EG UK Computer Graphics & Visual Computing (2023)
D. Hunter and P. Vangorp (Editors)

RPS-Net: Indoor Scene Point Cloud Completion using RBF-Point Sparse Convolution

Tao Wang, Jing Wu, Ze Ji and Yu-Kun Lai

Cardiff University, UK

Abstract

We introduce a novel approach to the completion of 3D scenes, which is a practically important task as captured point clouds of 3D scenes tend to be incomplete due to limited sensor range and occlusion. We address this problem by utilising sparse convolutions, commonly used for recognition tasks, to this content generation task, which can well capture the spatial relationships while ensuring high efficiency, as only samples near the surface need to be processed. Moreover, traditional sparse convolutions only consider grid occupancies, which cannot accurately locate surface points, with unavoidable quantisation errors. Observing that local surface patches have common patterns, we propose to sample a Radial Basis Function (RBF) field within each grid which is then compactly represented using a Point Encoder-Decoder (PED) network. This further provides a compact and effective representation for 3D completion, and the decoded latent feature includes important information of the local area of the point cloud for more accurate, sub-voxel level completion. Extensive experiments demonstrate that our method outperforms state-of-the-art methods by a large margin.

CCS Concepts

• ***Computing methodologies*** → *Shape representations; Point-based models;*

1. Introduction

In the fields of computer graphics and computer vision, it is important to obtain high-fidelity 3D data of real-world objects and scenes. Such data is useful for a wide range of practical applications. For instance, high-quality indoor scene data is beneficial to AR/VR (augmented reality and virtual reality) applications. However, directly acquired 3D models by scanning are often flawed, because of the unavoidable occlusion, limited sensor range and noise during data capture. Techniques to improve the captured data are thus highly demanded, to facilitate downstream processing and applications.

Among different 3D data enhancement techniques, 3D completion is particularly important, which infers the missing part from partial input to deliver completed results of the original models or scenes. It has a firm connection with traditional 3D reconstruction that builds a digital representation for real-world objects and scenes. In the process of reconstruction, it is common that certain parts are missing due to unavoidable occlusions or sensor limits (e.g. objects being too close or too far away from the sensor). This can cause significant problems for downstream applications, especially when large parts are missing. 3D completion is therefore not only an important task on its own, but also often seen as an integral component in a 3D reconstruction pipeline.

Despite great effort [YKH*18, YFST18, TKR*19], 3D completion still faces several significant challenges to be tackled. First, both the (local) geometric shapes and their distributions and spatial relationships are crucial to providing clues for completing missing parts. This applies both at the object level where existing parts give clues for completing missing parts, and at the scene level where the contextual information is crucial for scene understanding and modelling [CLH15]. Existing learning-based methods such as those based on point clouds (e.g. [YKH*18]) often struggle to capture both geometric shape information and spatial relationships well. Second, it is challenging to recover fine geometric details for missing parts. This would normally require high-resolution representations, which can be prohibitively expensive, especially for volumetric and implicit representations (e.g. [DQN17]). The above challenges are further exacerbated when addressing 3D scene completion, which can involve a huge amount of data and require significant computational power to process.

In this paper, we address the above challenges in the challenging 3D scene completion task. Our proposed RPS-Net (RBF-Point Sparse Convolution Net) is based on a volumetric representation. However, as the majority of space is empty, we propose to use sparse convolutions [CGS20] such that only voxels close to the surface need to be calculated, significantly reducing the computational and memory costs. Sparse convolutions have been successful for shape/scene understanding tasks, achieving state-of-the-art performance [GEV18, CPK19], but to the best of our knowledge, it has not been fully demonstrated for generative tasks, in particular 3D

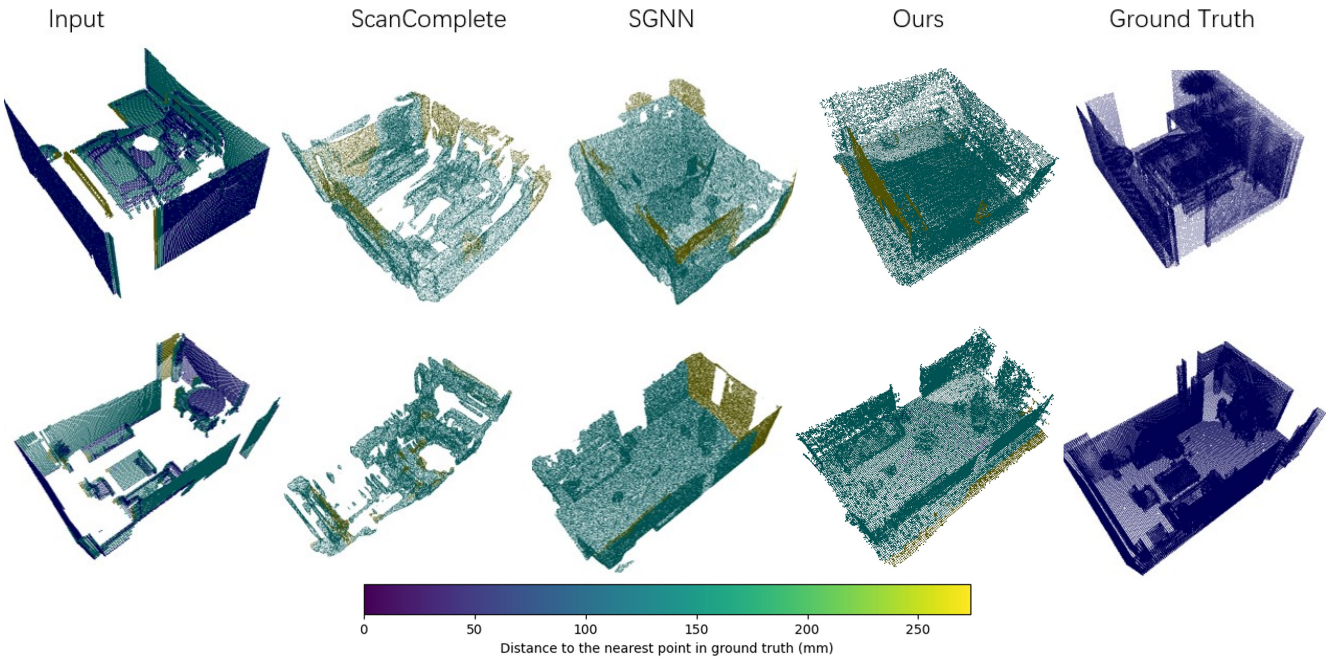


Figure 1: *Comparison of completion results between our method and state-of-the-art methods SGNN [DDN20] and ScanComplete [DCS*17]. The colour coding is used to show the distance of a point to the nearest point in the ground truth, where darker means closer. Note that the same colour scheme is applied throughout the paper.*

