DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Probabilistic Object Detection and Reconstruction from a single RGB-D frame

Kerem Yıldırır

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Probabilistic Object Detection and Reconstruction from a single RGB-D frame

Probabilistische Objekterkennung und Rekonstruktion aus einem einzigen RGB-D Frame

Author:	Kerem Yıldırır
Supervisor:	Prof. Angela Dai
Advisor:	Prof. Angela Dai
Submission Date:	15.09.2022

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.09.2022 Kerem Yıldırır

Acknowledgments

I would like to thank my supervisor Angela Dai for her valuable guidance and feedback throughout the duration of this thesis. I would also like to convey my thanks and gratitude to my family, without whom none of these would be possible. I am really grateful for their love and support. Last but not least, I would like to thank my friends at the Visual Computing Lab for all the valuable academic and culinary discussions and Marc Benedí for the fruitful coffee breaks throughout the Master's.

Danke.
Thank you.
Teşekkürler.

Kerem Yıldırır
Munich, September of 2022

Abstract

Semantic scene understanding is crucial aspect of modern day robotic applications. With the recent advances in deep learning and the increased availability of large scale richly annotated datasets, popularity of 3D scene understanding tasks has increased rapidly.

In this work, we present a probabilistic hybrid solution with point-based and volumetric components to jointly localize, classify and complete the object instances given a single RGB-D frame. Our reconstructions are not restricted by the camera field of view, and aims to complete the object geometry even outside the camera frustum where the input signal is significantly weaker compared to the rest of the scene.

Our probabilistic detection approach aims to combat ambiguous scenarios where objects in the scene have weak representation, and multiple completions are plausible for an object. We show that instead of outputting directly regressed bounding boxes, learning a distribution of bounding boxes can help us generate alternative suggestions for every detection proposal, which improves both detection and completion performance when utilized.

Contents

1 Introduction

In the last years following the emergence of multiple of large-scale 3D datasets, semantic scene reconstruction has gained significant momentum in the research community. The task focuses on recovering semantic classes, geometry and object poses from partial inputs signals (3D scans or images). Although existing methods for reconstruction from monocular images benefit from the significant progress of 2D CNN's and are able to achieve plausible results, they still face the main bottleneck of depth ambiguity. [1],[2],[3] [4]. Another type of input, real world 3D scans, are also problematic, as they often include imperfect or missing, geometry due to various factors such as occlusions and bad illumination. Existing works have explored many different strategies to address these imperfections and recover the missing geometric features, [5] used TSDF(truncated signed distance field) grids of the whole scene and completed the missing geometric attributes of objects to assist with object localization, [6] leveraged the sparse and compact nature of point clouds to operate in high resolution and both detect and complete 3D objects from scenes with missing geometry. Retrieval methods like [7][8][9] reformulated the problem as a matching problem and for each detected entity, aimed to retrieve and align a CAD model from a predefined database. [10] united the tasks of scene completion and semantic segmentation, and proposed semantic scene completion (SSC), the task of jointly estimating both scene geometry and semantic information of a scene from a given sparse partial input, a single depth frame.

In this work, we tackle the task Semantic Instance Completion (SIC), which is slightly different from SSC in the sense that instead of semantically segmenting the whole scene, it only focuses on the object instances in the scene with background classes such as wall, ceiling and floor are ignored. Object localization is required to identify regions of interest and recover missing geometry in a per-instance basis, SSC, on the other hand treats all instances of the objects belonging to the same semantic class the same and complete the scene as a whole. Both tasks show significant relevance in real life scenarios where semantic understanding of the scene is crucial, such as robot navigation and interior design. The choice of which problem to tackle depends on the constraints enforced by the application. To give an example in the robotics domain, imagine an agent is deployed in an environment where it needs to analyze and interact with the scene. While for some cases it is enough for the agent to perceive the object classes in the scene, another scenario where further interaction with the scene is needed might require additional information such as how many instances of an object exists in the scene, or with which instance the agent is currently interacting.

We take inspiration from the single depth frame setting, and formulate the SIC problem as localizing, classifying and completing all the object instances represented

in a single depth frame, while also extending the completion scope beyond the camera field of view. To our knowledge, existing methods have formulated this task to perform completion and classification only in the camera field of view. This extension brings additional ambiguity to the task, especially for the objects that lie mainly outside the camera frustum, where multiple pose, geometry and classification possibilities become plausible due to lack of information from the data.

Deterministic methods tend to suffer from environments with high uncertainty, and deep learning methods are not an exception, and they have been shown to be often over-confident in their predictions [11] which could potentially lead to troubling outcomes. To address this issue, recent methods introduced uncertainty quantification in their deep neural networks, and by leveraging the quantified information, improved their dedicated tasks. Uncertainty quantification not only improves performance in the tasks of semantic classification and depth estimation [12], but also provides a strong signal on when the model is performing poorly, and can be used for weighting during optimization to penalize noisy residuals, making the model more robust to noisy environments [13] [2][1].

To this end, we present a probabilistic hybrid framework for semantic instance completion from a single RGB-D frame, utilizing both point cloud and voxel grid representations for 3D scenes. Our method is capable of detecting and classifying object instances in the scene, while recovering the missing geometry globally from both inside and outside the camera frustum. We also show that due to the probabilistic nature of our method, we are able to generate multiple plausible detection and reconstructions per proposal, and we show that utilizing multiple suggestions lead to better localization and completion, especially when objects are underrepresented in the input.

The rest of the thesis is structured like the following:

Chapter 2 gives an overview of the existing 3D representations, followed by a brief review of 3D Object Detection, SSC and SIC approaches and concludes with a recap on uncertainty quantification in computer vision problems.

Chapter 3 presents our proposed method and the main ideas behind it, and describes how our framework operates in practice.

Chapter 4 provides our implementation details, qualitative and quantitative results of our method, comparison with our SSC baseline, followed by an analysis of our method under different constraints.

Chapter 5 concludes this work by summarizing our contributions and potential future improvements.

2 Related Work

In this chapter, we give a brief history of the developments in the relevant areas to our work. We introduce some of the popular 3D data representations, followed by methods for 3D object detection, semantic scene and instance completion, and uncertainty quantification in computer vision.

2.1 3D Data Representations

While for humans it is trivial to perceive our surroundings and perform complex analysis in the environment, replicating these tasks with computers require proper representation of the captured 3D space. Accurately representing 3D is a challenging task, and many different representations have been used to encode 3D data for many tasks. For applications that require semantic scene understanding, the most popular representations are point clouds and volumetric grids.

Point clouds are unordered set of points in the euclidean space, with optionally other attributes than position, and volumetric grids are the intuitive extension of a 2D image to 3D. The space is partitioned with 3D unit cells which are named voxels, to the desired resolution and every cell encodes information about the 3D unit volume it is representing, this information can simply be occupancy, or in the more sophisticated scenario where the scene is encoded as an implicit surface, a distance field, it can contain the distance to the surface in the scene, which provides rich information for processing. Volumetric representations have the regular grid structure with efficient neighbor access, which is beneficial for feature extraction using 3D convolutional nerual networks (CNN). However these benefits come with the cost of heavy memory usage and computation requirements, as the grid size grows cubicly with the resolution. Point clouds do not offer any regular structure and efficient neighbor access, but are much more lightweight and less demanding in terms of resources. Since 3D convolutions cannot be directly applied to point clouds, classical methods for handcrafted features **??**, or more recently, deep neural networks such as [14] are used to extract per-point features from the point cloud, which led to many deep learning based methods utilizing point clouds for many different tasks, ranging from object detection [15], semantic segmentation [14], classification etc.

In the context of this thesis, we examine methods utilizing both representations, as our method makes use of both point clouds and TSDF grids.

2.2 3D Object Detection

Object detection is the task of localizing object instances in a given an input signal. For the case of 2D, the signal comes from the image domain, and although it lacks the depth information, encodes dense information about the scene and its semantic properties, and modern methods like [16] are robust and accurate enough to be deployed at commercial autonomous driving systems. For the case of 3D, the input signal often comes from point clouds or volumetric grids, and with the increasing popularity of the field and richly annotated synthetic [17] and real world datasets [18] [19], many approaches for object detection emerged over the years.

Inspired by the success of 2D methods, some earlier works on 3D object detection leveraged RGB-D images in their pipeline. [20] proposed regions in 2D, then lifted them to 3D, then using the points from the depth point cloud inside the proposed region, used PointNet for further processing. Following this work, [21] used 2D CNNs for feature extraction, backprojected the extracted 2D features into 3D grids, performed semantic instance segmentation with a 3D CNN architecture. Taking it a step further, [5] encoded the RGB-D scans as TSDFs to be coupled with color information from the RGB images, and proposed a fully convolutional 3D architecture for both predicting missing object geometry and localizing objects in the scene. However, these methods had to be restricted to low resolution, because of the large memory requirements of volumetric methods. Overcoming this bottleneck, [15] presented a robust approach which utilizes only geometric features from point clouds, yet still outperformed the state of the art detection algorithms in indoor scenes, and following works continued to use the point cloud representation for object detection, driving state of the art further. Recent works using only RGB images show promising results [22], but suffer from the depth ambiguity to predict accurate bounding boxes in 3D space. In this work, we choose the point based VoteNet [15] architecture as our detection backbone and build our framework on top of its foundations.

