Adversarial Semantic Scene Completion from a Single Depth Image

Yida Wang, David Joseph Tan, Nassir Navab, Federico Tombari
Technische Universität München
Boltzmannstraße 3, 85748 Garching bei München

Abstract

We propose a method to reconstruct, complete and semantically label a 3D scene from a single input depth image. We improve the accuracy of the regressed semantic 3D maps by a novel architecture based on adversarial learning. In particular, we suggest using multiple adversarial loss terms that not only enforce realistic outputs with respect to the ground truth, but also an effective embedding of the internal features. This is done by correlating the latent features of the encoder working on partial 2.5D data with the latent features extracted from a variational 3D auto-encoder trained to reconstruct the complete semantic scene. In addition, differently from other approaches that operate entirely through 3D convolutions, at test time we retain the original 2.5D structure of the input during downsampling to improve the effectiveness of the internal representation of our model. We test our approach on the main benchmark datasets for semantic scene completion to qualitatively and quantitatively assess the effectiveness of our proposal.

1. Introduction

Inspired by the way humans can imagine the structure of a room by looking at an image, we propose an algorithm that reconstructs the entire scene geometry and semantics from a single depth image. By directly reconstructing the scene from one view, the challenge is to plausibly complete the scene in place of the hidden structures that are not visible from the input depth image. To this end, we utilize a learning strategy that allows the algorithm to simultaneously perceive the objects in the scene and use its contextual shape to fill the hidden structures. In addition, we simultaneously estimate a semantic segmentation of the completed 3D scene geometry.

Reconstructing the environmental information in 3D space from a single viewpoint is relevant for a lot of tasks in the field of augmented reality [24], robotic perception [14] and scene understanding [15], where users and autonomous agents often have only a limited set of observations of the surrounding, and would benefit from a complete semantic

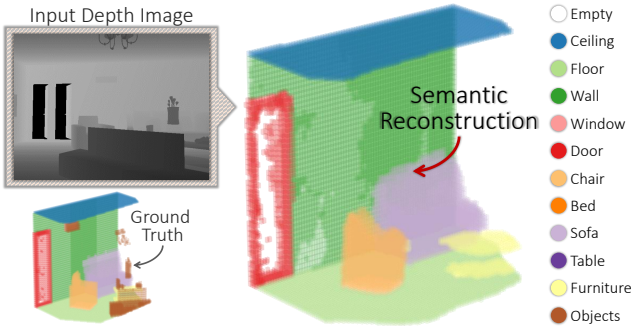

Figure 1: The input depth image and the output semantic 3D reconstruction.

reconstruction of the scene geometry. To push the scientific effort along these directions, recently large-scale benchmark datasets, such as SUNCG [22], NYU [20], ScanNet [5] and SceneNet [8], have been proposed to evaluate different visual scene understanding tasks including those of scene completion and semantic segmentation.

A few methods have recently been proposed in the direction of 3D shape completion. In particular, SSCNet [22] demonstrated good results in the joint task of scene completion and semantic segmentation by means of a CNN [23, 9]. They encode the depth image into volumetric space using the Truncated Signed Distance Function (TSDF) from KinectFusion [17]. Differently, 3D-RecGAN++ [26] suggests using an adversarial approach to learn how to realistically complete partial object shapes from common classes. Generative models have also been proposed to generate 3D data directly from 2D images such as the 3D Inductor [7].

In this work, we focus on the data acquired from depth cameras, with the goal of reconstructing and semantically labeling the whole scene from one single range image. As a scene may contain small objects and complicated shapes, we apply a generative adversarial model for this semantic completion task. Combined with an encoder and a generator, our architecture uses depth images directly as the input information and generate 3D volumetric data whose elements are labeled with object categories. Specifically, we use two discriminators to train the architecture to back-

project the depth information into the 3D volumetric space with semantic labels. One discriminator is used to optimize the entire architecture by comparing the reconstructed semantic scene with the ground truth. Since the 3D variational auto-encoders models the latent features of the volumetric data very well, we designed our architecture such that our encoder for depth images learns similar latent features. To do so, we introduce another discriminator for optimizing the learnt latent features.

To summarize, we have two main contributions. Firstly, we propose the first generative adversarial network aimed at semantic 3D scene completion, and we demonstrate how the adversarial approach is a meaningful choice for the task at hand. Secondly, we enforce adversarial learning not just on the output reconstruction, but also on the latent space to improve the quality of the results. We evaluate our approach on the main benchmark dataset for semantic scene completion to qualitatively and quantitatively assess the effectiveness of our proposal.

2. Related work

Being particularly difficult and training intensive, the task of shape completion from 2.5D has only, with the recent explosion of deep learning, started to become a main research trend in the community. SSCNet [22] proposes a CNN-based architecture that carries out jointly the 3D scene completion and the semantic labeling from a single depth image. A voxel-wise softmax loss function is proposed as the optimizer for learning semantic segmentation of volumetric elements. For training, the method assumes to know the viewpoint as well as the alignment of the depth maps and the reconstructed volumes to a common 3D reference frame. Differently, our approach drops such assumptions and can work without the information regarding the camera pose or the global alignment. On a different task, 3D-RecGAN++ [26] suggests learning a 3D adversarial generative model to complete partial 3D shapes of common object classes. The use of the adversarial loss is motivated to provide realistic and plausible interpolations of the missing shape parts.

