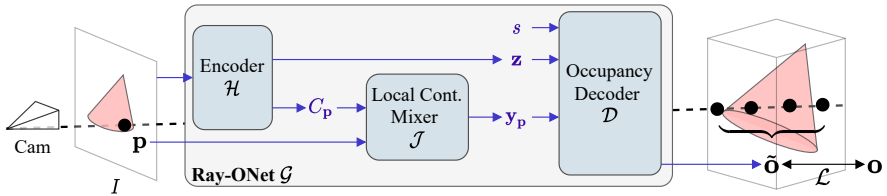where $\mathbf{z}$ is a latent code that encodes an input image I.

Figure 2: **Ray-ONet Overview.** Given an RGB image, Ray-ONet first maps the image to a global feature descriptor $\mathbf{z}$ and a dense feature map C via an image encoder $\mathcal{H}$. The local feature $C_\mathbf{p}$ (feature at position $\mathbf{p}$) and the position $\mathbf{p}$, taken by the local context mixer $\mathcal{J}$, are mapped to a local context descriptor $\mathbf{y_p}$. Finally, the occupancy decoder $\mathcal{D}$ takes as input $\mathbf{y_p}$, $\mathbf{z}$ and a scale factor s and predicts occupancies of 3D points along the back-projected ray from position $\mathbf{p}$.

This mapping function, however, suffers from slow inference speed because the network needs to query a large number of 3D positions to recover a high-resolution 3D reconstruction, which leads to $O(N^3)$ complexity if the 3D positions are sampled in a regular 3D grid. Although more sophisticated sampling strategies such as Multi-resolution IsoSurface Extraction (MISE) can be applied as in ONet [15], the inference speed is still largely restrained by this query-every-3D-point strategy. This motivates us to develop a novel ray-based pipeline that reduces the complexity to $O(N^2)$.

4 Method

Given an input image I, we aim to build a 3D reconstruction in the camera space of I, instead of in the canonical space as in ONet, by predicting series of occupancies along rays that are back-projected for all image pixels.

In other words, for each ray points to a 2D position $\mathbf{p}$ on the image plane, we predict a series of occupancies along the ray behind the 2D position in one forward pass, hence reducing the complexity to $O(N^2)$. As a ray direction can be uniquely defined by the 2D position $\mathbf{p} = (p_x, p_y)$ in the viewer space, our Ray-ONet, which conditions the occupancy prediction on the ray direction, can be considered as conditioned on the 2D position $\mathbf{p}$. Fig. 2 presents an overview of our method.

In the following sections, we first formalise our occupancy prediction function in Section 4.1, followed by how we assemble it with image encoders and fusion modules (Section 4.2) and a detail description of the scale calibration factor that facilitates training given various scales of ground-truth object shapes (Section 4.3). Lastly, we describe our training and inference pipeline in Section 4.4 and Section 4.5, including the training loss and the mesh extracting process etc.

4.1 Occupancy Prediction

We now detail our occupancy prediction function. In Ray-ONet, we propose a function $\mathcal{G}$:

$$\mathcal{G}(\overbrace{\mathbf{p}, C_\mathbf{p}}^{\text{Local}}, \overbrace{\mathbf{z}, s}^{\text{Global}}) = \tilde{\mathbf{o}}, \tag{2}$$

which maps a 2D position $\mathbf{p}$ on an image I to a number of M occupancy estimations $\tilde{\mathbf{o}} \in \mathbb{R}^M$, conditioning on a global image feature $\mathbf{z}$, a local image feature $C_\mathbf{p}$ and a scale calibration

factor s. Technically, the occupancy prediction function $\mathcal{G}$ is parameterised by a Multi-Layer Perceptron (MLP), and image global features $\mathbf{z}$ and local features $C_{\mathbf{p}}$ are encoded by a shared CNN encoder. From now on, with a slight abuse of notation, we also refer $\mathcal{G}$ as our ray occupancy network, as the network is just a parameterisation of the occupancy prediction function.