scene completion. Unlike unstructured point clouds which largely depend on Multi-Layer Perceptron (MLP) that cannot fully capture spatial relationships, sparse convolutions effectively capture geometric information and spatial relationships as spatial neighbourhoods of different scales are taken into account. To achieve this, we need to extend the original sparse convolutions to address the overall dimensions of the full scene not known in advance. To further capture long-range dependencies (e.g., the shape of a chair in a scene may help complete other chairs in the same scene), we further incorporate an attention module in the network architecture. To recover fine geometric details, instead of using voxels for occupancy, we further propose to encode point distributions within each voxel using a compact representation. This is based on the observation that geometric shapes of local volumetric regions are not arbitrary, but have common patterns. Such patterns are effectively encoded using an encoder-decoder structure network to encode sub-voxel radial basis function (RBF) distribution from nearby points. Previous work used a Variational Autoencoder (VAE) to compactly represent shape details within voxels, but their approach is more suitable and only used for recognition [MGLM19]. Our idea is to instead develop a compact representation for *generation* tasks, and our Point Encoder-Decoder (PED) network not only compactly represents local shapes within voxels, but also exploits the latent code for inferring sub-voxel level geometric details, thus helping to reconstruct high-quality detailed geometry. Some examples of our method along with comparisons with state of the arts are shown in Figure 1, demonstrating the superior performance of our method.

In summary, our contributions are:

(1) Sparse convolution plays an important role in our method to process large-scale point clouds. To our knowledge, our method is the first supervised method to perform generative tasks with full 3D sparse convolutions.

(2) By compactly encoding the point distribution in local areas using a dedicated network based on a smooth RBF field, our volumetric representation contains richer geometric details within voxels than the binary occupancy maps and signed distances. This helps our method better predict point positions at the sub-voxel level for more accurate completion.

(3) Because there is no public dataset that provides paired partial/complete 3D scenes anymore, we built a new dataset named IPS (Indoor Partial Scene) dataset that fills this gap. Our dataset will be made available for research purposes.

2. Related Work

In the early years, methods based on radial basis functions achieved great results in several tasks, including reconstruction and representation of 3D models [CBC*01]. Recently, several attempts were made with shared weight MLP autoencoders [YKH*18, YFST18] for single object point cloud completion. However, local information is considerably lost as such methods have to use symmetric functions in the feature space to avoid the requirement of ordering. Convolutional neural networks [SYZ*16] have been proposed for 3D completion since it is a natural extension from 2D inpainting [LRS*18] to the 3D domain. Recently, sparse convolution [CGS20] has shown its excellent performance in processing 3D

data for applications such as classification and segmentation. However, generative tasks with full sparse convolution have not been applied, which is addressed in our work.

2.1. Sparse convolution

Sparse convolution directly conducts convolution on sparse tensors which saves huge computational resources. Graham [Gra14] uses 2D sparse convolution on handwriting recognition and image classification which achieved promising results. Moreover, in the 3D feature extraction task, sparse convolution outperforms other methods by higher accuracy and much less time [CPK19]. A generative paradigm was attempted in [GCS20] for the object detection task, instead of generating new or missing content. Sparse Generative Neural Network (SGNN) [DDN20] exploits sparse convolutions for scene completion; however, the method still contains dense components. The method is self-supervised by removing partial data from the already incomplete scan input and learning to recover the removed data. However, as the original scan is incomplete, the evaluation process has fundamental problems that would penalise correct completion that adds missing parts in the original scans. In our work, we propose a scene completion method that fully benefits from sparse convolutions. Along with detail encoding within voxels and the attention module, our method improves 3D scene completion both at structure and detail levels. We further develop a dataset based on synthetic 3D scenes, ensuring the ground truth does not have missing data.

2.2. Object and scene completion

Object completion. Following classic PointNet [QSMG17], many papers [DQN17, YFST18, GFK*18, LBBM18, CCM20, WLHL20, WXH*21, WXH*22, XWL*22] try to address single object completion. PCN (Point Completion Network) [YKH*18] is applied in a coarse-to-fine structure which is widely used in voxel grid completion. This network is based on FoldingNet [YFST18] but the folding happens in a small region specific to each point in the coarse result. Liu et al. [LSY*20] similarly deform 2D planes into a 3D shape with two stages. However, the above MLP-based methods require the input point cloud to have the same number of points, and consequently a fixed number of points are outputted. Thus, their generalisation capability for unseen objects is limited. Moreover, due to the fully connected nature of these methods, large-scale point cloud input is hard to handle.

