SceneCAD: Predicting Object Alignments and Layouts in RGB-D Scans

Armen Avetisyan[1], Tatiana Khanova[3], Christopher Choy[2], Denver Dash[3], Angela Dai[1], and Matthias Nießner[1]

[1] Technical University of Munich [2] Stanford University [3] Occipital Inc.

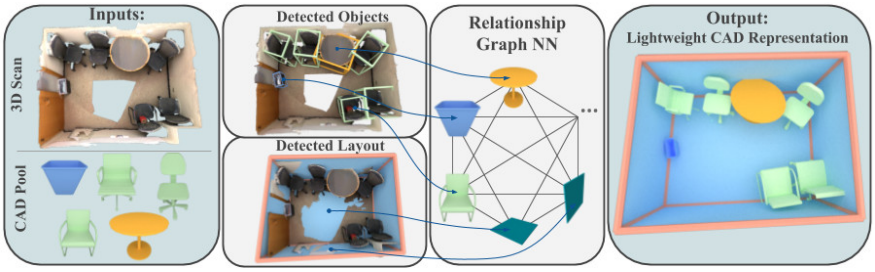

Fig. 1. Our method takes as input a 3D scan and a set of CAD models. We jointly detect objects and layout elements in the scene. Each detected object or layout component then forms a node in a graph neural network which estimates object-object relationships and object-layout relationships. This holistic understanding of the scene enables results in a lightweight CAD-based representation of the scene.

Abstract. We present a novel approach to reconstructing lightweight, CAD-based representations of scanned 3D environments from commodity RGB-D sensors. Our key idea is to jointly optimize for both CAD model alignments as well as layout estimations of the scanned scene, explicitly modeling inter-relationships between objects-to-objects and objects-to-layout. Since object arrangement and scene layout are intrinsically coupled, we show that treating the problem jointly significantly helps to produce globally-consistent representations of a scene. Object CAD models are aligned to the scene by establishing dense correspondences between geometry, and we introduce a hierarchical layout prediction approach to estimate layout planes from corners and edges of the scene. To this end, we propose a message-passing graph neural network to model the inter-relationships between objects and layout, guiding generation of a globally object alignment in a scene. By considering the global scene layout, we achieve significantly improved CAD alignments compared to state-of-the-art methods, improving from 41.83% to 58.41% alignment accuracy on SUNCG and from 50.05% to 61.24% on ScanNet, respectively. The resulting CAD-based representations makes our method well-suited for applications in content creation such as augmented- or virtual reality.

1 Introduction

The recent progress of 3D reconstruction of real-world environments from commodity range sensors has spurred interest in using such captured 3D data for applications across many fields, such as content creation, mixed reality, or robotics. State-of-the-art 3D reconstruction approaches can now produce impressively-robust camera tracking and surface reconstruction [29, 30, 7, 11].

Unfortunately, the resulting 3D reconstructions are not well-suited for direct use with many applications, as the geometric reconstructions remain incomplete (e.g., due to occlusions and sensor limitations), are often noisy or oversmoothed, and often consume a large memory footprint due to high density of triangles or points used to represent a surface at high resolution. There still remains a notable gap between these reconstructions and artist-modeled 3D content, which are clean, complete, and lightweight [16].

Inspired by these attributes of artist-created 3D content, we aim to construct a CAD-based scene representation of an input RGB-D scan, with objects represented by individual CAD models and scene layout represented by lightweight meshes. In contrast to previous approaches which have individually tackled the tasks of CAD model alignment [22, 2, 3] and of layout estimation [28, 25, 6], we observe that object arrangement is typically tightly correlated with the scene layout. We thus propose to jointly optimize for CAD model alignment and scene layout to produce a globally-consistent CAD-based representation of the scene.

From an input RGB-D scan along with a CAD model pool, we align CAD models to the scanned scene by establishing dense correspondences. To estimate the scene layout, we characterize the layout into planar elements, and propose a hierarchical layout prediction by first detecting corner locations, then predicting scene edges, and from sets of edges potentially presenting a layout plane, predicting the final layout. We then propose a graph neural network architecture for optimizing the relationships between objects and layout by predicting object-object relative poses as well as object-layout support relationships. This optimization guides both object and layout arrangement to be consistent with each other. Our approach is fully-convolutional and trained end-to-end, generating a CAD-based scene representation of a scan in a single forward pass.

In summary, we present the following contributions:

- We formulate a lightweight heuristic-free 3D layout prediction algorithm that hierarchically predicts corners, edges and then planes in an end-to-end fashion consisting of only $\approx 1M$ trainable parameters generating satisfactory layouts without the need for extensive heuristics.
- We present a scene graph network that learns relationships between objects and scene layout, enabling globally consistent CAD model alignments and results in a significant increase in prediction performance in both synthetic as well as real-world datasets.
- We introduce a new richly-annotated real-world scene layout dataset consisting of 1151 CAD shells and wireframes on top of the ScanNet RGB-D dataset, allowing large-scale data-driven training for layout estimation.

2 Related Work

CAD model alignment Aligning an expert-generated 3D model or a 3D template to 3D scan data has been studied widely due to its wide range of applications, for instance motion capture [4], 3D object detection and localization [12, 13, 39], and scene registration [35]. Our aim is to leverage large-scale datasets of CAD models to reconstruct a lightweight, semantically-informed, high-quality CAD representation of an RGB-D scan of a scene. Several approaches have been developed to retrieve and align CAD models from a shape database and align them in real time to a scan during the 3D scanning process [19, 22], although their use of handcrafted features for geometric scan-to-CAD matching limit robustness.

