Holistic 3D Scene Understanding from a Single Image with Implicit Representation

Cheng Zhang[2*] Zhaopeng Cui[1*] Yinda Zhang[3*] Bing Zeng[2] Marc Pollefeys[4] Shuaicheng Liu[2†]
[1] State Key Lab of CAD & CG, Zhejiang University
[2] University of Electronic Science and Technology of China [3] Google [4]ETH Zürich

Abstract

We present a new pipeline for holistic 3D scene understanding from a single image, which could predict object shapes, object poses, and scene layout. As it is a highly ill-posed problem, existing methods usually suffer from inaccurate estimation of both shapes and layout especially for the cluttered scene due to the heavy occlusion between objects. We propose to utilize the latest deep implicit representation to solve this challenge. We not only propose an image-based local structured implicit network to improve the object shape estimation, but also refine the 3D object pose and scene layout via a novel implicit scene graph neural network that exploits the implicit local object features. A novel physical violation loss is also proposed to avoid incorrect context between objects. Extensive experiments demonstrate that our method outperforms the state-of-the-art methods in terms of object shape, scene layout estimation, and 3D object detection.

1. Introduction

3D indoor scene understanding is a long-lasting computer vision problem and has tremendous impact on several applications, e.g., robotics, virtual reality. Given a single color image, the goal is to reconstruct the room layout as well as each individual object and estimate its semantic type in the 3D space. Over decades, there are plenty of works consistently improving the performance of such a task over two focal points of the competition. One is the **3D shape representation** preserving fine-grained geometry details, evolving from the 3D bounding box, 3D volume, point cloud, to the recent triangulation mesh. The other is the joint inference of multiple objects and layout in the scene leveraging **contextual information**, such as co-occurring or relative locations among objects of multiple categories. However, the cluttered scene is a double-blade sword, which unfortunately increases the complexity of 3D scene understanding by introducing large variations in object poses and scales, and heavy occlusion. Therefore, the overall performance is still far from satisfactory.

Figure 1: Our proposed pipeline takes a single image as input, estimates layout and object poses, then reconstructs the scene with Signed Distance Function (SDF) representation.

In this work, we propose a deep learning system for holistic 3D scene understanding, which predicts and refines object shapes, object poses, and scene layout jointly with **deep implicit representation**. At first, similar to previous methods, we exploit standard Convolutional Neural Networks (CNN) to learn an initial estimation of 3D object poses, scene layout as well as 3D shapes. Different from previous methods using explicit 3D representation like volume or mesh, we utilize the local structured implicit representation of shapes motivated by [12]. Instead of taking depth images as input like [12], we design a new local implicit shape embedding network to learn the latent shape code directly from images, which can be further decoded to generate the implicit function for 3D shapes. Due to the power of implicit representation, the 3D shape of each object can be reconstructed with higher accuracy and finer surface details compared to other representations.

Then, we propose a novel graph-based scene context network to gather information from local objects, i.e., bottom-up features extracted from the initial predictions, and learn to refine the initial 3D pose and scene layout via scene context information with the implicit representation. Being one of the core topics studied in scene understanding, the context has been achieved in the era of deep learning mainly

*Equal contribution
†Corresponding author
Project webpage: https://chengzhag.github.io/publication/im3d/

from two aspects - the model architecture and the loss function. From the perspective of model design, we exploit the graph-based convolutional neural network (GCN) to learn context since it has shown competitive performance to learn context [58]. With the deep implicit representation, the learned local shape latent vectors are naturally a compact and informative feature measuring of the object geometries, which results in more effective context models compared to features extracted from other representations such as mesh.

Not only the architecture, but the deep implicit representation also benefits the context learning on the loss function. One of the most basic contextual information yet still missing in many previous works - objects should not intersect with each other, could be easily applied as supervision by penalizing the existence of 3D locations with negative predicted SDF in more than one objects[1]. We define this constraint as a novel physical violation loss and find it particularly helpful in preventing intersecting objects and producing reasonable object layouts.

Overall, our contributions are mainly in four aspects. First, we design a two-stage single image-based holistic 3D scene understanding system that could predict object shapes, object poses, and scene layout with deep implicit representation, then optimize the later two. Second, a new image-based local implicit shape embedding network is proposed to extract latent shape information which leads to superior geometry accuracy. Third, we propose a novel GCN-based scene context network to refine the object arrangement which well exploits the latent and implicit features from the initial estimation. Last but not least, we design a physical violation loss, thanks to the implicit representation, to effectively prevent the object intersection. Extensive experiments show that our model achieves the state-of-the-art performance on the standard benchmark.

2. Related works

Single Image Scene Reconstruction. As a highly ill-posed problem, single image scene reconstruction sets a high bar for learning-based algorithms, especially in a cluttered scene with heavy occlusion. The problem can be divided into layout estimation, object detection and pose estimation, and 3D object reconstruction. A simple version of the first problem is to simplify the room layout as a bounding box [19, 27, 30, 8, 38]. To detect objects and estimate poses in 3D space, Recent works [10, 21, 5] try to infer 3D bounding boxes from 2D detection by exploiting relationships among objects with a graph or physical simulation. At the same time, other works [24, 23, 22] further extend the idea to align a CAD model with similar style to each detected object. Still, the results are limited by the size of the CAD model database which results in an inaccurate representation of the scene. To tackle the above limitations of previous works, Total3D [33] is proposed as an end-to-end solution to jointly estimate the layout box and object poses while reconstructing each object from the detection and utilizing the reconstruction to supervise the pose estimation learning. However, they only exploit relationships among objects with features based on appearance and 2D geometry.