Scene completion. Traditional methods to achieve scene completion can be categorised into geometric approaches and template-based approaches. Symmetry [PMW*08] is a common feature in geometric approaches to analyse the existing shape, and then complete the missing parts. Template-based approaches mainly exploit similar shapes [PMG05] in the database. The template works as a reference for inferring missing parts in incomplete shapes. However, traditional methods could fail because either symmetry does not exist or similar templates are missing in the database. Deep learning-based scene completion surpasses traditional methods by their efficiency and accuracy. Song et al. [SYZ*16] take a depth image along with semantic information as input and only output the voxel grid within the camera frustum. The method requires semantic information, which is not often provided and can be difficult to obtain. The works [WTNT19, CLQ*20, WTNT20, CCZ*21] continued this work and the task can be classified as semantic scene completion (SSC), which needs to either provide or predict semantic information. The task we focus on is different from SSC: we do not need semantic information, and the input is the partial point cloud of the whole scene rather than a single depth image. Some works [DRB*18, DDN20] have attempted to complete a partial scene with Truncated Signed Distance Fields (TSDF), but limited by the input voxel resolution, both results are unsatisfactory. Our method does not need semantic information so does not require additional semantic labelling for training. With the capacity of sparse convolution and the attention module, our method can recover more high-quality details.

3. RPS-Net

Our method applies sparse convolution to the generation task, along with compact encoding of local geometry as an RBF field to get promising results, as shown in Figure 1. Sparse convolution makes it possible to conduct convolutions on high-resolution voxel grids. Meanwhile, the embedding feature helps the network get richer geometric information for both the initial input and the generated completion results. The details of the pipeline will be described in the following subsections.

3.1. Bounding box prediction

As a voxel-based method, it is essential to know the dimension of the whole volume to cover the completed 3D scene. However, the input 3D scene may have large parts missing, so directly using the bounding box of the input point cloud may mistakenly cut off the output result. To address this, we take the Frustum-PointNet [QLW*17] architecture, and use it to analyse the global features to regress the centre and size (a 6-dim vector) of the bounding box of the *completed* 3D scene through an added fully connected (FC) layer.

3.2. Voxelisation and RBF field computation

Our approach is based on sparse convolution on a volumetric grid, so we need to first turn point cloud input to voxels. Feature extraction is then performed based on points within each voxel. We split the point cloud by an appropriate voxel size to get a regular grid. Convolution intrinsically better exploits local geometry information than MLP as it learns features of neighbouring voxels. However, dense 3D CNN's resolution is typically limited to $64 \times 64 \times 64$ due to the curse of dimensionality. We use two strategies to address this fundamental challenge: Thanks to sparse convolution where only occupied voxels are stored and processed, we are able to increase the resolution of the voxel grid, such that the longest dimension is split into 128 voxels, and other dimensions are split accordingly based on the voxel size, i.e., the grid resolution is $n_x \times n_y \times n_z$, where $\max(n_x, n_y, n_z) = 128$. To retain useful information within each voxel, instead of using occupancy or distance field, we further divide each voxel into $s \times s \times s$ subvoxels ($s = 4$ in our experiments), and the RBF value reflecting point distribution is

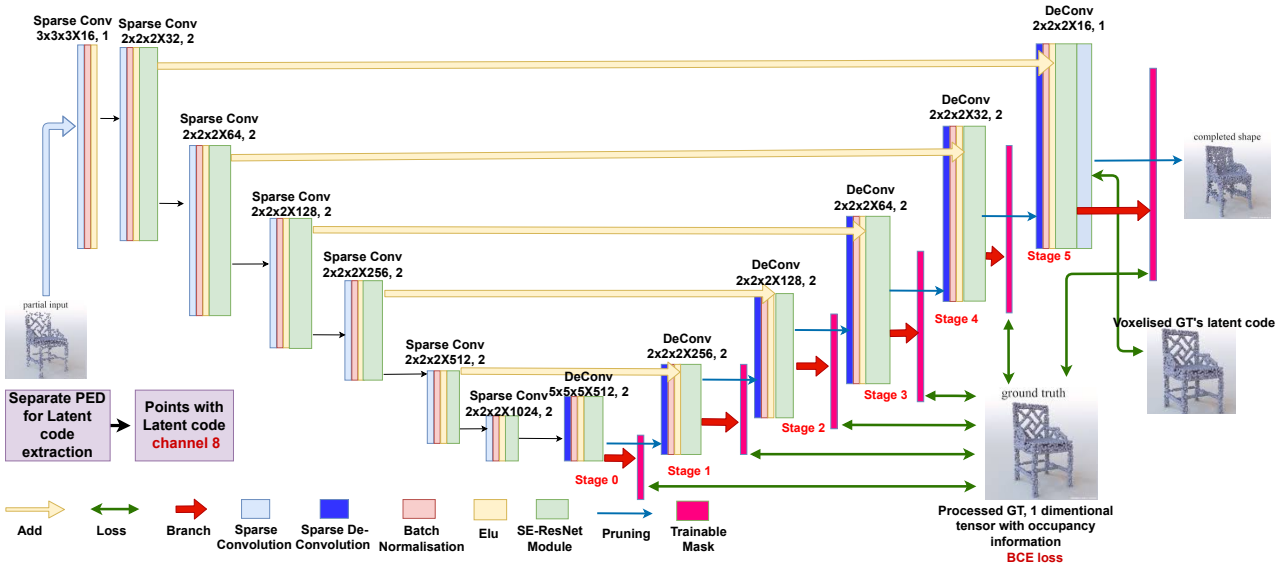


Figure 2: *Pipeline of our RPS-Net (using a single chair as the example), where a learnable mask is predicted in a separate branch in each stage and used as a reference to prune redundant voxels in the pruning process.*

calculated for each subvoxel as follows:

$$f(v) = \max_{p \in P} \left(\exp \frac{-\|p - v\|_2^2}{2\sigma^2} \right) \quad (1)$$

where v indicates the centre coordinates of a subvoxel, P indicates the set of points in the point cloud, and p is one of the points. The Gaussian kernel is applied in our RBF, where the value is determined by the closest point near the centre of the subvoxel, which has the dominant influence. σ is the Gaussian kernel size. Compared with binary occupancy at the subvoxel level, RBF is smooth so can be easier to learn. Storing all the subvoxel RBF values as multi-channels in each voxel is prohibitively expensive. To reduce the complexity and avoid potential overfitting, a network named PED (Point Encoder-Decoder) as shown in Figure 3 is utilised to embed subvoxel RBF values within a voxel to a latent code ℓ using the encoder, and the decoder aims to recover point positions at the sub-voxel grid. PED is trained by taking the RBF values as input and outputting the closest point to the centre of each subvoxel. Unlike [MGLM19] which only uses such variational latent for analysis using a VAE, we develop PED that not only compactly embeds local geometry within a voxel, but also allows flexible and more accurate point positions to be recovered. It is incorporated in our sparse convolution architecture for subvoxel-level representation, utilised for representing both the input with richer information, and the output to better recover details.