Zeng et al. [40] developed a learned feature extractor using a siamese network design for geometric feature matching, which can be employed for scan-to-CAD feature matching, though this remains difficult due to the domain gap between synthetic CAD models and real-world scans. Avetisyan et al. [2] proposed a scan-to-CAD retrieval and alignment approach leveraging learned features to detect objects in a 3D scan and establish correspondences across the domain gap of scan and CAD. They later built upon this work to develop a fully end-to-end trainable approach for this CAD alignment task [3]. For such approaches, each object is considered independently, whereas our approach exploits contextual information from object-object and object-layout to produce globally consistent CAD model alignment and layout estimation.

Other approaches retrieve and align CAD models to RGB images [23, 38, 33]; our work instead focuses on geometric alignment of CAD models and layout.

Graph neural networks and relational inference in 3D. Recent developments in graph inference and graph neural networks have shown significant promise for inference on 3D data. Recently, various approaches have viewed 3D meshes as graphs in order find correspondences between 3D shapes [5], deform a template mesh to fit an image observation of a shape [36], or generate a mesh model of an object [10], among other applications. Learning on graphs has also shown promise for estimating higher-level relational information in scenes, as a scene graph. 3D-RelNet [21] predicts 3D shapes and poses from single RGB images and establish pairwise pose constraints between objects to improve overall prediction quality. Our approach is similarly inspired to establish relationships between objects; we additionally employ relationships between objects and structural components (i.e., walls, floors, and ceilings), which considerably inform object arrangement. Armeni et al. [1] propose a unified hierarchical structure that hosts building, room, and object relationships into one 3D scene graph. They leverage this graph structure to generate scene graphs from 2D images. Our approach focuses on leveraging relational information to reconstruct imperfect scans with a CAD-based representation for each object and layout element.

Layout estimation. Various layout estimation approaches have been developed to infer structural information from RGB and RGB-D data. Scan2BIM [28] generates building information models (BIM) from 3D scans by detecting planes and finding

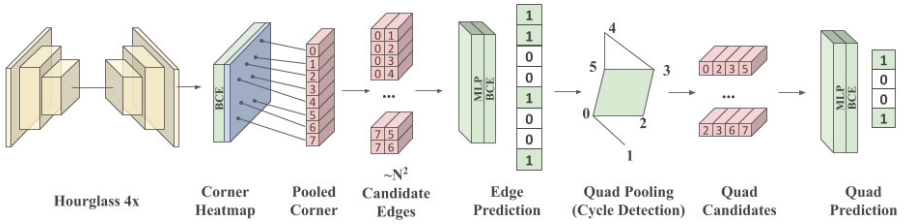

Fig. 2. Layout estimation as planar quad structures. Layout components are characterized as planar elements which are detected hierarchically. From an input scene, corners of these layout elements are predicted in heatmap fashion leveraging non-maximum suppression. From these predicted corners, edges are then predicted for each possible pair of corners as a binary classification task. From the predicted edge candidates, valid quads of four connected edges are considered as candidate layout elements, with a binary classification used to produce the final layout prediction.

plausible intersections to produce room-level segmentation of floors, ceilings and walls under Manhattan-style constraints. PlaneRCNN [24] and PlaneNet [26] propose deep neural network architectures to detect planes from RGB images and estimate their 3D parameters. FloorNet [25] estimates a 2D Manhattan-style floorplan representation for an input RGB-D scan using a point-based neural network architecture. Floor-SP [6] relaxes the Manhattan constraints with an integer programming formulation, and produces more robust floorplan estimation. In contrast to these layout estimation approaches, our focus lies in leveraging global scene relations between objects as well as structural elements in order to produce a CAD-based representation of the scene.

Single view 3D reconstruction. Holistic 3D Scene Parsing [18] parses a single RGB image and reconstruct a holistic 3D arrangements of CAD models jointly optimizing for 3D object detection, scene layout and hidden human context. Zou et al. [41] infers a complete interpretation of the scene from a single RGBD frame where objects and scene layout are predicted in data-driven fashion. In contrast to single view reconstruction, our approach aims towards holistic scene understanding that can operate on large-scale 3D scenes while consuming only a few seconds of runtime at test time.

3 SceneCAD: Joint Object Alignment and Layout Estimation

The input scan is represented as a sparse 3D voxel grid of the occupied surface geometry carrying fused RGB data. The scan is first encoded by a series of sparse 3D convolutional layers [8] to produce a feature volume F'. The sparse output F' is then densified into a dense 3D feature grid $F \in \mathbb{R}^{N_f \times N_x \times N_y \times N_z}$ where N_F is the number of channels in the feature and N_x, N_y, and N_z are the resolution

of the feature along x, y, and z axis respectively. Note that the encoder serves as backbone for proceeding modules. Hence, F is the input to the CAD alignment module as well as the layout estimation module.

Based on F, we detect objects along with their bounding box in the object detection module and layout planes in the layout detection module. We then establish our relational inference by formulating a message-passing graph neural network on the predicted objects and layout planes, where each node represents an object or layout plane, with losses on edge relationships representing relative poses and support. Finally, we predict a set of retrieved CAD models along with their 9-DoF poses (3 translation, 3 rotation, and 3 scale) for every detected object.

The message-passing graph neural network helps to inform objects of both relations between other objects as well as with the scene layout, e.g., certain types of furniture such as beds and chairs are typically directly supported by a floor, chairs near a table often face the table. This joint optimization thus helps to enable globally consistent CAD model alignment in the final output.

3.1 Layout Prediction

The indoor scene of interest in our problem consists of planar or quadrilateral components such as walls, floors, and ceilings. However, some of these planar elements create complex geometry such as bars, beams, or other structures that effectively make template-matching approach to find the room layout challenging. Thus, we propose a bottom-up approach that predicts corners, edges, and planar elements sequentially to predict the room layout. Our layout prediction pipeline is structured hierarchically: first predicting the corner locations, then predicting edges between the corners, and finally extracting quads from the predicted edges. We visualize the overview of the pipeline on Figure 2.

