DDIT: Semantic Scene Completion via Deformable Deep Implicit Templates

Haoang Li[1,2,*] Jinhu Dong[3,*] Binghui Wen[1,*] Ming Gao[1,2,*]
Tianyu Huang[3] Yun-Hui Liu[3] Daniel Cremers[1,2,4]
[1]Technical University of Munich [2]Munich Center for Machine Learning (MCML)
[3]The Chinese University of Hong Kong [4]University of Oxford
{haoang.li, ming.gao, bh.wen, cremers}@tum.de {jhdong, tyhuang, yhliu}@mae.cuhk.edu.hk

Abstract

Scene reconstructions are often incomplete due to occlusions and limited viewpoints. There have been efforts to use semantic information for scene completion. However, the completed shapes may be rough and imprecise since respective methods rely on 3D convolution and/or lack effective shape constraints. To overcome these limitations, we propose a semantic scene completion method based on deformable deep implicit templates (DDIT). Specifically, we complete each segmented instance in a scene by deforming a template with a latent code. Such a template is expressed by a deep implicit function in the canonical frame. It abstracts the shape prior of a category, and thus can provide constraints on the overall shape of an instance. Latent code controls the deformation of template to guarantee fine details of an instance. For code prediction, we design a neural network that leverages both intra- and inter-instance information. We also introduce an algorithm to transform instances between the world and canonical frames based on geometric constraints and a hierarchical tree. To further improve accuracy, we jointly optimize the latent code and transformation by enforcing the zero-valued isosurface constraint. In addition, we establish a new dataset to solve different problems of existing datasets. Experiments showed that our DDIT outperforms state-of-the-art approaches.

1. Introduction

3D scene reconstruction has been widely studied in the past decades. However, in practice, reconstructed scenes are often incomplete due to occlusions and limited viewpoints. Missing structures affect various practical applications [32, 45, 43, 22, 23, 6, 41]. To complete a scene, several methods have been proposed. Some of them [17, 30, 10] only consider geometric information and are prone to result in noisy results. Recent work [15, 27] demonstrated that semantic information can improve the completion quality. In this paper, we investigate semantic scene completion.

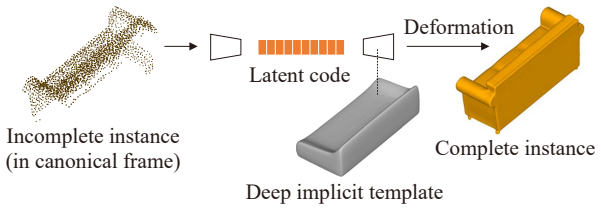

Figure 1. We complete a point set of an instance by deforming a deep implicit template with a latent code.

Existing semantic scene completion methods can be roughly classified into two main categories, i.e., scene-level [35, 40, 11, 31] and instance-level [26, 15, 27, 4, 16]. The scene-level methods directly complete the whole environment, while the instance-level approaches separately complete each detected/segmented instance. The instance-level approaches are typically more accurate than the scene-level methods since per-instance completion can better handle different structures of instances. However, existing instance-level approaches still have some limitations. They may generate rough and unreasonable meshes since they rely on 3D convolution and/or lack effective shape constraints. To overcome these limitations, we propose an instance-level method by deforming deep implicit templates with latent codes. As shown in Fig. 1, such a template abstracts the shape prior of a category, and thus can provide constraints on the overall shape of an instance. Latent code controls the deformation of template to guarantee fine details of an instance.

As shown in Fig. 2, given an incomplete point cloud of a scene in the world frame, we first conduct instance segmentation. Then we transform each segmented point set from the world frame to the canonical frame. Our transformation estimation is based on geometric constraints and a latent code-based hierarchical tree. It is more reliable and/or efficient than existing algorithms [27, 1, 2]. In the canonical frame, as mentioned above, we complete each point set by deforming a deep implicit template [46] with a latent

*Haoang Li, Jinhu Dong, Binghui Wen and Ming Gao contributed equally to this work.

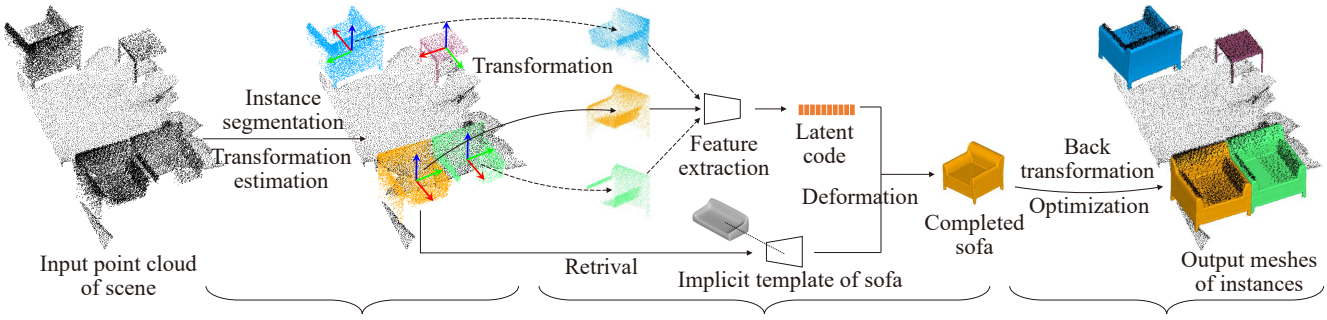

Figure 2. Pipeline of our DDIT. (a) Given an incomplete point cloud of a scene, we conduct instance segmentation and transform each segmented point set from the world frame to the canonical frame. (b) For each point set (let us take the orange one for example), we use its semantic label to retrieve a pre-trained deep implicit template. We also feed this point set and its neighbors to a neural network to predict a latent code. By deforming the retrieved template with the predicted code, we obtain a complete mesh. (c) We transform the complete mesh back into the world frame, followed by jointly optimizing its associated latent code and transformation.

code. This template is expressed by a deep implicit function and pre-trained using numerous CAD models from the same category. For latent code prediction, we design a neural network that considers both intra- and inter-instance information. Specifically, given a point set of an instance, we first divide it into several patches and use these patches to predict the initial code. Then we exploit point sets of neighboring instances from the same category to refine the initial code. Intuitively, an instance and its neighbors may have similar overall shapes but different reconstructed parts (see different sides of sofas in Fig. 2). Different parts can provide multi-view constraints on the deformation of template, improving the reliability of latent code.