Scene and object completion has been investigated also from RGB data. MarrNet [25] proposes to reconstruct 3D object from 2.5D sketches with normal, depth and silhouette information extracted from 2D images. Inspired by MarrNet, the encoder of our architecture is mainly composed of 2D convolutional operators while the generator is mainly composed of 3D deconvolutional operators. The difference lies in the fact that the latent variables of our model are learned to be similar to the feature extracted from a 3D VAE trained on the complete volumetric data.

PointOutNet [6] proposes an encoder-decoder deep architecture to complete 3D objects from RGB images in the form of 3D coordinates. 3D-R^2N^2 [3] tries to reconstruct a volumetric representation of an object from an RGB image by training a recurrent neural network over a latent representation of the RGB data. In addition, by combining scene reconstruction and GAN, 3D-Scene-GAN [27] is introduced for reconstructing complicated 3D scenes from RGB views with mesh and texture by applying a discriminator to distinguish between the rendered 2D images of the scene and real ones.

On a different topic, feature representations for generative models has been often deployed for reconstruction tasks, *e.g.* by means of Variational Auto-Encoders (VAE) [1, 13] and conditional VAE (CVAE) [12, 21], which are two popular methods to learn features from an input data in continuous latent spaces trained via variational inference. 3D VAE [2] is also introduced by replacing 2D convolutional kernels with 3D kernels for auto-encoding voxel data.

3. Semantic reconstruction

The semantic reconstruction algorithm takes a single view of the scene, depicted by a depth image x, to predict its 3D volumetric representation y. The voxels of y are semantically labeled with N_c object classes, denoted as an $N_c \times 1$ one-hot vector, *i.e.* a binary vector where one of its element has a value of 1 to indicate the object category while the other elements remain zero. Considering that the image has a limited view of the entire scene, constrained by the sensor's viewpoint, the objective of our deep learning approach is also to complete the scene by revealing the hidden structures that are not visible in the input. Therefore, simulateneously learning the geometric structure and the semantic information allows the algorithm to learn the contextual cues that can in turn represent the objects in the reconstruction.

Specifically, the depth image is a 640×480 image that represents the z-axis of the camera coordinate system. As input to our deep learning architecture, this image is downsampled to 320×240 in order to conserve GPU memory. The resulting volumetric reconstruction is represented by N_c grids of size $40 \times 80 \times 80$ filled with binary elements presenting the labels for each of the N_c objects. For simplicity, we denote this 4D data as $40 \times 80 \times 80 \times N_c$.

From the depth image to the 3D volume, our architecture is a concatenation of an encoder $\mathcal{E}_{\text{dep}}$ with 2D convolutional operators that convert the input depth image into a lower-dimensional latent feature l_{dep}; and, a generator $\mathcal{G}$ with 3D deconvolutional kernels that takes l_{dep} to build the semantic reconstruction. This architecture is illustrated in Fig. 2.

Encoder for depth image. The encoder $\mathcal{E}_{\text{dep}}$ compresses the depth image into a feature in the latent space. Its architecture is a concatenated network that sequentially combines 2D convolutional layers and max-pooling layers. The operators for the paired convolutional and pooling layers

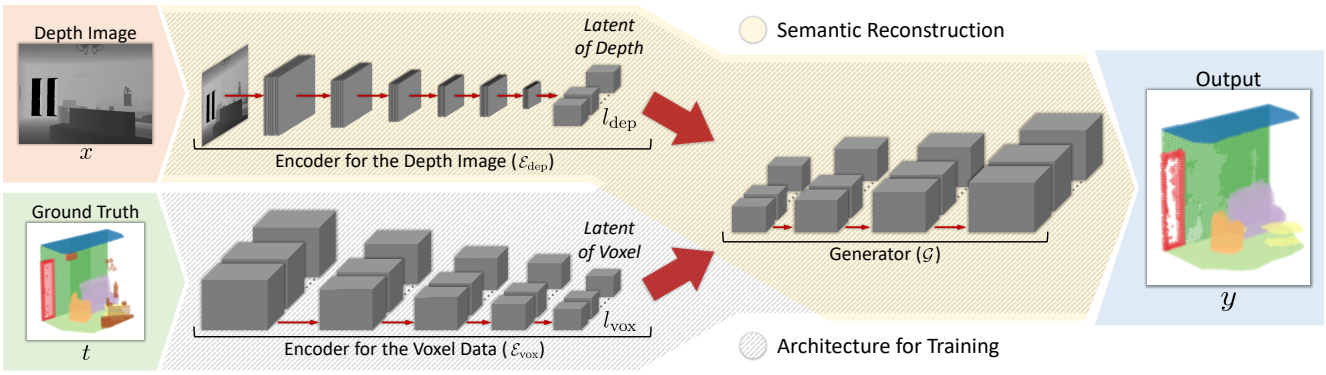

Figure 2: Deep architecture for the semantic reconstruction in Sec. 3 and for the training procedure in Sec. 4. The former is the concatenated architecture of the encoder $\mathcal{E}_{\text{dep}}$ and the generator $\mathcal{G}$ reconstructs from the depth image to a voxel data while the latter is the concatenated architecture of $\mathcal{E}_{\text{vox}}$ and $\mathcal{G}$ is a 3D variational auto-encoder [2] for self-reconstruction.