Shape Representation. In the field of computer graphics, traditional shape representation methods include mesh, voxel, and point cloud. Some of the learning-based works try to encode the shape prior into a feature vector but stick to the traditional representations by decoding the vector into mesh [17, 50, 34, 42, 14], voxel [54, 7, 3, 52, 44] or point cloud [29, 1, 57]. Others try to learn structured representations which decompose the shape into simple shapes [28, 11, 36]. Recently, implicit surface function [31, 35, 56, 39, 37, 40] has been widely used as a new representation method to overcome the disadvantages of traditional methods (i.e. unfriendly data structure to neural network of mesh and point cloud, low resolution and large memory consumption of voxel). Most recent works [13, 12, 53] try to combine the structured and implicit representation which provides a physically meaningful feature vector while introducing significant improvement on the details of the decoded shape.

Graph Convolutional Networks. Proposed by [15], graph neural networks or GCNs have been widely used to learn from graph-structured data. Inspired by convolutional neural networks, convolutional operation has been introduced to graph either on spectral domain [4, 9, 25] or non-spectral domain [18] which performs convolution with a message passing neural network to gather information from the neighboring nodes. Attention mechanism has also been introduced to GCN and has been proved to be efficient on tasks like node classification [48], scene graph generation [58] and feature matching [41]. Recently, GCN has been even used on super-resolution [59] which is usually the territory of CNN. In the 3D world which interests us most, GCN has been used on classification [51] and segmentation [46, 49, 51] on point cloud, which is usually an enemy representation to traditional neural networks. The most related application scenario of GCN with us is 3D object detection on points cloud. Recent work shows the ability of GCN to predict relationship [2] or 3D object detections [32] from point cloud data.

3. Our method

As shown in Fig. 2, the proposed system consists of two stages, i.e., the initial estimation stage, and the refinement stage. In the initial estimation stage, similar to [21, 33], a 2D detector is first adopted to extract the 2D bounding

[1]The object interior is with negative SDF, and thus no location should be inside of two objects.

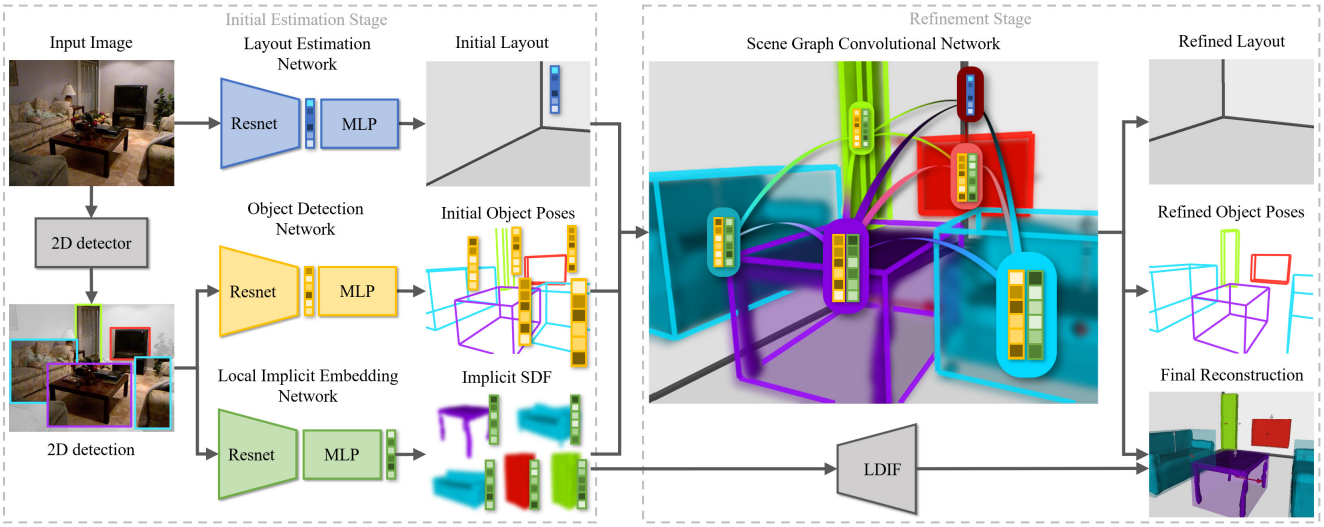


Figure 2: Our proposed pipeline. We initialize the layout estimation and 3D object poses with LEN and ODN from prior work [33], then refine them with Scene Graph Convolutional Network (SGCN). We utilize a Local Implicit Embedding Network (LIEN) to encode latent code for LDIF decoder [12] and to extract implicit features for SGCN. With the help of LDIF and marching cube algorithm, object meshes are extracted then rotated, scaled, and put into places to construct the scene.

box from the input image, followed by an Object Detection Network (ODN) to recover the object poses as 3D bounding boxes and a new Local Implicit Embedding Network (LIEN) to extract the implicit local shape information from the image directly, which can further be decoded to infer 3D geometry. The input image is also fed into a Layout Estimation Network (LEN) to produce a 3D layout bounding box and relative camera pose. In the refinement stage, a novel Scene Graph Convolutional Network (SGCN) is designed to refine the initial predictions via the scene context information. As 2D detector, LEN, ODN has the standard architecture similar to prior works [21, 33], in this section, we will describe the details of the novel SGCN and LIEN in detail. Please refer to our supplementary materials for the details of our 2D detector, LEN, ODN.