After generating a complete mesh, we transform it back into the world frame. To further improve accuracy, we jointly optimize the latent code and transformation. We leverage the fact that surface points of an instance in the world frame correspond to the zero-valued signed distance function (SDF) in the canonical frame. In addition, existing datasets [1, 14, 42] for semantic scene completion have some limitations, e.g., non-alignment between point sets and meshes, irrational relationship between meshes, and lack of partial data. We establish a new dataset to solve these problems. Overall, our main contributions are

- We propose to complete instances in a scene by deforming deep implicit templates with latent codes.
- We design a neural network to predict latent codes based on intra- and inter-instance information.
- We estimate the instance transformation using geometric constraints and a code-based hierarchical tree.
- We jointly optimize the latent code and transformation based on the zero-valued SDF constraint.
- We establish a new dataset for semantic scene completion to overcome the limitations of existing ones.

Experiments showed that our DDIT outperforms state-of-the-art approaches [27, 36]. Our generated meshes show more reasonable overall shapes and finer details, and also are better aligned to the observed incomplete point sets.

2. Related Work

We classify existing semantic scene completion methods into two categories, i.e., the scene-level and instance-level.

Scene-level. Early works take a single depth image as input. Song et al. [35] proposed to jointly estimate the geometric structures and semantic labels based on 3D convolution. Each voxel is associated with an $(N+1)$-dimensional probability vector where "N" denotes the number of categories and "1" corresponds to the free space. Wang et al. [40] improved the accuracy based on a shared latent space of geometric and semantic information. Dai et al. [11] extended the above single-view case to full 3D scans using a kernel invariant to the scene size. A common limitation of the above methods is that the resolution of their completed scenes is relatively low. The reason is that 3D convolution is computationally expensive and thus the number of voxels cannot be too large. To overcome this limitation, a recent work [31] leverages deep implicit function. In a continuous space, this method can predict an $(N+1)$-dimensional vector mentioned above for an arbitrary position. However, since this method neglects different structures of instances, it may result in unreasonable overall shapes of instances.

Instance-level. Hou et al. [15] designed a method to sequentially detect, classify, and complete instances. This method outperforms the above scene-level approaches, but suffers from the resolution problem caused by 3D convolution. By contrast, some methods [27, 26] jointly detect and complete instances. In particular, they exploit deep implicit function instead of 3D convolution for shape generation, improving the smoothness of surface. However, details of the completed instances may be unsatisfactory due

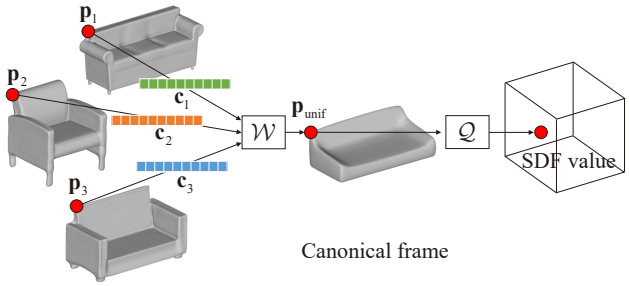

Figure 3. Illustration of deep implicit template. Position warping sub-network $\mathcal{W}$ takes different latent codes $\{\mathbf{c}_i\}_{i=1}^N$ as input and maps different positions $\{\mathbf{p}_i\}_{i=1}^N$ into a unified position $\mathbf{p}_{\text{unif}}$. SDF query sub-network $\mathcal{Q}$ predicts SDF value of the position $\mathbf{p}_{\text{unif}}$.

to the lack of effective shape constraints. Tang et al. [36] first pre-trained a decoder to express complete meshes in a latent space. Then they used an encoder to map incomplete instances into the same latent space. Accordingly, an incomplete instance is associated with a complete mesh by a latent code. Some recent work [4, 13] simultaneously conduct instance-level and scene-level completions. While they can predict reasonable overall shapes of instances, their resolution is unsatisfactory due to voxel representation. In addition, several methods [2, 16] first retrieve CAD models in a database to replace incomplete shapes, and then optionally deform CAD models for a better alignment with these shapes. Their performance is subject to the database size.

Due to limited space, please refer to a recent survey paper [32] for detailed introduction and comparison.

3. Background of Deep Implicit Template

Deep implicit template [46] is a variant of the well-known DeepSDF [29]. Such a template is expressed by a deep implicit function consisting of two cascading networks, i.e., position warping sub-network $\mathcal{W}$ and SDF sub-query network $\mathcal{Q}$. Each category corresponds to a unique template. To model such a template, N instances from the same category are normalized into a cube-shaped space called the canonical space/frame [5]. In this space, we denote an arbitrary position with respect to the i-th instance by $\mathbf{p}_i$. Here, let us take the corner of sofa for example. As shown in Fig. 3, while $\{\mathbf{p}_i\}_{i=1}^N$ have different coordinates in the canonical frame, they have the same semantic meaning, i.e., they represent the corners of different sofas.

Position warping sub-network $\mathcal{W}$ takes a latent code $\mathbf{c}_i$ of the i-th instance as input, and maps the position $\mathbf{p}_i$ into a new position $\mathbf{p}_{\text{unif}}$, i.e., $\mathbf{p}_{\text{unif}} = \mathcal{W}(\mathbf{c}_i, \mathbf{p}_i)$. This network features mapping different positions $\{\mathbf{p}_i\}_{i=1}^N$ into a unified position $\mathbf{p}_{\text{unif}}$. The position $\mathbf{p}_{\text{unif}}$ is with respect to the template, and semantically represents the corner of the sofa template (see Fig. 3). SDF query sub-network $\mathcal{Q}$ looks up the SDF value of the unified position $\mathbf{p}_{\text{unif}}$, and treats this value as the SDF value v_i of the position $\mathbf{p}_i$, i.e.,

$$v_i = \mathcal{Q}(\mathbf{p}_{\text{unif}}) \Rightarrow v_i = \mathcal{Q}\Big(\mathcal{W}(\mathbf{c}_i, \mathbf{p}_i)\Big). \tag{1}$$

Intuitively, position warping sub-network $\mathcal{W}$ and latent code $\mathbf{c}_i$ control the deformation between the i-th instance and the template. SDF query sub-network $\mathcal{Q}$ encapsulates the shape of template. This warp-and-query strategy is more accurate than direct regression of SDF value [29] for two main reasons. First, the sub-network $\mathcal{Q}$ abstracts the shape prior of a category and thus provides constraints on the overall shape of an instance. Second, the sub-network $\mathcal{W}$ and code $\mathbf{c}_i$ guarantee fine details of an instance based on flexible shape deformation.