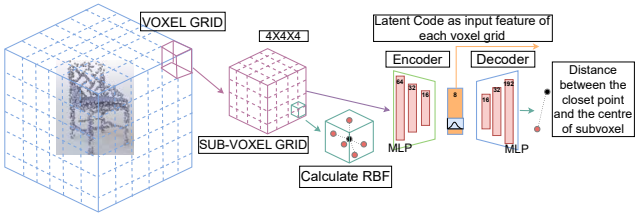


Figure 3: *PED: a separate encoder-decoder network that encodes RBF values at subvoxels within a voxel into a compact latent code, which can then be decoded into points' coordinates.*

3.3. Network architecture

RPS-Net is based on an encoder-decoder architecture, with skip connections to retain more information from the input, as illustrated in Figure 2. To better extract information at different scales and complete missing parts, a coarse-to-fine architecture is introduced in the decoder, which contains six cascade stages. In each stage, the deconvolution layer expands the potentially occupied (i.e., non-zero) voxels by duplicating each voxel from the previous stage in space according to the kernel size and stride. Through this step, new non-zero entries will be added in the sparse tensor to get more voxels generated. This is then followed by a pruning layer [TDB17] as not all the expanded voxels are needed. The resolution of output is gradually increased as illustrated in Figure 4. A learnable mask is formed to guide the pruning layer to remove voxels generated by deconvolution. As training progresses, the mask is gradually refined to be like the ground truth to prune unneeded voxels produced in deconvolution. Correspondingly, in the early stage, the structure of the complete 3D object is recovered. With multiple layers of deconvolution and pruning of redundant voxels, the resolution of the recovered object increases to acquire more details. Attention mechanism has been shown to improve the generalisation capability in existing works [VSP*17,HSS18]. As different filters in convolution extract different feature maps, the channel attention module assigns different weights to them according to their importance and selects

the most useful feature maps for the task. In practice, we utilise SE-ResNet block [HSS18] for the channel attention module.

3.4. Prediction of point distribution within each voxel

The use of information-rich latent code for the input voxels enhances the representation of detail. The coarse output by the encoder-decoder structure (i.e., only considering the non-zero voxels in the last layer) can fill in large missing or occluded parts of the input object. However, details in the output are not sufficiently retained. Since the output of the network also encodes voxels with latent code, this enables point distribution within each voxel to be recovered to get a fine result. We supervise the output at the last stage with the ground truth voxel latent code ℓ. By passing the predicted latent code through the decoder of the PED, we can recover the point distribution at the subvoxel level. In the inference stage, the predicted latent code is finally decoded by PED to the coordinates of points, allowing more accurate sub-voxel level point positions to be regressed. As this process aims to predict more accurate point locations within subvoxels, the predicted point is removed if it is too far away from the centre of its corresponding subvoxel.

3.5. Loss function

We implement the sparse convolution and deconvolution using Minkowski Engine [CGS20]. Binary Cross Entropy Loss (BCE) is used in the coarse (i.e., voxel-level) completion. The voxelisation makes the input partial object bounded in a certain box and it is easy to depict the voxelised ground truth as a binary format with its occupancy in the sparse voxel grid. Therefore, BCE loss in each stage k measures the difference between output and ground truth during model training. There are $K = 6$ stages in the decoder, and $\mathcal{L}_k$ is individually calculated in each stage. Through experiments, we found that the model is more easily trained in stages, i.e., as training progresses, the weights of layers in previous stages are frozen. This also avoids introducing additional hyperparameters in the combined stage loss.

In the fine completion stage, the subvoxel points prediction is dominated by the voxel latent code. Mean squared error (MSE) as Equation 4 is used to evaluate the difference between the predicted and ground truth latent codes, which makes the model regress the voxel latent code that implicitly describes point distribution within each voxel. Therefore, the loss function in the final stage is as follows:

$$\mathcal{L} = \mathcal{L}_{k=6} + \lambda \mathcal{L}_{fine} \tag{2}$$

The BCE loss guarantees the occupancy of sparse voxel grids and MSE ensures their features reflect the distribution of subvoxel points. $\lambda = 1.0$ is used in our experiments. For detailed formulas of L_k and L_{fine}, Equation 3 is the loss function used in the phase of coarse completion.

$$\mathcal{L}_k = -\frac{1}{N_k} \sum_{i=1}^{N_k} [y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i})] \tag{3}$$

where $\hat{y_i}$ indicates the 1-channel feature of each voxel representing occupancy delivered by another branch in each stage, and a sigmoid function makes it clamped within $[0, 1]$, y_i indicates the target occupancy of corresponding voxels. k indicates the current stage ($1 \leq k \leq K$). N_k is the total number of voxels at stage k.

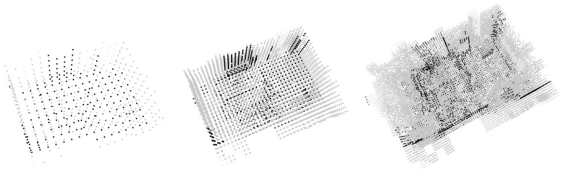


Figure 4: *Output progresses to higher resolution with stage proceeding.*

$$\mathcal{L}_{fine} = \frac{1}{N} \sum_{i=1}^{N} \|\ell_i - \hat{\ell}_i\|^2 \tag{4}$$

where $\hat{\ell}_i$ indicates output latent code of each voxel and ℓ_i is the voxel latent code obtained from ground truth. N is the total number of voxels in the last stage.