3.1. Scene Graph Convolutional Network

As shown in Fig. 2, motivated by Graph R-CNN [58], we model the whole 3D scene as a graph G, in which the nodes represent the objects, the scene layout, and their relationships. The graph is constructed starting from a complete graph with undirected edges between all objects and layout nodes, which allows information to flow among objects and the scene layout. Then, we add relation nodes to each pair of neighboring object/layout nodes. Considering the nature of directional relation [26], we add two relation nodes between each pair of neighbors in different directions.

It is well known that the input features are the key to an effective GCN [50]. For different types of nodes, we design features carefully from different sources as follows. For each node, features from different sources are flattened and concatenated into a vector, then embedded into a node representation vector with the same length using MLP.

Layout Node. We use the feature from the image encoder of LEN, which encodes the appearance of layout, and the parameterized output of layout bounding box and camera pose from LEN, as layout node features. We also concatenate the camera intrinsic parameters normalized by the image height into the feature to add camera priors.

Object Node. We collect the appearance-relationship feature [33] from ODN, and the parameterized output of object bounding box from ODN, along with the element centers in the world coordinate and analytic code from LIEN (which we will further describe in the next section). We also use the one-hot category label from the 2D detector to introduce semantic information to SGCN.

Relationship Node. For nodes connecting two different objects, the geometry feature [20, 47] of 2D object bounding boxes and the box corner coordinates of both connected objects normalized by the image height and width are used as features. The coordinates are flattened and concatenated in the order of source-destination, which differentiate the relationships of different directions. For nodes connecting objects and layouts, since the relationship is presumably different from object-object relationship, we initialize the representations with constant values, leaving the job of inferring reasonable relationship representation to SGCN.

For a graph with N objects and 1 layout, object-layout nodes and relationship nodes can then be put into two matrixes $Z^o \in \mathbb{R}^{d\times(N+1)}$ and $Z^r \in \mathbb{R}^{d\times(N+1)^2}$.

Since the graph is modeled with different types of nodes, which makes a difference in the information needed from different sources to destinations, we define independent message passing weights for each of the source-destination types. We denote the linear transformation from source node to destination node with type a and b as W^{ab}, in which

node types can be source object (or layout) s, destination object (or layout) d, and relationships r. With adjacent matrix $\alpha^{sr} = \alpha^{dr} = 1 - I_{N+1}$, the representation of object and layout nodes can be updated as

$$z_i^o = \sigma(z_{i-1}^o + \overbrace{W^{\mathtt{sd}} Z^o}^{\substack{\text{Message from} \\ \text{Layout or Objects}}} + \overbrace{W^{sr} Z^r \alpha^{sr} + W^{dr} Z^r \alpha^{dr}}^{\substack{\text{Messages from} \\ \text{Neighboring Relationships}}}), \tag{1}$$

and the relationship node representations can be updated as

$$z_i^r = \sigma(z_{i-1}^r + \underbrace{W^{rs} Z^o \alpha^{rs} + W^{rd} Z^o \alpha^{rd}}_{\text{Messages from Layout or Neighboring Objects}}), \tag{2}$$

After four steps of message passing, independent MLPs are used to decode object node representations into residuals for corresponding object bounding box parameters (δ, d, s, θ), and layout node representation into residuals for initial layout box $\left(C, s^l, \theta^l\right)$ and camera pose $\mathbf{R}\,(\beta, \gamma)$. Please refer to our supplementary or [33] for the details of the definition. The shape codes can be also refined in the scene graph, while we find that it doesn't improve empirically as much as for the layout and object poses in our pipeline because our local implicit embedding network, which will be introduced in the following, is powerful enough to learn accurate shapes.

3.2. Local Implicit Embedding Network

With a graph constructed for each scene, we naturally ask what features help SGCN effectively capture contextual information among objects. Intuitively, we expect features that well describe 3D object geometry and their relationship in 3D space. Motivated by Genova *et al*. [12], we propose to utilize the local deep implicit representation as the features embedding object shapes due to its superior performance for single object reconstruction. In their model, the function is a combination of 32 3D elements (16 with symmetry constraints), with each element described with 10 Gaussian function parameters analytic code and 32-dim latent variables (latent code). The Gaussian parameters describe the scale constant, center point, radii, and Euler angle of every Gaussian function, which contains structured information of the 3D geometry. We use analytic code as a feature for object nodes in SGCN, which should provide information on the local object structure. Furthermore, since the centers of the Gaussian functions presumably correspond to centers of different parts of an object, we also transform them from the object coordinate system to the world coordinate system as a feature for every object node in SGCN. The transformation provides global information about the scene, which makes SGCN easier to infer relationships between objects. The above two features make up the implicit features of LIEN.

As LDIF [12] is designed for 3D object reconstruction from one or multiple depth images, we design a new image-based Local Implicit Embedding Network (LIEN) to learn the 3D latent shape representation from the image which is obviously a more challenging problem. Our LIEN consists of a Resnet-18 as image encoder, along with a three-layer MLP to get the analytic and latent code. Additionally, in order to learn the latent features effectively, we concatenate the category code with the image feature from the encoder to introduce shape priors to the LIEN, which improves the performance greatly. Please refer to our supplementary material for the detailed architecture of the proposed LIEN.