Corners are predicted by a convolutional network that decodes F to its original dimension by predicting a heatmap; i.e. a voxel-wise score that indicates a *cornerness* likeliness. The loss for this predicted heatmap is a voxel-wise binary cross-entropy classification loss in conjunction with a softmax and a negative log-likelihood over the entire voxel grid where the problem is formulated as a spatial multi-class problem. This is structured as an encoder-decoder, where the bottleneck lies at a spatial reduction of $4\times$. Note that we make predictions for corners which have not been observed in the input scan (e.g., due to occlusions, c.f.). See supplemental material for a visual illustration of the layout prediction pipeline. From the output corner heatmap, we apply a non-maximum suppression to filter out weak responses, and define the final corner predictions as a set of xyz coordinates $\mathcal{V} = \{\mathbf{v}_i\}_i$, $\mathbf{v}_i = [x_i, y_i, z_i]$.

We the predict the layout edges from the predicted corners $\mathcal{V}$. We construct the candidate set of edges by taking all pair-wise combinations of corners $\mathbf{e}_{ij} = (\mathbf{v}_i, \mathbf{v}_j)$ for all $i \in [1, ..., |\mathcal{V}|]$ and $j \in [1, ..., i-1]$. We denote all edges as $\mathcal{E} = \{\mathbf{e}_{ij}\}_{ij}$. From the pool of candidate edges we predict a set of edges that belongs to the scene structure using a graph neural network. Specifically, for each potential edge $\mathbf{e}_{ij} = (\mathbf{v}_i, \mathbf{v}_j)$, we extract corresponding features from the vertex prediction

convolutional network, $F[\mathbf{v}_i], F[\mathbf{v}_j]$ where $F[\cdot]$ denotes the feature vector at the specified x, y, z coordinate. We concatenate these features along with the normalized coordinates to form an input feature vector for each edge $\mathbf{f}_{\mathbf{e}_{ij}} = [F[\mathbf{v}_i], F[\mathbf{v}_j], \mathbf{N}(\mathbf{v}_i), \mathbf{N}(\mathbf{v}_j)]$. For each edge we construct two feature descriptors with alternating order of corner features $\mathbf{f}_{\mathbf{e}_{ji}}$ to mitigate the effect of order dependency. We feed these concatenated features into a graph network, which we train with edge-wise binary cross entropy loss against ground truth edges. As the vertex predictions have uncertainty, we label edges with predicted vertices within a certain radius from the ground truth layout vertices to be positives. This edge prediction limits the set of candidate layout quads which would otherwise be $O\left(\binom{|\mathcal{V}|}{4}\right)$.

From these predicted edges, we then compute the set of candidate layout quads as the set of planar, valid 4-cycles within these edges $\mathbf{q}_{ijkl} = \{\mathbf{e}_{ij}, \mathbf{e}_{jk}, \mathbf{e}_{kl}, \mathbf{e}_{li}\}$. To detect valid cycles, we use the depth-first-search cycle detection algorithm We predict the final set of layout quads as either positive or negative where the positive predictions constitute the scene layout, decomposed as quads. The feature descriptor for a candidate quad is constructed by concatenating the features from F corresponding to the corner locations of its vertices and normalized corner locations, $\mathbf{q}_{ijkl} = [F[\mathbf{v}_i], F[\mathbf{v}_j], F[\mathbf{v}_k], F[\mathbf{v}_l], \mathbf{N}(\mathbf{v}_i), \mathbf{N}(\mathbf{v}_j), \mathbf{N}(\mathbf{v}_k), \mathbf{N}(\mathbf{v}_l)]$. Similar to the edge features, every quad feature descriptor is 4-way permuted $\mathbf{q}_{jkli}, \mathbf{q}_{klij}$, and $\mathbf{q}_{lijk}$ in order to mitigate order-dependency. This feature is input to an MLP followed by a binary cross entropy loss. From these predicted quads, we recover the scene layouts without heuristic post-processing.

3.2 CAD Model Alignment

Along with the room layout, we aim to find and align light-weight CAD models to objects in the scanned scene. To this end, we propose a CAD model alignment pipeline that detects objects, retrieves CAD models, and finds transformations that aligns the CAD model to the scanned scene. First, we use a single-shot anchor-based object detector to identify objects [17], using the features from the backbone we extracted ($\mathbf{F}$) from the previous stage. We then filter the predicted anchors with non-maximum suppression following the standard single-shot object detection pipeline [27]. Given this set of object bounding boxes $\mathcal{B}$, we extract $N_d \times N_d \times N_d$ feature volume F_o for all $o \in [1, ..., |\mathcal{B}|]$ from the feature map F around the object anchor a_o. We use this feature volume for CAD model retrieval and alignment. A corresponding CAD model is retrieved by calculating an object descriptor of length 512 and searching the nearest neighbor CAD model from an shared embedding space. This shared embedding space is established by minimizing the distance between descriptors of scanned objects and their CAD counterpart with an L1 loss during training.

Finally, given the nearest CAD model for all object anchors, we find dense correspondences between the CAD model and the feature volume F_o. Dense correspondences are trained through an explicit voxel-wise L1 regression loss. We use Procrutes [15] to estimate a rotation matrix and an L1 distance loss with

respect to the groundtruth rotation matrix to further enhance correspondence quality. Note that the Procrutes method yields a transformation matrix through the Singular Value Decomposition which is differentiable, allowing for end-to-end training.

3.3 Learning Object and Layout Relationships

From our layout prediction and CAD model alignment, we obtain a set of layout quads and aligned CAD models, both obtained independently from the same backbone features. However, this can result in globally inconsistent arrangements; for instance, objects passing through the ground floor, or shelves misaligned with walls. We thus propose to learn the object-layout as well as object-object relationships as a proxy loss used to guide the CAD model alignments and layout quads into a globally consistent arrangement.