2.3 Semantic Scene Completion

Semantic Scene Completion (SSC) is the task of jointly estimating scene geometry and semantic class information from a partial sparse input. The two tasks were treated independent until it is proposed by [10] that they can be jointly tackled and are beneficial to each other and proposed a 3D CNN architecture to perform this task from a single depth frame. Inferring a dense scene with partial input is an ill posed problem, since the input does not contain enough information to recover all missing characteristics of the scene. Most of the existing methods rely on deep learning, and benefit from different 3d representations and architecture types. In the scope of this thesis, we will constrain this section to describing some of the recent approaches and representations, for a more in depth explanation of the problem and existing approaches, we refer the reader to [23].

Majority of the existing methods rely on 3D grid based representations for this task,

as it is convenient to represent the scene as a binary occupancy grid or an implicit surface representation such as truncated signed distance fields, and its variants[24] [25] . In this work, we use a completion backbone which works with TSDF grids, as they encode richer information about the contextual state of the scene, which we belive is necessary for our ambiguous problem setting. Following the initial work of [10], following methods approached the task from various perspectives. [26] and [27] aimed to reduce the computation on dense voxel grids, [28] proposed a coarse to fine approach for refining geometric details. Other methods utitilize RGB images as a complementary input to the depth, and aimed to fuse the two feautures together. We make use of the 3D-Sketch architecture by [29] which instead of implicitly encoding the information in a feature space proposes an explicit geometric embedding of the scene, and together with the RGB image features, guides the process of reconstruction and classification of the input scene. Some of the existing works also show promising results by only using RGB images [3], they are unable to compete with the geometrical approaches, as they use even less information for the task. Following the existing work, we adapt the single RGB-D input setting, but different than existing methods, we define our task as retrieving the complete global object geometry, regardless of the completed area being inside camera field of view or not. Hence the end goal of our method falls under the task of semantic instance completion rather than semantic scene completion.

2.4 Semantic Instance Completion

Semantic Instance Completion (SIC) focuses on classifying, localizing and completing all the object instances in the scene. In the context of this thesis, we define our "scene" as a single depth frame, where the existing approaches make use of the whole indoor scene scan as their inputs.

Existing work on SIC rely on instance segmentation or object detection to localize the object, typically, generating annotated ground truth data per 3D object instance is very expensive, and to often infeasable to put into motion. Some methods approach the completion problem as retrieval, and after locating the objects, align a matching CAD model from the database to the objects, leading to a clean CAD model of the scene. [9][7]. Other methods including ours use a completion module to isolate the region inside the localized area and recover missing geometry in the designated region. Also, more recent approaches attempt to perform multiple object detection and completion from single RGB frames.[30][3]

From the existing works RfD-Net by [6] shows strong resemblance to our problem, as they jointly detect and complete object geometries from directly from point clouds in an end-to-end fashion. They follow the work of [5], which hallucinates the missing geometry in the scene and uses it to enhance detection performance. Although end goal is quite similar, both of these methods operate on a different setting from our pipeline. They benefit from the whole 3D scan of the scene while we restrict ourselves to a single RGB-D frame, carrying a much weaker input signal.

2.5 Uncertainty quantification in computer vision

Deep learning models need to be robust against ambiguous input or unseen data, however this is rarely the case. This is only natural, as the data the model trained with has a big impact on how that model performs under real world circumstances, and it is infeasible to include every single variation of a task in a dataset. As deep learning methods continue to improve and outperform and replace classical methods in computer vision, their roles in real life applications are becoming more and more important. In the context of computer vision, there is already a variety of existing deep learning based solutions for problems such as object detection, classification, autonomous driving, depth estimation are getting integrated into our lives.

With the increasing use cases in real life, the need of uncertainty quantification becomes more imminent. The deployed models should be able to distinguish when their predictions become less reliable and invoke the necessary action accordingly, rather than being overconfident about them. To give an example, let us imagine an autonomous driving scenario where the 3d surroundings are identified through images. If a vehicle has unusual color or material, which affect the way its perceived by the camera, it could lead into faulty detections. Assuming the model is not trained for scenarios like this, the desirable situation would be notifying the driver to take action in regarding this lack of confidence in model predictions. If this issue is not addressed, it could lead to catastrophic outcomes. Uncertainty quantification in statistical modeling is not a newly addressed problem, existing work investigate and capture the uncertainty in many tasks in classical methods [31], however it is still yet to become a standard in modern computer vision applications.

For the case of computer vision and deep learning, the work of [12] investigate the types of uncertainty and how to capture them in deep learning methods for popular tasks and provide a practical approach to enable any deep neural network to capture the uncertainty, which was deemed as infeasible before. In their work, the uncertainty is quantified under two categories, aleatoric uncertainty and epistemic uncertainty. Epistemic uncertainty is the uncertainty of the model parameters, and can be reduced with more data usage. It is useful for detecting out of distribution inputs and it is also beneficial as a selection criteria in active learning scenarios to select the most informative fractions of the data include in the training set. Aleatoric uncertainty is the uncertainty present inherently in the data and stems from the nature of the input, which cannot be explained away with more data, and is useful to capture in regression tasks for robustness. One can model it in a homoscedastic fashion where constant noise is assumed over the whole data, or in a heteroscedastic fashion, where the learned uncertainty is modeled as a function of the data, and varies from input to input.

Existing works demonstrate that captured aleatoric uncertainty can be used for robust optimization against noisy input, [2][13], or as a refinement criteria for object detection [32]. Capturing aleatoric uncertainty is typically done by assuming the data comes from Gaussian distribution, and adding a head to the network to estimate the variance of the distribution input data is coming from. The predicted variance is supervised

using a Gaussian Negative Log Likelihood loss function, as suggested by [12]. In our proposed method, we employ this strategy to learn the distributions for object sizes and centers, but instead of using the predicted values as weighting factors, we use the learned distribution to generate multiple suggestions.

3 Method Overview

In this chapter we introduce our problem statement and constraints, and describe our proposed object detection and completion pipeline.

3.1 Overview

Given a 3D scene, the task of semantic instance completion focuses on recovering oriented bounding boxes, semantic classes and their missing geometric features [23].

We further increase the difficulty of this task by constraining our input signal to a single RGB-D frame, but still attempting to complete global object geometry for each instance. This introduces many ambiguous scenarios where underrepresented objects are required to be localized and completed correctly.

We present a hybrid two stage approach with a point-based probabilistic detection module followed by a volumetric completion module.

In the first stage, the detection head takes the generated point cloud from the RGB-D frame as the input, and outputs object proposals for the scene. For each proposed box, our method also captures the heteroscedastic aleatoric uncertainty for box size and box center, which we leverage at test time to generate multiple suggestions per output.

In the second stage, for each object proposal coming from the first stage, we isolate the corresponding region in the input TSDF and point cloud, and feed it to our completion head to predict occupancy inside the detected region.

3.2 Object detection

We employ VoteNet [15] as our detection backbone which utilizes point features from [14], and learns to cast votes for object centers for each point, then clusters these votes and regresses bounding box information using the extracted features and aggregated clusters. We follow the approach of [12] to extend this initial architecture with two additional 1D convolution heads to capture the heteroscedastic aleatoric uncertainty σ for box size and centers. With this extension, we model the bounding box sizes and centers as Gaussian distributions $\mathcal{N}(\mathcal{S}, \sigma_{size})$ and $\mathcal{N}(\mathcal{C}, \sigma_{center})$ with initial size and center predictions $\mathcal{S}$ and $\mathcal{C}$ as the means and the predicted σs as variances. As a result of this modeling, our method learns bounding box center and size distributions as a function of the data, enabling us to sample from the learned distribution at test time to generate multiple plausible suggestions for bounding box parameters during inference. The sampling process for a set of bounding boxes K is done by first finding ground truth

association K' where each of the boxes in K has a target in K', then sampling N sizes and centers from $\mathcal{N}(\mathcal{S}_j, \sigma_{size})$ and $\mathcal{N}(\mathcal{C}_j, \sigma_{center})$ for every jth box in K, then finally out of all the samples, select the one closest to corresponding ground truth S' and C' where S' and C' are the sizes and centers of the ground truth boxes. Figure 3.1 illustrates a set of box samples.

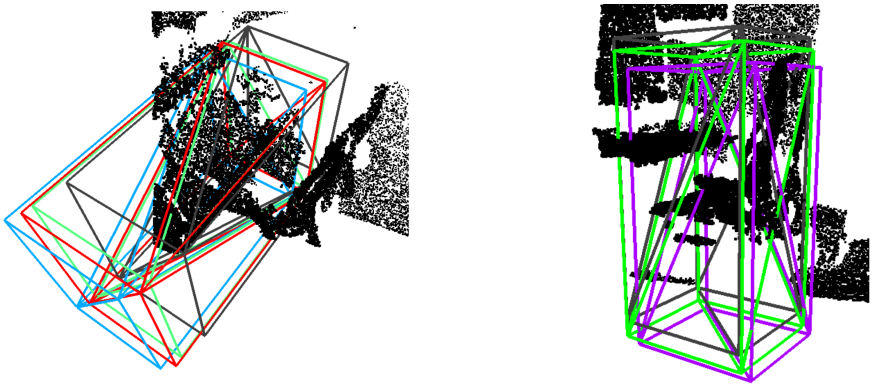

Figure 3.1: Visualization of sampled bounding boxes.

Our main motivation for this extension is to address the problem of ambiguity inherently present in the data, and provide an approach to tackle this by presenting alternative outputs rather than making deterministic point estimates by showing we can generate better suggestions from the learned distributions.