3.3. Loss Function

Losses for Initialization Modules. When training LIEN along with LDIF decoder individually, we follow [12] to use the shape element center loss $\mathcal{L}_c$ with weight λ_c and point sample loss,

$$\mathcal{L}_p = \lambda_{ns}\mathcal{L}_{ns} + \lambda_{us}\mathcal{L}_{us}, \tag{3}$$

where $\mathcal{L}_{ns}$ and $\mathcal{L}_{us}$ evaluates L_2 losses for near-surface samples and uniformly sampled points. When training LEN and ODN, we follow [21, 33] to use classification and regression loss for every output parameter of LEN and ODN,

$$\mathcal{L}_{LEN} = \sum_{y \in \{\beta, \gamma, C, s^l, \theta^l\}} \lambda_y \mathcal{L}_y, \tag{4}$$

$$\mathcal{L}_{ODN} = \sum_{x \in \{\delta, d, s, \theta\}} \lambda_x \mathcal{L}_x. \tag{5}$$

Joint Refinement with Object Physical Violation Loss. For the refinement stage, we aim to optimize the scene layout and object poses using the scene context information by minimizing the following loss function,

$$\mathcal{L}_j = \mathcal{L}_{LEN} + \mathcal{L}_{ODN} + \lambda_{co}\mathcal{L}_{co} + \lambda_{phy}\mathcal{L}_{phy}. \tag{6}$$

Besides $\mathcal{L}_{LEN}$, $\mathcal{L}_{ODN}$ and cooperative loss $\mathcal{L}_{co}$ [33], we propose a novel physical violation loss as a part of joint loss for the scene graph convolutional network to make sure that objects will not intersect with each other. The neural SDF representation used by local implicit representation gives us a convenient way to propagate gradient from undesired geometry intersection back to the object pose estimation. To achieve this, we first sample points inside objects. For each object i, we randomly sample points inside the bounding box of each object, along with the center points of Gaussian elements as point candidates. We then queue these candidates into LDIF decoder of the object and filter out points outside object surfaces to get inside point samples $\mathbb{S}_i$. Finally, we queue $\mathbb{S}_i$ into the LDIF decoder of the k-nearest objects $\mathbb{N}_i$ to verify if they have intersection with other objects (if the predicted label is "inside"). We follow [12] to compute a L_2 loss between the predicted labels of intersected points with the ground truth surface label (where we

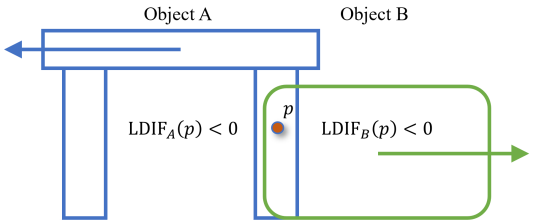


Figure 3: Object physical violation loss. Based on the insight that objects should not intersect, we punish points inside neighboring objects (demonstrated as p, which has negative LDIF values in both object A and object B). With error back-propagated through the LDIF decoder, intersected objects should be pushed back from each other, reducing intersection resulting from bad object pose estimation.

use 1, 0, 0.5 for "outside", "inside", "surface" labels). The object physical violation loss can be defined as:

$$\mathcal{L}_{phy} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathbb{S}_i|}\sum_{\mathbf{x}\in\mathbb{S}_i}\|\texttt{relu}(0.5 - \texttt{sig}(\alpha \text{LDIF}_i(\mathbf{x})))\|, \tag{7}$$

where $\text{LDIF}_i(\mathbf{x})$ is the LDIF for object i to decode a world coordinate point $\mathbf{x}$ into LDIF value. A sigmoid is applied on the LDIF value (scaled by α) to get the predicted labels, and a ReLU is applied to consider only the intersected points. As shown in Fig. 3, the loss punishes intersected sample points thus push both objects away from each other to prevent intersections.

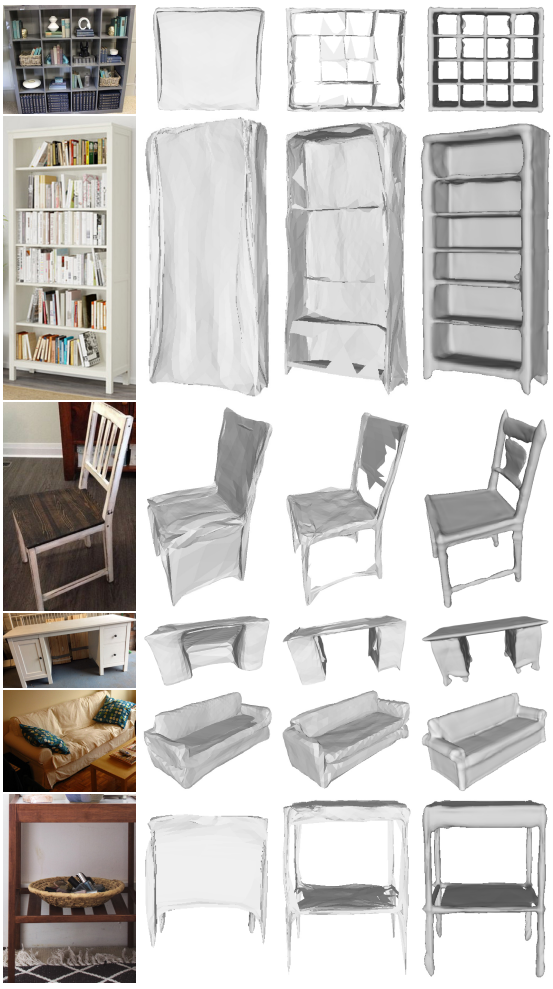
4. Experiments

In this section, we compare our method with state-of-the-art 3D scene understanding methods in various aspects and provide an ablation study to highlight the effectiveness of major components.