We construct this relationship learning as a graph problem, where the set of objects and layout quads form the nodes of the graph. Edges are constructed between every object-object node-pair and every object-quad node-pair, forming a graph on which we formulate a message-passing graph neural network.

Each node of the graph is characterized by a feature vector of length 128. For objects this feature vector is obtained by pooling the object feature volume to 8^3 resolution, followed by linearization. For layout quads, this feature vector is constructed by concatenating the features from F or the associated corner locations, upon which an MLP is applied to obtain a 128-dimensional vector.

Figure 3 shows an overview of our message-passing network. Messages are passed from nodes to edges for a graph $G = (V, E)$, with nodes $v_i \in V$ and edges $e_{j,k} = (v_j, v_k) \in E$. We define the message passing similar to [14, 20, 10, 37]:

$$v \rightarrow e : \mathbf{h}_{i,j}^{t+1} = f_e(\text{concat}(\mathbf{h}_i^t, \mathbf{h}_j^t - \mathbf{h}_i^t))$$

where $\mathbf{h}_i^t$ is the feature corresponding to vertex v_i at message passing step t, $\mathbf{h}_{i,j}^t$ is the feature corresponding edge $e_{i,j}$ at step t, and f_e represents an MLP. That is, edges features are computed as the concatenation of its constituent vertices.

We then take these output edge features from the message passing and perform a classification of various relationships using a cross entropy loss. We describe the relationships as follows, which we chose as they do not require extra manual annotation effort given existing ground truth CAD alignments and scene layout; see Section 4.2 for more detail regarding extraction of ground truth object and layout relationships. For object-layout relationships, we formulate a 3-class classification task for support relations, predicting *horizontal support*, *vertical support*, or no support. Only one relationship per object-layout pair is allowed. For object-object relationships, we predict the angular difference between the front-facing vectors of the respective objects, in order to recognize common relative arrangements of objects (e.g., chairs often face tables). This is trained with a 6-class cross entropy loss where the angular deviation up to 180$\circ$ is discretized into 6 bins.

Here, the relationship prediction adds a proxy loss to the model in Figure 2 which inter-correlates object and layout alignments, implicitly guiding the CAD model alignment and layout quad estimation to become more globally consistent.

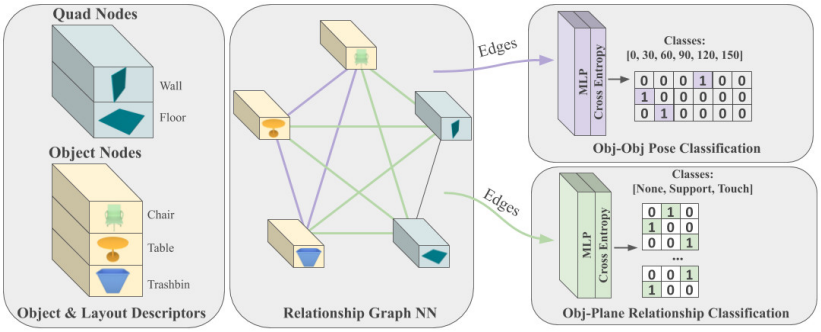

Fig. 3. Object and layout relational prediction. We establish a message-passing neural network in order to predict object-object and object-layout relations. The inputs are feature descriptors of detected objects and quads pooled to the same size, and the output is relationship classification between objects and layout elements, as well as pose relations between objects. Note this relational inference is fully differentiable, enabling end-to-end prediction.

4 Object+Layout Dataset

To train and evaluate our method, we introduce a new dataset of 1151 CAD layout annotations to the real-world RGB-D scans of the ScanNet dataset [9]. These layout annotations, in addition to the CAD annotations of Scan2CAD [2] to ScanNet scenes, inform our method and evaluation on real-world scan data.

In order to obtain these room layout annotations, we use a semi-automated annotation process. We then automatically extract the object-object and object-layout relations.

4.1 Extraction of Scene Layouts

We performed a semi-automatic layout annotation for ScanNet scene data. First, large planar surfaces are detected using RANSAC on the reconstructed scans. We then employ a manual refinement step to modify potential errors in the automatic extraction. The surface extraction is preceded by a semantic instance segmentation to obtain wall, floor, ceiling, window, door, etc. instances. RANSAC is then applied to extract 3D planes from each instance. Planes that fall below a threshold will be merged or connected. All planes are projected onto the floor plane and through a set of various heuristics the most plausible intersection

points are selected to ultimately become corner points for the final layout. The room height is either estimated by the maximum height of the detected wall instances or is spanned by the ceiling.

Following the proposals given by RANSAC, we then manually verified which proposals were plausible. This step is relatively quick ($\approx$ 2min per scene) and indicated that the RANSAC produced 1151 plausible initial layouts. These layouts were then refined through a manual annotation process. We developed a Blender[4]-based tool was introduced for the layout refinement, allowing annotators to edit/merge/delete corner junctions as well as add or modify edges and planes. All automatically generated layouts were verified and refined by two student annotators ($\approx 15min$ per scene). An illustration of layouts annotation samples on ScanNet can be found in the supplemental.

4.2 Extraction of Object and Layout Relationships

To support learning global scene relationships, we extract object and layout relations to supervised our message-passing approach to learning relationships. We opt to learn relations which can be automatically extracted from given CAD model and layout annotations.

We extract object-object and object-layout relationships. For the object-object case, we compute the angular difference between the front-facing vectors of each object where symmetrical properties are ignored; in practice, we compute this on-the-fly during the training process.

Relationships between objects and layout elements are established by support:

- A vertical **support** relationship between a layout element and an object is valid if the bottom side of the bounding box of the object within close proximity to and close to parallel to the layout element.
- A horizontal **touch** relationship is valid if the left, right, front or back side of the bounding box of the object is within close proximity to and close to parallel to the layout element.