4. Experiments and Implementation

4.1. Datasets

Point clouds are a popular representation for 3D feature learning as depth cameras and other 3D scanning devices are becoming more accessible. For single object completion, Completion3D [TKR*19] is one of the most popular datasets. There are 8 different categories of point cloud objects. The numbers of points in the input point cloud and ground truth both are 2048. For such a small number of points within the point cloud, it is hard to evaluate the genuine capability of models for 3D scene completion. Moreover, each instance of the partial point cloud in Completion3D has the exact same number of points which does not conform to real-world situations. Therefore, more attention has been paid to large-scale point clouds in which a varied number of points exist.

For the indoor scene dataset, SUNCG dataset [SYZ*16] was popular in this regard but is no longer accessible. There is no comprehensive indoor synthesised dataset with accurate ground truth. We therefore created the IPS (Indoor Partial Scene) dataset. It fills the gap that there is no publicly available paired partial and complete indoor scene point cloud dataset. It is worth noting that datasets like NYUv2 [SHKF12], Matterport3D [CDF*17] and SemanticKITTI [BGM*19] are all **real-world scanned** datasets and not dedicated to 3D completion. The real-world scanned datasets, both indoors and outdoors, are not perfectly complete, due to unavoidable occlusion and noise. Such "ground truth" may be acceptable for other tasks like classification. However, these occluded areas with missing data could have a huge impact on the completion task, both for training and evaluation. Therefore, evaluations on these datasets are not recommended for our problem, as a method that correctly recovers parts that are missing in imperfect "ground truth" would be unfairly penalised.

4.1.1. SceneNet [MHLJ17]

SceneNet is also a labelled synthesised indoor scene dataset, but there are only 57 room instances in this dataset. Although we also

use this dataset for testing scene completion methods, it is inherently limited and may not be sufficient as a practical dataset due to such a small number of rooms.

4.1.2. Indoor partial scene (IPS) Dataset

There are 4216 houses with selected 6264 rooms in this dataset, the training and test split is 7 : 3. The variety of rooms ranges from bedrooms to kitchens which fits the real-world situation and ensures the generalisation capability to be evaluated. These 6264 synthesised rooms are extracted from the 3D-FRONT dataset [FCG*21] including annotations. Each room is centred by geometry. A camera is then placed in the centre with a height of 0.6 metres and looks down by 45 degrees. By rotating the camera around the y-axis at an interval of 40°, 9 depth maps are captured. Occlusions naturally occur in these depth maps. The 9 depth maps are then converted to the partial point cloud.

4.2. Training details

The learning rate is 1e-2 and weight decay is 1e-4. The batch size is set to 4. There are six up-sampling stages in the decoder to expand the occupied voxels. For each stage, the output voxels are increased through deconvolution based on its kernel size, and we use early stopping with a patience parameter $\eta = 10$ to make the model automatically progress to further stages when the BCE loss of each stage does not decrease over a few epochs. In addition, after each stage, the weights of layers in previous stages are frozen.

5. Evaluation and Results

In terms of PCN [YKH*18] and other PointNet-based methods [YFST18,TKR*19,LSY*20,CCM20,WXH*21], there is a major drawback that their result has an identical number of points in the point cloud as input, which is restrictive. Furthermore, indoor scene point clouds contain hundreds of thousands of points, these methods could not take such a large number of points with their fully connected layers due to the excessive number of learnable parameters. Therefore, these methods fail to effectively handle 3D scene completion. In our work, the output has been pruned by a trained mask which decreases the size of output while it varies to fit different situations of indoor scenes.

5.1. Metrics

To measure the accuracy of the completion network, Chamfer distance and Earth Mover's distance are popular metrics to measure the difference between the output point cloud and the ground truth. Due to the unordered nature of the point cloud, the metric should be invariant to the permutation of points. The requirement to use EMD distance is point clouds S_1, S_2 have to be of the same size. However, the scene completion task in our setting does not assume this, so EMD is not applicable to our method. **Chamfer Distance (CD)** as Equation 5 is a classic metric to compare two point clouds and is used in this work. Chamfer distance is different from EMD, which does not require S_1, S_2 to be the same. CD measures the squared distance between each point in one set to its nearest neighbour in another set.

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2 \quad (5)$$

Note that previous works [DDN20, DRB*18] also used L_1 error of TSDF values between output and target. However, it is highly related to the voxel size and only applicable to TSDF-based methods. The point cloud is more in common use and widely used in 3D object completion. Therefore, the mesh outputs of ScanComplete and SGNN have been sampled to point clouds for comparison.

There are several differences between our work and scene completion methods like [CCZ*21, XZS*19]. Such methods require semantic labels for training, which are expensive to acquire. Furthermore, there is no fixed camera or viewpoint for our work, and the completion of the entire scene geometry is our focus rather than just a single depth image.

Method	IPS	SceneNet [MHLJ17]
ScanComplete [DRB*18]	226.38	506.72
SGNN [DDN20]	35.45	145.60
Ours	**20.06**	**106.55**

Table 1: *Chamfer distance (lower is better) between output and ground truth.*

5.2. Results

Our method achieves competitive results as shown in Table 1 and outperforms other methods by a large margin quantitatively with two datasets; qualitative comparison also is shown in Figure 5. The comparison is conducted with methods that do not use semantic information. This attribute makes methods applicable to broader settings. ScanComplete is limited by the low resolution of the grid and occupancy features. SGNN tries to use sparse grids but dense blocks and incomplete "ground truth" impede the capability of their model. Compared to other methods, our method compelete more parts in terms of floor and wall. Figure 6 shows the floor in bedroom is still missing in SGNN, and this missing part has been completed by our method, same as the wall in another living room. In contrast to the TSDF value used in SGNN, the subvoxel RBF value indicates the distribution of points in higher resolution. Ablation studies are conducted in the following section.

