Convolutional Occupancy Networks

Songyou Peng[1,2]   Michael Niemeyer[2,3]   Lars Mescheder[2,4⋆]
Marc Pollefeys[1,5]   Andreas Geiger[2,3]

[1]ETH Zurich   [2]Max Planck Institute for Intelligent Systems, Tübingen
[3]University of Tübingen   [4]Amazon, Tübingen   [5]Microsoft

Abstract. Recently, implicit neural representations have gained popularity for learning-based 3D reconstruction. While demonstrating promising results, most implicit approaches are limited to comparably simple geometry of single objects and do not scale to more complicated or large-scale scenes. The key limiting factor of implicit methods is their simple fully-connected network architecture which does not allow for integrating local information in the observations or incorporating inductive biases such as translational equivariance. In this paper, we propose Convolutional Occupancy Networks, a more flexible implicit representation for detailed reconstruction of objects and 3D scenes. By combining convolutional encoders with implicit occupancy decoders, our model incorporates inductive biases, enabling structured reasoning in 3D space. We investigate the effectiveness of the proposed representation by reconstructing complex geometry from noisy point clouds and low-resolution voxel representations. We empirically find that our method enables the fine-grained implicit 3D reconstruction of single objects, scales to large indoor scenes, and generalizes well from synthetic to real data.

1 Introduction

3D reconstruction is a fundamental problem in computer vision with numerous applications. An ideal representation of 3D geometry should have the following properties: a) encode complex geometries and arbitrary topologies, b) scale to large scenes, c) encapsulate local and global information, and d) be tractable in terms of memory and computation.

Unfortunately, current representations for 3D reconstruction do not satisfy all of these requirements. Volumetric representations [25] are limited in terms of resolution due to their large memory requirements. Point clouds [9] are lightweight 3D representations but discard topological relations. Mesh-based representations [13] are often hard to predict using neural networks.

Recently, several works [3, 26, 27, 31] have introduced deep implicit representations which represent 3D structures using learned occupancy or signed distance functions. In contrast to explicit representations, implicit methods do not discretize 3D space during training, thus resulting in continuous representations of 3D geometry without topology restrictions. While inspiring many follow-up

---

⋆ This work was done prior to joining Amazon.

(a) Occupancy Network [26]

(b) Conv. Occupancy Network
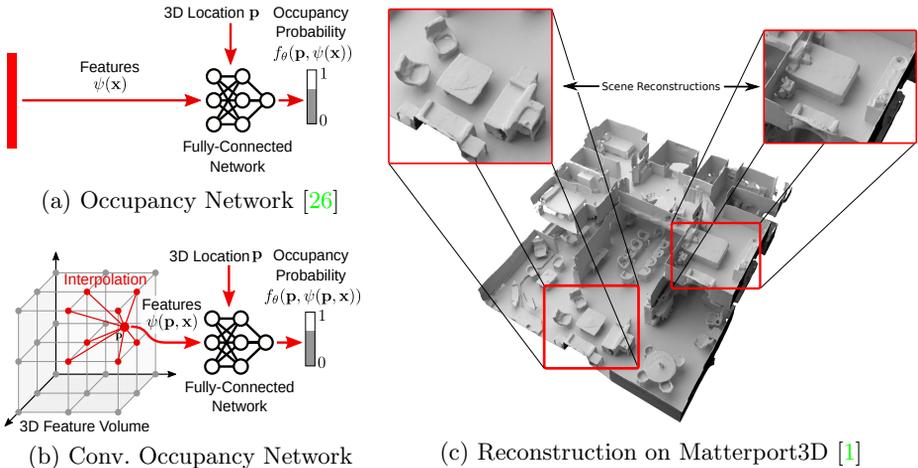
(c) Reconstruction on Matterport3D [1]

Fig. 1: **Convolutional Occupancy Networks.** Traditional implicit models (a) are limited in their expressiveness due to their fully-connected network structure. We propose Convolutional Occupancy Networks (b) which exploit convolutions, resulting in scalable and equivariant implicit representations. We query the convolutional features at 3D locations $\mathbf{p} \in \mathbb{R}^3$ using linear interpolation. In contrast to Occupancy Networks (ONet) [26], the proposed feature representation $\psi(\mathbf{p}, \mathbf{x})$ therefore depends on *both* the input $\mathbf{x}$ and the 3D location $\mathbf{p}$. Fig. (c) shows a reconstruction of a two-floor building from a noisy point cloud on the Matterport3D dataset [1].

works [10, 11, 23, 24, 28–30, 41], all existing approaches are limited to single objects and do not scale to larger scenes. The key limiting factor of most implicit models is their simple fully-connected network architecture [26, 31] which neither allows for integrating local information in the observations, nor for incorporating inductive biases such as translation equivariance into the model. This prevents these methods from performing *structured reasoning* as they only act globally and result in overly smooth surface reconstructions.

In contrast, translation equivariant convolutional neural networks (CNNs) have demonstrated great success across many 2D recognition tasks including object detection and image segmentation. Moreover, CNNs naturally encode information in a hierarchical manner in different network layers [50, 51]. Exploiting these inductive biases is expected to not only benefit 2D but also 3D tasks, e.g., reconstructing 3D shapes of multiple similar chairs located in the same room. In this work, we seek to combine the complementary strengths of convolutional neural networks with those of implicit representations.