Comparing with the $\mathcal{F}(\mathbf{x}, \mathbf{z})$ in the classical ONet in Eq. (1), which maps a 3D position $\mathbf{x}$ to an occupancy estimation $\tilde{o}$ while conditioning on a global image feature, Ray-ONet uses $\mathcal{G}(\mathbf{p}, C_{\mathbf{p}}, \mathbf{z}, s)$ to map a 2D position $\mathbf{p}$ to a series of occupancy estimations $\tilde{\mathbf{o}}$ by conditioning the occupancy function on both local and global context. Ray-ONet, therefore, can predict all M occupancies along a ray in a single network forward pass, reducing the complexity to $O(N^2)$. Moreover, as both our occupancy prediction function $\mathcal{G}$ and the classical $\mathcal{F}$ in ONet are parameterised in a similar MLP structure, our method maintains a similar memory footprint during inference.

4.2 Individual Modules

Given an input image I and its intrinsic and extrinsic camera parameters, we aim to extract relevant information for the occupancy prediction function $\mathcal{G}$. Specifically, we split the occupancy prediction function $\mathcal{G}$ to three modules: First, an image encoder $\mathcal{H}$, which extracts image global feature $\mathbf{z}$ and local feature $C_{\mathbf{p}}$; Second, a local context mixer $\mathcal{J}$, which fuses local image feature $C_{\mathbf{p}}$ with a ray that is defined by the 2D position $\mathbf{p}$ that it points to; and Third, an occupancy decoder $\mathcal{D}$, which decodes a series of occupancy estimations from both local and global context.

Image Encoder $\mathcal{H}$. We extract a global feature vector $\mathbf{z}$ and a feature map C from the input image I using a ResNet-fashion CNN [] $\mathcal{H} : I \rightarrow (\mathbf{z}, C)$. Specifically, the global latent code $\mathbf{z}$ is an output of a ResNet-18, and the feature map $C \in \mathbb{R}^{H \times W \times D}$ is a result of concatenating bilinear upsampled activations after the ResNet blocks, where H and W denote input image resolution, and D denotes the feature dimension.

Local Context Mixer $\mathcal{J}$. We parameterise the local context mixer module with an MLP $\mathcal{J} : (C_{\mathbf{p}}, \mathbf{p}) \rightarrow \mathbf{y}_{\mathbf{p}}$, which aggregates a 2D location $\mathbf{p}$ and an interpolated image feature $C_{\mathbf{p}}$ to a fused local context feature $\mathbf{y}_{\mathbf{p}}$.

Occupancy Decoder $\mathcal{D}$. For a local feature $\mathbf{y}_{\mathbf{p}}$ at point $\mathbf{p}$, we aim to predict a series of occupancies, conditioning on a global feature $\mathbf{z}$ and a scale calibration factor s, via a decoder $\mathcal{D} : (\mathbf{y}_{\mathbf{p}}, \mathbf{z}, s) \rightarrow \tilde{\mathbf{o}}$. Similar to ONet, we parameterise the decoder $\mathcal{D}$ in an MLP network with skip links and Conditional Batch Normalisation (CBN).

To summarise this section, we assemble the occupancy prediction network $\mathcal{G}$ with three modules: an image encoder $\mathcal{H}$, a local context mixer $\mathcal{J}$ and an occupancy decoder $\mathcal{D}$:

$$\mathcal{G}(\overbrace{\mathbf{p}, C_{\mathbf{p}}}^{\text{Local}}, \overbrace{\mathbf{z}, s}^{\text{Global}}) = \mathcal{D}(\mathcal{J}(\mathbf{p}, C_{\mathbf{p}}), \mathbf{z}, s) = \tilde{\mathbf{o}}, \tag{3}$$

where the global image feature $\mathbf{z}$ and the image feature map C are extracted the input image I from encoder $\mathcal{H}$.

4.3 Scale Calibration

The scale of the predictions is as important as the reconstructed shape because a correct shape prediction but in a different scale to the ground truth can result in a large training loss.