5.3. Generalisation

Further experiments have been done on real world scanned datasets like Matterport3D [CDF*17], we evaluate the generalisation capability of our model and SGNN, where the models are trained on our IPS dataset, and applied to the Matterport3D dataset, and the average Chamfer distances are reported in Table 2. Qualitative comparisons are shown in Figure 7. As can be seen, the "ground truth" in Matterport3D has large incomplete regions. Our method manages to complete the floor and wall which are missing in the "ground truth", which is plausible, but can significantly skew the quantitative analysis. To address this, we measure the Chamfer distance from the ground truth to output point clouds, and our method outperforms SGNN.

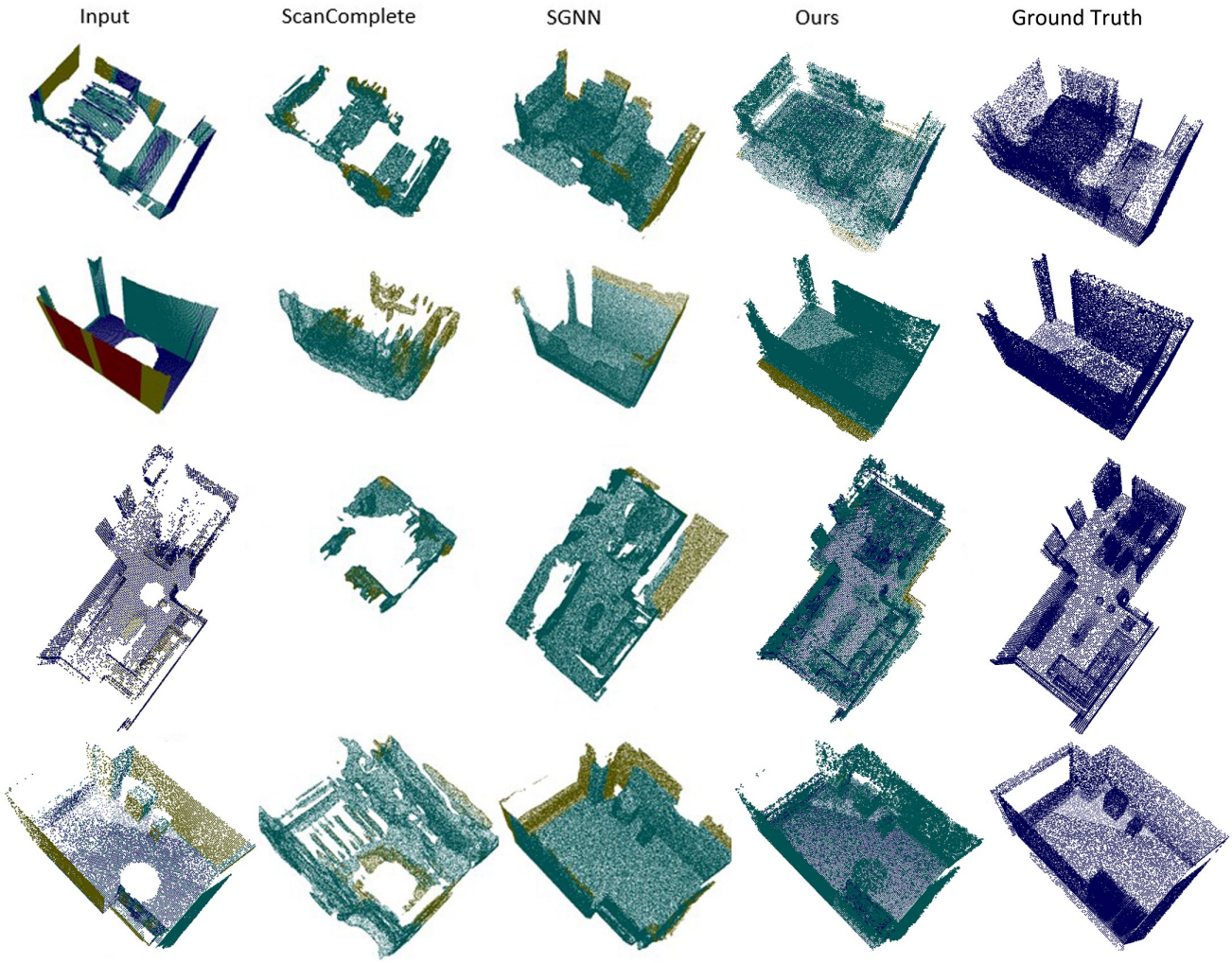
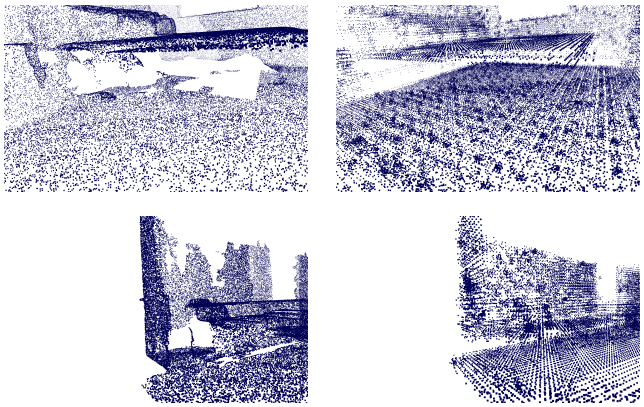


Figure 5: *More visual comparison between our method and others, showing more details.*



(a) SGNN (b) Ours

Figure 6: *Completion detail comparison with SGNN. ScanComplete is not applicable as it has a large area missing in its results.*

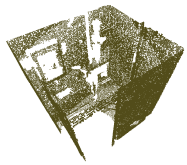
Method/Dataset	Matterport3D [CDF*17]
SGNN [DDN20]	57.90
Ours	**42.76**

Table 2: *Average Chamfer distances (from ground truth to output point clouds) comparison on methods both trained on IPS.*