In our context, for each category, we pre-train a template network using numerous CAD models from this category on ShapeNet dataset [5]. We also obtain the pre-trained latent codes of these CAD models as by-products that will be used for our transformation estimation. Note that these latent codes are not predicted by an encoder. Instead, they are pre-trained along with the weights of a template network.

4. Our Method

Fig. 2 shows an overview of our DDIT. Given an incomplete point cloud of a scene in the world frame, we first conduct instance segmentation based on the state-of-the-art method Mask3D [34]. Each segmented point set is associated with a semantic label. Then we introduce a reliable and efficient algorithm to transform each point set from the world frame to the canonical frame. For each transformed point set, we design a neural network to predict its latent code. By feeding this code to a pre-trained deep implicit template, we can predict SDF values of the densely sampled positions in the canonical space based on Eq. (1). After that, we apply the marching cubes algorithm [25] to these SDF values, obtaining a complete shape. Finally, we transform this shape back into the world frame, and also optimize the latent code and transformation for shape improvement.

Compared with the foreground instances, it is easier to complete the background layout typically composed of ceiling, wall, and floor. We achieve this based on Manhattan/Atlanta world assumption [8, 21, 33, 20] and plane fitting [37, 18].

4.1. Latent Code Prediction

We design a neural network for latent code prediction in the canonical frame.[1] To predict the code of the i-th point set, our network takes this point set and its neighbors with the same semantic label and similar sizes (this information has been obtained in the pre-processing step) as input. The

[1] Here, we assume that the segmented point sets have been transformed into the canonical frame based on the known transformations. Our transformation estimation will be introduced in the next subsection.

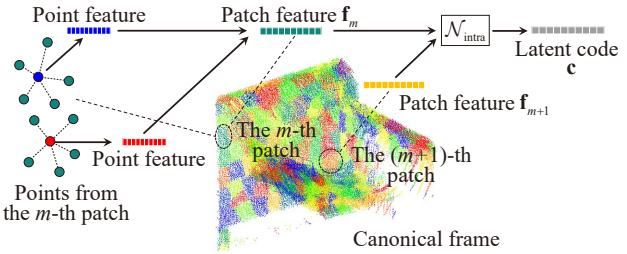

Figure 4. Illustration of our latent code prediction based on *intra-instance* information. Given a point set of an instance in the canonical frame, we first extract the feature of each point. Then we segment a point set into several patches, and extract the feature of each patch based on the features of points belonging to this patch. Finally, we aggregate features of patches to obtain a latent code.

main novelty of our network is that it considers both intra- and inter-instance information.

Intra-instance Information (see Fig. 4). Given a point set of an instance, we first use a sub-network $\mathcal{N}_{\text{point}}$ based on EdgeConv [39] to extract point features. Briefly, each root point and its K neighboring points define a graph with K edges. The root and its k-th neighbor are used to compute the feature of the k-th edge ($1 \leqslant k \leqslant K$). Then the features of all the edges are aggregated as the feature of root point.

To learn fine details, we segment the point set of an instance into several patches based on the VCCS algorithm [28]. This algorithm provides reasonable patch boundaries by considering spatial connectivity. We call the central point of a patch "control point". For all the points from the m-th patch, we aggregate their features (extracted above) as the feature of the m-th control point. Then we use the m-th control point and its neighboring control points to define a graph. This graph covers larger areas than the above graph defined by the ordinary points, and thus is suitable to extract more global features. We apply EdgeConv again to this control point-based graph to update the feature of the m-th control point. We treat the updated feature as the feature $\mathbf{f}_m$ of the m-th patch.

We integrate the above patch features $\{\mathbf{f}_m\}$ of an instance to predict the latent code $\mathbf{c}$ by designing a sub-network $\mathcal{N}_{\text{intra}}$, i.e.,

$$\mathbf{c} = \mathcal{N}_{\text{intra}}(\{\mathbf{f}_m\}). \tag{2}$$

The sub-network $\mathcal{N}_{\text{intra}}$ first exploits the attention mechanism [38, 44] to integrate the patch features $\{\mathbf{f}_m\}$ as an improved feature. Then it employs the multi-layer perceptron (MLP) to map the improved feature into a 256-dimensional latent code $\mathbf{c}$.

Inter-instance Information (see Fig. 5). Let us consider the i-th instance and its neighbors, i.e., the $(i+1)$-th and $(i+2)$-th instances for illustration. These instances belong to the same category and have similar overall shapes. However, their reconstructed parts are different due to different

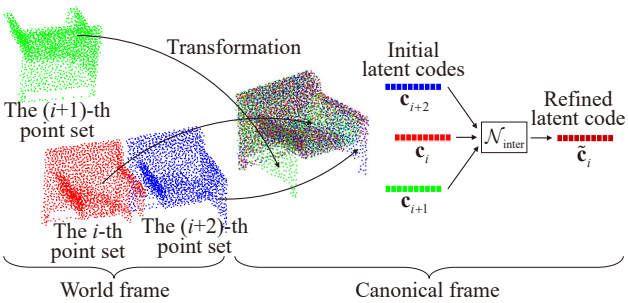

Figure 5. Illustration of our latent code prediction based on *inter-instance* information. Three input point sets in the world frame correspond to similar one-seat sofas. We transform them into the canonical frame and predict their initial latent codes separately. Then we refine the latent code of the i-th instance by integrating the initial codes of this instance and its neighbors.

viewpoints and occlusions. For example, the point set of the i-th instance lacks two front legs. The point sets of the $(i+1)$-th and $(i+2)$-th instances lack the left and right front legs, respectively. Given that an incomplete point set is reconstructed from limited view directions, it can hardly provide sufficient constraints on the deformation of template. Accordingly, the latent code predicted based on such a point set may not be reliable enough. To solve this problem, we propose to integrate the related latent codes that encapsulate the constraints from different view directions.