are 2D convolutional kernels with, respectively, the size of 3×3 and stride of 1×1 and the size of 2×2 with stride of 2×2. Each of these paired layers is processed by a leaky ReLU activation function [16]. Therefore, the output of every ReLU activation is a multi-channel 2D image. After six convolutions operations, the result is an 80-channel 5×3 image which is reshaped into a set of 3D volume of size $5 \times 3 \times 5 \times 16$. The output of the encoder represents the latent feature l_{dep} of the semantic reconstruction architecture.

Generator. With the goal of regressing the semantic reconstruction, the generator $\mathcal{G}$ unwraps the latent feature to a higher dimensional voxel data. We assemble the generator with 3D deconvolutional layers with the size of $3 \times 3 \times 3$ and stride of $2 \times 2 \times 2$ which are processed by the ReLU function as activation. After four deconvolutional layers, the output of the generator is the voxel-wise classification y. By doing this, y is presented in the shape of $80 \times 48 \times 80 \times N_c$.

4. Architecture for training

Although our semantic reconstruction algorithm in Sec. 3 could be optimized only with encoder $\mathcal{E}_{\text{dep}}$ and generator $\mathcal{G}$, the performance after training this way is subpar (see Sec. 7.1). Hence, we include three components during the training process to improve the performance – (1) the encoder for the voxel data, (2) the discriminator for the reconstruction and (3) the discriminator for the latent features.

Specifically, we introduce another encoder $\mathcal{E}_{\text{vox}}$ to extract the feature l_{vox} such that the latent feature from the encoder $\mathcal{E}_{\text{dep}}$ is driven to be similar to a feature extracted from $\mathcal{E}_{\text{vox}}$. Thus, a discriminator $\mathcal{D}_l$ is used to optimize this similarity as illustrated in Fig. 3 and consequently updates the parameters in $\mathcal{E}_{\text{dep}}$. Notably, $\mathcal{E}_{\text{vox}}$ is optimized together with the generator $\mathcal{G}$ as a 3D variational auto-encoder (3D VAE) [2] to learn meaningful weights from training samples representing complete 3D semantic volumes.

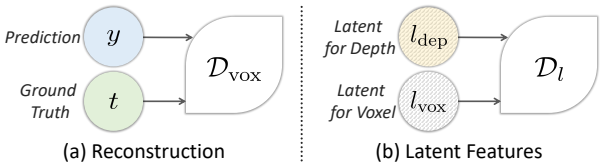

Figure 3: Variables associated to each discriminator.

Encoder for the voxel data. Since reconstructing from one image has a restrictive view of the scene, we want to make the latent features l_{dep}, extracted from the depth image, to be similar to the complete volumetric data in order to incorporate structures that are not visible from the input. We introduce another encoder $\mathcal{E}_{\text{vox}}$ to extract the feature l_{vox} into the architecture for learning. The input of $\mathcal{E}_{\text{vox}}$ is the ground truth volumentric data with semantic labels such that all the operators in its architecure are 3D convolution kernels with the size of $3 \times 3 \times 3$ and stride of $2 \times 2 \times 2$ as shown in Fig. 2. The last layer of the encoder $\mathcal{E}_{\text{vox}}$ produces 16 blocks. The size of the output from $\mathcal{E}_{\text{vox}}$ is set to be the same as $\mathcal{E}_{\text{dep}}$ because we want to make the latent representation of the input depth images to be as similar as possible to that of the ground truth volumetric representations. Thus, measuring the similarity between l_{dep} and l_{vox} is possible because the latent representation compresses the results for both the depth and voxel data. As illustrated in Fig. 2, the latent features from both the encoders $\mathcal{E}_{\text{dep}}$ and $\mathcal{E}_{\text{vox}}$ go through the same generator $\mathcal{G}$ to predict semantic volumetric data.

Discriminator for the reconstruction. Encouraged by the benefits of the generative models trained with adversarial techniques [4], we introduce the discriminator $\mathcal{D}_{\text{vox}}$ in training to optimize our semantic reconstruction by comparing our prediction against the ground truth as shown in Fig. 3. The architecture of $\mathcal{D}_{\text{vox}}$ is similar to the encoder $\mathcal{E}_{\text{vox}}$ except for the last layer. In Fig. 4 (a), all the four 3D