Towards this goal, we introduce *Convolutional Occupancy Networks*, a novel representation for accurate large-scale 3D reconstruction[1] with continuous implicit representations (Fig. 1). We demonstrate that this representation not only

---

[1] With 3D reconstruction, we refer to 3D surface reconstruction throughout the paper.

preserves fine geometric details, but also enables the reconstruction of complex indoor scenes at scale. Our key idea is to establish rich input features, incorporating inductive biases and integrating local as well as global information. More specifically, we exploit convolutional operations to obtain translation equivariance and exploit the local self-similarity of 3D structures. We systematically investigate multiple design choices, ranging from canonical planes to volumetric representations. Our contributions are summarized as follows:

- We identify major limitations of current implicit 3D reconstruction methods.
- We propose a flexible translation equivariant architecture which enables accurate 3D reconstruction from object to scene level.
- We demonstrate that our model enables generalization from synthetic to real scenes as well as to novel object categories and scenes.

Our code and data are provided at https://github.com/autonomousvision/convolutional_occupancy_networks.

2   Related Work

Learning-based 3D reconstruction methods can be broadly categorized by the output representation they use.

Voxels: Voxel representations are amongst the earliest representations for learning-based 3D reconstruction [5,46,47]. Due to the cubic memory requirements of voxel-based representations, several works proposed to operate on multiple scales or use octrees for efficient space partitioning [8,14,25,37,38,42]. However, even when using adaptive data structures, voxel-based techniques are still limited in terms of memory and computation.

Point Clouds: An alternative output representation for 3D reconstruction is 3D point clouds which have been used in [9,21,34,49]. However, point cloud-based representations are typically limited in terms of the number of points they can handle. Furthermore, they cannot represent topological relations.

Meshes: A popular alternative is to directly regress the vertices and faces of a mesh [12,13,17,20,22,44,45] using a neural network. While some of these works require deforming a template mesh of fixed topology, others result in non-watertight reconstructions with self-intersecting mesh faces.

Implicit Representations: More recent implicit occupancy [3,26] and distance field [27,31] models use a neural network to infer an occupancy probability or distance value given any 3D point as input. In contrast to the aforementioned explicit representations which require discretization (e.g., in terms of the number of voxels, points or vertices), implicit models represent shapes continuously and naturally handle complicated shape topologies. Implicit models have been adopted for learning implicit representations from images [23,24,29,41], for encoding texture information [30], for 4D reconstruction [28] as well as for primitive-based reconstruction [10,11,15,32]. Unfortunately, all these methods are limited to

comparably simple 3D geometry of single objects and do not scale to more complicated or large-scale scenes. The key limiting factor is the simple fully-connected network architecture which does not allow for integrating local features or incorporating inductive biases such as translation equivariance.

Notable exceptions are PIFu [40] and DISN [48] which use pixel-aligned implicit representations to reconstruct people in clothing [40] or ShapeNet objects [48]. While these methods also exploit convolutions, all operations are performed in the *2D image domain*, restricting these models to image-based inputs and reconstruction of single objects. In contrast, in this work, we propose to aggregate features in *physical 3D space*, exploiting both 2D and 3D convolutions. Thus, our world-centric representation is independent of the camera viewpoint and input representation. Moreover, we demonstrate the feasibility of implicit 3D reconstruction at scene-level as illustrated in Fig. 1c.

In concurrent work, Chibane et al. [4] present a model similar to our convolutional volume decoder. In contrast to us, they only consider a single variant of convolutional feature embeddings (3D), use lossy discretization for the 3D point cloud encoding and only demonstrate results on single objects and humans, as opposed to full scenes. In another concurrent work, Jiang et al. [16] leverage shape priors for scene-level implicit 3D reconstruction. In contrast to us, they use 3D point normals as input and require optimization at inference time.

3   Method

Our goal is to make implicit 3D representations more expressive. An overview of our model is provided in Fig. 2. We first encode the input **x** (e.g., a point cloud) into a 2D or 3D feature grid (left). These features are processed using convolutional networks and decoded into occupancy probabilities via a fully-connected network. We investigate planar representations (a+c+d), volumetric representations (b+e) as well as combinations thereof in our experiments. In the following, we explain the encoder (Section 3.1), the decoder (Section 3.2), the occupancy prediction (Section 3.3) and the training procedure (Section 3.4) in more detail.

3.1   Encoder

While our method is independent of the input representation, we focus on 3D inputs to demonstrate the ability of our model in recovering fine details and scaling to large scenes. More specifically, we assume a noisy sparse point cloud (e.g., from structure-from-motion or laser scans) or a coarse occupancy grid as input **x**.