These relations are extracted through an exhaustive search. That is, each pair of object-layout is checked for vertical support or horizontal touch. To estimate proximity of objects, we expand the bounding box of the objects by τ_p, and expand the sides of the bounding boxes of the layout elements by τ_p. We then consider the object and layout element to be in close proximity if their expanded bounding boxes overlap. We use $\tau_p = 0.2$ meters for all experiments.

4.3 Synthetic Data

We additionally evaluate our approach on synthetic data, where CAD object and layout ground truth are provided in the construction of the synthetic 3D scenes. We use synthetic scenes from the SUNCG dataset [32]. SUNCG contains models of indoor building environments including CAD models and room layouts. Layout

[4] https://www.blender.org

	bathtub	bookshelf	cabinet	chair	display	other	sofa	table	trashbin	class avg.	avg.
FPFH (Rusu et al. [31])	0.00	1.92	0.00	10.00	0.00	5.41	2.04	1.75	2.00	2.57	4.45
SHOT (Tombari et al. [34])	0.00	1.43	1.16	7.08	0.59	3.57	1.47	0.44	0.75	1.83	3.14
Li et al. [22]	0.85	0.95	1.17	14.08	0.59	6.25	2.95	1.32	1.50	3.30	6.03
3DMatch (Zeng et al. [40])	0.00	5.67	2.86	21.25	2.41	10.91	6.98	3.62	4.65	6.48	10.29
Scan2CAD (Avetisyan et al. [2])	36.20	36.40	34.00	44.26	17.89	**70.63**	30.66	30.11	20.60	35.64	31.68
End2End (Avetisyan et al. [3])	38.89	41.46	51.52	73.04	26.53	26.83	76.92	**48.15**	18.18	44.61	50.72
Ours (dense)	33.33	39.39	**58.62**	70.76	28.57	33.72	50.00	34.55	23.73	41.41	51.05
Ours (dense) + obj-obj	44.44	**54.55**	49.15	68.05	37.50	36.05	61.11	42.01	27.12	46.66	52.97
Ours (dense) + layout	**54.55**	47.37	38.33	71.11	32.88	28.05	62.86	37.91	**32.26**	45.04	52.06
Ours (dense) full	39.39	42.11	48.33	74.32	42.47	36.59	62.86	36.26	30.65	45.89	54.33
Ours (sparse)	42.42	39.47	51.67	77.28	45.21	28.05	77.14	37.91	25.81	47.22	55.77
Ours (sparse) + obj-obj	42.42	44.74	50.00	77.53	43.84	30.49	74.29	39.56	**32.26**	48.35	56.70
Ours (sparse) + layout	45.45	42.11	48.33	78.27	42.47	31.71	77.14	37.36	27.42	47.81	56.29
Ours (sparse) full	42.42	36.84	58.33	**81.23**	**50.68**	40.24	**82.86**	45.60	**32.26**	**52.27**	**61.24**

Table 1. CAD alignment evaluation on ScanNet Scan2CAD data [9, 2]. Our final method (last row), incorporating contextual information from both object-object relationships and object-layout relationships, outperforms the baseline by a notable margin of 10.52%.

components are given and hence extraction into planar quads can be performed automatically. To generate the input partial scans, we virtually scan the scenes to produce input scans similar to real-world scenarios, following previous approaches to generate synthetic partial scan data [17].

Object and layout relational information was extracted following the same procedure for ScanNet data.

5 Results

5.1 CAD Alignment Performance

We evaluate our method on synthetic SUNCG [32] scans as well as real-world ScanNet [9] scans in Tables 3 and 1, respectively. We follow the CAD alignment evaluation metric proposed by [2], which measures alignment accuracy where an alignment is considered successful if it falls within 20cm, 20°, and 20% scale of the ground truth. On both SUNCG and ScanNet scans we compare to several state-of-the-art handcrafted geometric feature matching approaches [31, 34, 22] and learned approaches [40, 2, 3]. We additionally show qualitative comparisons in Figures 6 and 5.2. On synthetic scan data we outperform the strongest baseline by 16.58%, and improve by 10.52% on real scan data. This demonstrates the benefit of leveraging global information regarding object and layout relations in improving object alignments.

We also perform an ablation study on the various design choices and impact of relation information. We evaluate a dense convolutional backbone for our network architecture (*dense*) in contrast to our final sparse convolutional backbone leveraging the sparse convolutions proposed by [8]. We additionally show that the object-to-object relational inference (*obj-obj*) as well as layout estimation (*layout*) improve upon no relational inference, and our full method incorporating both object and layout relational inference, the most contextual information, yields the best performance.