4.1. Experiment Setup

Datasets. We follow [33] to use two datasets to train each module individually and jointly. We use two datasets for training and evaluation. 1) **Pix3D** dataset [45] is presented as a benchmark for shape-related tasks including reconstruction, providing 9 categories of 395 furniture models and 10,069 images with precise alignment. We use the mesh fusion pipeline from Occupancy Network [31] to get watertight meshes for LIEN training and evaluate LIEN on original meshes. 2) **SUN RGB-D** dataset [43] contains 10K RGB-D indoor images captured by four different sensors and is densely annotated with 2D segmentation, semantic labels, 3D room layout, and 3D bounding boxes with object orientations. Follow Total3D [33], we use the train/test split from [14] on the Pix3D dataset and the official train/test split on the SUN RGB-D dataset. The object labels are mapped from NYU-37 to Pix3D as presented by [33].

Metrics. We adopt the same evaluation metrics with [21, 33], including average 3D Intersection over Union (IoU) for layout estimation; mean absolute error for camera pose; average precision (AP) for object detection; and chamfer distance for single-object mesh generation from single image.



(a) Input (b) AtlasNet (c) MGN (d) Ours

Figure 4: Object reconstruction qualitative comparison. We use the implementation from [33] for AtlasNet [16]. Our results contain finer details and have more smooth surfaces.

Implementation. We use the outputs of the 2D detector from Total3D as the input of our model. We also adopted the same structure of ODN and LEN from Total3D. LIEN is trained with LDIF decoder on Pix3D with watertight mesh, using Adam optimizer with a batch size of 24 and learning rate decaying from 2e-4 (scaled by 0.5 if the test loss stops decreasing for 50 epochs, 400 epochs in total) and evaluated on the original non-watertight mesh. SGCN is trained on SUN RGB-D, using Adam optimizer with a batch size of 2 and learning rate decaying from 1e-4 (scaled by 0.5 every 5 epochs after epoch 18, 30 epochs in total). We follow [33] to train each module individually then jointly. When training SGCN individually, we use $\mathcal{L}_j$ without $\mathcal{L}_{phy}$, and put it into the full model with pre-trained weights of other modules. In joint training, we adopt the observation from

Category	bed	bookcase	chair	desk	sofa	table	tool	wardrobe	misc	mean
AtlasNet [16]	9.03	6.91	8.37	8.59	6.24	19.46	6.95	4.78	40.05	12.26
TMN [34]	7.78	5.93	6.86	7.08	4.25	17.42	4.13	4.09	**23.68**	9.03
MGN [33]	5.99	6.56	**5.32**	**5.93**	**3.36**	14.19	3.12	**3.83**	26.93	8.36
Ours	**4.11**	**3.96**	5.45	7.85	5.61	**11.73**	**2.39**	4.31	24.65	**6.72**

Table 1: Object reconstruction comparison. We report the Chamfer distance scaled with the factor of 10^3. We follow [33] to align the reconstructed mesh to ground-truth with ICP then sample 10K points from the output and the ground-truth meshes. Although trained on watertight meshes with more noise, our results still shows better results.

Method	bed	chair	sofa	table	desk	dresser	nightstand	sink	cabinet	lamp	mAP
3DGP [6]	5.62	2.31	3.24	1.23	-	-	-	-	-	-	-
HoPR [22]	58.29	13.56	28.37	12.12	4.79	13.71	8.80	2.18	0.48	2.41	14.47
CooP [21]	57.71	15.21	36.67	31.16	19.90	15.98	11.36	15.95	10.47	3.28	21.77
Total3D [33]	60.65	17.55	44.90	36.48	27.93	21.19	17.01	18.50	14.51	5.04	26.38
Ours	**89.32**	**35.14**	**69.10**	**57.37**	**49.03**	**29.27**	**41.34**	**33.81**	**33.93**	**11.90**	**45.21**

Table 2: 3D object detection comparison. For CooP, we report the better results from [33] trained on NYU-37 object labels. Our method outperforms SOTA, benefiting from a better understanding of the object relationships and the scene context.

[33] that object reconstruction depends on clean mesh for supervision, to fix the weights of LIEN and LDIF decoder.

4.2. Comparison to State-of-the-art

In this section, we compare to the state-of-the-art methods for holistic scene understand from aspects including object reconstruction, 3D object detection, layout estimation, camera pose prediction, and scene mesh reconstruction.

3D Object Reconstruction. We first compare the performance of LIEN with previous methods, including AtlasNet [16], TMN [34], and Total3D [33], for the accuracy of the predicted geometry on Pix3D dataset. All the methods take as input a crop of image of the object and produce 3D geometry. To make a fair comparison, the one-hot object category code is also concatenated with the appearance feature for AtlasNet [16] and TMN [34]. For our method, we run a marching cube algorithm on 256 resolution to reconstruct the mesh. The quantitative comparison is shown in Table 1. Our method produces the most accurate 3D shape compared to other methods, yielding the lowest mean Chamfer Distance across all categories. Qualitative results are shown in Fig. 4. AtlasNet produces results in limited topology and thus generates many undesired surfaces. MGN mitigates the issue with the capability of topology modification, which improves the results but still leaves obvious artifacts and unsmooth surface due to the limited representation capacity of the triangular mesh. In contrast, our method produces 3D shape with correct topology, smooth surface, and fine-grained details, which clearly shows the advantage of the deep implicit representation.