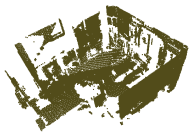
6. Ablation studies

6.1. Input voxel latent code ℓ

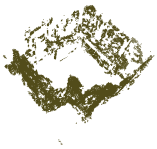
The use of PED latent code ℓ directly improves the quality of coarse output, as more details of the input point cloud are captured. Compared to the occupancy feature, ℓ extends the learning capability of the convolutional neural network. Table 3 indicates the coarse output's quality in terms of average Chamfer distances when using our proposed PED latent code feature and the occupancy feature. As a comparison, we also show the performance when using the VAE representation from VV-Net [MGLM19]. Although the VAE repre-



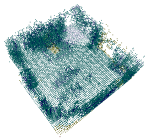
(a) Incomplete "ground truth"



(b) Input



(c) SGNN



(d) Ours

Figure 7: *Rooms in Matterport3D dataset are extracted by signed distance from scanned frames, many of them are inherently incomplete, so not generally suitable for the 3D scene completion task.*

Method or Category	Completion3D [TKR*19]	IPS
Occupancy	0.0015	50.29
VAE latent code	0.0012	24.25
PED latent code ℓ	**0.0012**	**23.15**

Table 3: *Voxel latent code ℓ has considerable impact on coarse output. We show the results of single category (chair) test on the Completion3D dataset [TKR*19] where shapes have been normalised, as well as results on IPS dataset.*

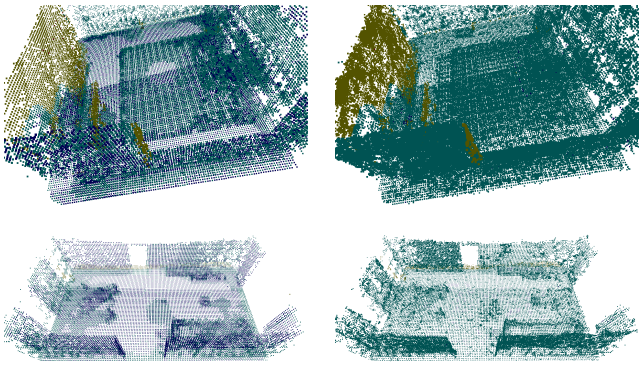
sentation could embed the RBF distribution within each voxel into a compact code, it loses the ability to recover more accurate point locations. Our representation achieves similar performance as VAE latent code for the single object case, but clearly better performance on the (arguably more challenging) Indoor scene dataset.

6.2. Fine recovery with PED latent prediction

Voxel latent code ℓ contributes largely to the final output of our method. The abundant geometry information within subvoxel ℓ value enlarges the capacity of representation of the deep network model. The ablation study compares the coarse output and our PED latent code improved output. Comparison in Table 4 indicates PED latent code is essential in terms of internal point prediction, which improves the quality of final output. Figure 8 also reveals this point.

6.3. Attention Module

We demonstrate that the attention module improves the completion performance by quantitative comparison. These benefits can be recognised from Table 4. It also helps the training process converge better.



(a) Coarse output

(b) Fine output

Figure 8: *Comparison of coarse output (left) and fine output after subvoxel point prediction (right).*

Method or Category	IPS
RPS-Net w/o Attention Module	105.69
RPS-Net w/o subvoxel points prediction	23.15
RPS-Net	**20.06**

Table 4: *Further modification of coarse output improves the Chamfer distance, the attention module also has a considerable impact on output.*

7. Conclusion and Future work

We have presented a novel method for point cloud completion. By utilising sparse convolutions, we are able to process high-resolution volumetric grids efficiently. Moreover, the PED-encoded RBF fields within voxels make it possible for our method to recover subvoxel details. We further create the IPS dataset for point cloud completion. On both the IPS and SceneNet [MHLJ17] datasets our method has shown its superiority to fill in the missing part of the input partial point cloud, outperforming state-of-the-art methods. Matterport3D [CDF*17] a test dataset verify that the generalisation of our model also is better than other methods. As future work, we would like to extend our approach to larger scenarios like city scale to further explore the potential of sparse convolution.

Acknowledgements

The authors gratefully acknowledge the support of China Scholarship Council (CSC), and would like to thank Advanced Research Computing at Cardiff (ARCCA) for the support of computing facilities.

References

[BGM*19] BEHLEY J., GARBADE M., MILIOTO A., QUENZEL J., BEHNKE S., STACHNISS C., GALL J.: SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *ICCV* (2019). 5

[CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL

T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH* (2001), p. 67–76. 2

[CCM20] CHEN X., CHEN B., MITRA N. J.: Unpaired Point Cloud Completion on Real Scans using Adversarial Training. *ICLR* (mar 2020). 3, 6

[CCZ*21] CAI Y., CHEN X., ZHANG C., LIN K.-Y., WANG X., LI H.: Semantic scene completion via integrating instances and scene in-the-loop. In *CVPR* (2021), pp. 324–333. 3, 6

[CDF*17] CHANG A., DAI A., FUNKHOUSER T., HALBER M., NIESSNER M., SAVVA M., SONG S., ZENG A., ZHANG Y.: Matterport3D: Learning from RGB-D data in indoor environments. *arXiv:1709.06158* (2017). 5, 6, 7, 8

[CGS20] CHOY C., GWAK J., SAVARESE S.: 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In *CVPR* (2020). 1, 2, 5

[CLH15] CHEN K., LAI Y., HU S.: 3D indoor scene modeling from RGB-D data: a survey. *Comput. Vis. Media 1*, 4 (2015), 267–278. 1

[CLQ*20] CHEN X., LIN K.-Y., QIAN C., ZENG G., LI H.: 3D sketch-aware semantic scene completion via semi-supervised structure prior. In *CVPR* (2020), pp. 4193–4202. 3

[CPK19] CHOY C., PARK J., KOLTUN V.: Fully convolutional geometric features. In *ICCV* (2019). 1, 3

[DCS*17] DAI A., CHANG A. X., SAVVA M., HALBER M., FUNKHOUSER T., NIESSNER M.: ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR* (2017). 2