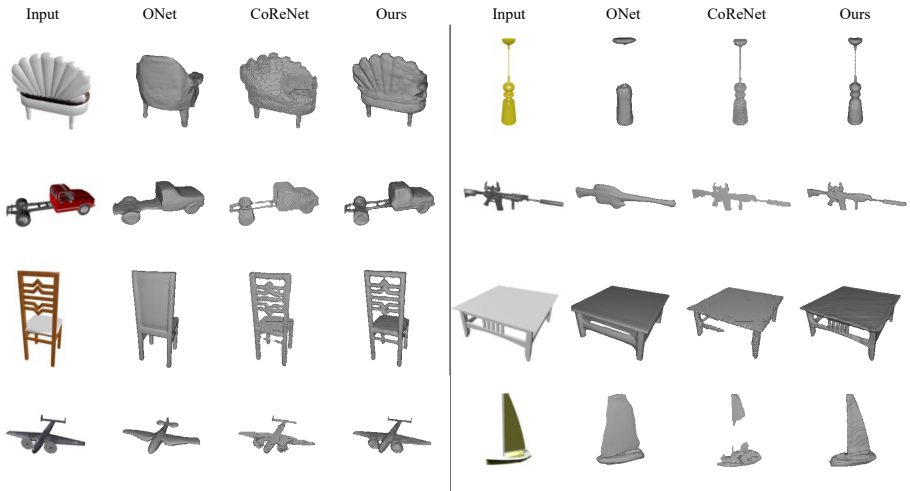| Input	ONet	CoReNet	Ours	Input	ONet	CoReNet	Ours

Figure 3: **Qualitative Reconstruction Results on ShapeNet.** Note that Ray-ONet is able to reconstruct fine details of object shapes, such as the complicated pattern on the chair back.

Instead of normalising all ground-truth shapes, we introduce the scale calibration factor s as an extra dimension for the global feature to facilitate network training given ground-truth shapes in different scales. Specifically, we define our scale calibration factor $s = \|\mathbf{c}\|_2/f$, where $\|\mathbf{c}\|_2$ denotes the distance between a camera and a ground truth 3D model, and f denotes the camera focal length.

This scale calibration factor s conveys that the predicted shape scale should be proportional to camera-object distance and inverse proportional to the camera focal length. In the case of using a fixed focal length for the entire training set, the scale calibration factor can be further reduced to $\|\mathbf{c}\|_2$. Note that this scale calibration factor is only needed during training.

4.4   Training

We use a binary cross-entropy loss $\mathcal{L}$ between predicted occupancies $\tilde{\mathbf{o}}$ and ground truth occupancies $\mathbf{o}$ to supervise the training of the encoder $\mathcal{H}$, the mixer $\mathcal{J}$ and the decoder $\mathcal{D}$:

$$\mathcal{L}(\tilde{\mathbf{o}}, \mathbf{o}) = -[\mathbf{o}\log(\tilde{\mathbf{o}}) + (1-\mathbf{o})\log(1-\tilde{\mathbf{o}})]. \tag{4}$$

As we predict 3D models in the camera space of an input image, we rotate ground truth 3D models so that the projection of a ground truth 3D model on the input image plane matches the observations on I. This step is necessary to acquire ground truth occupancies $\mathbf{o}$ in the camera space of an input image.

4.5   Inference

During inference, as we do not have access to camera-object distance, nor camera intrinsics, the scale calibration factor can be set to a random value in the range of the scaling factor used during training. As a result, our method recovers correct shapes in an unknown scale due to the inherent scale/depth ambiguity in the single-view 3D reconstruction task.

Before extracting meshes using the Marching Cubes algorithm [14], we apply an additional re-sampling process on the Ray-ONet occupancy predictions, which is in the shape of

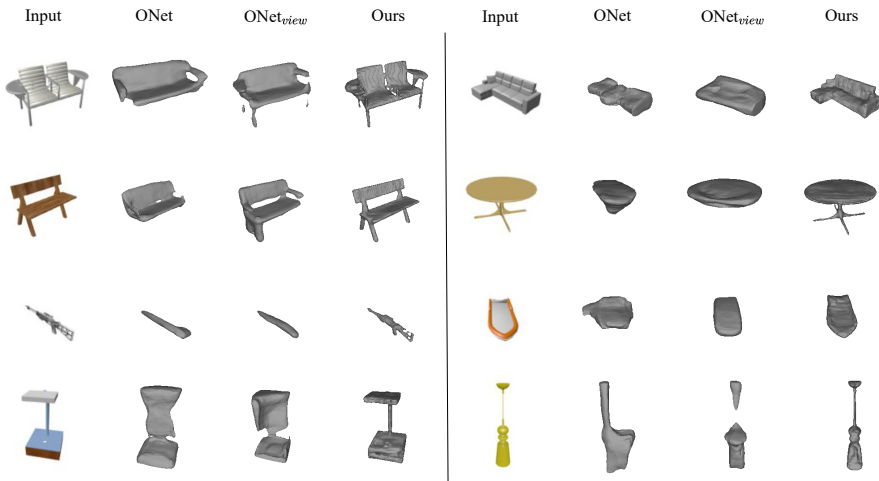| Input | ONet | ONet$_{view}$ | Ours | Input | ONet | ONet$_{view}$ | Ours |

Figure 4: **Qualitative Reconstruction Results on Unseen Categories.** We show Ray-ONet can generalise better than our baselines on unseen categories, in which object shapes are significantly different from the training classes (airplane, car, and chair).

a camera frustum due to the nature of our ray-based method, to get occupancies in a regular 3D grid. This re-sampling process can be simply done by trilinear interpolation.