3D Object Detection. We then evaluate the 3D object detection performance of our model. Follow [33, 21], we use mean average precision (mAP) with the threshold of 3D bounding box IoU set at 0.15 as the evaluation metric. The quantitative comparison to state-of-the-art methods [6, 22, 21, 33] is shown in Table 2. Our method performs consistently the best over all semantic categories and significantly outperforms the state-of-the-art (i.e. improving AP by 18.83%). Figure 5 shows some qualitative comparison. Note how our method produces object layout not only more accurate but also in reasonable context compared to Total3D, e.g. objects are parallel to wall direction.

Method	Layout IoU	Cam pitch	Cam roll
3DGP [6]	19.2	-	-
Hedau [19]	-	33.85	3.45
HoPR [22]	54.9	7.60	3.12
CooP [21]	56.9	3.28	2.19
Total3D [33]	59.2	3.15	**2.09**
Ours	**64.4**	**2.98**	2.11

Table 3: 3D layout and camera pose estimation comparison. Our method outperforms SOTA by 5.2% in layout estimation while on par with SOTA on camera pose estimation.

Layout Estimation. We also compare the 3D room layout estimation with Total3D [33] and other state-of-the-arts [6, 22, 21]. The quantitative evaluation is shown in Table 3 (Layout IoU). Overall, our method outperforms all the baseline methods. This indicates that the GCN is effective in measuring the relation between layout and objects and thus benefits the layout prediction.

Camera Pose Estimation. Table 3 also shows the comparison over camera pose prediction, following the evaluation protocol of Total3D. Our method achieves 5% better camera pitch and slightly worse camera roll.

Holistic Scene Reconstruction. To our best knowledge, Total3D [33] is the only work achieving holistic scene reconstruction from a single RGB, and thus we compare to it. Since no ground truth is presented in SUN RGB-D dataset, we mainly show qualitative comparison in Fig. 5. Compared to Total3D, our model has less intersection and estimates more reasonable object layout and direction. We consider this as a benefit from a better understanding of scene context by GCN. Our proposed physical violation loss $\mathcal{L}_{phy}$ also contributes to less intersection.