For the i-th, $(i+1)$-th, and $(i+2)$-th instances in the canonical frame, we first use the above sub-network $\mathcal{N}_{\text{intra}}$ to predict their latent codes $\mathbf{c}_i$, $\mathbf{c}_{i+1}$, and $\mathbf{c}_{i+2}$, respectively. Then based on the attention mechanism [38], we design a sub-network $\mathcal{N}_{\text{inter}}$ to refine $\mathbf{c}_i$ using $\mathbf{c}_{i+1}$, and $\mathbf{c}_{i+2}$ as

$$\tilde{\mathbf{c}}_i = \mathcal{N}_{\text{inter}}(\mathbf{c}_i, \mathbf{c}_{i+1}, \mathbf{c}_{i+2}). \tag{3}$$

Note that we do not directly treat the output of the attention module as the refined code $\tilde{\mathbf{c}}_i$. Instead, we first concatenate such output and $\mathbf{c}_i$, followed by mapping the result back into the 256-dimensional code $\tilde{\mathbf{c}}_i$ based on MLP. This operation maintains the key role of the i-th point set to predict the code $\tilde{\mathbf{c}}_i$. The refined latent code $\tilde{\mathbf{c}}_i$ encapsulates multi-view constraints on the deformation of template, providing higher reliability than the original code $\mathbf{c}_i$. In practice, for each instance, we find up to three neighbors with the same semantic label and similar sizes. The size similarly is measured by the volume of bounding boxes in the world frame. Overall, the above sub-networks $\mathcal{N}_{\text{point}}$, $\mathcal{N}_{\text{intra}}$, and $\mathcal{N}_{\text{inter}}$ constitute our code prediction network. Details of network architecture are available in the supplementary material.

Loss Function. We follow [29] to train our network using SDF values as supervision. Briefly, for the i-th training instance, we sample Z positions $\{\mathbf{p}_i^z\}_{z=1}^Z$ in the canonical space. We use the ground truth mesh of this instance to obtain the ground truth SDF values $\{\hat{v}_i^z\}_{z=1}^Z$ at the sampled positions. We adopt L_1 loss function to minimize the dif-

ference between each pair of the estimated SDF value v_i^z in Eq. (1) and ground truth SDF value $\hat{v}_i^z$, i.e.,

$$L_1 = \sum_z \left| \mathcal{Q}\big(\mathcal{W}(\tilde{\mathbf{c}}_i, \mathbf{p}_i^z)\big) - \hat{v}_i^z \right|. \tag{4}$$

Eq. (4) is with respect to our latent code prediction network and deep implicit template $\{\mathcal{W}, \mathcal{Q}\}$. In addition to training our code prediction network, we fine-tune the pre-trained warping sub-network $\mathcal{W}$, but fix the pre-trained query sub-network $\mathcal{Q}$. The reason is that the latent code and warping sub-network $\mathcal{W}$ jointly determine the coordinate warping.

4.2. Transformation Estimation

Given a point set of an instance in the world frame, we aim to estimate its transformation between the world and canonical frames. Existing methods can be classified into two main categories. The first type of methods [27, 15] relies on the regressed bounding boxes, and is prone to be unreliable due to the lack of geometric constraints. The second type of methods [1, 2] is based on the point correspondences between the input point set and a CAD model in the canonical frame. To establish such a pair, [1] exhaustively searches for the best-matched CAD model in the database and thus results in unsatisfactory efficiency. [2] introduces a neural network to directly retrieve a CAD model, but can hardly guarantee the reliability of retrieval. To overcome these limitations, we propose a correspondence-based method using a coarse-to-fine geometric search strategy.

Recall that we pre-train a template network using numerous CAD models from the same category. As a by-product, each model is associated with a pre-trained latent code. We employ K-Means algorithm [24] to cluster these models based on their latent codes. We repeat clustering on each newly generated group using a loose-to-tight threshold. Accordingly, we generate a hierarchical tree of CAD models (see the supplementary material for details). Along this tree, we search for the best-matched CAD model against the input point set based on geometric evaluation. Specifically, given the input point set and a candidate CAD model, we first use FCGF [7] to establish putative point correspondences. Then we find inliers of these correspondences by combing PointDSC [3] for spatial consistency and RANSAC [12] for parametric consistency. Accordingly, each candidate CAD model is associated with an inlier ratio. The CAD model achieving the highest inlier ratio corresponds to the best-matched model. We treat the transformation between this model and the input point set as the transformation of the input point set.

Overall, our algorithm is efficient by considering the pre-trained latent codes to avoid an exhaustive search. Moreover, it is reliable thanks to geometric constraints.

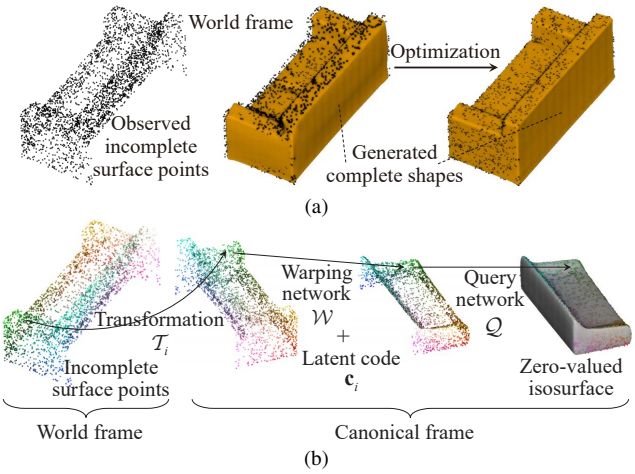

Figure 6. Illustration of our algorithm to jointly optimize the transformation and latent code. (a) In the world frame, the generated shape is better aligned to the observed surface points thanks to our optimization. (b) Surface points in the world frame correspond to the zero-valued isosurface of SDF in the canonical frame.