We first process the input **x** with a task-specific neural network to obtain a feature encoding for every point or voxel. We use a one-layer 3D CNN for voxelized inputs, and a shallow PointNet [35] with local pooling for 3D point clouds. Given these features, we construct planar and volumetric feature representations in order to encapsulate local neighborhood information as follows.
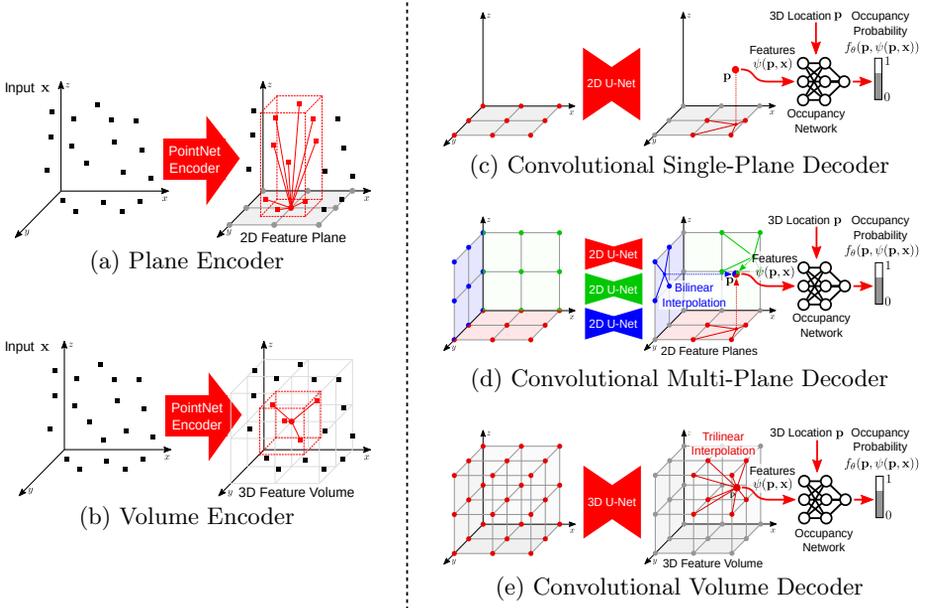
Fig. 2: **Model Overview.** The **encoder (left)** first converts the 3D input **x** (e.g., noisy point clouds or coarse voxel grids) into features using task-specific neural networks. Next, the features are projected onto one or multiple planes (Fig. 2a) or into a volume (Fig. 2b) using average pooling. The **convolutional decoder (right)** processes the resulting feature planes/volume using 2D/3D U-Nets to aggregate local and global information. For a query point $\mathbf{p} \in \mathbb{R}^3$, the point-wise feature vector $\psi(\mathbf{x}, \mathbf{p})$ is obtained via bilinear (Fig. 2c and Fig. 2d) or trilinear (Fig. 2e) interpolation. Given feature vector $\psi(\mathbf{x}, \mathbf{p})$ at location $\mathbf{p}$, the occupancy probability is predicted using a fully-connected network $f_\theta(\mathbf{p}, \psi(\mathbf{p}, \mathbf{x}))$.

Plane Encoder: As illustrated in Fig. 2a, for each input point, we perform an orthographic projection onto a canonical plane (i.e., a plane aligned with the axes of the coordinate frame) which we discretize at a resolution of $H \times W$ pixel cells. For voxel inputs, we treat the voxel center as a point and project it to the plane. We aggregate features projecting onto the same pixel using average pooling, resulting in planar features with dimensionality $H \times W \times d$, where d is the feature dimension.

In our experiments, we analyze two variants of our model: one variant where features are projected onto the ground plane, and one variant where features are projected to all three canonical planes. While the former is computationally more efficient, the latter allows for recovering richer geometric structure in the z dimension.

Volume Encoder: While planar feature representations allow for encoding at large spatial resolution (128^2 pixels and beyond), they are restricted to two dimensions. Therefore, we also consider volumetric encodings (see Fig. 2b) which

better represent 3D information, but are restricted to smaller resolutions (typically 32^3 voxels in our experiments). Similar to the plane encoder, we perform average pooling, but this time over all features falling into the same *voxel* cell, resulting in a feature volume of dimensionality $H \times W \times D \times d$.

3.2 Decoder

We endow our model with translation equivariance by processing the feature planes and the feature volume from the encoder using 2D and 3D convolutional hourglass (U-Net) networks [6,39] which are composed of a series of down- and upsampling convolutions with skip connections to integrate both local and global information. We choose the depth of the U-Net such that the receptive field becomes equal to the size of the respective feature plane or volume.

Our single-plane decoder (Fig. 2c) processes the ground plane features with a 2D U-Net. The multi-plane decoder (Fig. 2d) processes each feature plane separately using 2D U-Nets with shared weights. Our volume decoder (Fig. 2e) uses a 3D U-Net. Since convolution operations are translational equivariant, our output features are also translation equivariant, enabling structured reasoning. Moreover, convolutional operations are able to "inpaint" features while preserving global information, enabling reconstruction from sparse inputs.

3.3 Occupancy Prediction

Given the aggregated feature maps, our goal is to estimate the occupancy probability of any point $\mathbf{p}$ in 3D space. For the single-plane decoder, we project each point $\mathbf{p}$ orthographically onto the ground plane and query the feature value through bilinear interpolation (Fig. 2c). For the multi-plane decoder (Fig. 2d), we aggregate information from the 3 canonical planes by summing the features of all 3 planes. For the volume decoder, we use trilinear interpolation (Fig. 2e).

Denoting the feature vector for input $\mathbf{x}$ at point $\mathbf{p}$ as $\psi(\mathbf{p}, \mathbf{x})$, we predict the occupancy of $\mathbf{p}$ using a small fully-connected occupancy network:

$$f_\theta(\mathbf{p}, \psi(\mathbf{p}, \mathbf{x})) \to [0, 1] \tag{1}$$

The network comprises multiple ResNet blocks. We use the network architecture of [29], adding ψ to the input features of every ResNet block instead of the more memory intensive batch normalization operation proposed in earlier works [26]. In contrast to [29], we use a feature dimension of 32 for the hidden layers. Details about the network architecture can be found in the supplementary.

3.4 Training and Inference

At training time, we uniformly sample query points $\mathbf{p} \in \mathbb{R}^3$ within the volume of interest and predict their occupancy values. We apply the binary cross-entropy loss between the predicted $\hat{o}_{\mathbf{p}}$ and the true occupancy values $o_{\mathbf{p}}$:

$$\mathcal{L}(\hat{o}_{\mathbf{p}}, o_{\mathbf{p}}) = -[o_{\mathbf{p}} \cdot \log(\hat{o}_{\mathbf{p}}) + (1 - o_{\mathbf{p}}) \cdot \log(1 - \hat{o}_{\mathbf{p}})] \tag{2}$$

We implement all models in PyTorch [33] and use the Adam optimizer [19] with a learning rate of 10^{-4}. During inference, we apply *Multiresolution IsoSurface Extraction* (MISE) [26] to extract meshes given an input **x**. As our model is fully-convolutional, we are able to reconstruct large scenes by applying it in a "sliding-window" fashion at inference time. We exploit this property to obtain reconstructions of entire apartments (see Fig. 1).