5.2 Layout Prediction

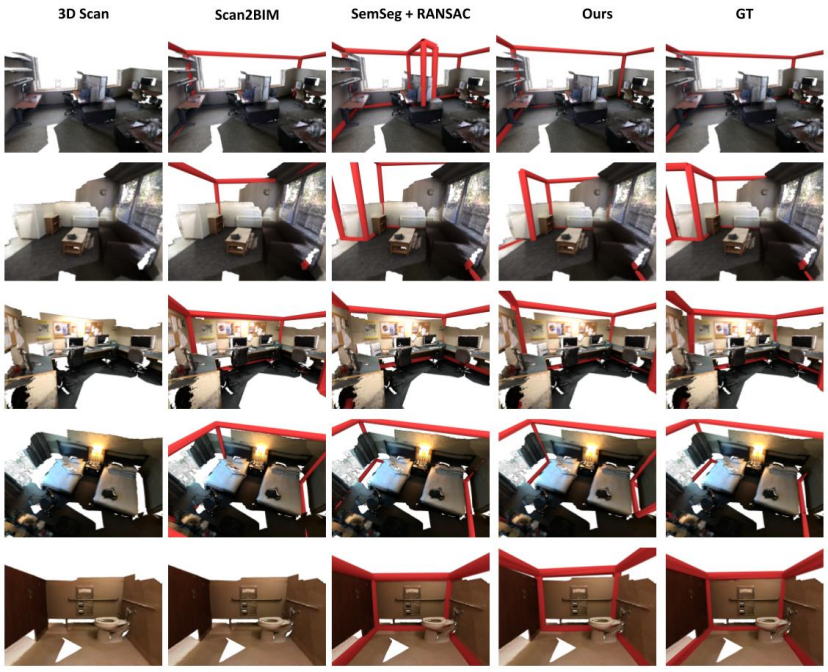

Fig. 4. Qualitative comparison of our layout estimation on the ScanNet dataset [9]. Layout elements are highlighted with their wireframes. Our method provides a very lightweight, learned approach ($\approx 1M$ trainable parameters) for layout estimation.

For the final quad prediction we achieve a F1-score of 37.9% on ScanNet and 69.6% on SUNCG. Corners are considered as successfully detected if the predicted corner is within a radius of $40cm$ from the ground truth corner. Edges are considered as correctly predicted if they connect the same corners as the ground truth edges. Similarly, correctly predicted quads are spanned by the same 4 corners as the associated ground truth quad. We aim to achieve a high recall for corners and edges due to our hierarchical prediction. We achieve robust results on both datasets, although ScanNet is notably more difficult as many scenes can miss views of entire layout components (e.g., missing ceilings).

6 Limitations

While the focus of this work was to show improved scene understanding through joint prediction of objects **and** layouts, we believe there is potential for further achievements. For instance, our layout prediction method is bound to predict

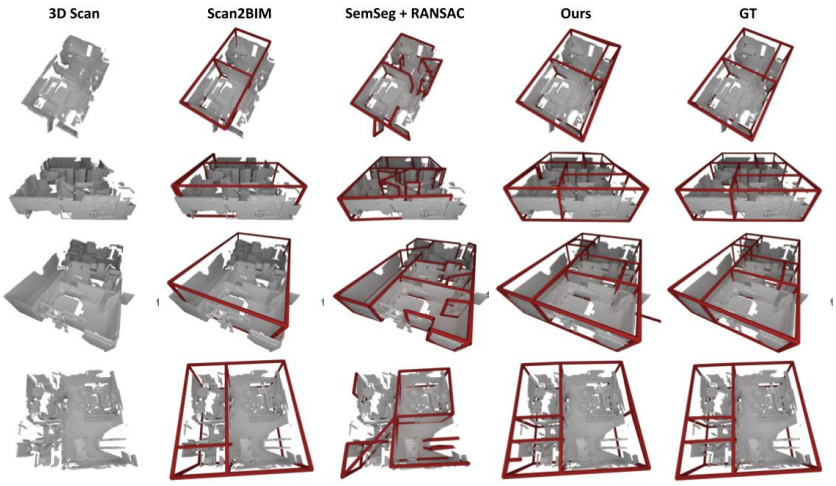

Fig. 5. Layout estimation on SUNCG [32] scans. Layout elements are highlighted with their wireframes. Our method excels with its simplicity, especially for very large and complex scenes where heuristics to determine intersections tend to struggle.

# voxels	18K	42K	71K
Scene extent	$2.6m^2 \times 2.4m^2$	$3.2m^2 \times 3.5m^2$	$7.5m^2 \times 6.2m^2$
# objects	1	5	26
Timing	**1.9s**	**2.0s**	**2.60s**

Table 2. Runtime (seconds) of our approach on different test scenes categorized into small, medium and large.

	bed	cabinet	chair	desk	dresser	other	shelves	sofa	table	class avg.	avg.
SHOT (Tombari et al. [34])	13.43	3.23	10.18	2.78	0.00	0.00	1.75	3.61	11.93	5.21	6.30
FPFH (Rusu et al. [31])	38.81	3.23	7.64	11.11	3.85	13.21	0.00	21.69	11.93	12.39	9.94
Scan2CAD (Avetisyan et al. [2])	52.24	17.97	36.00	30.56	3.85	20.75	7.89	40.96	43.12	28.15	29.23
End2End (Avetisyan et al. [3])	71.64	32.72	48.73	27.78	38.46	37.74	14.04	67.47	45.87	42.72	41.83
Ours (dense)	63.89	35.16	56.82	39.02	30.00	38.85	29.17	**76.67**	31.03	44.51	44.48
Ours (dense) + obj-obj	77.78	36.26	53.03	41.46	40.00	47.48	20.83	**76.67**	25.86	46.60	46.41
Ours (dense) + layout	75.00	37.04	60.68	37.14	38.89	45.53	**33.33**	72.41	32.08	48.01	48.33
Ours (dense) full	**81.25**	40.00	51.92	45.45	41.18	49.17	31.58	75.86	**46.00**	51.38	50.41
Ours (sparse)	54.29	42.55	66.67	48.57	**44.44**	57.60	27.27	57.89	36.84	48.46	52.31
Ours (sparse) + obj-obj	74.29	40.43	70.09	**65.71**	27.78	**60.80**	27.27	55.26	38.60	51.14	55.27
Ours (sparse) + layout	65.71	42.55	**77.78**	54.29	38.89	**60.80**	22.73	57.89	45.61	51.81	57.12
Ours (sparse) full	71.43	**43.62**	**77.78**	54.29	38.89	**60.80**	22.73	68.42	45.61	**53.73**	**58.41**

Table 3. CAD alignment accuracy on SUNCG [32] scans. Our final method (last row) goes beyond considering only objects and jointly estimates room layout and object and layout relationships, resulting in significantly improved performance.