4.3. Optimization

In this section, we aim to optimize the latent codes and transformations estimated above. Let us first consider the i-th instance to illustrate the motivation of optimization. Given an incomplete point set in the world frame (see Fig. 6(a)-left), we estimate its transformation $\mathcal{T}_i$ and latent code $\mathbf{c}_i$ to generate a complete shape in the world frame (see Fig. 6(a)-middle). Due to the effect of noise, the generated shape is not strictly aligned with the observed point set. As shown in Fig. 6(a)-right, we propose to jointly optimize the latent code $\mathbf{c}_i$ and transformation $\mathcal{T}_i$ to better fit the generated shape to the observed point set.

As shown in Fig. 6(b), we leverage the fact that surface points $\{\mathbf{s}_i^j\}_{j=1}^J$ in the world frame should correspond to the zero-valued isosurface of SDF in the canonical frame (this isosurface is exactly the surface of template). Specifically, we first use the estimated transformation $\mathcal{T}_i$ to transform each point $\mathbf{s}_i^j$ into the canonical frame. Then for each transformed point $\mathcal{T}_i(\mathbf{s}_i^j)$, we estimate its SDF value based on latent code $\mathbf{c}_i$ and deep implicit template $\{\mathcal{W}, \mathcal{Q}\}$ (see Eq. (1)). We define an objective function to enforce the

Table 1. Comparisons between various datasets for semantic scene completion. Detailed illustrations are available in the supplementary material.

	Alignment	Rationality	Completeness	Watertightness
Scan2CAD [1]	✗	✓✗	✗	✗
SOSPC [14]	✓	✗	✗	✗
SCFusion [42]	✓	✓✗	✗	✗
ScanARCW (our)	✓	✓	✓	✓

zero-valued SDF constraint, i.e.,

$$\min_{\mathcal{T}_i, \mathbf{c}_i} \sum_j \left| \mathcal{Q}\Big(\mathcal{W}\big(\mathbf{c}_i, \mathcal{T}_i(\mathbf{s}_i^j)\big)\Big) \right|. \tag{5}$$

This objective function is with respect to both latent code $\mathbf{c}_i$ and transformation $\mathcal{T}_i$. To minimize it, we leverage Adam algorithm [19]. We initialize the latent code $\mathbf{c}_i$ and transformation $\mathcal{T}_i$ by the results obtained in Sections 4.1 and 4.2, and fix the weights of networks $\{\mathcal{W}, \mathcal{Q}\}$. During minimization, we limit the variation ranges of transformation and each element of latent code to achieve a moderate update.

5. Our New Dataset

Existing datasets for semantic scene completion include Scan2CAD [1], SOSPC [14], and SCFusion [42] datasets. They are all built on ScanNet dataset [9]. As shown in Table 1, they have at least one of the following limitations. 1) An observed incomplete point set may not be strictly aligned with its ground truth complete mesh. 2) Relationship between ground truth meshes may be irrational, e.g., intersection or floating. 3) Ground truth meshes of the background point cloud, e.g., ceiling, wall, and floor are lacking. 4) Ground truth meshes may not be watertight, which affects the SDF-based research. To overcome these limitations, we establish a new dataset achieving Alignment, Rationality, Completeness, and Watertightness. We call it "ScanARCW" dataset.

We briefly introduce the procedure of our dataset establishment. First, we place CAD models of both foreground instances and background layout in the world frame. These models are newly generated (e.g., background layout), or re-used from Scan2CAD dataset. We process these models in batches for watertightness, and partly edit their poses for rationality. The results are treated as ground truth complete meshes. Then we project these meshes with semantic labels by the cameras of ScanNet dataset, obtaining a set of depth maps and semantic label maps. After that, we back-project these maps to obtain an incomplete point cloud of the scene. This step guarantees that the ground truth meshes and observed point cloud are strictly aligned.

Additional information about our dataset is available in the supplementary material. We release our dataset on the project website[2].

6. Experiments

We first introduce our experimental setup in Section 6.1. Then we compare our approach with state-of-the-art methods in Section 6.2, followed by presenting ablation study in Section 6.3.

6.1. Experimental Setup

Dataset and Categories. As mentioned above, on our ScanARCW dataset, the observed point cloud and ground truth meshes are strictly aligned. Therefore, we conduct experiments on this dataset for an unbiased evaluation. We consider dominant categories in indoor environments, including sofa, chair, table, cabinet, bookshelf, bathtub, and bed. We additionally complete the background composed of ceiling, wall, and floor. For a fair comparison with [27, 36], we only report the results of foreground completion in the main manuscript. The results of background completion are available in the supplementary material.

Evaluation Criteria. We adopt 3D intersection-over-union (IoU) [27] and point coverage ratio (PCR) [36] as our evaluation metrics. Specifically, in the world frame, we voxelize the generated and ground truth meshes respectively, and compute IoU (%) by $\frac{\text{Volume of overlap}}{\text{Volume of Union}}$; For each point from an observed point set, we compute its smallest distance to the generated mesh. We treat a point whose distance is smaller than the threshold as an inlier and compute PCR (%) by $\frac{\text{Number of Inliers}}{\text{Number of Points}}$. Considering the quality of instance segmentation or object detection, we follow [27, 36] to report the average precision with respect to IoU and PCR. Briefly, IoU or PCR determines true/false based on a threshold such as 0.25, 0.5, or 0.75. Confidence score of segmentation or detection determines positive/negative.

Implementation Details. We use Adam [19] to minimize our loss in Eq. (4). The learning rate is 10^{-5}, batch size is 8, and number of epochs is 1000. The number of edges K used in EdgeConv is 8. We follow the setup of [46] to pretrain deep implicit templates.

[2] https://sites.google.com/view/haoangli/projects/ddit

Table 2. Quantitative comparisons with state-of-the-art methods. For each category, we report the average precision with respect to PCR and IoU at the threshold of 0.5.