4   Experiments

We conduct three types of experiments to evaluate our method. First, we perform **object-level reconstruction** on ShapeNet [2] chairs, considering noisy point clouds and low-resolution occupancy grids as inputs. Next, we compare our approach against several baselines on the task of **scene-level reconstruction** using a synthetic indoor dataset of various objects. Finally, we demonstrate **synthetic-to-real generalization** by evaluating our model on real indoor scenes [1,7].

Datasets:

ShapeNet [2]: We use all 13 classes of the ShapeNet subset, voxelizations, and train/val/test split from Choy et al. [5]. Per-class results can be found in supplementary.

Synthetic Indoor Scene Dataset: We create a synthetic dataset of 5000 scenes with multiple objects from ShapeNet (chair, sofa, lamp, cabinet, table). A scene consists of a ground plane with randomly sampled width-length ratio, multiple objects with random rotation and scale, and randomly sampled walls.

ScanNet v2 [7]: This dataset contains 1513 real-world rooms captured with an RGB-D camera. We sample point clouds from the provided meshes for testing.

Matterport3D [1]: Matterport3D contains 90 buildings with multiple rooms on different floors captured using a Matterport Pro Camera. Similar to ScanNet, we sample point clouds for evaluating our model on Matterport3D.

Baselines:

ONet [26]: Occupancy Networks is a state-of-the-art implicit 3D reconstruction model. It uses a fully-connected network architecture and a global encoding of the input. We compare against this method in all of our experiments.

PointConv: We construct another simple baseline by extracting point-wise features using PointNet++ [36], interpolating them using Gaussian kernel regression and feeding them into the same fully-connected network used in our approach. While this baseline uses local information, it does not exploit convolutions.

SPSR [18]: Screened Poisson Surface Reconstruction (SPSR) is a traditional 3D reconstruction technique which operates on oriented point clouds as input. Note that in contrast to all other methods, SPSR requires additional surface normals which are often hard to obtain for real-world scenarios.

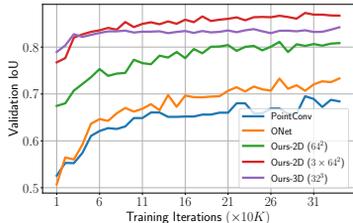|	GPU Memory	IoU	Chamfer-L_1	Normal C.	F-Score
PointConv	5.1G	0.689	0.126	0.858	0.644
ONet [26]	7.7G	0.761	0.087	0.891	0.785
Ours-2D (64^2)	1.6G	0.833	0.059	0.914	0.887
Ours-2D (3×64^2)	2.4G	**0.884**	**0.044**	**0.938**	**0.942**
Ours-3D (32^3)	5.9G	0.870	0.048	0.937	0.933

Table 1: **Object-Level 3D Reconstruction from Point Clouds.** Left: We report GPU memory, IoU, Chamfer-L_1 distance, Normal Consistency and F-Score for our approach (2D plane and 3D voxel grid dimensions in brackets), the baselines ONet [26] and PointConv on ShapeNet (mean over all 13 classes). Right: The training progression plot shows that our method converges faster than the baselines.

Metrics:

Following [26], we consider Volumetric IoU, Chamfer Distance, Normal Consistency for evaluation. We further report F-Score [43] with the default threshold value of 1% unless otherwise specified. Details can be found in the supplementary.

4.1   Object-Level Reconstruction

We first evaluate our method on the single object reconstruction task on ShapeNet [2]. We consider two different types of 3D inputs: noisy point clouds and low-resolution voxels. For the former, we sample 3000 points from the mesh and apply Gaussian noise with zero mean and standard deviation 0.05. For the latter, we use the coarse 32^3 voxelizations from [26]. For the query points (i.e., for which supervision is provided), we follow [26] and uniformly sample 2048 and 1024 points for noisy point clouds and low-resolution voxels, respectively. Due to the different encoder architectures for these two tasks, we set the batch size to 32 and 64, respectively.

Reconstruction from Point Clouds: Table 1 and Fig. 3 show quantitative and qualitative results. Compared to the baselines, all variants of our method achieve equal or better results on all three metrics. As evidenced by the training progression plot on the right, our method reaches a high validation IoU after only few iterations. This verifies our hypothesis that leveraging convolutions and local features benefits 3D reconstruction in terms of both accuracy and efficiency. The results show that, in comparison to PointConv which directly aggregates features from point clouds, projecting point-features to planes or volumes followed by 2D/3D CNNs is more effective. In addition, decomposing 3D representations from volumes into three planes with higher resolution (64^2 vs. 32^3) improves performance while at the same time requiring less GPU memory. More results can be found in supplementary.

Voxel Super-Resolution: Besides noisy point clouds, we also evaluate on the task of voxel super-resolution. Here, the goal is to recover high-resolution details
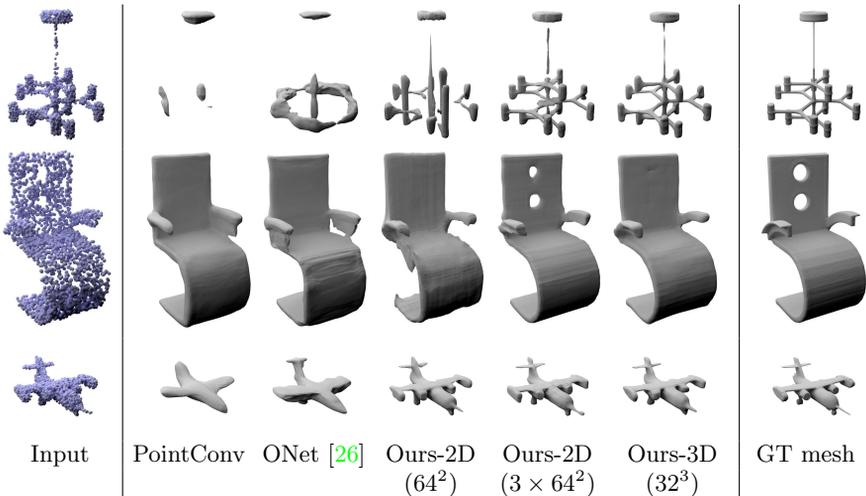
Fig. 3: **Object-Level 3D Reconstruction from Point Clouds.** Comparison of our convolutional representation to ONet and PointConv on ShapeNet.