quad planes only and hence more sophisticated methods could be used for

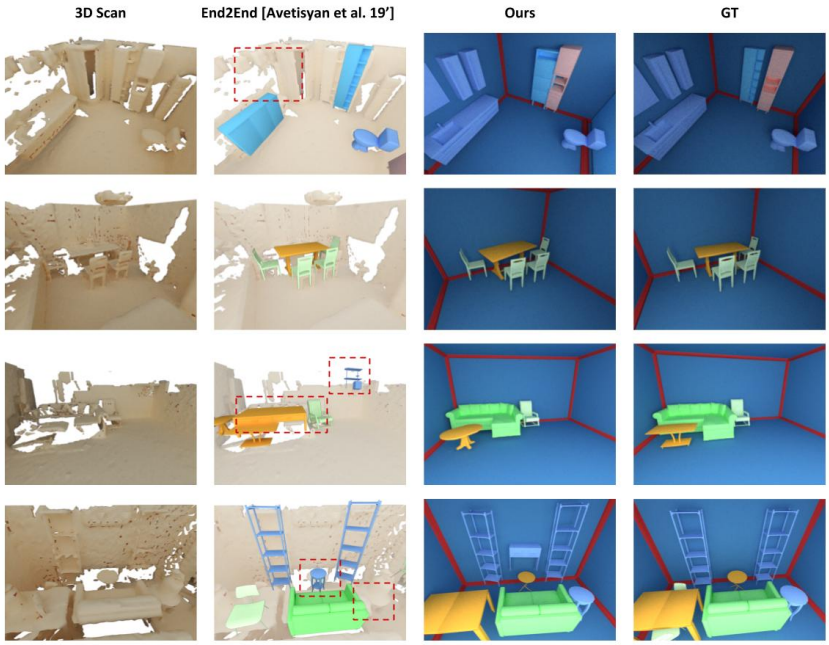

Fig. 6. Qualitative CAD alignment and layout estimation results on SUNCG [32] scans. Our joint estimation approach produces more globally consistent CAD alignments and generates additionally room layout applicable for VR/AR applications.

more accurate layout estimation. Also, we used a very lightweight graph neural network for message passing. One could use a more sophisticated method for more accurate relationship prediction and a richer set of relationships that may contain functionality relationships, spatial relationships or room semantic relationships.

7 Conclusion

In this work we formulated a method to digitize 3D scans that goes beyond the focus of objects in the scene. We propose a novel method that estimates the layout of the scene by sequentially predicting corners, then edges and finally quads in a fully differentiable way. The estimated layout is used in conjunction with an object detector to predict contact relationships between objects and the layout and ultimately to predict a CAD arrangement of the scene. We can show that objects and the surrounding (scene layout) go hand in hand and are a crucial factor towards full scene digitization and scene understanding. Objects in the scene are often not arbitrarily arranged, for instance often cabinets are leaned at walls or a table is surrounded by chairs in a dining room, hence we leverage the inherent coupling between objects and layout structure in the learning process. Our approach improves global CAD alignment accuracy by

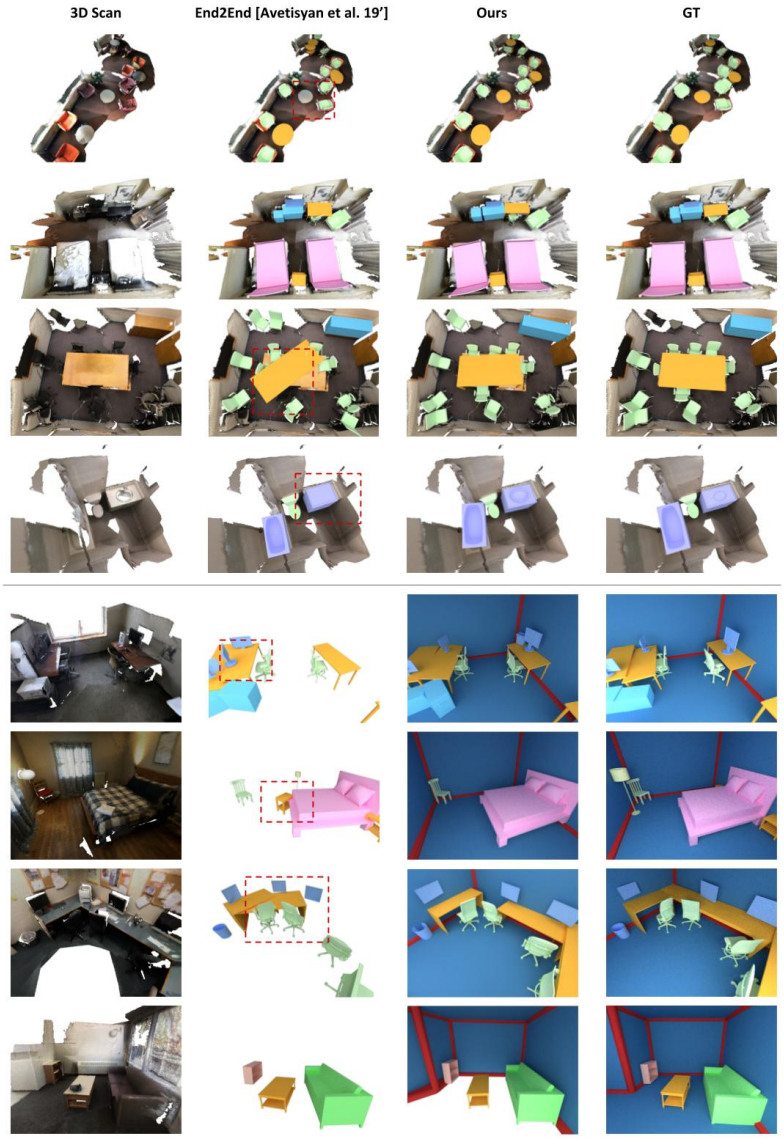

Fig. 7. Qualitative CAD alignment and layout estimation results on ScanNet [9] scans (zoomed in views on the bottom). Our approach incorporating object and layout relationships produces globally consistent alignments along with the room layout.

learning those patterns on both real and synthetic scans. We hope that we can encourage further research towards this avenue, and see as next immediate steps for future work the necessity of texturing digitized shapes in order to enhance the immersive experience in VR environments.