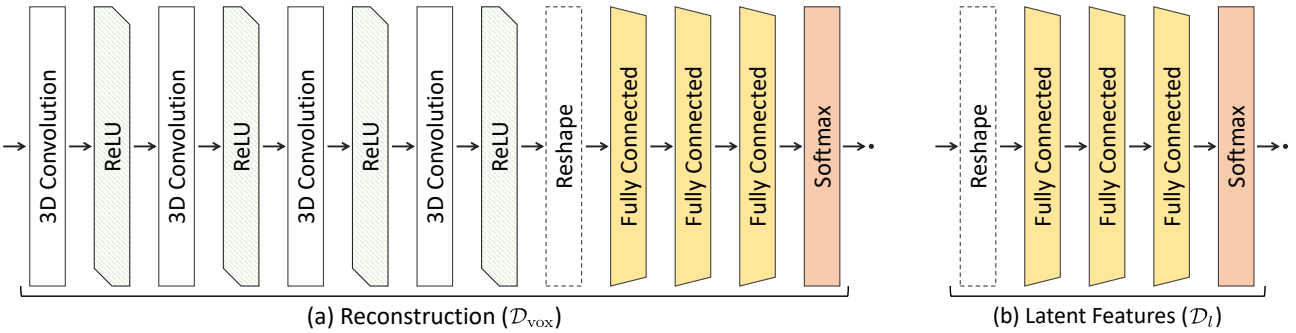

Figure 4: Architecture of the two discriminators.

convolutions have $3 \times 3 \times 3$ kernels with stride of $2 \times 2 \times 2$. Then, the output of the last convolutional layer with the size of $5 \times 3 \times 5 \times 16$ is reshaped to a vector of 1200 dimensions. This is processed by three fully-connected layers with output sizes, respectively, of 256, 128 and 1. Hence, the final logit is a binary indicator to determine whether the predicted volumetric data is the expected ones or not, which is widely used in GAN [19].

Discriminator for the latent features. Since the output of $\mathcal{E}_{\text{dep}}$ and $\mathcal{E}_{\text{vox}}$ are passed to the same generator $\mathcal{G}$, the resulting latent feature from the depth image l_{dep} is driven to be similar to the feature extracted from the ground truth volumetric data l_{vox}. We introduce another discriminator $\mathcal{D}_l$ aiming at distinguishing the latent descriptors illustrated in Fig. 3 and consequently updating the parameters in $\mathcal{E}_{\text{vox}}$. As input to $\mathcal{D}_l$, the latent variables are reshaped from $5 \times 3 \times 5 \times 16$ to a vector of 1200 dimensions. The architecture of $\mathcal{D}_l$ in Fig. 4 (b) is constructed purely by three fully-connected layers with output sizes of 256, 128 and 1. Finally, the output of the discriminator is also a logit where 1 indicates that the latent feature from the depth image is similar to the feature of the 3D VAE; otherwise, the value is zero.

5. Optimization

The goal of the optimization is to enforce the latent features from the depth image (l_{dep}) and the predicted reconstruction (y) to resemble the latent features of the 3D VAE (l_{vox}) and the ground truth volumetric data (t), respectively. Since the architecture for the semantic reconstruction and the 3D VAE share the same generator (see Fig. 2), we distinguish their results by denoting y_x as the prediction from the semantic reconstruction while y_t from the 3D VAE.

Loss functions. When we solely consider the semantic reconstruction architecture (see Sec. 3), the loss that compares the prediction and the ground truth for all the N_c objects is represented as

$$\mathcal{L}_{x \to y}(\mathcal{E}_{\text{dep}}, \mathcal{G}) = \sum_{c=1}^{N_c} [\epsilon(y_x(c), t(c))] \tag{1}$$

where we define the per-object error as

$$\epsilon(q, r) = -\gamma r \log q - (1-\gamma)(1-r) \log(1-q) \tag{2}$$

with γ as the hyper-parameter which weighs the relative importance of false positives against false negatives. Consequently, the error penalizes when the prediction and the ground truth are distinct.

To further improve the reconstruction performance using GAN (see Sec. 4), we include an adversarial loss

$$\mathcal{L}_{\text{GAN-}y}(\mathcal{E}_{\text{dep}}, \mathcal{G}) = -\log(\mathcal{D}_{\text{vox}}(y_x)) \tag{3}$$

based on the trained discriminator $\mathcal{D}_{\text{vox}}$ that optimizes the semantic reconstruction architecture by updating the parameters of its encoder and generator. On the other hand, training for the parameters in $\mathcal{D}_{\text{vox}}$ entails a loss function

$$\mathcal{L}_{\text{GAN-}y}(\mathcal{D}_{\text{vox}}) = -\log(\mathcal{D}_{\text{vox}}(t)) - \log(1 - \mathcal{D}_{\text{vox}}(y_x)) \tag{4}$$

so that the discriminator $\mathcal{D}_{\text{vox}}$ could be further optimized to be capable of distinguishing the generated volumetric data from the ground truth.

As for the 3D VAE, we can train this architecture by minimizing a loss similar to (1). However, since we use the ground truth reconstruction as the input, the loss function

$$\mathcal{L}_{t \to y}(\mathcal{E}_{\text{vox}}, \mathcal{G}) = \sum_{c=1}^{N_c} [\epsilon(y_t(c), t(c))] \tag{5}$$

enforces the predicted reconstruction y_t to be similar to its input. By training with variational inference by optimizing the evidence lower bound (ELBO) [1, 13], the latent variables are distributed in a simple Gaussian distribution.