	GPU Memory	IoU	Chamfer-L_1	Normal C.	F-Score
Input	-	0.631	0.136	0.810	0.440
ONet [26]	4.8G	0.703	0.110	0.879	0.656
Ours-2D (64^2)	2.4G	0.652	0.145	0.861	0.592
Ours-2D (3×64^2)	4.0G	**0.752**	0.092	0.905	**0.735**
Ours-3D (32^3)	10.8G	**0.752**	**0.091**	**0.912**	0.729

Table 2: **Voxel Super-Resolution.** 3D reconstruction results from low resolution voxelized inputs (32^3 voxels) on the ShapeNet dataset (mean over 13 classes).

from coarse (32^3) voxelizations of the shape. Table 2 and Fig. 4 show that our method with three planes achieves comparable results over our volumetric method while requiring only 37% of the GPU memory. In contrast to reconstruction from point clouds, our single-plane approach fails on this task. We hypothesize that a single plane is not sufficient for resolving ambiguities in the coarse but regularly structured voxel input.

4.2   Scene-Level Reconstruction

To analyze whether our approach can scale to larger scenes, we now reconstruct 3D geometry from point clouds on our synthetic indoor scene dataset. Due to the increasing complexity of the scene, we uniformly sample 10000 points as input point cloud and apply Gaussian noise with standard deviation of 0.05. During training, we sample 2048 query points, similar to object-level reconstruction. For our plane-based methods, we use a resolution to 128^2. For our volumetric approach, we investigate both 32^3 and 64^3 resolutions. Hypothesizing that the
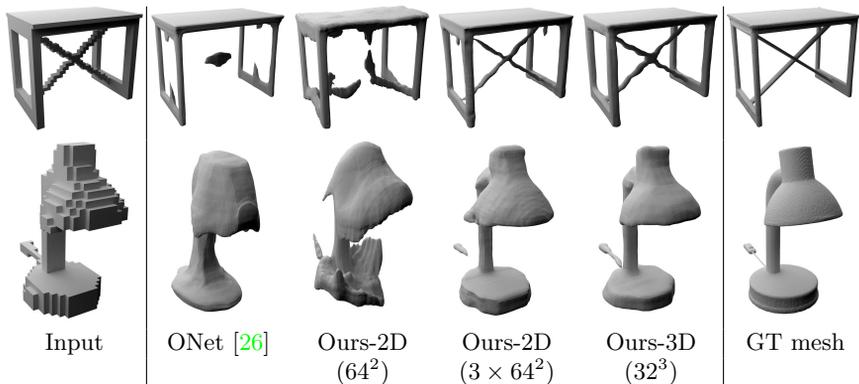
Fig. 4: **Voxel Super-Resolution.** Qualitative comparison between our method and ONet using coarse voxelized inputs at resolution 32^3 voxels.

	IoU	Chamfer-L_1	Normal Consistency	F-Score
ONet [26]	0.475	0.203	0.783	0.541
PointConv	0.523	0.165	0.811	0.790
SPSR [18]	-	0.223	0.866	0.810
SPSR [18] (trimmed)	-	0.069	0.890	0.892
Ours-2D (128^2)	0.795	0.047	0.889	0.937
Ours-2D (3×128^2)	0.805	0.044	0.903	0.948
Ours-3D (32^3)	0.782	0.047	0.902	0.941
Ours-3D (64^3)	**0.849**	**0.042**	**0.915**	**0.964**
Ours-2D-3D ($3 \times 128^2 + 32^3$)	0.816	0.044	0.905	0.952

Table 3: **Scene-Level Reconstruction on Synthetic Rooms.** Quantitative comparison for reconstruction from noisy point clouds. We do not report IoU for SPSR as SPSR generates only a single surface for walls and the ground plane. To ensure a fair comparison to SPSR, we compare all methods with only a single surface for walls/ground planes when calculating Chamfer-L_1 and F-Score.

plane and volumetric features are complementary, we also test the combination of the multi-plane and volumetric variants.

Table 3 and Fig. 5 show our results. All variants of our method are able to reconstruct geometric details of the scenes and lead to smooth results. In contrast, ONet and PointConv suffer from low accuracy while SPSR leads to noisy surfaces. While high-resolution canonical plane features capture fine details they are prone to noise. Low-resolution volumetric features are instead more robust to noise, yet produce smoother surfaces. Combining complementary volumetric and plane features improves results compared to considering them in isolation. This confirms our hypothesis that plane-based and volumetric features are complementary. However, the best results in this setting are achieved when increasing the resolution of the volumetric features to 64^3.