3.3 Instance Completion

For the task of instance completion, we make use of the 3D-Sketch architecture proposed by [29] for the task of SSC, and rearrange it such that only occupancy is predicted as the semantic classification is done in the previous stage. The completion is performed in the predicted object boundaries, meaning that objects partially outside of the camera frustum is also expected to be completed, if their boundaries are detected correctly.

The output of the whole pipeline is a completed object inside the voxel grid for each given input detected region. This component takes a TSDF grid and an RGB image as input, and predicts occupancy as output using a learned 3d geometric prior referred as the "sketch" in the original paper. Staying faithful to the original grid resolution,we also keep the grid resolution at (60,36,60) and perform completion. An illustration of our completion module can be seen in Figure 3.2.

3.4 Joint Detection and Completion

For the task of jointly detecting and completing objects, we put the two networks together. Both our detection and completion stages suffer from the inherent uncertainty

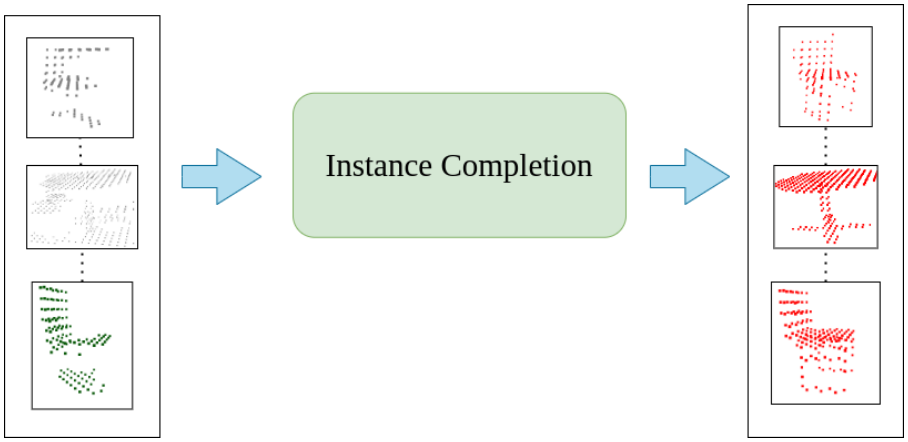

Figure 3.2: Visualization of our instance completion module

of the task for the objects with poor representation. To tackle this, we leverage the probabilistic nature of our detections, and generate multiple suggestions per bounding boxes to generate multiple suggestions for the completion regions, leading to alternative completions.

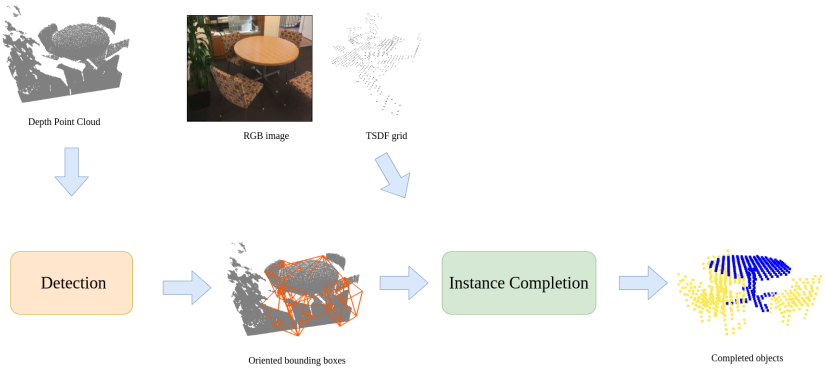

Figure 3.3: Visualization of our pipeline

3.5 Loss functions

We follow the VoteNet implementation for the detection stage and use their proposed loss functions, which are cross entropy for objectness and semantic classification, and Huber loss for regressing box size, center and heading angle. To supervise the learned distribution parameters, we add Gaussian Negative Log Likelihood(GNLL) $\mathcal{L}_{\mathcal{D}}$ which is defined in (Equation 3.1) where D corresponds to the number of valid object proposals, $\sigma(x_i)$ corresponds to the ith predicted σ, and y_i and $\mu(x_i)$ denote the ground truth value and predicted value, respectively. For both box size and center, we compute GNLL and

add it to the final loss term with coefficient $\lambda = 2$.

$$\mathcal{L}_D = \frac{1}{2} \sum_{i=0}^{|D|} \left[\log\ \sigma(x_i)^2 + \frac{(y_i - \mu(x_i))^2}{\sigma(x_i)^2} \right] \tag{3.1}$$

For the second stage, we use the proposed loss functions in the original 3D-Sketch implementation, but since we removed the semantic classification, we swap the proposed categorical cross entropy loss with weighted binary cross entropy loss for scene occupancy prediction. Our weighting scheme aims to gives higher priority to the unrepresented parts of objects to capture the thinner object structures such as chair and table legs, which are often ignored due to the class imbalance in the task, as empty voxels tend to dominate a voxel grid.

4 Experiments and Results

In this chapter, we give an overview of our data generation process, the metrics for evaluation, our training strategy and results for a series of qualitative and quantitative experiments to evaluate the effectiveness of our method under different constraints.

4.1 Data

For our task, we need singular RGB-D frames, coupled with object instances represented in this frame, with oriented bounding boxes and complete ground truth geometry. With these constraints taken into consideration, we generate data for our method from two datasets:

- ScanNet v2 [18], which consists of 1,513 richly annotated real world scene scans the corresponding RGB-D frames used to reconstruct them.
- Scan2CAD [7] aligns the ShapeNet [33] models with the object instances per scene in ScanNet, and provides the object meshes.

We first preprocess the depth frames of ScanNet to align them to the scene they belong. The 3D scene reconstructions in ScanNet are generated using real data, which sometimes results in incomplete or incorrect semantic labeling in a per point level, which could hinder the quality of the ground truth for instance completion, hence we include Scan2CAD models into our pipeline.

After having the frame-scene association, we align object meshes from Scan2CAD to the frame and generate ground truth point clouds for instance completion by sampling 5000 points from the mesh surface. This enables us with the "ground truth" for the object's geometry in the scene, as well as the oriented bounding boxes, where the original ScanNet only provided axis aligned bounding boxes.

At this stage, our dataset consist of depth frames per scene, and the object instances transferred to the frame coordinate frame. However both ScanNet and Scan2CAD annotations are on complete indoor scene scans, not individual frames. Hence we still lack the association of which object is represented in which frame.

To address this issue, we define a '*visibility*' metric for every object in the scene for the current frame, with the purpose of measuring how much the object represented in a depth frame in the scene. We define this quantity as the overlap percentage of the ground truth point cloud with the input point cloud, and compute it by first carrying both the frame and the object ground truth to the same voxel grid coordinates in (60,36,60) resolution, then measuring the overlap. We refrain from using the view frustum for this

task, because an object can be inside the camera frustum but not fully represented in the point cloud due to sensor noise, occlusions etc. For our task, we select a lower bound of $\theta = 0.2$ for deciding object-frame association. Although we keep this quantity per object to perform further analysis on our method at different θ ranges.

Furthermore, we filter out the frames which have no or smaller than θ object overlap with the input frame, and only use the classes chairs and tables.

Our dataset consists of total 8561 frames, 7180 chairs and 5519 tables in the train split, and 2315 frames, 2107 chairs and 1488 tables in the validation split.

We use the input frames belonging to the scenes in official train/test split for all the experiments and use chairs and tables as our semantic classes. For validation, we use a subset of randomly selected 500 elements from the original validation set for our experiments. We refer the reader to Table 4.1 and Table 4.2 for more statistics of our dataset, and some example annotated frames can be seen in Figure 4.1.

Object Class \Visibility	**0 - 0.2**	**0.2 - 0.5**	**0.5 - 1.0**
Chairs	1708	1737	370
Tables	1043	1155	333
Total	2751	2892	703

Table 4.1: Distribution of classes and their visibilities in the validation split

Object Class \Visibility	**0 - 0.2**	**0.2 - 0.5**	**0.5 - 1.0**
Chairs	5605	5777	1403
Tables	3626	4216	1303
Total	9231	9993	2706

Table 4.2: Distribution of classes and their visibilities in the train split

4.2 Implementation Details

4.2.1 Training strategy

We train both of our modules separately until convergence on a single Nvidia GeForce GTX 1080 Ti. For detection, we select batch size 12 and we use the Adam optimizer with one cycle learning rate scheduler with range 1e-4 to 1e-3 proposed in [34] for the detection module, and another Adam optimizer for with static learning rate of 1e-5 for the center and box size uncertainty heads for 120 epochs. We also employ rotation augmentations up to 45 degrees to improve generalization of our method in scenes with objects in different orientations.