In reference to semantic reconstruction architecture, the 3D VAE influences the latent variable l_{dep} to be as similar

to l_{vox} as possible by using discriminator $\mathcal{D}_l$ to determine whether l_{dep} is presented similar to l_{vox}. Therefore, similar to $\mathcal{D}_{\text{vox}}$, optimizing the similarity between the latent features uses another discriminator $\mathcal{D}_l$ such that the loss function to update the encoder $\mathcal{E}_{\text{dep}}$ is

$$\mathcal{L}_{\text{GAN-}l}(\mathcal{E}_{\text{dep}}) = -\log(\mathcal{D}_l(l_{\text{dep}})) \tag{6}$$

while training for $\mathcal{D}_l$ involves

$$\mathcal{L}_{\text{GAN-}l}(\mathcal{D}_l) = -\log(\mathcal{D}_l(l_{\text{vox}})) - \log(1 - \mathcal{D}_l(l_{\text{dep}}))\,. \tag{7}$$

Minimization. Now that we have all the loss functions, our optimization is defined as a combination of five components. The first two are based on the architecture for training in Fig. 2. From the depth image x and the ground truth t, we separately train them one after the other for the samples in a mini-batch with

$$\min(\mathcal{L}_{x\to y}(\mathcal{E}_{\text{dep}}, \mathcal{G})) \text{ and} \tag{8}$$
$$\min(\mathcal{L}_{t\to y}(\mathcal{E}_{\text{vox}}, \mathcal{G})) \tag{9}$$

so that the parameters of the architectures are updated alternatively. At the same time, the variational inference sets a constraint on the latent variables as a Gaussian distribution which makes it easier for the output of both of the encoders to match with each other.

Assuming that the discriminators are trained, we can fix their parametric model in order to update the encoder to move towards

$$\min(\mathcal{L}_{\text{GAN-}l}(\mathcal{E}_{\text{dep}})) \tag{10}$$

while update both the encoder and the generator toward

$$\min(\mathcal{L}_{\text{GAN-}y}(\mathcal{E}_{\text{dep}}, \mathcal{G})) \tag{11}$$

which are also optimized alternatively.

Finally, the two discriminators are trained by minimizing

$$\min(\mathcal{L}_{\text{GAN-}y}(\mathcal{D}_{\text{vox}})) \text{ and} \tag{12}$$
$$\min(\mathcal{L}_{\text{GAN-}l}(\mathcal{D}_l)) \tag{13}$$

such that the former is used to penalize poorly reconstructed voxel data in reference to the ground truth while the latter makes the latent codes computed from the depth image similar to the latent feature extracted from a well trained 3D VAE. Notably, the discriminators are updated when the accuracy in distinguishing the generated outputs are lower than specific level [4]. We set this threshold to 15% in our experiments.

In practice, we use the Adam optimizer [11] with a learning rate of 0.0001.

6. Implementation details

We use the paired depth image and semantically labeled volumes provided by SUNCG [22] and NYU [20]. The size of volumetric data with the object labels is $240 \times 240 \times 240 \times N_c$ where N_c is set to 12. Due to the limited GPU memory, we down-sample the data to $80 \times 48 \times 80 \times N_c$ by max-pooling with $3 \times 3 \times 3$ kernel and $3 \times 3 \times 3$ stride. In this manner, the original volumetric data is presented in a space with a lower resolution which is suitable for training in a single GPU with no more than 12 GB memory. In our experiments, we use a single NVIDIA TITAN Xp for training and the batch size is set to be 8. The depth images are also resized from 640×480 to 320×240 with a bilinear interpolation.

The 12 object classes in our experiments are based on SUNCG [22] that includes: empty space, ceiling, floor, wall, window, door, chair, bed, sofa, table, furniture and small objects. Since the ratios of samples in each categories are not balanced, we redesign the evaluation strategy in Sec. 7 to concentrate on reconstructing important objects in the indoor condition with small amount of voxels such as furnitures and small objects.

7. Experiments

We evaluated on the SUNCG dataset [22] that includes pairs of depth images and the corresponding semantically labelled 3D reconstructions.

Evaluation Strategy. Considering that this dataset is for the indoor environments, over 90% of the reconstructed scene is empty. Then, when we exclude the empty spaces, simple structures such as the wall, floor and ceiling dominate the voxels in the scene. This means that the ratio of the number of voxels for different object classes is not balanced. For instance, we noticed that the SUNCG test sets [22] do not have enough small objects and furnitures. In this case, if the learned architecture enhances its ability to predict the empty spaces and the simple structures, their accuracy is significantly higher than the results predicted by an architecture that focuses on distinguishing the other object classes.