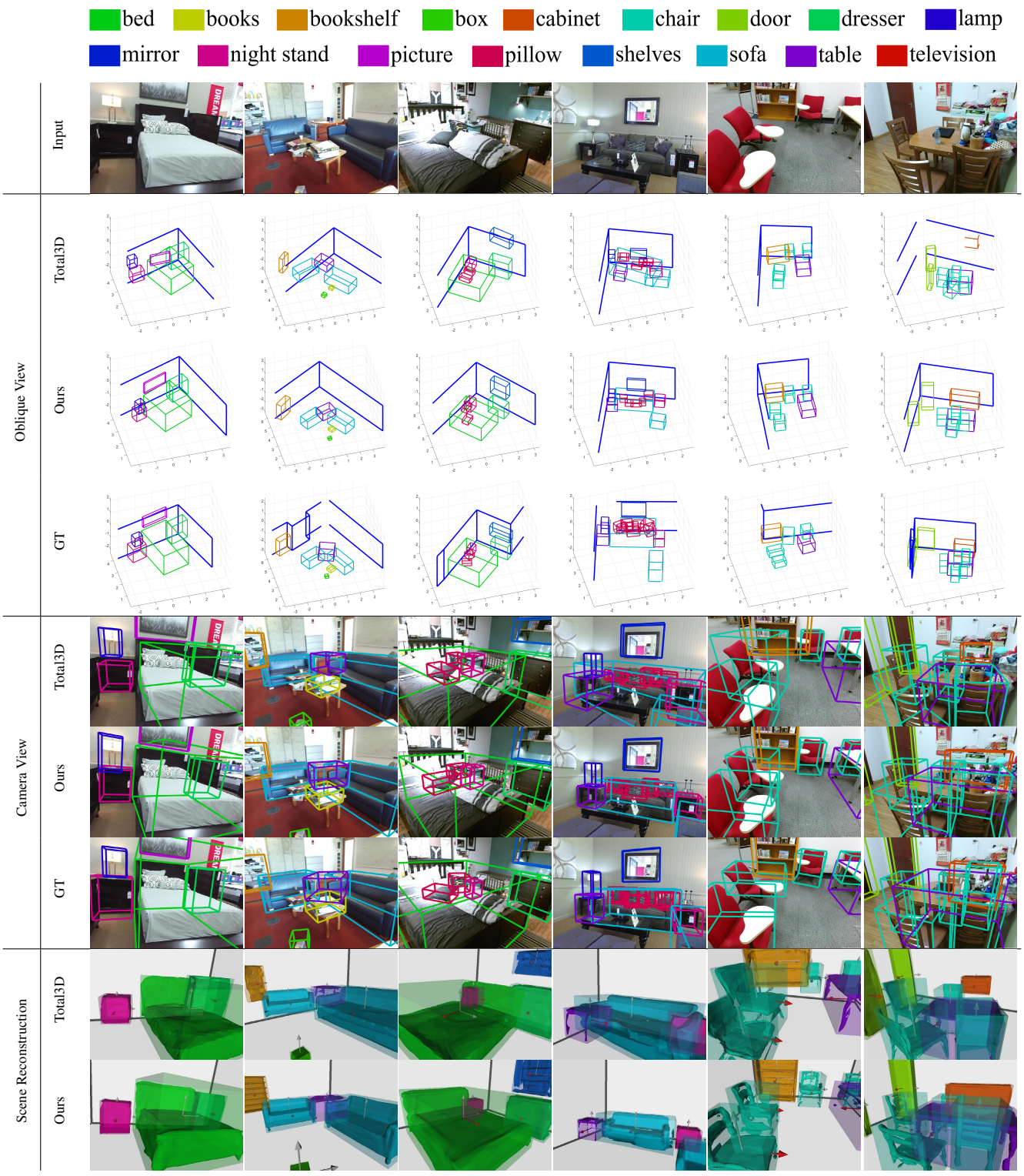


Figure 5: Qualitative comparison on object detection and scene reconstruction. We compare object detection results with Total3D [33] and ground truth in both oblique view and camera view. The results show that our method gives more accurate bounding box estimation and with less intersection. We compare scene reconstruction results with Total3D in camera view and observe more reasonable object poses.

4.3. Ablation Study

In this section, we verify the effectiveness of the proposed components for holistic scene understanding. As shown in Table 4, we disable certain components and evaluate the model for 3D layout estimation and 3D object detection, We do not evaluate the 3D object reconstruction since it is highly related to the usage of deep implicit representation, which has been already evaluated in Section 4.2.

Does GCN Matter? To show the effectiveness of GCN, we first attach the GCN to the original Total3D to improve the object and scene layout (Table 4, Total3D+GCN). For the difference between MGN of Total3D and LIEN of ours, we replace deep implicit features with the feature from image encoder of MGN and use their proposed partial Chamfer loss $\mathcal{L}_g$ instead of $\mathcal{L}_{phy}$. Both object bounding box and scene layout are improved. We also train a version of our model without the GCN (Ours-GCN), and the performance drops significantly. Both experiments show that GCN is effective in capturing scene context.

Does Deep Implicit Feature Matter? As introduced in Section 3.2, the LDIF representation provides informative node features for the GCN. Here we demonstrate the contribution from each component of the latent representation. Particularly, we remove either element centers or analytic code from the GCN node feature (Ours-element, Ours-analytic), and find both hurts the performance. This indicates that the complete latent representation is helpful in pursuing better scene understanding performance.

Does Physical Violation Loss Matter? Additionally, we evaluate the effectiveness of the physical violation loss. We train our model without it (Ours-$\mathcal{L}_{phy}$), and also observe performance drop for both scene layout and object 3D bounding box in Table 4. We refer to supplementary material for qualitative comparison.

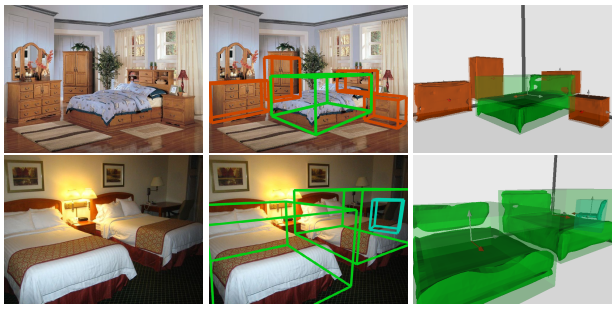
Evaluating on Other Metrics. We also test our method in other aspects including supporting relation, geometry accuracy, and room layout as shown in Table 5. 1) We calculate the mean distance between the predicted bottom of on-floor objects and the ground truth floor to measure the supporting relationship. As ground truth, an object is considered to be on-floor if its bottom surface is within 15cm to the floor. While GCN significantly improves the metric, $\mathcal{L}_{phy}$ slightly hurts possibly because it tends to push objects away. Further qualitative results are shown in the supplementary material. Besides, we also measure the average volume of the collision per scene between objects (Coll Vol), and our full model effectively prevent collision. 2) We follow Total3D [33] to evaluate the alignment between scene reconstruction and ground truth depth map with global loss $\mathcal{L}_g$, and our full model performs the best. 3) We also project the predicted layout onto the image and evaluate with image based metrics [8, 38]. Our full model achieves the best on both corner and pixel errors. Overall, the GCN and $\mathcal{L}_{phy}$ benefit on all these aspects.

Setting	Layout IoU (↑)	Detection mAP (↑)
Total3D	59.25	26.38
Total3D+GCN	62.49	37.04
Ours-GCN	60.04	27.47
Ours-element	64.22	42.05
Ours-analytic	63.76	43.10
Ours-$\mathcal{L}_{phy}$	63.52	43.33
Full	**64.41**	**45.21**

Table 4: Ablation study. We evaluate layout estimation with layout IoU and 3D object detection with mAP.

Setting	Sup Err (cm)	$\mathcal{L}_g$	Coll Vol ($dm^3/scene$)	Corner Err (%)	Pixel Err (%)
Total3D	26.72	1.43	-	13.29	20.51
Ours-GCN	24.18	1.41	16.64	13.17	20.05
Ours-$\mathcal{L}_{phy}$	**13.35**	1.14	13.65	11.60	17.91
Full	14.71	**1.11**	**13.55**	**11.45**	**17.60**

Table 5: Ablation study on other metrics. We compare on supporting error, $\mathcal{L}_g$ (in units of 10^{-2}), average collision volume, corner error, and pixel error.



(a) Input (b) 3D Detection (c) Reconstruction

Figure 6: Qualitative results on ObjectNet3D dataset [55] (row 1) and the layout estimation dataset in [19] (row 2).