5 Experiments

We conduct experiments to evaluate our method on two benchmarks, showing our method achieves state-of-the-art performance while being much faster during inference. This section is organised as follows, Section 5.1 reports the setup of our experiments. We compare our reconstruction quality with baselines in Section 5.2. Memory and time consumption are reported in Section 5.3. Lastly, we conduct an ablation study in Section 5.4.

5.1 Setup

Dataset. We evaluate Ray-ONet on the ShapeNet dataset [2], which is a large-scale dataset containing 3D synthetic object meshes with category labels. To make fair comparisons with prior works, we follow the evaluation protocol in ONet [15], taking a pre-processed subset from 3D-R2N2 [4], which includes image renderings from 13 classes in ShapeNetCore v1 [2] and adopting the same train/test splits.

Metrics. Following ONet [15], we adopt the volumetric IoU, the Chamfer-L1 distance, and the normal consistency score (NC) as the evaluation metrics. The volumetric IoU is evaluated on 100k randomly sampled points from the bounding volume. Additionally, we compute IoU on voxelised meshes in order to compare with some other baseline models. Methods with different IoU evaluation methods are separately listed in Table 1. Excepting from Table 1, our paper follows ONet's IoU evaluation method by default.

Implementation Details. Network initialisation: Apart from the image encoder $\mathcal{H}$, which is fine tuned from an ImageNet[5]-pretrained ResNet-18 backbone[9] with a random initialised fully-connected layer head, the local context mixer $\mathcal{J}$ and the occupancy decoder $\mathcal{D}$ are both

trained from scratch. Training details: During training, we randomly sample 1024 pixels on an input image and predict occupancies for 128 points that are equally spaced between a defined depth range on the rays. We use a batch size of 64, with an Adam[10] optimiser and a learning rate of 10^{-4}. Network architecture: A detailed description of network architectures is available in the supplementary material.

5.2 Reconstruction Accuracy

Standard Reconstruction. We evaluate the performance of Ray-ONet on the standard 3D reconstruction task, training/testing on all 13 categories. We compare our model with prior works including ONet [15], AtlasNet [7], CoReNet [18], DmifNet [12], 3D43D [1], DISN [28] and Ladybird [29]. Note that ONet, AtlasNet, DmifNet and DISN are object-centred models and the others are viewer-centred. DISN and Ladybird have different object scale from ours, hence we only evaluate IoU, which is scale-invariant. Table 1 reports that our model achieves state-of-the-art 3D reconstruction accuracy on the ShapeNet benchmark.