4.3 Ablation Study

In this section, we investigate on our synthetic indoor scene dataset different feature aggregation strategies at similar GPU memory consumption as well as different feature interpolation strategies.
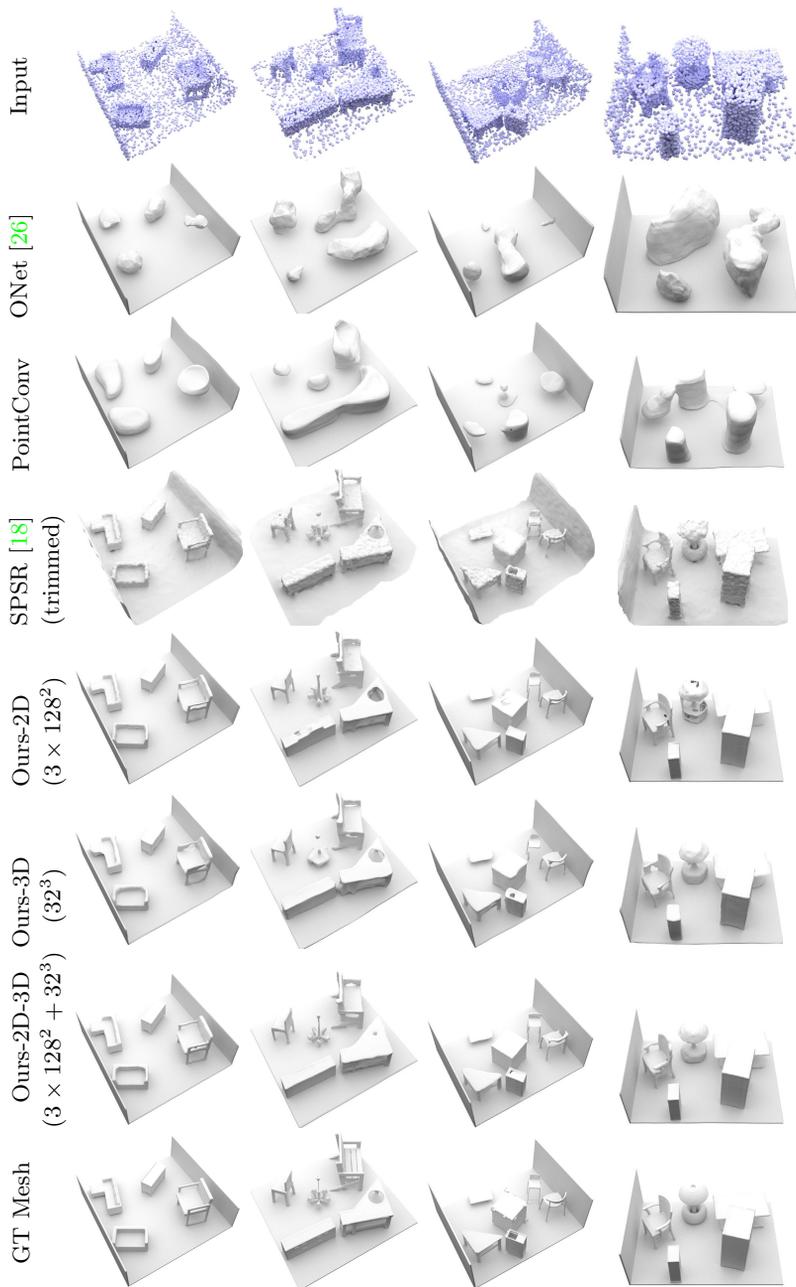
Fig. 5: **Scene-Level Reconstruction on Synthetic Rooms.** Qualitative comparison for point-cloud based reconstruction on the synthetic indoor scene dataset.

	GPU Memory	IoU	Chamfer-L_1	Normal C.	F-Score
Ours-2D (192^2)	9.5GB	0.773	0.047	0.889	0.937
Ours-2D (3×128^2)	9.3GB	**0.805**	**0.044**	**0.903**	**0.948**
Ours-3D (32^3)	8.5GB	0.782	0.047	0.902	0.941

	IoU	Chamfer-L_1	Normal C.	F-Score
Nearest Neighbor	0.766	0.052	0.885	0.920
Bilinear	**0.805**	**0.044**	**0.903**	**0.948**

(a) Performance at similar GPU Memory   (b) Interpolation Strategy

Table 4: **Ablation Study on Synthetic Rooms.** We compare the performance of different feature aggregation strategies at similar GPU memory in Table 4a and evaluate two different sampling strategies in Table 4b.

Performance at Similar GPU Memory: Table 4a shows a comparison of different feature aggregation strategies at similar GPU memory utilization. Our multi-plane approach slightly outperforms the single plane and the volumetric approach in this setting. Moreover, the increase in plane resolution for the single plane variant does not result in a clear performance boost, demonstrating that higher resolution does not necessarily guarantee better performance.

Feature Interpolation Strategy: To analyze the effect of the feature interpolation strategy in the convolutional decoder of our method, we compare nearest neighbor and bilinear interpolation for our multi-plane variant. The results in Table 4b clearly demonstrate the benefit of bilinear interpolation.

4.4   Reconstruction from Point Clouds on Real-World Datasets

Next, we investigate the generalization capabilities of our method. Towards this goal, we evaluate our models trained on the synthetic indoor scene dataset on the real world datasets ScanNet v2 [7] and Matterport3D [1]. Similar to our previous experiments, we use 10000 points sampled from the meshes as input.

ScanNet v2: Our results in Table 5 show that among all our variants, the volumetric-based models perform best, indicating that the plane-based approaches are more affected by the domain shift. We find that 3D CNNs are more robust to noise as they aggregate features from all neighbors which results in smooth outputs. Moreover, all variants outperform the learning-based baselines by a significant margin.

The qualitative comparison in Fig. 6 shows that our model is able to smoothly reconstruct scenes with geometric details at various scales. While Screened PSR [18] also produces reasonable reconstructions, it tends to close the resulting meshes and hence requires a carefully chosen trimming parameter. In contrast, our method does not require additional hyperparameters.