	Sofa		Chair		Table		Cabinet		Bookshelf		Bathtub		Bed		Mean	
	PCR	IoU	PCR	IoU	PCR	IoU	PCR	IoU	PCR	IoU	PCR	IoU	PCR	IoU	PCR	IoU
RfD-Net [27]	23.37	36.21	76.78	21.01	32.06	9.75	23.15	23.59	29.59	11.60	53.48	31.97	45.37	49.29	40.54	26.20
DIMR [36]	**62.11**	39.54	**77.50**	14.27	49.14	10.83	18.31	19.49	27.88	10.58	**75.60**	**34.36**	49.02	51.73	51.36	25.82
DDIT (our)	61.59	**44.43**	75.04	**39.83**	**67.62**	**41.96**	**53.92**	**50.39**	**49.38**	**30.51**	43.23	20.49	**52.48**	**57.76**	**57.60**	**40.76**

6.2. Comparisons with State-of-the-art Methods

Methods for Comparison. We compare our DDIT with the state-of-the-art methods introduced in Section 2:

- RfD-Net [27] first conducts object detection and then extracts foreground point sets. Occupancies of 3D positions are implicitly predicted for mesh generation.
- DIMR [36] relies on instance segmentation. The latent code of a point set is mesh-aware. Such a code is fed to a pre-trained decoder to directly predict a mesh.

For a fair comparison, we re-train RfD-Net and DIMR on our ScanARCW dataset using their recommended parameters. We conduct all the tests on a computer equipped with NVIDIA RTX A6000 GPU.

Results. Table 2 and Fig. 7 show that RfD-Net may generate unreasonable shapes due to the lack of constraints on shape prior. Moreover, its object detection module is prone to result in some false positives (e.g., redundant tables). DIMR is more accurate than RfD-Net thanks to its mesh-aware latent codes. However, its generated shapes

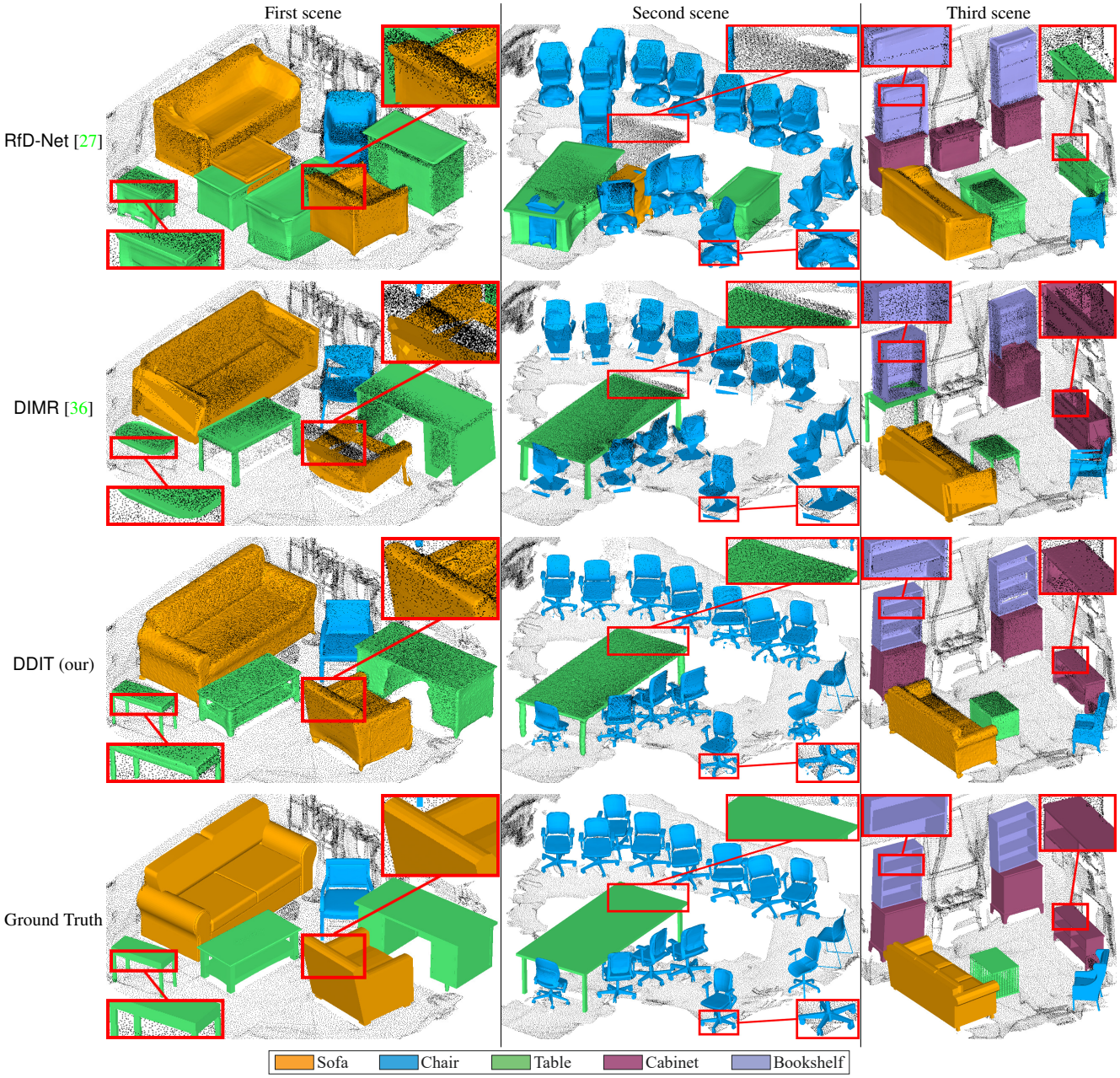

Figure 7. Qualitative comparisons with state-of-the-art methods in three representative scenes. Red bounding boxes correspond to the zoom-in view of some instances. Additional results are available in the supplementary material.

Table 3. Quantitative comparisons with state-of-the-art methods. We report the mean average precision over all categories with respect to PCR at the thresholds of 0.25, 0.5, and 0.75.

	PCR@0.25	PCR@0.5	PCR@0.75
RfD-Net [27]	63.33	40.54	20.61
DIMR [36]	66.41	51.36	29.55
DDIT (our)	**67.63**	**57.60**	**42.74**

Table 4. Ablation study of our inter-instance information. We report mean average precision with respect to IoU and PCR at the threshold of 0.5.