Since the ratio of voxels for small objects and furnitures in the training dataset are higher than the one in the test set in SUNCG [22], we design a 10-fold cross validation by splitting the training data which was introduced by [10]. The entire dataset is divided into ten folds with the same amount of samples, the evaluation procedure then uses 1 of the 10 folds as the test set and the remaining 9 as the training dataset. Thereafter, the final result is the average of the ten evaluations.

	empty	ceil.	floor	wall	win.	door	chair	bed	sofa	table	furn.	objs.	*Avg.*
3D VAE [2]	49.3	26.1	33.2	29.7	14.4	4.6	0.7	16.4	13.9	0.0	0.0	0.0	30.8
3D-RecGAN++ [26]	49.3	32.6	37.7	36.0	23.6	13.6	8.7	20.3	16.7	9.6	0.2	3.6	36.1
Ours without $\mathcal{D}_l$	49.6	42.0	35.9	44.8	28.5	25.5	15.4	28.6	20.1	21.5	11.5	6.5	42.7
Ours without $\mathcal{D}_{\text{vox}}$	49.6	39.0	35.7	43.4	26.8	23.8	18.5	29.2	22.4	16.8	10.4	5.3	41.7
Ours (*Proposed*)	49.7	41.4	37.7	45.8	26.5	26.4	21.8	25.4	23.7	20.1	16.2	5.7	**44.1**

Table 1: Semantic scene completion results on the SUNCG test set with depth map for IoU in percentage.

	empty	ceil.	floor	wall	win.	door	chair	bed	sofa	table	furn.	objs.	*Avg.*
3D VAE [2]	99.6	18.8	68.9	63.6	25.0	8.5	4.2	16.4	9.5	1.3	0.4	2.6	65.6
3D-RecGAN++ [26]	99.9	21.5	76.2	78.8	31.9	15.3	8.1	18.7	10.2	2.9	1.4	4.3	79.4
Ours without $\mathcal{D}_l$	100.0	29.1	72.8	92.9	29.7	20.2	9.9	20.8	13.5	2.6	6.2	3.0	92.3
Ours without $\mathcal{D}_{\text{vox}}$	99.9	28.6	70.3	91.5	28.3	18.8	9.1	20.2	12.7	2.6	4.9	2.6	90.1
Ours (*Proposed*)	100.0	29.1	76.2	94.2	32.0	22.7	11.4	21.9	14.2	3.1	7.6	3.6	**94.5**

Table 2: Semantic scene completion results on the SUNCG test set with depth map for mAP in percentage.

Metric. We evaluate the performance of the reconstructor based on the intersection over union (IoU) and the mean average precision (mAP) of the predicted voxel labels compared to ground truth labels [22] where we evaluate the IoU of each object classes on both the observed and occluded voxels for semantic scene completion. Notably, instead of taking the average IoU and mAP as the mean of the results from individual categories, we calculate the average with respect to the number of voxels in each category.

Comparison. We compare our results against 3D VAE [2] and 3D-RecGAN++ [26]. In order to directly estimate the volumetric reconstruction solely from the input depth image, we modify [2, 26] by scaling the surface generated by the depth image through bilinear interpolation to fit the $80 \times 48 \times 80$ volumetric grid which serves as the input to [2, 26]. Furthermore, we added the loss function from (1) in training to perform semantic segmentation. Notably, the U-Net [18] connection between encoder and decoder in 3D-RecGAN++ [26] are still applied by resizing the scale of every layers. In addition, we further investigate the advantage of the discriminators by evaluating our approach without $\mathcal{D}_l$ and $\mathcal{D}_{\text{vox}}$. Based on Sec. 5, when implementing our approach without $\mathcal{D}_l$, (10) and (13) are discarded in the optimization; while, the implementation without $\mathcal{D}_{\text{vox}}$ discards (11) and (12).

7.1. SUNCG

SUNCG [22] is a dataset of 3D scenes which contains pairs of depth image and its corresponding volumetric scene where all objects in the scene are semantically annotated. We implemented the 10-fold validation on the pairs for the 111,697 different scenes.

Comparison against other approaches. The evaluation on both the IoU and mAP in Table 1 and Table 2 shows that our generative model performs better than 3D VAE [2] and 3D-RecGAN++ [26] which are the recent works on 3D generative architectures. We acquired an IoU of 44.1% and an mAP of 94.5% that is 8% and 15.1% better than the next best performing approach.

Comparison on the architecture for learning. To understand the advantage of incorporating the components in learning, we investigate learning our method without the discriminators. Without the discriminator for the latent features, our performance decreases by 1.4% in IoU and 2.2% in mAP; while, without the discriminator for the reconstruction, the results decrease by 2.4% in IoU and 4.4% in mAPo. However, it is noteworthy to mention that, even without these discriminators, our method still achieves better results compared to both 3D VAE [2] and 3D-RecGAN++ [26].

Performance on smaller objects. If we look closely on Table 1, our approach has a significant improvement over 3D VAE [2] and 3D-RecGAN++ [26] on smaller objects like the class of table, furniture and objects wherein [2] produced an IoU of zero. The reason behind this improvement is because the adversarial training is especially helpful in reconstructing and completing small objects compared to 3D VAE [2]. Note that these results are also validated by evaluating the mAP in Table 2.