Matterport3D Dataset: Finally, we investigate the scalability of our method to larger scenes which comprise multiple rooms and multiple floors. For this experiment, we exploit the Matterport3D dataset. Unlike before, we implement a fully convolutional version of our 3D model that can be scaled to any size by running on overlapping crops of the input point cloud in a sliding window fashion. The overlap is determined by the size of the receptive field to ensure

	Chamfer-L_1	F-Score
ONet [26]	0.398	0.390
PointConv	0.316	0.439
SPSR [18]	0.293	0.731
SPSR [18] (trimmed)	0.086	0.847

	Chamfer-L_1	F-Score
Ours-2D (128^2)	0.139	0.747
Ours-2D (3×128^2)	0.142	0.776
Ours-3D (32^3)	0.095	0.837
Ours-3D (64^3)	**0.077**	**0.886**
Ours-2D-3D ($3 \times 128^2 + 32^3$)	0.099	0.847

Table 5: **Scene-Level Reconstruction on ScanNet.** Evaluation of point-based reconstruction on the real-world ScanNet dataset. As ScanNet does not provide watertight meshes, we trained all methods on the synthetic indoor scene dataset. Remark: In ScanNet, walls / floors are only observed from one side. To not wrongly penalize methods for predicting walls and floors with thickness (0.01 in our training set), we chose a F-Score threshold of 1.5% for this experiment.

correctness of the results. Fig. 1 shows the resulting 3D reconstruction. Our method reconstructs details inside each room while adhering to the room layout. Note that the geometry and point distribution of the Matterport3D dataset differs significantly from the synthetic indoor scene dataset which our model is trained on. This demonstrates that our method is able to generalize not only to unseen classes, but also novel room layouts and sensor characteristics. More implementation details and results can be found in supplementary.

5   Conclusion

We introduced Convolutional Occupancy Networks, a novel shape representation which combines the expressiveness of convolutional neural networks with the advantages of implicit representations. We analyzed the tradeoffs between 2D and 3D feature representations and found that incorporating convolutional operations facilitates generalization to unseen classes, novel room layouts and large-scale indoor spaces. We find that our 3-plane model is memory efficient, works well on synthetic scenes and allows for larger feature resolutions. Our volumetric model, in contrast, outperforms other variants on real-world scenarios while consuming more memory.

Finally, we remark that our method is not rotation equivariant and only translation equivariant with respect to translations that are multiples of the defined voxel size. Moreover, there is still a performance gap between synthetic and real data. While the focus of this work was on learning-based 3D reconstruction, in future work, we plan to apply our novel representation to other domains such as implicit appearance modeling and 4D reconstruction.
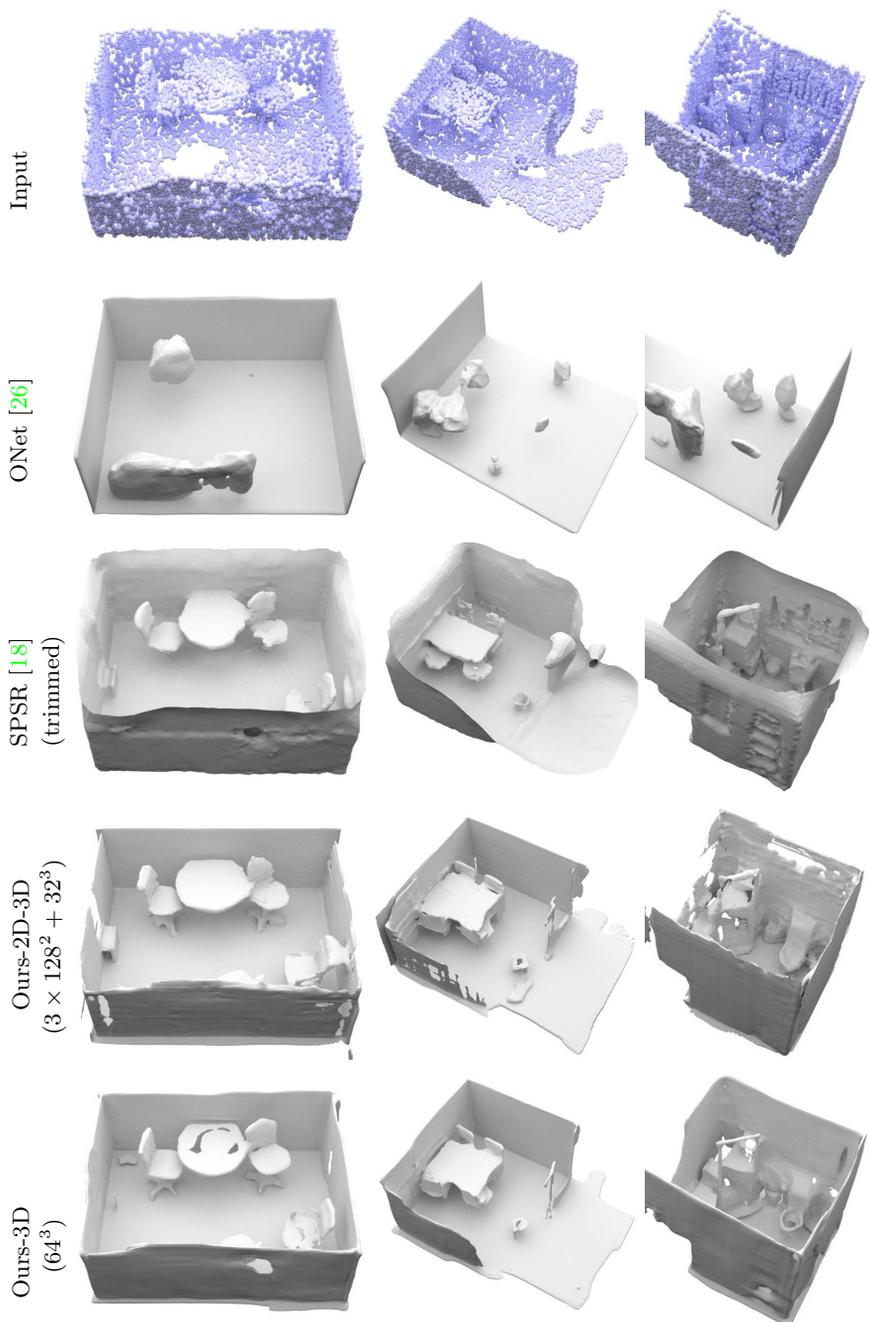
Fig. 6: **Scene-Level Reconstruction on ScanNet.** Qualitative results for point-based reconstruction on ScanNet [7]. All learning-based methods are trained on the synthetic room dataset and evaluated on ScanNet.