References

1. Armeni, I., He, Z.Y., Gwak, J., Zamir, A.R., Fischer, M., Malik, J., Savarese, S.: 3d scene graph: A structure for unified semantics, 3d space, and camera (2019)
2. Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A.X., Nießner, M.: Scan2cad: Learning cad model alignment in rgb-d scans. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2019)
3. Avetisyan, A., Dai, A., Nießner, M.: End-to-end cad model retrieval and 9dof alignment in 3d scans. arXiv preprint arXiv:1906.04201 (2019)
4. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In: European Conference on Computer Vision. pp. 561–578. Springer (2016)
5. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017)
6. Chen, J., Liu, C., Wu, J., Furukawa, Y.: Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2661–2670 (2019)
7. Choi, S., Zhou, Q.Y., Koltun, V.: Robust reconstruction of indoor scenes. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5556–5565. IEEE (2015)
8. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019)
9. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2017)
10. Dai, A., Nießner, M.: Scan2mesh: From unstructured range scans to 3d meshes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5574–5583 (2019)
11. Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. ACM Transactions on Graphics (TOG) **36**(3), 24 (2017)
12. Drost, B., Ilic, S.: 3d object detection and localization using multimodal point pair features. In: 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission. pp. 9–16. IEEE (2012)
13. Engelmann, F., Stückler, J., Leibe, B.: Joint object pose estimation and shape reconstruction in urban street scenes using 3d shape priors. In: German Conference on Pattern Recognition. pp. 219–230. Springer (2016)
14. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. arXiv preprint arXiv:1704.01212 (2017)
15. Goodall, C.: Procrustes methods in the statistical analysis of shape. Journal of the Royal Statistical Society: Series B (Methodological) **53**(2), 285–321 (1991)
16. Gupta, S., Arbeláez, P.A., Girshick, R.B., Malik, J.: Aligning 3d models to rgb-d images of cluttered scenes. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4731–4740 (2015)
17. Hou, J., Dai, A., Nießner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2019)
18. Huang, S., Qi, S., Zhu, Y., Xiao, Y., Xu, Y., Zhu, S.C.: Holistic 3d scene parsing and reconstruction from a single rgb image (2018)

19. Kim, Y.M., Mitra, N.J., Huang, Q., Guibas, L.: Guided real-time scanning of indoor objects. In: Computer Graphics Forum. vol. 32, pp. 177–186. Wiley Online Library (2013)
20. Kipf, T., Fetaya, E., Wang, K.C., Welling, M., Zemel, R.: Neural relational inference for interacting systems. arXiv preprint arXiv:1802.04687 (2018)
21. Kulkarni, N., Misra, I., Tulsiani, S., Gupta, A.: 3d-relnet: Joint object and relational network for 3d prediction (2019)
22. Li, Y., Dai, A., Guibas, L., Nießner, M.: Database-assisted object retrieval for real-time 3D reconstruction. In: Computer Graphics Forum. vol. 34, pp. 435–446. Wiley Online Library (2015)
23. Lim, J.J., Pirsiavash, H., Torralba, A.: Parsing ikea objects: Fine pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2992–2999 (2013)
24. Liu, C., Kim, K., Gu, J., Furukawa, Y., Kautz, J.: Planercnn: 3d plane detection and reconstruction from a single image (2018)
25. Liu, C., Wu, J., Furukawa, Y.: Floornet: A unified framework for floorplan reconstruction from 3d scans. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 201–217 (2018)
26. Liu, C., Yang, J., Ceylan, D., Yumer, E., Furukawa, Y.: Planenet: Piece-wise planar reconstruction from a single rgb image. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Jun 2018). https://doi.org/10.1109/cvpr.2018.00273, http://dx.doi.org/10.1109/CVPR.2018.00273
27. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
28. Murali, S., Speciale, P., Oswald, M.R., Pollefeys, M.: Indoor scan2bim: Building information models of house interiors. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6126–6133. IEEE (2017)
29. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on. pp. 127–136. IEEE (2011)
30. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. ACM Transactions on Graphics (TOG) (2013)
31. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. pp. 3212–3217. Citeseer (2009)
32. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image (2017)
33. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3d: Dataset and methods for single-image 3d shape modeling. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Jun 2018). https://doi.org/10.1109/cvpr.2018.00314, http://dx.doi.org/10.1109/CVPR.2018.00314
34. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision – ECCV 2010. pp. 356–369. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

35. Wald, J., Avetisyan, A., Navab, N., Tombari, F., Niessner, M.: Rio: 3d object instance re-localization in changing indoor environments. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
36. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 52–67 (2018)
37. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics **38**(5), 1–12 (Oct 2019). https://doi.org/10.1145/3326362, http://dx.doi.org/10.1145/3326362
38. Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C.B., Su, H., Mottaghi, R., Guibas, L.J., Savarese, S.: Objectnet3d: A large scale database for 3d object recognition. In: ECCV (2016)
39. Zakharov, S., Shugurov, I., Ilic, S.: Dpod: 6d pose object detector and refiner. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1941–1950 (2019)
40. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. pp. 199–208. IEEE (2017)
41. Zou, C., Guo, R., Li, Z., Hoiem, D.: Complete 3d scene parsing from an rgbd image (2017)