For completion, we set the batch size to 1 frame due to memory constraints, and process all the objects belonging to the frame in by mini-batching, accumulate gradients

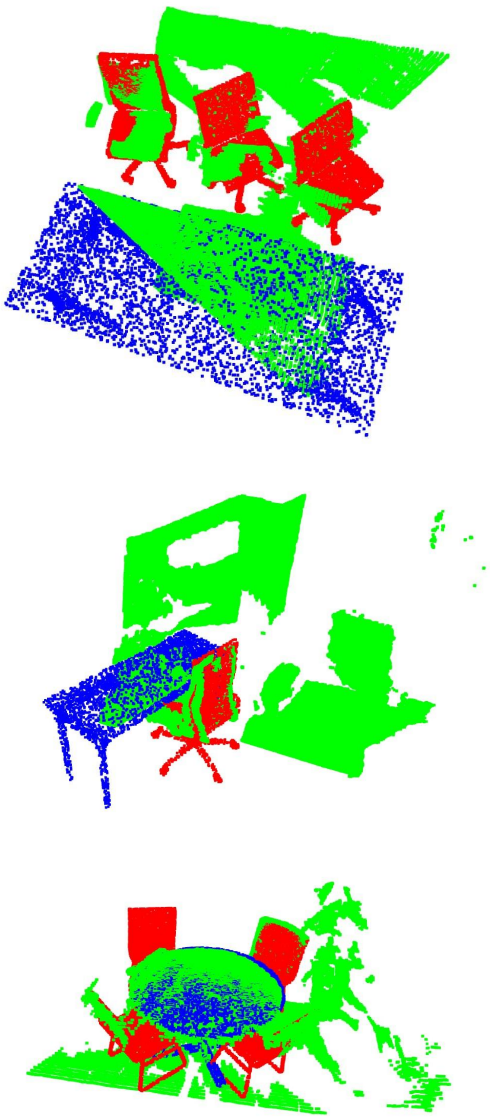

Figure 4.1: Visualization of some of our dataset elements, raw point cloud input is represented with green color, subsampled chairs and tables are shown with red and blue color, respectively.

for all the individual elements in the scene. During training, we limit our operating region to the limits of object bounding boxes , and assign higher weight of $\omega = 3$ to the voxels which belong to unseen ground truth regions to tackle class imbalance for our task as the majority of the operated region is empty space.

4.2.2 Inference

At test time, we put two networks together, and first perform detection on the input point cloud, getting the object proposals for the scene. We then filter out the boxes first with objectness confidence score 0.5, then using 3D NMS with overlap threshold 0.25, but keep the information of learned distribution parameters per box. At this stage, we can generate alternative bounding boxes by sampling from the learned size and center distributions. We then prepare completion input for each detected object (and their alternative suggestions if we are sampling) by extracting their region from the TSDF by setting the rest of the input as zero, then perform completion. Finally, we measure the quality of the each of the sampled boxes by computing the intersection over union with the corresponding ground truth, and select the best suggestion per initial proposal for evaluation.

4.3 Experiments

In this section, we provide a series of quantitative and qualitative evaluations for both of our modules, first separately, then jointly, and then conclude the chapter with ablation studies and limitations of our method.

4.3.1 Object Detection

Quantitative Evaluation

For the task of object detection, we use the standard detection performance metrics in the literature. We compute mAP(mean average precision) and AR(average recall) for each class with 3D IoU(intersection over union) thresholds 0.25 and 0.5.

Metric	**N = 0**	**N = 5**	**N = 10**	**N = 15**	**N = 20**
chair/AP	0.84	0.86	0.86	0.86	0.86
table/AP	0.54	0.58	0.59	0.60	0.61
mAP	0.69	0.72	0.73	0.73	0.74
chair/Recall	0.94	0.96	0.97	0.97	0.97
table/Recall	0.72	0.81	0.83	0.83	0.86
AR	0.83	0.89	0.90	0.90	0.91

Table 4.3: Object detection results with sampling thresholded IoU@0.25

Metric	N = 0	N = 5	N = 10	N = 15	N = 20
chair/AP	0.67	0.77	0.79	0.81	0.81
table/AP	0.11	0.16	0.20	0.22	0.22
mAP	0.39	0.47	0.50	0.52	0.52
chair/Recall	0.79	0.89	0.91	0.93	0.93
table/Recall	0.24	0.37	0.41	0.44	0.46
AR	0.52	0.63	0.66	0.69	0.69

Table 4.4: Object detection results with sampling thresholded IoU@0.5

Effect of sampling in object detection

We observe that our learned distribution is able to give us better center and size suggestions, and as a result, is able to improve total detection performance.We report the most visible improvement in the scores evaluated with IoU@0.5 threshold. Tables Table 4.3 and Table 4.4 show the increase in mAP and AR per class with respect to increasing number of samples when evaluated on our complete visibility range. We also provide ablations with the individual effects of only size sampling and only center sampling for detection. Our results with show that we are able to get 5% increase at mAP@0.25 and 13% increase at mAP@0.5 when we use our learned distribution to create multiple bounding boxes.

Visibility analysis

Due to the nature of our data, majority of the objects in our dataset are only partially represented in the input point cloud, making it harder for the network to predict object boundaries for objects that are poorly represented. We observe that the performance for tables drop by a margin of 5% on tables and 9% on chairs when we only consider objects poorly represented with visibility $\theta \in [0.2, 0.5)$.

The following tables show the detection results for $\theta \in [0.2, 0.5)$ and $\theta \in [0.5, 1.0]$.

Metric	N = 0	N = 5	N = 10	N = 15	N = 20
chair/AP	0.6	0.73	0.75	0.77	0.77
table/AP	0.08	0.12	0.15	0.16	0.17
mAP	0.34	0.42	0.45	0.46	0.47
chair/Recall	0.78	0.89	0.91	0.93	0.93
table/Recall	0.23	0.33	0.36	0.39	0.41
AR	0.5	0.61	0.64	0.66	0.67

Table 4.5: Effect of sampling with $\theta \in (0.2, 0.5]$ and IoU @0.5

Our results show that when we are in ambiguous scenarios with poorly represented objects, our proposed sampling scheme shows consistent gain as the number of samples increases. However when we analyze only the well represented objects, sampling

Metric	N = 0	N = 5	N = 10	N = 15	N = 20
chair/AP	0.8	0.82	0.83	0.83	0.83
table/AP	0.49	0.51	0.52	0.51	0.53
mAP	0.64	0.67	0.67	0.67	0.68
chair/Recall	0.95	0.97	0.97	0.97	0.97
table/Recall	0.77	0.79	0.81	0.82	0.82
AR	0.86	0.88	0.89	0.89	0.90

Table 4.6: Effect of sampling with $\theta \in (0.2, 0.5]$ and IoU @0.25

Metric	N = 0	N = 5	N = 10	N = 15	N = 20
chair/AP	0.75	0.69	0.69	0.69	0.66
table/AP	0.19	0.13	0.16	0.21	0.17
mAP	0.47	0.41	0.42	0.45	0.41
chair/Recall	0.86	0.81	0.80	0.81	0.79
table/Recall	0.33	0.25	0.27	0.36	0.27
AR	0.59	0.53	0.53	0.59	0.53

Table 4.7: Effect of sampling with $\theta \in (0.5, 1.0]$ and IoU @0.5

Metric	N = 0	N = 5	N = 10	N = 15	N = 20
chair/AP	0.86	0.86	0.83	0.85	0.84
table/AP	0.63	0.58	0.61	0.60	0.59
mAP	0.75	0.72	0.72	0.73	0.72
chair/Recall	0.97	0.97	0.95	0.96	0.95
table/Recall	0.85	0.79	0.84	0.83	0.83
AR	0.91	0.88	0.90	0.89	0.89

Table 4.8: Effect of sampling with $\theta \in (0.5, 1.0]$ and IoU @0.25

multiple suggestions actually shows a slight adversarial effect on the metrics on some cases. We believe this is due to the already good quality initial detections, and our learned distributions are unable to improve them any further.

4.3.2 Analysis of the learned distributions

With our probabilistic detection module, we model the box sizes and box centers as Gaussian distributions. As the ground truth distributions for these quantities are unknown, we use the following metrics to evaluate the quality of our learned distributions quantitatively:

- Negative log-likelihood(NLL) of N samples.
- The distance to the closest ground truth value out of N samples.
- Standard deviation of the samples to see the variance.

NLL is the minimization objective when trying to fit a Gaussian distribution to a given data, hence lower NLL values indicated a better learned distribution. When we examine Table 4.9, Table 4.10, Table 4.11, Table 4.12 we observe that the table size distribution is the most challenging one to model, as it outputs the highest NLL values out of all others. Intuitively, compared to chairs, tables come in much more different sizes and leading to a more complex distribution to model. Chair sizes on the other hand do not vary as much. These remarks are also supported by our other metrics, where we observe similar trends. Table sizes have the biggest distance value to the best ground truth size, and also have the highest standard deviation value. These findings also correlate with our ablation on individual samplings, where we see that sampling sizes benefit tables more than it benefits chairs. We also visit the scenario in where instead of using the learned σ, we sampled with the variance randomly sampled from $\mathcal{N}(0, 1)$, and show in Table 4.13 that our learned distribution is more meaningful than random selection.

Metric/# of samples	**N = 5**	**N = 10**	**N = 15**	**N = 20**
Best Distances	0.21	0.2	0.19	0.18
NLL	-7.14	-10.97	-14.80	-18.18
STD	0.06	0.06	0.06	0.06

Table 4.9: Stats for size for chairs distribution.

Qualitative Results

Here we present some qualitative results of our detection module, with the visualizations of some of the detected objects in a scene, and the change in boxes of after the sampling operation.

Metric/# of samples	N = 5	N = 10	N = 15	N = 20
Best Distances	0.32	0.3	0.29	0.28
NLL	-7.408	-11.014	-14.148	-16.852
STD	0.06	0.062	0.064	0.066

Table 4.10: Stats for center for chairs distribution.

Metric/# of samples	N = 5	N = 10	N = 15	N = 20
Best Distances	0.51	0.46	0.43	0.42
NLL	-4.08	-4.48	-5.28	-5.85
STD	0.11	0.12	0.12	0.12

Table 4.11: Stats for size for tables distribution.