IoU ↑	airplane	bench	cabinet	car	chair	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
ONet	0.591	0.492	0.750	0.746	0.530	0.518	0.400	0.677	0.480	0.693	0.542	0.746	0.547	0.593
DmifNet	**0.603**	0.512	0.753	**0.758**	0.542	0.560	0.416	0.675	0.493	0.701	0.550	0.750	0.574	0.607
CoReNet†	0.543	0.431	0.715	0.718	0.512	0.557	0.430	0.686	0.589	0.686	0.527	0.714	0.562	0.590
3D43D	0.571	0.502	0.761	0.741	0.564	0.548	0.453	**0.729**	0.529	0.718	0.574	0.740	0.588	0.621
Ours	0.574	**0.521**	**0.763**	0.755	**0.567**	**0.601**	**0.467**	0.726	**0.594**	**0.726**	0.578	**0.757**	**0.598**	**0.633**
DISN	0.617	0.542	0.531	0.770	0.549	0.577	0.397	0.559	0.680	0.671	0.489	0.736	0.602	0.594
Ladybird	**0.666**	0.603	0.564	0.802	**0.647**	0.637	0.485	0.615	**0.719**	0.735	0.581	**0.781**	**0.688**	0.656
Ours	0.630	**0.630**	**0.806**	**0.815**	0.631	**0.652**	**0.549**	**0.768**	0.662	**0.768**	**0.654**	0.773	0.658	**0.692**
Chamfer-L1 ↓	airplane	bench	cabinet	car	chair	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
AtlasNet*	0.104	0.138	0.175	0.141	0.209	0.198	0.305	0.245	0.115	0.177	0.190	0.128	**0.151**	0.175
ONet	0.134	0.150	0.153	0.149	0.206	0.258	0.368	0.266	0.143	0.181	0.182	0.127	0.201	0.194
DmifNet	0.131	0.141	0.149	0.142	0.203	0.220	0.351	0.263	0.135	0.181	0.173	0.124	0.189	0.185
CoReNet†	0.135	0.157	0.181	0.173	0.216	0.213	0.271	0.263	**0.091**	0.176	0.170	0.128	0.216	0.184
3D43D	**0.096**	**0.112**	**0.119**	**0.122**	0.193	**0.166**	0.561	0.229	0.248	**0.125**	0.146	**0.107**	0.175	0.184
Ours	0.121	0.124	0.140	0.145	**0.178**	0.183	**0.238**	**0.204**	0.094	0.151	**0.140**	0.109	0.163	**0.153**
NC ↑	airplane	bench	cabinet	car	chair	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
AtlasNet*	0.836	0.779	0.850	0.836	0.791	0.858	0.694	0.825	0.725	0.840	0.832	0.923	0.756	0.811
ONet	0.845	0.814	0.884	0.852	0.829	0.857	0.751	0.842	0.783	0.867	0.860	**0.939**	0.797	0.840
DmifNet	**0.853**	0.821	0.885	**0.857**	0.835	0.872	0.758	0.847	0.781	0.873	0.868	0.936	0.808	0.846
CoReNet†	0.752	0.742	0.814	0.771	0.764	0.815	0.713	0.786	0.762	0.801	0.790	0.870	0.722	0.777
3D43D	0.825	0.809	**0.886**	0.844	0.832	**0.883**	0.766	**0.868**	0.798	0.875	0.864	0.935	0.799	0.845
Ours	0.830	**0.828**	0.881	0.849	**0.844**	**0.883**	**0.780**	0.866	**0.837**	**0.881**	**0.873**	0.929	**0.815**	**0.854**

Table 1: **Reconstruction Accuracy on ShapeNet.** We show Ray-ONet achieves state-of-the-art reconstruction accuracy in all three evaluation metrics. For IoU evaluation, the first group (5 rows) follows ONet's evaluation pipeline, the second group (3 rows) follows DISN's pipeline. The notation * denotes results taken from [15] and † denotes results we train their public code and evaluate in ONet's protocol. All other results are taken from corresponding original publications.

Reconstruction on Unseen Categories. To evaluate the generalisation capability, we train our model on 3 classes (airplane, car, and chair) and test on the remaining 10 classes. To make a fair comparison, we additionally train an ONet model with the viewer-centred setup, named ONet$_{view}$. Table 2 reports that our model performs better than all baselines in this setup.

5.3 Memory and Inference Time

As shown in Table 3, Ray-ONet is over 20× faster than ONet even if the efficient Multi-resolution IsoSurface Extraction is used in ONet while maintaining a similar memory footprint. It is worth noting that 3D-R2N2 [4] uses a 3D auto-encoder architecture to predict on the discretised voxel grid. Although it is 4× faster than Ray-ONet, it produces inferior reconstruction accuracy and requires a much larger memory size with 20× parameters. Since