Since the latent space is continouos, this implies that it reserves regions for the object classes with a smaller amount of voxels in the scene or a fewer samples in the training dataset. Therefore, while all methods can reconstruct the common objects such as the ceiling, floor and walls with

	empty	ceil.	floor	wall	win.	door	chair	bed	sofa	table	furn.	objs.	*Avg.*
3D VAE [2]	49.4	33.3	25.3	32.4	16.9	9.3	5.6	19.2	14.7	1.1	0.0	0.0	31.5
3D-RecGAN++ [26]	49.6	35.1	31.8	39.2	23.7	17.9	11.5	26.1	22.6	18.1	5.1	3.0	37.7
Ours without $\mathcal{D}_l$	49.6	42.4	35.8	44.4	29.2	24.8	17.2	30.6	24.2	19.5	11.5	4.4	42.4
Ours without $\mathcal{D}_{\text{vox}}$	49.7	43.9	37.3	45.9	26.7	29.2	20.1	24.0	24.6	26.1	19.8	9.0	44.3
Ours (*Proposed*)	49.8	49.6	42.7	51.2	24.2	34.9	23.0	28.1	30.4	29.9	22.0	11.5	**51.4**

Table 3: Semantic scene completion results finetuned on the NYU training set with real world depth map for IoU in percentage.

	empty	ceil.	floor	wall	win.	door	chair	bed	sofa	table	furn.	objs.	*Avg.*
3D VAE [2]	99.8	25.0	53.8	70.9	19.3	7.4	4.2	14.3	9.4	1.1	1.2	0.9	68.4
3D-RecGAN++ [26]	99.9	27.3	67.5	87.6	27.0	15.8	8.0	19.2	12.0	2.2	3.4	1.8	86.5
Ours without $\mathcal{D}_l$	100.0	28.9	72.1	92.7	29.6	19.8	9.9	20.8	13.3	2.7	6.6	2.9	91.9
Ours without $\mathcal{D}_{\text{vox}}$	100.0	29.2	76.8	94.5	31.9	22.6	11.5	21.9	14.2	3.2	8.2	4.1	94.8
Ours (*Proposed*)	100.0	30.8	79.1	96.6	35.4	26.9	17.0	13.9	15.8	3.6	9.7	5.5	**97.2**

Table 4: Semantic scene completion results finetuned on the NYU training set with real world depth map for mAP in percentage.

correct labels as illustrated in both Table 1 and Table 2, the main advantage of our work is the capacity to reconstruct and classify every type of object labels.

Qualitative results. We illustrate the qualitative results in Fig. 5 and compare them with 3D VAE [2] and the ground truth. Based on these voxel representations, we can clearly visualize the superiority of our algorithm to reconstruct more detailed structures compared to [2]. Therefore, this confirms the advantage of our approach to reconstruct not only the larger structures but also the smaller objects in the scene.

7.2. Fine-tune with NYU

The objective of this section is to investigates whether an increase in the size of the learning dataset from a different source can improve the performance of the algorithm or confuse the learned model.

In this section, we include the NYU dataset [20] which is also an indoor scene dataset. It contains both the depth images captured by Kinect and the 3D models. This includes the volumetric 3D data with the annotated object labels for every voxels in 1,449 scenes. The semantic annotations for the volumetric data in this dataset consist of 33 objects in 7 categories. Note that, due to the limited amount of 1,449 volumetric scenes from the NYU dataset, this size is insufficient to learn a deep learning architecture. Thus, we only use the NYU to supplement our training dataset while testing on SUNCG for the 12 categories. This requires us to relabel the object classes of the volumetric data in NYU to match the labels provided by SUNCG dataset.

Comparison against other approaches. Similar to Sec. 7.1, this procedure is implemented on all the five approaches that we are comparing. While fine-tuning with the NYU dataset, our experiments show that the combination of the two datasets improve the perfomance of our algorithm. From Table 1 to Table 3 and Table 2 to Table 4, we experience an increase in IoU by 7.3% and in mAP by 2.7%. Although both the 3D VAE [2] and 3D-RecGAN++ [26] also experienced an increase in performance, the difference is not significant which counts for a maximum of 1.6% increase in IoU. Note that, in Table 3, the results on smaller objects for 3D VAE [2] remains close to zero or zero.

Comparison on the architecture for learning. When we learn our architecture with the discriminators, the effect of the improvement is negligible. Without the discriminator for the latent features, the IoU even decreased from 42.7% to 42.4%; while, without the discriminator for the reconstruction, the IoU increases only from 41.7% to 44.3%. Therefore, based on this experiment, we can attribute the significant improvement of our work's perfomance to the discriminators in the training architecture.

8. Conclusion

We have proposed a novel approach for semantic scene completion from a single depth map, which exploits the power of adversarial training to regress accurate reconstructions without the need of additional assumptions or the camera pose information. Our proposal relies on the enforcement of two adversarial losses – one aimed at mak-