Metric/# of samples	N = 5	N = 10	N = 15	N = 20
Best Distances	0.41	0.39	0.37	0.37
NLL	-7.15	-10.67	-13.61	-16.37
STD	0.06	0.06	0.06	0.07

Table 4.12: Stats for center distribution for tables.

Metric	Table/Size	Table/Center	Chair/Size	Chair/Center
Best Distances	0.9	0.6	0.4	0.46
NLL	170.587	1100.186	171.33	311.
STD	0.6	0.6	0.6	0.6

Table 4.13: Stats for random sampling.

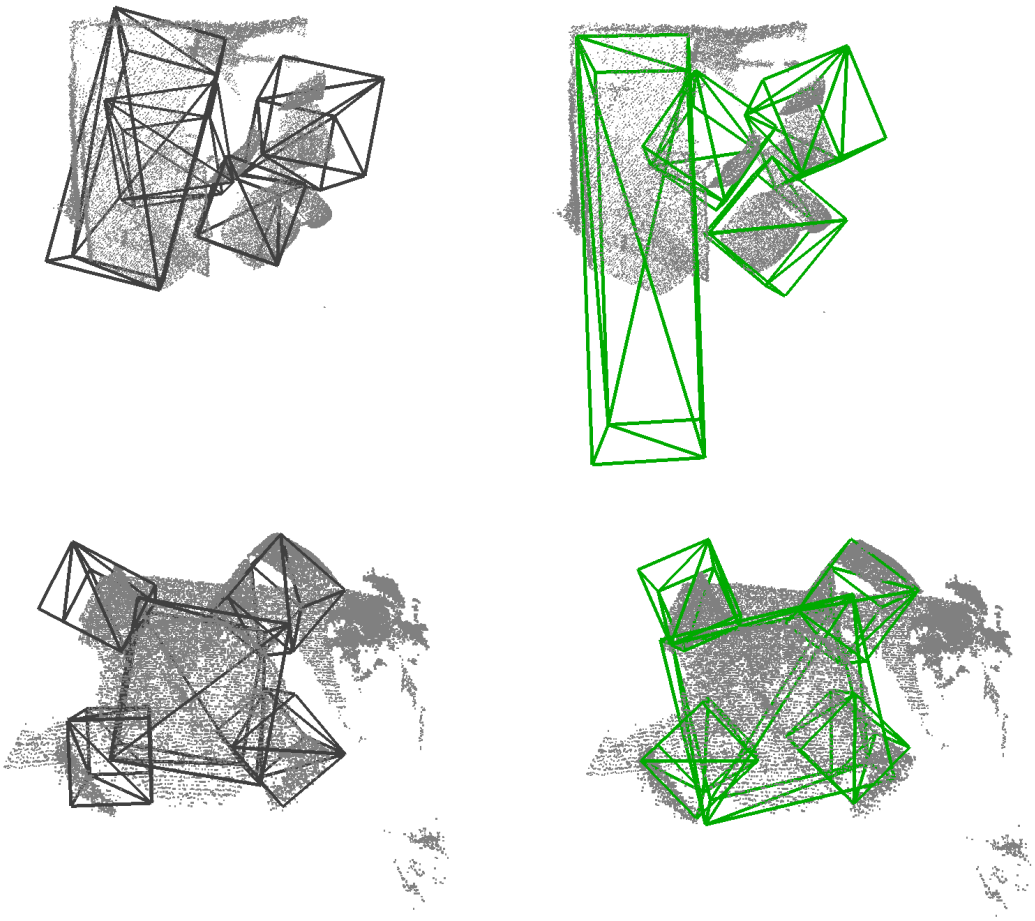

Figure 4.2: From left to right: Our predicted bounding boxes, ground truth bounding boxes.

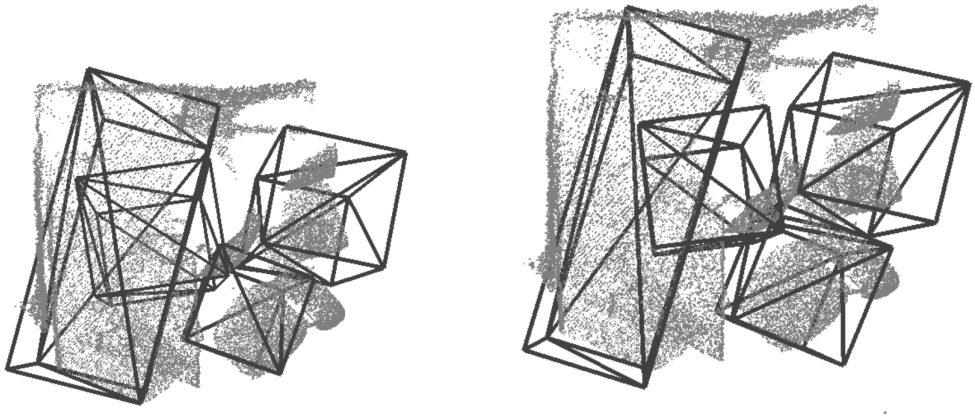

Figure 4.3: Before and after sampling boxes from the distribution, the refinement is most visible for the bottom right box.

4.3.3 Object Completion

Quantitative Evaluation

Our object completion module is class agnostic, it does not distinguish between classes. Hence we evaluate our completion module's binary occupancy outputs with scene IOU, voxelwise precision and recall in the completed areas inside ground truth bounding boxes. We use a TSDF grid and an RGB image features as information in this task, and evaluate quality of the completion only in the designated area inside bounding box. Even then we hit a IoU bottleneck with the completion quality, which is a limitation of our method, and discuss its potential causes in section 4.6

Moreover, we also analyze the performance for the objects in the areas inside and outside of the camera frustum, under different visibility thresholds. As expected, the completions are better for the areas inside the camera frustum, and with more visible inputs. These scores also act as a theoretical upper bound in our main task, as it illustrates the maximum level of completion we can attain in the case of perfectly aligned bounding boxes without any extra or missing boxes.

Qualitative Results

Our qualitative results in Figure 4.4 show that our network is able to generate visually pleasing completions given perfect object boundaries and the features in it. Even with cases with little information, it is able produce a chair-like outcome. However, this is hard to quantify with our current metrics as thin structures like furniture legs can be easily aligned counted as false predictions, even when the prediction as a whole was not completely wrong. This issue is also addressed in section 4.6. Overpredicting or underpredicting occupancy is related to the class weighting in our training strategy and

Metric/Visibility	0.2 - 0.5	0.5-1.0	0.2-1.0
chair/scIoU	0.44	0.47	0.44
chair/Prec	0.51	0.56	0.52
chair/Recall	0.75	0.75	0.75
table/scIoU	0.45	0.52	0.46
table/Prec	0.53	0.62	0.55
table/Recall	0.73	0.76	0.74

Table 4.14: Complete object geometry

Metric/Visibility	0.2 - 0.5	0.5-1.0	0.2-1.0
chair/scIoU	0.48	0.50	0.48
chair/Prec	0.55	0.59	0.55
chair/Recall	0.79	0.76	0.78
table/scIoU	0.48	0.57	0.50
table/Prec	0.58	0.68	0.60
table/Recall	0.74	0.78	0.75

Table 4.15: Object geometry only inside frustum

discussed with more results in the ablation study.

4.3.4 Semantic Instance Completion

Quantitative Evaluation

To evaluate our main task, SIC, we use global metrics mAP at IoU@0.25 and IoU@0.5 thresholds. However, different from the detection module, this time mAP is computed using scene IoU values instead of bounding box IoUs, and if IoU of a prediction and its corresponding ground truth is larger than the threshold, it is accepted as true positive. Additionally, we investigate the effect of sampling on the task of instance completion, and show that by sampling multiple suggestions, we can boost the completion performance.

We see a promising increase of mAP in the table Table 4.16, and infer that most of our detection + completions produce more than 0.25 IoU, but we once again face our completion bottleneck when we look at Table 4.17. As our theoretical bottleneck for completion was on average around 0.47, very few of the detection + completions are able to exceed the IoU threshold of 0.5 with the ground truth completions.

Sampling effect on completions with $\theta \in [0.2, 0.5)$

When we investigate only less visible objects in the scene we observe that the decrease in detection quality also impacts the completion quality together with the small amount of