IoU ↑	bench	cabinet	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
ONet	0.251	0.400	0.165	0.147	0.394	0.172	0.505	0.222	0.147	0.374	0.278
ONet$_{view}$	0.241	0.442	0.204	0.135	0.449	0.104	0.524	0.264	0.239	0.302	0.290
3D43D	0.302	0.502	0.243	0.223	0.507	0.236	0.559	0.313	0.271	**0.401**	0.356
Ours	**0.334**	**0.517**	**0.256**	**0.246**	**0.538**	**0.253**	**0.583**	**0.350**	**0.277**	0.394	**0.375**
Chamfer-L1 ↓	bench	cabinet	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
ONet	0.515	0.504	1.306	1.775	0.635	0.913	0.412	0.629	0.659	0.392	0.774
ONet$_{view}$	0.443	0.418	0.840	2.976	0.496	1.188	0.343	0.470	0.532	0.469	0.818
3D43D	0.357	0.529	1.389	1.997	0.744	0.707	0.421	0.583	0.996	0.521	0.824
Ours	**0.228**	**0.351**	**0.592**	**0.485**	**0.389**	**0.307**	**0.266**	**0.299**	**0.453**	**0.304**	**0.367**
NC ↑	bench	cabinet	display	lamp	loudspeaker	rifle	sofa	table	telephone	vessel	mean
ONet	0.694	0.697	0.527	0.563	0.693	0.583	0.779	0.701	0.605	0.694	0.654
ONet$_{view}$	0.717	**0.762**	0.600	0.508	0.740	0.505	0.796	0.740	0.673	0.679	0.672
3D43D	0.706	0.759	0.638	0.618	0.749	0.588	0.784	0.731	**0.700**	0.690	0.696
Ours	**0.736**	**0.762**	**0.640**	**0.642**	**0.762**	**0.609**	**0.806**	**0.764**	0.670	**0.703**	**0.709**

Table 2: **Reconstruction Accuracy on Unseen Categories.** We report the performance of different models on 10 unseen categories after trained on 3 categories. Our method outperforms all prior works in almost all categories.

SeqXY2SeqZ [8] has not released their code, we can only compare with them on complexity and accuracy. The inference time complexity of SeqXY2SeqZ is $O(N^2) \times l_{RNN}$, where l_{RNN} denotes the number of the recurrent steps. While being more efficient, we outperform SeqXY2SeqZ by 0.152 measured in IoU at 32^3 resolution (SeqXY2SeqZ reports its IoU at this resolution).

Methods	ONet	3D-R2N2	CoReNet	Ours
# Params (M)	13.4	282.0	36.1	14.1
Inference Memory (MB)	1449	5113	2506	1559
Inference time (ms)	333.1	3.4	37.8	13.8

Table 3: **Comparisons of Number of Parameters, Inference Memory and Inference Time at 128^3 Resolution**. The inference time measures the network forward time only and does not include any post-processing time.

5.4 Ablation Study

We analyse the effectiveness of the scale calibration factor s, the global and local image features as well as the architecture design.

Effect of the Scale Calibration Factor. Training with the scale calibration factor can lead to better reconstruction accuracy as reported in Table 4. This shows that decoupling scale from shape learning not only enables us to train with varying scales of ground-truth shapes but also helps in network convergence. As shown in Fig. 5, the scale factor helps to locate the object and reconstruct the object with correct scale.

Effect of Global and Local Image Features. Both global and local image features are important for reconstruction accuracy as suggested by the performance gap between w/ and w/o features in Table 4. Visualisation in Fig. 5 shows that local feature enables more details to be reconstructed, and global feature gives prior knowledge about object shape.

Effect of Architecture. We test a variation of our model by removing the Local Context Mixer and concatenating all inputs $(\mathbf{z}, s, C_{\mathbf{p}}, \mathbf{p})$ to feed into an Occupancy Decoder with MLPs. The evaluation mIoU for this variation is 0.610, showing our original design can achieve better performance.
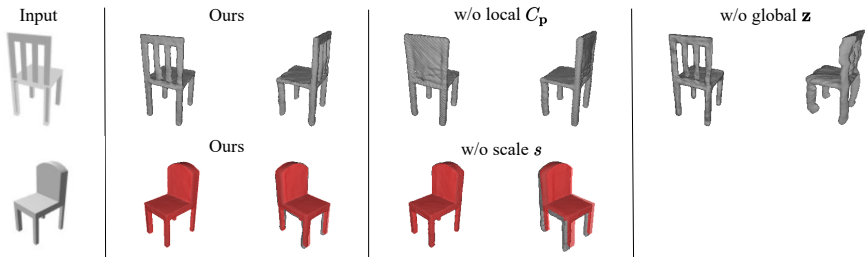
Figure 5: **Visualisation for Ablation Study.** Without the local feature, details of the object can not be recovered. Without the global feature, artifacts occur in other views. Without the scale factor, the scale of prediction (grey) can be different from ground truth (red).