4.4. Generalization to other datasets

We also show qualitative results of our method tested on the 3D detection dataset ObjectNet3D [55] and the layout estimation dataset in [19] without fine-tuning in Fig. 6. Our method shows good generalization capability and performs reasonably well on these unseen datasets.

5. Conclusion

We have presented a deep learning model for holistic scene understanding by leveraging deep implicit representation. Our model not only reconstructs accurate 3D object geometry, but also learns better scene context using GCN and a novel physical violation loss, which can deliver accurate scene and object layout. Extensive experiments show that our model improves various tasks in holistic scene understanding over existing methods. A promising future direction could be exploiting object functionalities for better 3D scene understanding.

Acknowledgement: This research was supported by National Natural Science Foundation of China (NSFC) under grants No.61872067 and No.61720106004.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.

[2] Armen Avetisyan, Tatiana Khanova, Christopher Choy, Denver Dash, Angela Dai, and Matthias Nießner. Scenecad: Predicting object alignments and layouts in rgb-d scans. *arXiv preprint arXiv:2003.12622*, 2020.

[3] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016.

[4] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[5] Yixin Chen, Siyuan Huang, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense. *arXiv preprint arXiv:1909.01507*, 2019.

[6] Wongun Choi, Yu-Wei Chao, Caroline Pantofaru, and Silvio Savarese. Understanding indoor scenes using 3d geometric phrases. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013.

[7] 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 *Eur. Conf. Comput. Vis.*, pages 628–644. Springer, 2016.

[8] Saumitro Dasgupta, Kuan Fang, Kevin Chen, and Silvio Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Adv. Neural Inform. Process. Syst.*, pages 3844–3852, 2016.

[10] Yilun Du, Zhijian Liu, Hector Basevi, Ales Leonardis, Bill Freeman, Josh Tenenbaum, and Jiajun Wu. Learning to exploit stability for 3d scene parsing. In *Adv. Neural Inform. Process. Syst.*, 2018.

[11] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. Sdm-net: Deep generative network for structured deformable mesh. *ACM Trans. Graph.*, 38(6):1–15, 2019.

[12] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4857–4866, 2020.

[13] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Int. Conf. Comput. Vis.*, pages 7154–7164, 2019.

[14] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Int. Conf. Comput. Vis.*, pages 9785–9795, 2019.

[15] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[16] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[17] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 216–224, 2018.

[18] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Adv. Neural Inform. Process. Syst.*, pages 1024–1034, 2017.

[19] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *Int. Conf. Comput. Vis.*, 2009.

[20] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[21] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation. In *Adv. Neural Inform. Process. Syst.*, 2018.

[22] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *Eur. Conf. Comput. Vis.*, 2018.

[23] Moos Hueting, Pradyumna Reddy, Vladimir Kim, Ersin Yumer, Nathan Carr, and Niloy Mitra. Seethrough: finding chairs in heavily occluded indoor scene images. *arXiv preprint arXiv:1710.10473*, 2017.

[24] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[25] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[26] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73, 2017.

[27] David C Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009.

[28] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Trans. Graph.*, 36(4):1–14, 2017.

[29] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. *arXiv preprint arXiv:1706.07036*, 2017.

[30] Arun Mallya and Svetlana Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *Int. Conf. Comput. Vis.*, 2015.

[31] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks:

Learning 3d reconstruction in function space. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4460–4470, 2019.

[32] Mahyar Najibi, Guangda Lai, Abhijit Kundu, Zhichao Lu, Vivek Rathod, Thomas Funkhouser, Caroline Pantofaru, David Ross, Larry S Davis, and Alireza Fathi. Dops: Learning to detect 3d objects and predict their 3d shapes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11913–11922, 2020.

[33] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[34] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Int. Conf. Comput. Vis.*, pages 9964–9973, 2019.

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *arXiv preprint arXiv:1901.05103*, 2019.

[36] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10344–10353, 2019.

[37] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *arXiv preprint arXiv:2003.04618*, 2020.

[38] Yuzhuo Ren, Shangwen Li, Chen Chen, and C-C Jay Kuo. A coarse-to-fine indoor layout estimation (cfile) method. In *ACCV*, 2016.

[39] 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 *Int. Conf. Comput. Vis.*, pages 2304–2314, 2019.

[40] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 84–93, 2020.

[41] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4938–4947, 2020.

[42] Edward J Smith, Scott Fujimoto, Adriana Romero, and David Meger. Geometrics: Exploiting geometric structure for graph-encoded objects. *arXiv preprint arXiv:1901.11461*, 2019.

[43] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[44] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1955–1964, 2018.

[45] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2974–2983, 2018.

[46] Gusi Te, Wei Hu, Amin Zheng, and Zongming Guo. Rgcnn: Regularized graph cnn for point cloud segmentation. In *ACM Int. Conf. Multimedia*, pages 746–754, 2018.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017.

[48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[49] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10296–10305, 2019.

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

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

[52] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Adv. Neural Inform. Process. Syst.*, pages 82–90, 2016.

[53] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 829–838, 2020.

[54] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1912–1920, 2015.

[55] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *Eur. Conf. Comput. Vis.*, pages 160–176. Springer, 2016.

[56] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711*, 2019.

[57] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Int. Conf. Comput. Vis.*, pages 4541–4550, 2019.

[58] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Eur. Conf. Comput. Vis.*, pages 670–685, 2018.

[59] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. *Adv. Neural Inform. Process. Syst.*, 33, 2020.