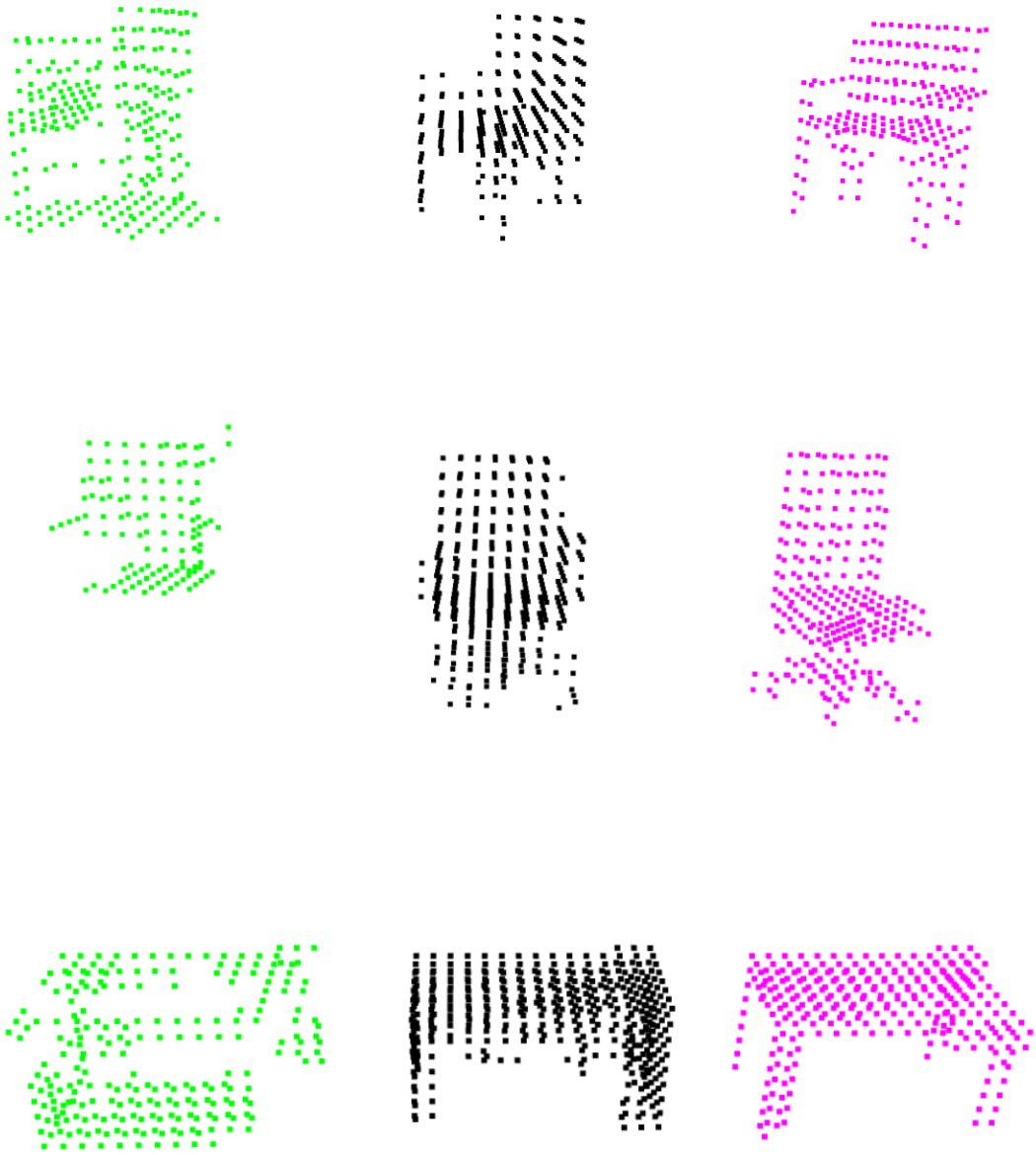

Figure 4.4: From left to right: Input scene, our prediction, ground truth

Metric/# of samples	N=0	N=3	N = 5	N = 10	N = 15	N = 20
chair/Recall	0.82	0.83	0.84	0.88	0.89	0.9
table/Recall	0.65	0.69	0.68	0.73	0.74	0.72
chair/AP	0.71	0.72	0.74	0.78	0.8	0.81
table/AP	0.46	0.5	0.49	0.54	0.55	0.53
mAP	0.59	0.61	0.62	0.66	0.67	0.67

Table 4.16: SIC results at IoU@0.25 and $\theta \in [0.2, 1.0]$

Metric/# of samples	N=0	N=3	N = 5	N = 10	N = 15	N = 20
chair/Recall	0.12	0.13	0.14	0.17	0.2	0.22
table/Recall	0.05	0.05	0.07	0.07	0.09	0.09
chair/AP	0.03	0.03	0.04	0.06	0.07	0.08
table/AP	0.01	0.01	0.01	0.01	0.01	0.01
mAP	0.02	0.02	0.02	0.03	0.04	0.05

Table 4.17: SIC results at IoU@0.5 and $\theta \in [0.2, 1.0]$

present information in the scene and leads to worse mAP scores, but a higher gain from our sampling, showing that refining detections elevates the completion performance, as more meaningful inputs are fed into the completion network.

Metric	N = 0	N = 5	N = 10	N = 15	N = 20
chair/AP	0.56	0.62	0.67	0.69	0.70
table/AP	0.50	0.54	0.57	0.57	0.58
chair/Recall	0.70	0.75	0.78	0.80	0.80
table/Recall	0.64	0.68	0.70	0.69	0.71
mAP	0.53	0.58	0.62	0.63	0.64

Table 4.18: SIC results at IoU@0.25 and $\theta \in [0.2, 0.5]$

Qualitative Results

Our qualitative results in show that our completion performance heavily relies on the quality of our detections, and our model is not very robust against noisy detections, imperfect detection directly leads to an imperfect completion. However this is expected due to the ambiguous nature of the problem, and our proposed sampling strategy is able to elevate the quality of our reconstructions compared to the initial predictions. An example case showing the different completions we can reach with sampling can be found in Figure 4.6.

Metric	N = 0	N = 5	N = 10	N = 15	N = 20
chair/AP	0.02	0.03	0.03	0.04	0.04
table/AP	0.00	0.01	0.00	0.01	0.01
chair/Recall	0.11	0.13	0.15	0.17	0.18
table/Recall	0.04	0.07	0.05	0.07	0.08
mAP	0.01	0.02	0.02	0.02	0.03

Table 4.19: SIC results at IoU@0.5 and $\theta \in [0.2, 0.5]$

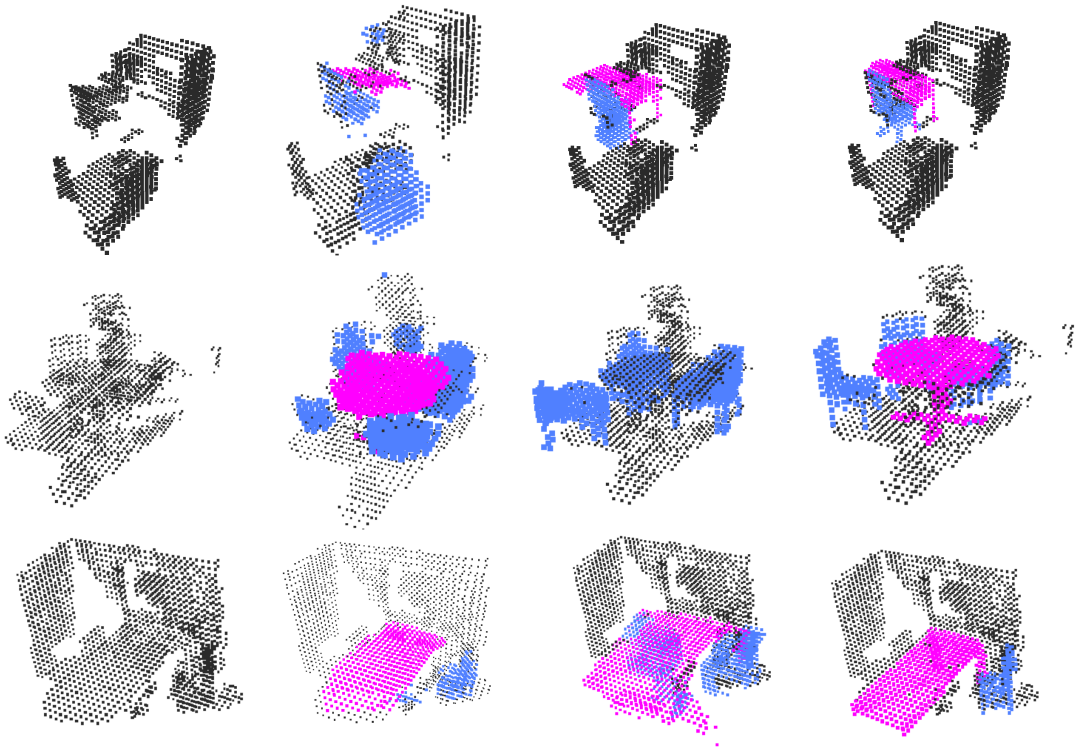

Figure 4.5: Semantic completion results from our joint network. Left to righ: Input depth frame, 3D-Sketch , ours and ground truth.