	IoU ↑	Chamfer-L_1 ↓	NC ↑
Ours	**0.633**	**0.153**	**0.854**
w/o scale s	0.545	0.226	0.818
w/o global $\mathbf{z}$	0.574	0.206	0.821
w/o local $C_{\mathbf{p}}$	0.525	0.224	0.802

Table 4: **Ablation Study.** We show the effectiveness of the scale calibration factor s, the global image feature $\mathbf{z}$ and the local image feature $C_{\mathbf{p}}$.

5.5 Discussion

We study the effect of changing number of samples in the image-plane dimension S_{plane} and along the ray direction M. Results: Fig. 6 shows that although the reconstruction quality relates to both sampling dimensions, having $M > 128$ along a ray cannot further improve reconstruction quality, whereas sampling more pixels in an image plane is still beneficial. Insights: These results are not surprising as deducing occluded space is an ill-posed problem and can only be resolved from learning shape priors during training, which is bottlnecked by training data rather than $M = 128$. On the other hand, since sampling more pixels in an image plane provides richer texture details, the reconstruction quality improves when S_{plane} increases. Setup: To study the effect of different S_{plane}, we fix $M = 128$ and test with $S_{plane} = 32^2|64^2|128^2|256^2$. To study the effect of different M, we fix $S_{plane} = 128^2$ and test with $M = 32|64|128|256$.
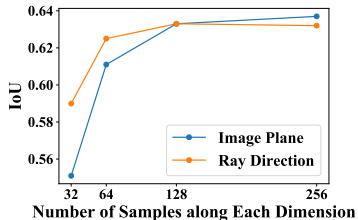

Figure 6: IoU of various No. of samples on a ray and in an image plane.

6 Conclusion

We proposed Ray-ONet, a deep neural network that reconstructs detailed object shapes efficiently given a single RGB image. We demonstrate that predicting a series of occupancies along back-projected rays from both global and local features can significantly reduce the time complexity at inference while improving the reconstruction quality. Experiments show that Ray-ONet is able to achieve state-of-the-art performance for both seen and unseen categories in ShapeNet with a significant evaluation time reduction. As our single-view object reconstruction results are encouraging, we believe an interesting direction in the future is to extend Ray-ONet to tasks such as scene-level reconstructions or novel view synthesis.

Acknowledgements

References

[1] Miguel Angel Bautista, Walter Talbott, Shuangfei Zhai, Nitish Srivastava, and Joshua M Susskind. On the generalization of learning-based 3d reconstruction. In WACV, 2021.

[2] 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.

[3] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In CVPR, 2019.

[4] 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 ECCV, 2016.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009.

[6] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In CVPR, 2017.

[7] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In CVPR, 2018.

[8] Zhizhong Han, Guanhui Qiao, Yu-Shen Liu, and Matthias Zwicker. Seqxy2seqz: Structure learning for 3d shapes by sequentially predicting 1d occupancy segments from 2d coordinates. In ECCV, 2020.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015.

[11] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. TPAMI, 1994.

[12] Lei Li and Suping Wu. Dmifnet: 3d shape reconstruction based on dynamic multi-branch information fusion. In International Conference on Pattern Recognition (ICPR), 2021.

[13] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In AAAI, 2018.

[14] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. ACM SIGGRAPH Computer Graphics, 1987.

[15] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In CVPR, 2019.

[16] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In CVPR, 2019.

[17] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In ECCV, 2020.

[18] Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. Corenet: Coherent 3d scene reconstruction from a single rgb image. In ECCV, 2020.

[19] Omid Poursaeed, Matthew Fisher, Noam Aigerman, and Vladimir G Kim. Coupling explicit and implicit surface representations for generative 3d modeling. In ECCV, 2020.

[20] Martin Runz, Kejie Li, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, et al. Frodo: From detections to 3d objects. In CVPR, 2020.

[21] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In CVPR, 2019.

[22] Daeyun Shin, Charless C Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In CVPR, 2018.

[23] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In ICCV, 2017.

[24] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In CVPR, 2019.

[25] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In ECCV, 2018.

[26] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning shape priors for single-view 3d completion and reconstruction. In ECCV, 2018.

[27] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In ICCV, 2019.

[28] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In NeurIPS, 2019.

[29] Yifan Xu, Tianqi Fan, Yi Yuan, and Gurprit Singh. Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry. In ECCV, 2020.

[30] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. TPAMI, 1999.