References

1. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. Proc. of the International Conf. on 3D Vision (3DV) (2017) 2, 7, 12
2. Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. arXiv.org **1512.03012** (2015) 7, 8
3. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 3
4. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 4
5. 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: Proc. of the European Conf. on Computer Vision (ECCV) (2016) 3, 7
6. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: Learning dense volumetric segmentation from sparse annotation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2016) 6
7. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Niessner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2017) 7, 12, 14
8. Dai, A., Qi, C.R., Nießner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2017) 3
9. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2017) 1, 3
10. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.A.: Local deep implicit functions for 3d shape. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 2, 3
11. Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 2, 3
12. Gkioxari, G., Malik, J., Johnson, J.: Mesh R-CNN. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 3
13. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: AtlasNet: A papier-mâché approach to learning 3d surface generation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2018) 1, 3
14. Hane, C., Tulsiani, S., Malik, J.: Hierarchical surface prediction for 3d object reconstruction. In: Proc. of the International Conf. on 3D Vision (3DV) (2017) 3
15. Jeruzalski, T., Deng, B., Norouzi, M., Lewis, J.P., Hinton, G.E., Tagliasacchi, A.: NASA: neural articulated shape approximation. In: Proc. of the European Conf. on Computer Vision (ECCV) (2020) 3
16. Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T.: Local implicit grid representations for 3d scenes. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 4
17. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Proc. of the European Conf. on Computer Vision (ECCV) (2018) 3

18. Kazhdan, M.M., Hoppe, H.: Screened poisson surface reconstruction. ACM Trans. on Graphics **32**(3), 29 (2013) 7, 10, 11, 12, 13, 14

19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proc. of the International Conf. on Machine learning (ICML) (2015) 7

20. Liao, Y., Donne, S., Geiger, A.: Deep marching cubes: Learning explicit surface representations. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2018) 3

21. Lin, C., Kong, C., Lucey, S.: Learning efficient point cloud generation for dense 3d object reconstruction. In: Proc. of the Conf. on Artificial Intelligence (AAAI) (2018) 3

22. Lin, C., Wang, O., Russell, B.C., Shechtman, E., Kim, V.G., Fisher, M., Lucey, S.: Photometric mesh optimization for video-aligned 3d object reconstruction. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019) 3

23. Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., Cui, Z.: DIST: rendering deep implicit signed distance function with differentiable sphere tracing. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 2, 3

24. Liu, S., Saito, S., Chen, W., Li, H.: Learning to infer implicit surfaces without 3d supervision. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) 2, 3

25. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS) (2015) 1, 3

26. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 2, 3, 6, 7, 8, 9, 10, 11, 13, 14

27. Michalkiewicz, M., Pontes, J.K., Jack, D., Baktashmotlagh, M., Eriksson, A.: Implicit surface representations as layers in neural networks. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 1, 3

28. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Occupancy flow: 4d reconstruction by learning particle dynamics. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 2, 3

29. Niemeyer, M., Mescheder, L.M., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 2, 3, 6

30. Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T., Geiger, A.: Texture fields: Learning texture representations in function space. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 2, 3

31. Park, J.J., Florence, P., Straub, J., Newcombe, R.A., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 2, 3

32. Paschalidou, D., van Gool, L., Geiger, A.: Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) 3

33. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) 7

34. Prokudin, S., Lassner, C., Romero, J.: Efficient learning on point clouds with basis point sets. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 3
35. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2017) 4
36. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems (NeurIPS) (2017) 7
37. Riegler, G., Ulusoy, A.O., Bischof, H., Geiger, A.: OctNetFusion: Learning depth fusion from data. In: Proc. of the International Conf. on 3D Vision (3DV) (2017) 3
38. Riegler, G., Ulusoy, A.O., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2017) 3
39. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2015) 6
40. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 4
41. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) 2, 3
42. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2017) 3
43. Tatarchenko, M., Richter, S.R., Ranftl, R., Li, Z., Koltun, V., Brox, T.: What do single-view 3d reconstruction networks learn? In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019) 8
44. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proc. of the European Conf. on Computer Vision (ECCV) (2018) 3
45. Wen, C., Zhang, Y., Li, Z., Fu, Y.: Pixel2mesh++: Multi-view 3d mesh generation via deformation. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 3
46. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Advances in Neural Information Processing Systems (NeurIPS) (2016) 3
47. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2015) 3
48. Xu, Q., Wang, W., Ceylan, D., Mech, R., Neumann, U.: DISN: deep implicit surface network for high-quality single-view 3d reconstruction. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) 4
49. Yang, G., Huang, X., Hao, Z., Liu, M., Belongie, S.J., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) 3
50. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Proc. of the European Conf. on Computer Vision (ECCV) (2014) 2
51. Zhang, Q., Wu, Y.N., Zhu, S.: Interpretable convolutional neural networks. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2018) 2