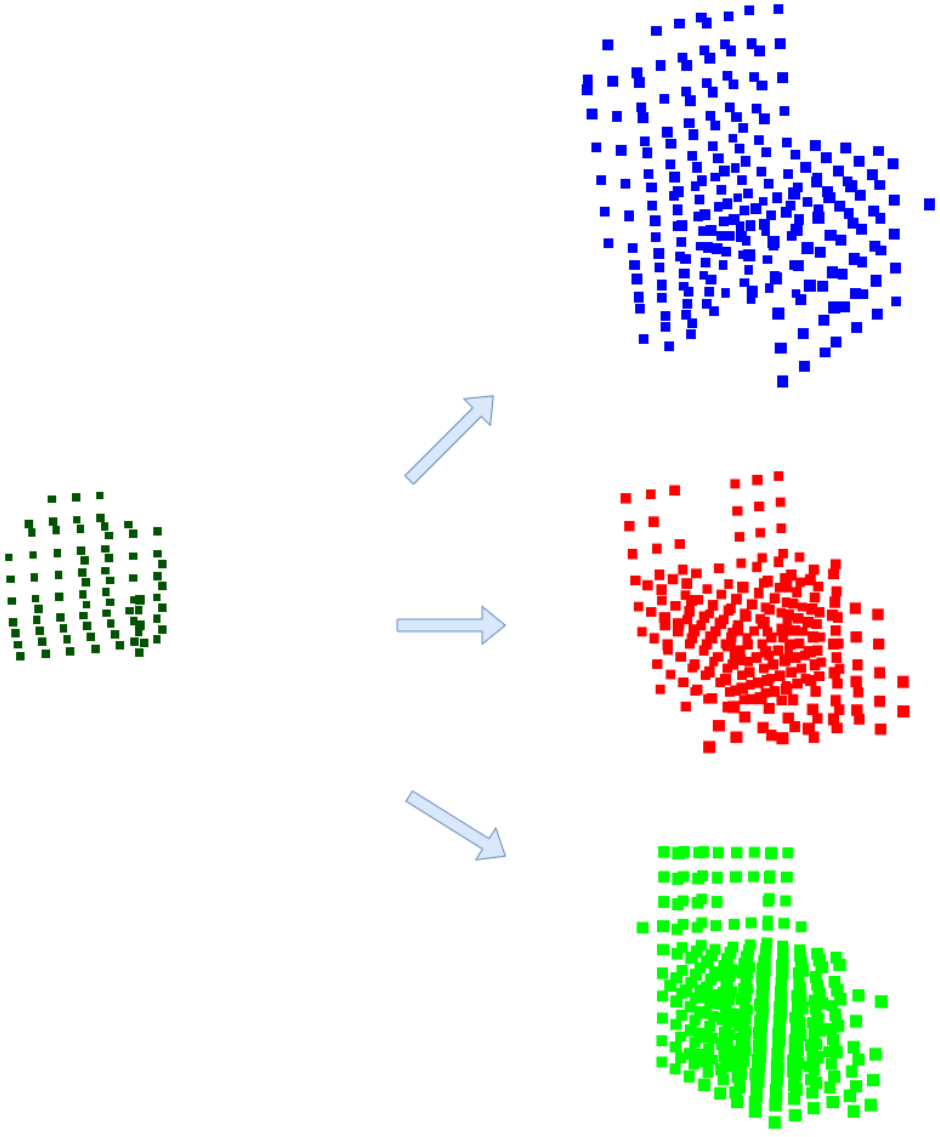

Figure 4.6: Example completion suggestions made with sampled bounding boxes.

4.4 Baseline Comparison

We compare our SIC framework with our original completion baseline, 3D-Sketch architecture for SSC. Since 3D-Sketch is designed for the task of SSC and does not provide instance level outputs, for a fair comparison between the two, we carry our pipeline over to SSC domain by aggregating the detected proposals for objects in the scene and creating a single scene representation where voxels are labeled with their predicted semantic class.

4.4.1 Quantitative Comparison

Method	scIOU	sscmIOU	scPrec	scRec
3D-Sketch Inside Frustum	0.35	0.34	0.45	0.63
Ours Inside Frustum	0.25	0.23	0.28	0.57
Ours Inside Frustum with Sampling N = 10	0.26	0.23	0.29	0.60
3D-Sketch Outside Frustum	0.34	0.33	0.44	0.58
Ours Outside Frustum	0.25	0.22	0.30	0.57
Ours Outside Frustum with Sampling N = 10	0.27	0.25	0.33	0.60

Table 4.20: Comparison of our methods with our baseline

We observe that for the task of semantic scene completion, 3D sketch outperform our method for most of the metrics. We believe the main advantage of 3D-Sketch against our model is that it does not rely on the localization of objects in the scene, and it has access to contextual information from the scene and TSDF and RGB features, giving it more information regarding the scene. As our method heavily relies on object localization, misaligned bounding boxes can lead to imperfect reconstructions, or missed detections directly lead to missed reconstructions.

4.4.2 Qualitative Comparison

Qualitative comparison of SSC with our method can be seen in Figure 4.5. While we are able recover the missing geometry better than 3D-Sketch, it is able to perform better classification and segmentation to the scene, leading to more accurate overall predictions. The biggest impact of this design difference can be seen in the middle row, while our method generated plausible chair geometries, it was unable to detect the table, hence no output for a table in the scene is generated. 3D-Sketch also failed to complete any missing limbs from the table, but was able to extract semantic information from the scene and output the correct class for those pixel.

4.5 Ablation study

In this section, we examine our choices for some hyperparameters used in both detection (λ) and completion (ω).

4.5.1 Different weighting of false negatives on completion from ground truth

We perform ablations on how to weight the voxel occupancy classes to address the problem of class imbalance during completion. Since most of the region to be completed is consisting of empty space, in order to not lose the thin geometric details such as chair and table legs. However if we increase this weight too much while the model recovers more geometric details, it also outputs more false positives, as false negatives are punished harder than false positives. We explore a range of ωs to weight the unseen object regions higher during the loss computation. and force the network to output more predictions by penalizing missing geometry more than incorrectly classified occupancies. We also experiment with Ω, which is the parameter for weighting all the false negatives, without any enforcement the structural information.

Metric	Precision	Recall	IoU
$\Omega = 1, \omega = 1$	0.68	0.63	0.47
$\Omega = 1, \omega = 3$	0.57	0.73	0.47
$\Omega = 1, \omega = 5$	0.46	0.82	0.42
$\Omega = 3, \omega = 1$	0.55	0.78	0.48
$\Omega = 3, \omega = 3$	0.41	0.88	0.39

Table 4.21: Different weighting of the false negatives and their effects on completion with GT boxes

We interpret the results from Figure 4.7 and Table 4.21 and deduct that punishing false negatives more forces the network to predict more occupied cells, and this leads to blockier reconstructions with more false positives giving us a higher recall score, but a lower precision. On the other hand, treating false positives and false negatives the same is also not a viable option due to class imbalance. We are getting way higher precision results, but lower recall scores, unable to predict finer geometric details. We choose ω as 3 and Ω as 1 as a middle ground. Even though we get better quantitative scores when Ω = 3, investigating qualitative results show that uplifting only unseen object structures lead to better reconstructions.

4.5.2 Sampling only size vs only center

We further investigate the individual effects of sampling only size versus sampl, hybrid, and no sampling. We perform N=10 sampling and compare objecte detection metrics for the different configurations.

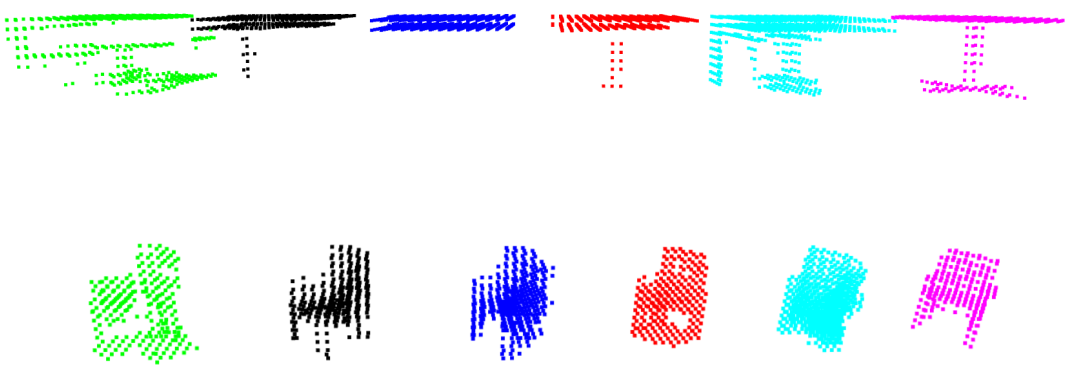

Figure 4.7: Qualitative results for the effect of different weightings. Green corresponds to the input, purple refers to the ground truth and black, red and cyan refer to the different weighting schemes described in , in that order except for the case both ω and Ω is equal to 1.

	Size Sampling	**Center Sampling**	**Dual Sampling**	**No Sampling**
chair/AP	0.80	0.80	0.86	0.84
table/AP	0.57	0.52	0.59	0.54
mAP	0.71	0.67	0.72	0.69
chair/Recall	0.88	0.88	0.97	0.94
table/Recall	0.72	0.69	0.83	0.72
AR	0.80	0.79	0.90	0.83

Table 4.22: Individual sampling comparison at IoU@0.25 and full visibility.

	Size Sampling	**Center Sampling**	**Dual Sampling**	**No Sampling**
chair/AP	0.67	0.68	0.79	0.67
table/AP	0.14	0.18	0.20	0.11
mAP	0.41	0.46	0.50	0.39
chair/Recall	0.78	0.83	0.91	0.79
table/Recall	0.26	0.33	0.41	0.24
AR	0.52	0.58	0.66	0.52

Table 4.23: Individual sampling comparison at IoU@0.5 and full visibility.

Results in Table 4.22 and Table 4.23 indicate that it is hard to entirely separate size sampling and center sampling, as both outperform the other in certain scenarios and using hybrid is the most beneficial approach for benefiting the advantages of both size and center sampling.

4.6 Limitations

One issue with our pipeline is the strong reliance on object detections for completion and the weak connection between our two components, obviously, locating the object in the scene is part of the problem definition, but in our current setting, we operate on a single coordinate system for completion. In order to have a perfect reconstruction in our evaluation, one must generate the object exactly at the place it is located as a ground truth. This forces the network to learn to generate objects in multiple regions of the grid and since we complete objects on a singular, class agnostic basis, we do not infer any contextual information. One of the strong arguments for the task of SSC was that the network should utilize the information in the scene when jointly inferring geometry and semantic class, it should learn that tables tend to have chairs around them.

Another issue stems from the ill-posed nature of our problem, our input signal does not carry rich information about the scene, which is one of the potential reasons our completion has bottleneck. With many different sized tables in the dataset when only partially seeing a table in a frame even humans would have trouble identifying the exact size and geometry of this table, making it very hard for the network to generalize. As a side effect of using dense volumetric grids, we further decrease the amount of information our pipeline processes by reducing the operated resolution.