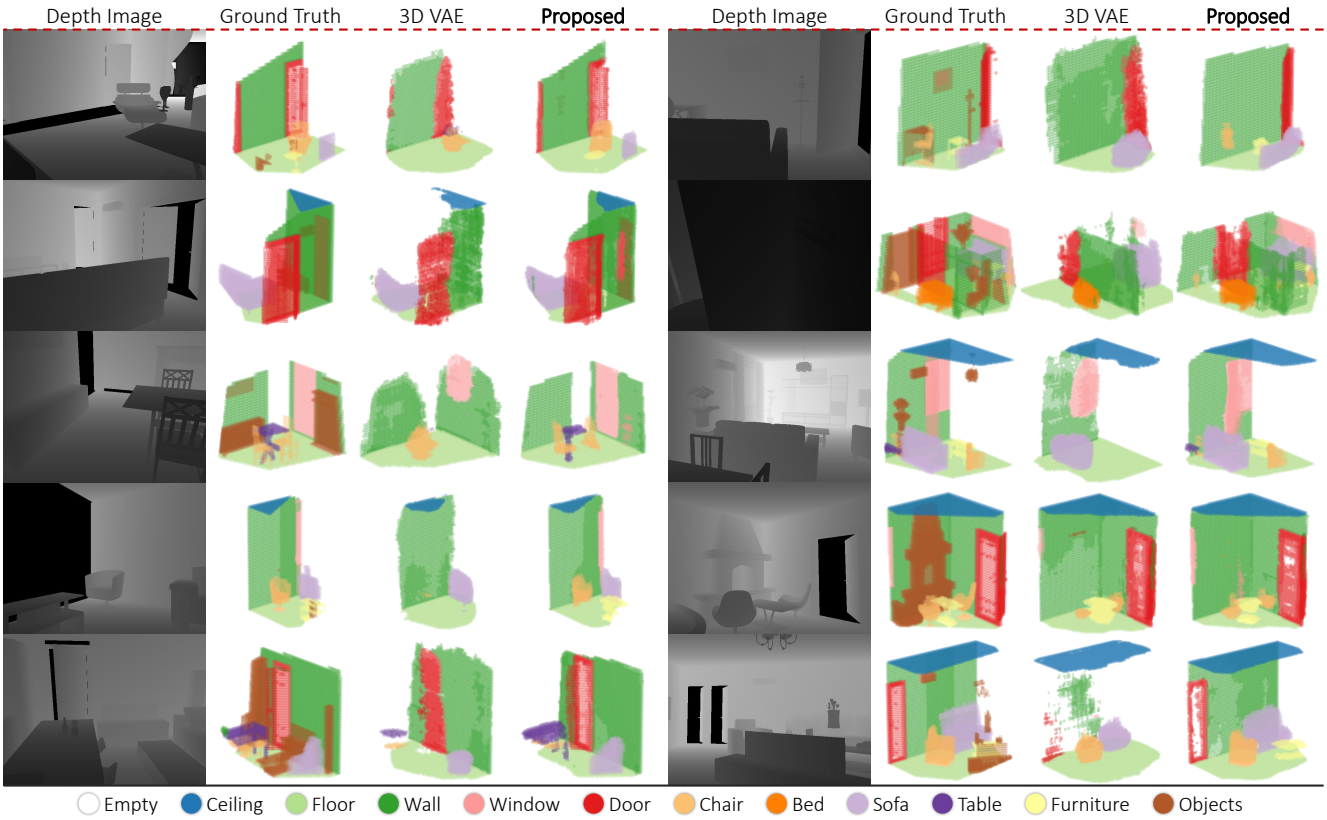

Figure 5: GAN for semantic 3D reconstruction from depth images.

ing the output realistc; while, the other aimed at imitating the embedding learned via auto-encoder from the complete volumetric data. We have demonstrated the effectiveness of our approach on a reference benchmark dataset such as SUNCG. The future work aims at modifying our architecture to overcome the memory limitation so to process higher resolution samples, this allowing a direct comparison with approaches such as SSCNet [22].

References

[1] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, (just-accepted), 2017. 2, 4

[2] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016. 2, 3, 6, 7

[3] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2

[4] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018. 3, 5

[5] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2017. 1

[6] H. Fan, H. Su, and L. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 38, 2017. 2

[7] M. Gadelha, S. Maji, and R. Wang. 3d shape induction from 2d views of multiple objects. *arXiv preprint arXiv:1612.05872*, 2016. 1

[8] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Scenenet: Understanding real world indoor scenes with synthetic data. *arXiv preprint arXiv:1511.07041*, 2015. 1

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[10] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007. 5

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[12] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. *Advances in Neural Information Processing Systems*, 4:3581–3589, 2014. 2

[13] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 4

[14] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013. 1

[15] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2036–2043. IEEE, 2009. 1

[16] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 3

[17] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011. 1

[18] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 6

[19] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017. 4

[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 1, 5, 7

[21] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3483–3491. Curran Associates, Inc., 2015. 2

[22] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 5, 6, 8

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 1

[24] D. Van Krevelen and R. Poelman. A survey of augmented reality technologies, applications and limitations. *International journal of virtual reality*, 9(2):1, 2010. 1

[25] J. Wu, Y. Wang, T. Xue, X. Sun, W. T. Freeman, and J. B. Tenenbaum. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *Advances In Neural Information Processing Systems*, 2017. 2

[26] B. Yang, S. Rosa, A. Markham, N. Trigoni, and H. Wen. 3d object dense reconstruction from a single depth view. *arXiv preprint arXiv:1802.00411*, 2018. 1, 2, 6, 7

[27] C. Yu and Y. Wang. 3d-scene-gan: Three-dimensional scene reconstruction with generative adversarial networks. 2018. 2