	PCR	IoU
Intra	55.74	37.90
Intra+Inter	**57.60**	**40.76**

Table 5. Ablation study of our optimization strategy. We report mean average precision with respect to IoU and PCR at the threshold of 0.5.

	PCR	IoU
No-optim	56.89	38.82
Optim	**57.60**	**40.76**

may show unsatisfactory details. For example, an instance appears to be assembled by a set of simple primitives without smooth transition (see bases of office chairs in Fig. 7). Moreover, the generated shapes may be not well-aligned to the observed point sets. The reason is that the instance transformation is directly regressed by a network without strict geometric constraints and thus may be unreliable. Our DDIT achieves the highest accuracy since the template leads to reasonable overall shape and latent code guarantees fine details. Note that for the categories of chair and table, average precisions with respect to IoU are significantly lower than average precisions with respect to PCR. The reason is that IoU computation is sensitive to the noise of thin parts of an instance, e.g., chair leg and table surface [27]. Such difference in average precision is relatively small for our DDIT thanks to a better alignment between the generated and ground truth meshes.

As shown in Table 3, the performance gap between the above methods becomes larger at a higher threshold. The reason is that the meshed generated by our DDIT is better aligned to the observed point sets. For example, assume that Mesh 1 generated by DIMR and Mesh 2 generated by our DDIT lead to PCR of 0.6 and 0.8, respectively. Mesh 1 is a true positive at the threshold of 0.5, but a false positive at the threshold of 0.75. By contrast, Mesh 2 remains a true positive at different thresholds. A larger number of true positives lead to a higher mean precision. In addition, the instance segmentation-based DIMR and DDIT is more accurate than the object detection-based RfD-Net. This partly demonstrates the superiority of instance segmentation in semantic scene completion task, as discussed in [36].

6.3. Ablation Study

Inter-instance Information (see Section 4.1). We denote our method using only intra-instance information by Intra,

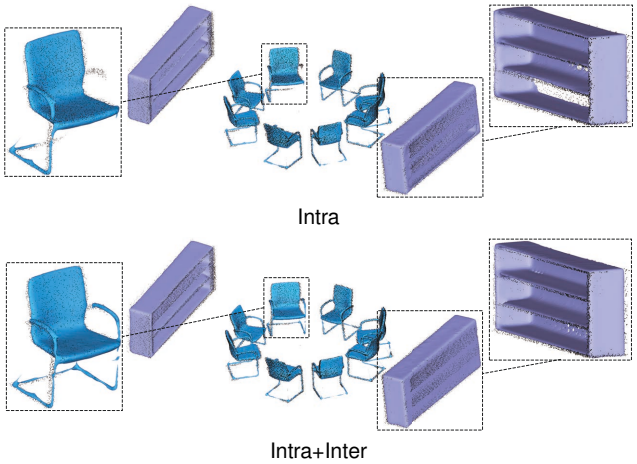

Figure 8. Ablation study of our inter-instance information. We present a qualitative comparison in a representative scene. Additional results are available in the supplementary material.

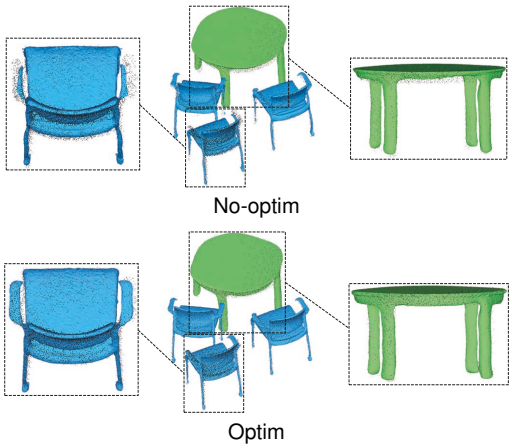

Figure 9. Ablation study of our optimization strategy. We present a qualitative comparison in a representative scene. Additional results are available in the supplementary material.

and our method using both inter- and intra-instance information by Intra+Inter. As shown in Table 4 and Fig. 8, the meshes generated by Intra may still lack some parts. Intra+Inter generates more complete meshes than Intra, demonstrating the effectiveness of our inter-instance information. The reason is that neighboring instances commonly have similar shapes but different reconstructed parts. Combining their information can enforce multi-view constraints on a generated mesh.

Optimization (see Section 4.3). We denote our method without optimization by No-optim, and our method using optimization by Optim. Table 5 and Fig. 9 show that for No-optim, there is still room for accuracy improvement. The reasons are that the predicted latent code may be partly affected by the uneven density of the input point set, and the computed transformation is inevitably affected by the noise of point correspondences. Optim improves the accuracy of both latent code and transformation. Accordingly, the gen-

erated shapes and observed point sets are better aligned.

Please note that our method still outperforms the state-of-the-art approaches RfD-Net and DIMR even without using inter-instance information (see Tables 3 and 4), or optimization (see Tables 3 and 5).

7. Conclusions

We presented DDIT, a semantic scene completion method by deforming deep implicit templates with latent codes. Our completed instances show reasonable overall shapes and fine details, and also are well-aligned to the observed point cloud. This is owed to our reliable estimation and optimization of both latent code and transformation. In addition, we established a new dataset to overcome the limitations of existing ones. Experiments showed that our method outperforms state-of-the-art approaches.