5 Conclusion

With this thesis, we presented a two-stage probabilistic hybrid method that can locate, classify and complete objects from a single depth frame. Our pipeline utilizes both point cloud and volumetric grid representations and is able to recover object geometries partially represented in the input. Furthermore, we challenge the ambiguity of the task with a probabilistic detection method, and demonstrate that with our learned distribution, we can boost both detection and completion performance.

While there are still significant limitations that introduce performance bottlenecks, we have showed that a single depth image can be used for reconstructing object multiple instances in the scene, and inherent ambiguities of the task can be tackled with uncertainty quantification.

Future work could address the previously mentioned limitations: Weak connection between the two modules, can be tackled by building a stronger internal connection with the network modules by using a uniform data representation, or by keeping the different representations but enforcing a flow of contextual features between the two modules to improve their awareness. The performance bottleneck due to low resolution data can be improved by switching to a data structure capable of handling higher resolution 3D data, such as sparse voxel grids.

List of Figures

List of Tables

Bibliography

[1] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz. "Neural RGB®D Sensing: Depth and Uncertainty From a Video Camera". en. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 10978–10987. ISBN: 978-1-72813-293-8. DOI: 10.1109/CVPR.2019.01124. URL: https://ieeexplore.ieee.org/document/8953381/.

[2] N. Yang, L. von Stumberg, R. Wang, and D. Cremers. "D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry". en. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 1278–1289. ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.00136. URL: https://ieeexplore.ieee.org/document/9157454/.

[3] A.-Q. Cao and R. de Charette. "MonoScene: Monocular 3D Semantic Scene Completion". In: arXiv:2112.00726 (Mar. 2022). arXiv:2112.00726 [cs]. URL: http://arxiv.org/abs/2112.00726.

[4] A. Božič, P. Palafox, J. Thies, A. Dai, and M. Nießner. "TransformerFusion: Monocular RGB Scene Reconstruction using Transformers". In: arXiv:2107.02191 (July 2021). arXiv:2107.02191 [cs]. URL: http://arxiv.org/abs/2107.02191.

[5] J. Hou, A. Dai, and M. Nießner. "3D-SIC: 3D Semantic Instance Completion for RGB-D Scans". In: *CoRR* abs/1904.12012 (2019). arXiv: 1904.12012. URL: http://arxiv.org/abs/1904.12012.

[6] Y. Nie, J. Hou, X. Han, and M. Nießner. *RfD-Net: Point Scene Understanding by Semantic Instance Reconstruction*. arXiv:2011.14744 [cs]. Nov. 2020. URL: http://arxiv.org/abs/2011.14744 (visited on 07/13/2022).

[7] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner. *Scan2CAD: Learning CAD Model Alignment in RGB-D Scans*. Tech. rep. arXiv:1811.11187. arXiv:1811.11187 [cs] version: 1 type: article. arXiv, Nov. 2018. URL: http://arxiv.org/abs/1811.11187 (visited on 06/09/2022).

[8] A. Avetisyan, T. Khanova, C. Choy, D. Dash, A. Dai, and M. Nießner. "SceneCAD: Predicting Object Alignments and Layouts in RGB-D Scans". In: arXiv:2003.12622 (Mar. 2020). arXiv:2003.12622 [cs]. URL: http://arxiv.org/abs/2003.12622.

[9] C. Gümeli, A. Dai, and M. Nießner. "ROCA: Robust CAD Model Retrieval and Alignment from a Single Image". In: arXiv:2112.01988 (Dec. 2021). arXiv:2112.01988 [cs]. URL: http://arxiv.org/abs/2112.01988.

[10] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. "Semantic Scene Completion from a Single Depth Image". In: *arXiv:1611.08974 [cs]* (Nov. 2016). arXiv: 1611.08974 version: 1. URL: http://arxiv.org/abs/1611.08974 (visited on 11/13/2021).

[11] M. A. Alcorn, Q. Li, Z. Gong, C. Wang, L. Mai, W.-S. Ku, and A. Nguyen. "Strike (with) a Pose: Neural Networks Are Easily Fooled by Strange Poses of Familiar Objects". In: arXiv:1811.11553 (Apr. 2019). arXiv:1811.11553 [cs]. DOI: 10.48550/arXiv.1811.11553. URL: http://arxiv.org/abs/1811.11553.

[12] A. Kendall and Y. Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://papers.nips.cc/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html (visited on 04/19/2022).

[13] B. Roessle, J. T. Barron, B. Mildenhall, P. P. Srinivasan, and M. Nießner. "Dense Depth Priors for Neural Radiance Fields from Sparse Input Views". In: *arXiv:2112.03288 [cs]* (Dec. 2021). arXiv: 2112.03288. URL: http://arxiv.org/abs/2112.03288 (visited on 02/15/2022).

[14] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space". In: *arXiv:1706.02413 [cs]* (June 2017). arXiv: 1706.02413. URL: http://arxiv.org/abs/1706.02413 (visited on 04/08/2022).

[15] C. R. Qi, O. Litany, K. He, and L. Guibas. "Deep Hough Voting for 3D Object Detection in Point Clouds". en. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 9276–9285. ISBN: 978-1-72814-803-8. DOI: 10.1109/ICCV.2019.00937. URL: https://ieeexplore.ieee.org/document/9008567/ (visited on 02/26/2022).

[16] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors". In: *arXiv preprint arXiv:2207.02696* (2022).

[17] S. Song, S. P. Lichtenberg, and J. Xiao. "SUN RGB-D: A RGB-D scene understanding benchmark suite". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 567–576. DOI: 10.1109/CVPR.2015.7298655.

[18] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. *ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes*. 2017. arXiv: 1702.04405v2 [cs.CV].

[19] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[20] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. "Frustum PointNets for 3D Object Detection From RGB-D Data". en. In: (), p. 10.

[21] J. Hou, A. Dai, and M. Nießner. "3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans". In: *CoRR* abs/1812.07003 (2018). arXiv: 1812.07003. URL: http://arxiv.org/abs/1812.07003.

[22] D. Rukhovich, A. Vorontsova, and A. Konushin. "Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 2397–2406.

[23] L. Roldao, R. de Charette, and A. Verroust-Blondet. "3D Semantic Scene Completion: a Survey". In: *arXiv:2103.07466 [cs]* (July 2021). arXiv: 2103.07466 version: 3. URL: http://arxiv.org/abs/2103.07466 (visited on 12/28/2021).

[24] S.-C. Wu, K. Tateno, N. Navab, and F. Tombari. "SCFusion: Real-time Incremental Scene Reconstruction with Semantic Completion". In: *2020 International Conference on 3D Vision (3DV)*. arXiv:2010.13662 [cs]. Nov. 2020, pp. 801–810. DOI: 10.1109/3DV50981.2020.00090. URL: http://arxiv.org/abs/2010.13662.

[25] M. Garbade, Y.-T. Chen, J. Sawatzky, and J. Gall. "Two Stream 3D Semantic Scene Completion". In: arXiv:1804.03550 (May 2019). arXiv:1804.03550 [cs]. URL: http://arxiv.org/abs/1804.03550.

[26] J. Zhang, H. Zhao, A. Yao, Y. Chen, L. Zhang, and H. Liao. "Efficient Semantic Scene Completion Network with Spatial Group Convolution". In: arXiv:1907.05091 (July 2019). arXiv:1907.05091 [cs]. URL: http://arxiv.org/abs/1907.05091.

[27] Y.-X. Guo and X. Tong. "View-volume network for semantic scene completion from a single depth image". In: *IJCAI*. 2018.

[28] P. Zhang, W. Liu, Y. Lei, H. Lu, and X. Yang. "Cascaded Context Pyramid for Full-Resolution 3D Semantic Scene Completion". In: *arXiv preprint arXiv:1908.00382* (2019).

[29] X. Chen, K.-Y. Lin, C. Qian, G. Zeng, and H. Li. "3D Sketch-aware Semantic Scene Completion via Semi-supervised Structure Prior". In: *arXiv:2003.14052 [cs]* (Mar. 2020). arXiv: 2003.14052. URL: http://arxiv.org/abs/2003.14052 (visited on 11/13/2021).

[30] Y. Nie, X. Han, S. Guo, Y. Zheng, J. Chang, and J. J. Zhang. "Total3DUnderstanding: Joint Layout, Object Pose and Mesh Reconstruction for Indoor Scenes From a Single Image". en. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 52–61. ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.00013. URL: https://ieeexplore.ieee.org/document/9157512/.

[31] X. He, R. Zemel, and M. Carreira-Perpinan. "Multiscale conditional random fields for image labeling". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 2. 2004, pp. II–II. DOI: 10.1109/CVPR.2004.1315232.

[32] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang. "Bounding Box Regression With Uncertainty for Accurate Object Detection". en. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 2883–2892. ISBN: 978-1-72813-293-8. DOI: 10.1109/CVPR.2019.00300. URL: https://ieeexplore.ieee.org/document/8953889/ (visited on 02/23/2022).

[33] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. *ShapeNet: An Information-Rich 3D Model Repository*. 2015. arXiv: 1512.03012v1 [cs.GR].

[34] L. N. Smith. "A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay". In: *CoRR* abs/1803.09820 (2018). arXiv: 1803.09820. URL: http://arxiv.org/abs/1803.09820.