[DDN20] DAI A., DILLER C., NIESSNER M.: SG-NN: Sparse generative neural networks for self-supervised scene completion of RGB-D scans. In *CVPR* (2020), pp. 849–858. 2, 3, 6, 7

[DQN17] DAI A., QI C. R., NIESSNER M.: Shape completion using 3D-encoder-predictor CNNs and shape synthesis. *CVPR* (2017), 6545–6554. 1, 3

[DRB*18] DAI A., RITCHIE D., BOKELOH M., REED S., STURM J., NIEBNER M.: ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans. In *CVPR* (2018), pp. 4578–4587. 3, 6

[FCG*21] FU H., CAI B., GAO L., ZHANG L.-X., WANG J., LI C., ZENG Q., SUN C., JIA R., ZHAO B., ET AL.: 3D-FRONT: 3D furnished rooms with layouts and semantics. In *ICCV* (2021), pp. 10933–10942. 6

[GCS20] GWAK J., CHOY C., SAVARESE S.: Generative sparse detection networks for 3D single-shot object detection. In *ECCV* (2020). 3

[GEV18] GRAHAM B., ENGELCKE M., VAN DER MAATEN L.: 3D semantic segmentation with submanifold sparse convolutional networks. In *CVPR* (2018). 1

[GFK*18] GROUEIX T., FISHER M., KIM V. G., RUSSELL B. C., AUBRY M.: A papier-mâché approach to learning 3D surface generation. In *CVPR* (2018), pp. 216–224. 3

[Gra14] GRAHAM B.: Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070* (2014). 3

[HSS18] HU J., SHEN L., SUN G.: Squeeze-and-excitation networks. In *CVPR* (2018), pp. 7132–7141. 4, 5

[LBBM18] LITANY O., BRONSTEIN A., BRONSTEIN M., MAKADIA A.: Deformable Shape Completion with Graph Convolutional Autoencoders. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018). 3

[LRS*18] LIU G., REDA F. A., SHIH K. J., WANG T.-C., TAO A., CATANZARO B.: Image Inpainting for Irregular Holes Using Partial Convolutions. 89–105. 2

[LSY*20] LIU M., SHENG L., YANG S., SHAO J., HU S.-M.: Morphing and sampling network for dense point cloud completion. In *AAAI* (2020). 3, 6

[MGLM19] MENG H.-Y., GAO L., LAI Y.-K., MANOCHA D.: VV-Net: Voxel VAE net with group convolutions for point cloud segmentation. In *ICCV* (2019). 2, 4, 7

[MHLJ17] MCCORMAC J., HANDA A., LEUTENEGGER S., J.DAVISON A.: Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? 5, 6, 8

[PMG05] PAULY M., MITRA N., GIESEN J.: Example-based 3D scan completion. In *Symp. Geometry Processing* (2005). 3

[PMW*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L. J.: Discovering structural regularity in 3D geometry. *ACM Trans. Graph. 27*, 3 (2008), 43:1–11. 3

[QLW*17] QI C. R., LIU W., WU C., SU H., GUIBAS L. J.: Frustum PointNets for 3D object detection from RGB-D data. In *CVPR* (2017), pp. 918–927. 3

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR* (2017). 3

[SHKF12] SILBERMAN N., HOIEM D., KOHLI P., FERGUS R.: Indoor segmentation and support inference from rgbd images. In *ECCV* (2012). 5

[SYZ*16] SONG S., YU F., ZENG A., CHANG A. X., SAVVA M., FUNKHOUSER T.: Semantic scene completion from a single depth image. *CVPR* (2016). 2, 3, 5

[TDB17] TATARCHENKO M., DOSOVITSKIY A., BROX T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *ICCV* (2017), pp. 2088–2096. 4

[TKR*19] TCHAPMI L. P., KOSARAJU V., REZATOFIGHI H., REID I., SAVARESE S.: TopNet: Structural point cloud decoder. In *CVPR* (2019), pp. 383–392. 1, 5, 6, 8

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł., POLOSUKHIN I.: Attention is all you need. *NeurIPS 30* (2017). 4

[WLHL20] WEN X., LI T., HAN Z., LIU Y.-S.: Point Cloud Completion by Skip-attention Network with Hierarchical Folding. 3

[WTNT19] WANG Y., TAN D. J., NAVAB N., TOMBARI F.: ForkNet: Multi-branch volumetric semantic completion from a single depth image. In *ICCV* (2019), pp. 8608–8617. 3

[WTNT20] WU S.-C., TATENO K., NAVAB N., TOMBARI F.: SCFusion: Real-time incremental scene reconstruction with semantic completion. In *3DV* (2020), pp. 801–810. 3

[WXH*21] WEN X., XIANG P., HAN Z., CAO Y.-P., WAN P., ZHENG W., LIU Y.-S.: PMP-Net: Point Cloud Completion by Learning Multi-step Point Moving Paths. In *CVPR* (dec 2021). 3, 6

[WXH*22] WEN X., XIANG P., HAN Z., CAO Y.-P., WAN P., ZHENG W., LIU Y.-S.: Pmp-net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*, 1 (2022), 852–867. 3

[XWL*22] XIANG P., WEN X., LIU Y.-S., CAO Y.-P., WAN P., ZHENG W., HAN Z.: Snowflake point deconvolution for point cloud completion and generation with skip-transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*, 5 (2022), 6320–6338. 3

[XZS*19] XU Y., ZHU X., SHI J., ZHANG G., BAO H., LI H.: Depth completion from sparse lidar data with depth-normal constraints. In *ICCV* (2019), pp. 2811–2820. 6

[YFST18] YANG Y., FENG C., SHEN Y., TIAN D.: FoldingNet: Point cloud auto-encoder via deep grid deformation. In *CVPR* (2018), pp. 206–215. 1, 2, 3, 6

[YKH*18] YUAN W., KHOT T., HELD D., MERTZ C., HEBERT M.: PCN: Point completion network. *3DV* (2018), 728–737. 1, 2, 3, 6