Acknowledgments. The work of Daniel Cremers was supported by the ERC Advanced Grant SIMULACRON, by the Munich Center for Machine Learning and by the EPSRC Programme Grant VisualAI EP/T028572/1. The work of Yun-Hui Liu was supported by the Shenzhen Portion of Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone under HZQBKCZYB-20200089, the InnoHK of the Government of Hong Kong via the Hong Kong Centre for Logistics Robotics, and the CUHK T-Sone Robotics Institute. We thank Yihao Wang and Kai Chen for fruitful discussion.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Niessner. Scan2CAD: Learning CAD model alignment in RGB-D scans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2614–2623, 2019. 1, 2, 5, 6

[2] Armen Avetisyan, Angela Dai, and Matthias Nießner. End-to-end CAD model retrieval and 9DOF alignment in 3D scans. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2551–2560, 2019. 1, 3, 5

[3] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. PointDSC: Robust point cloud registration using deep spatial consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15859–15869, 2021. 5

[4] Yingjie Cai, Xuesong Chen, Chao Zhang, Kwan-Yee Lin, Xiaogang Wang, and Hongsheng Li. Semantic scene completion via integrating instances and scene in-the-loop. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 324–333, 2021. 1, 3

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[6] Wen Chen, Haoang Li, Qiang Nie, and Yun-Hui Liu. Deterministic point cloud registration via novel transformation decomposition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6338–6346, 2022. 1

[7] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8957–8965, 2019. 5

[8] James Coughlan and Alan Yuille. Manhattan world: Compass direction from a single image by Bayesian inference. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 941–947, 1999. 3

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017. 6

[10] Angela Dai, Christian Diller, and Matthias Nießner. SG-NN: Sparse generative neural networks for self-supervised scene completion of RGB-D scans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 849–858, 2020. 1

[11] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2018. 1, 2

[12] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 5

[13] Ruochong Fu, Hang Wu, Mengxiang Hao, and Yubin Miao. Semantic scene completion through multi-level feature fusion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8399–8406, 2022. 3

[14] Shreyas Hampali, Sinisa Stekovic, Sayan Deb Sarkar, Chetan Srinivasa Kumar, Friedrich Fraundorfer, and Vincent Lepetit. Monte Carlo scene search for 3D scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13804–13813, 2021. 2, 5, 6

[15] Ji Hou, Angela Dai, and Matthias Niessner. RevealNet: Seeing behind objects in RGB-D scans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2098–2107, 2020. 1, 2, 5

[16] Vladislav Ishimtsev, Alexey Bokhovkin, Alexey Artemov, Savva Ignatyev, Matthias Niessner, Denis Zorin, and Evgeny Burnaev. CAD-Deform: Deformable fitting of CAD models to 3D scans. In *European Conference on Computer Vision (ECCV)*, pages 599–628, 2020. 1, 3

[17] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):1–13, 2013. 1

[18] Pyojin Kim, Haoang Li, and Kyungdon Joo. Quasi-globally optimal and real-time visual compass in Manhattan structured environments. *IEEE Robotics and Automation Letters (RAL)*, 7(2):2613–2620, 2022. 3

[19] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 6

[20] Haoang Li, Yazhou Xing, Ji Zhao, Jean-Charles Bazin, Zhe Liu, and Yun-Hui Liu. Leveraging structural regularity

of Atlanta world for monocular SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2412–2418, 2019. 3

[21] Haoang Li, Jian Yao, Jean-Charles Bazin, Xiaohu Lu, Yazhou Xing, and Kang Liu. A monocular SLAM system leveraging structural regularity in Manhattan world. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2518–2525, 2018. 3

[22] Haoang Li, Ji Zhao, Jean-Charles Bazin, Pyojin Kim, Kyungdon Joo, Zhenjun Zhao, and Yun-Hui Liu. Hong Kong World: Leveraging structural regularity for line-based SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1–18, 2023. 1

[23] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yunhui Liu. LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2831–2840, 2019. 1

[24] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 5

[25] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987. 3

[26] Mahyar Najibi, Guangda Lai, Abhijit Kundu, Zhichao Lu, Vivek Rathod, Thomas Funkhouser, Caroline Pantofaru, David Ross, Larry S Davis, and Alireza Fathi. DOPS: Learning to detect 3D objects and predict their 3D shapes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11913–11922, 2020. 1, 2

[27] Yinyu Nie, Ji Hou, Xiaoguang Han, and Matthias Niessner. RfD-Net: Point scene understanding by semantic instance reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4608–4618, 2021. 1, 2, 5, 6, 7, 8

[28] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2027–2034, 2013. 4

[29] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 3, 4

[30] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision (ECCV)*, pages 523–540, 2020. 1

[31] Christoph Rist, David Emmerichs, Markus Enzweiler, and Dariu Gavrila. Semantic scene completion using local deep implicit functions on LiDAR data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(10):7205–7218, 2021. 1, 2

[32] Luis Roldao, Raoul De Charette, and Anne Verroust-Blondet. 3D semantic scene completion: A survey. *International Journal of Computer Vision (IJCV)*, 130(8):1978–2005, 2022. 1, 3

[33] Grant Schindler and Frank Dellaert. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 203–209, 2004. 3

[34] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D for 3D semantic instance segmentation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 3

[35] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1746–1754, 2017. 1, 2

[36] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. Point scene understanding via disentangled instance mesh reconstruction. In *European Conference on Computer Vision (ECCV)*, pages 684–701, 2022. 2, 3, 6, 7, 8

[37] Philip Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding (CVIU)*, 78(1):138–156, 2000. 3

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. 4

[39] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):1–12, 2019. 4

[40] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. ForkNet: Multi-branch volumetric semantic completion from a single depth image. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8608–8617, 2019. 1, 2

[41] Huanshu Wei, Zhijian Qiao, Zhe Liu, Chuanzhe Suo, Peng Yin, Yueling Shen, Haoang Li, and Hesheng Wang. End-to-end 3D point cloud learning for registration task using virtual correspondences. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2678–2683, 2020. 1

[42] Shun-Cheng Wu, Keisuke Tateno, Nassir Navab, and Federico Tombari. SCFusion: Real-time incremental scene reconstruction with semantic completion. In *International Conference on 3D Vision (3DV)*, pages 801–810, 2020. 2, 5, 6

[43] Yaqi Xia, Yan Xia, Wei Li, Rui Song, Kailang Cao, and Uwe Stilla. ASFM-Net: Asymmetrical siamese feature matching network for point completion. In *ACM International Conference on Multimedia (MM)*, pages 1938–1947, 2021. 1

[44] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. SOE-Net: A self-attention and orientation encoding network for point cloud based place recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11348–11357, 2021. 4

[45] Yan Xia, Yusheng Xu, Cheng Wang, and Uwe Stilla. VPC-Net: Completion of 3D vehicles from MLS point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 174:166–181, 2021. 1

[46] Zerong Zheng, Tao Yu, Qionghai Dai, and Yebin Liu. Deep implicit templates for 3D shape representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1429–1439, 2021. 1, 3, 6