



**HAL**  
open science

## 3D Semantic Scene Completion: a Survey

Luis Roldão, Raoul de Charette, Anne Verroust-Blondet

► **To cite this version:**

Luis Roldão, Raoul de Charette, Anne Verroust-Blondet. 3D Semantic Scene Completion: a Survey. International Journal of Computer Vision, 2021, 10.1007/s11263-021-01504-5 . hal-03324932

**HAL Id: hal-03324932**

**<https://hal.science/hal-03324932>**

Submitted on 24 Aug 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# 3D Semantic Scene Completion: a Survey

Luis Roldão<sup>1,2</sup> · Raoul de Charette<sup>1</sup> · Anne Verroust-Blondet<sup>1</sup>

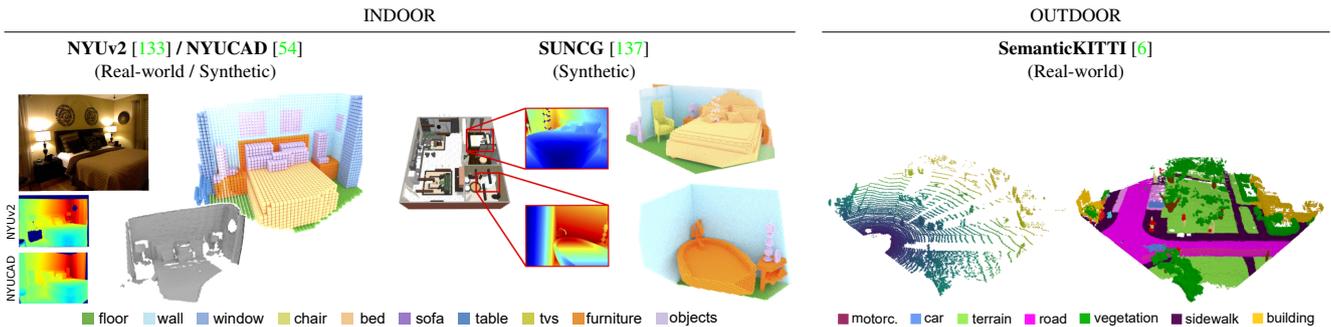


Fig. 1: **Popular datasets for Semantic Scene Completion (SSC)**. From an incomplete input view, the SSC task consists in the joint estimation of both geometry and semantics of the scene. The figure shows the 4 most popular datasets for SSC, each showing input data and ground truth. The complexity of SSC lies in the completion of unobserved / occluded regions and in the sparse supervision signal (notice that real ground truth is incomplete). (Images source [124, 137]).

**Abstract** Semantic Scene Completion (SSC) aims to jointly estimate the complete geometry and semantics of a scene, assuming partial sparse input. In the last years following the multiplication of large-scale 3D datasets, SSC has gained significant momentum in the research community because it holds unresolved challenges. Specifically, SSC lies in the ambiguous completion of large unobserved areas and the weak supervision signal of the ground truth. This led to a substantially increasing number of papers on the matter. This survey aims to identify, compare and analyze the techniques providing a critical analysis of the SSC literature on both methods and datasets. Throughout the paper, we provide an in-depth analysis of the existing works covering all choices made by the authors while highlighting the remaining avenues of research. SSC performance of the SoA on the most popular datasets is also evaluated and analyzed.

**Keywords** Semantic scene completion, 3D Vision, Semantic Segmentation, Scene understanding, Scene reconstruction, Point cloud.

L. Roldão (luis.rol dao@inria.fr / luis.rol dao@akka.eu).  
R. de Charette (raoul.de-charette@inria.fr).  
A. Verroust-Blondet (anne.verroust@inria.fr).

<sup>1</sup> Inria, Paris, France.

<sup>2</sup> AKKA Research, Guyancourt, France.

## 1 Introduction

Understanding the 3D surroundings is a natural ability for humans, who are capable to leverage prior knowledge to estimate geometry and semantics, even in large occluded areas. This proves more difficult for computers, which has drawn wide interest from computer vision researchers in recent years [53]. Indeed, 3D scene understanding is a crucial feature for many applications, such as robotic navigation or augmented reality, where geometrical and semantics understanding is key to leverage interaction with the real world [44]. Nonetheless, vision sensors only provide partial observations of the world given their limited field of view, sparse sensing, and measurement noise, capturing a partial and incomplete representation of the scene.

To address this, Scene Completion (SC) has been used for a long time to infer the complete geometry of a scene given one or more 2D/3D observations, historically using a wide variety of more or less sophisticated interpolation methods. With recent advances of 3D deep learning, Semantic Scene Completion (SSC) has been introduced as an extension of SC, where semantics and geometry are *jointly* inferred for the whole scene, departing from the idea that they are entangled [137]. Consequently, along with the addition of semantics, the SSC task has significantly departed from original SC in terms of nature and sparsity of the input data.

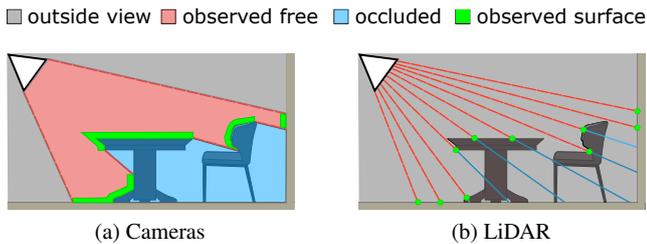


Fig. 2: **Scene acquisition.** A camera (RGB, RGB-D, Depth) senses dense volumes but produces noisy depth measurements (a), while LiDAR – more accurate – is significantly sparser (b). (Inspired from: [137]).

Fig. 1 shows samples of input and ground truth for the four most popular SSC datasets. The complexity of the SSC task lies in the sparsity of the input data (see holes in depth or LiDAR input), and the incomplete ground truth (resulting of frame aggregation) providing a rather weak guidance. Different from object-level completion [4, 172] or from scene reconstruction where multiple views are aggregated [109, 182], SSC requires in-depth understanding of the entire scene heavily relying on learned priors to resolve ambiguities. The increasing number of large scale datasets [6, 133, 137, 140] have encouraged new SSC works in the last years. On connex topics, 3D (deep) vision has been thoroughly reviewed [53, 82, 88, 91] including surveys on 3D representations [2] and task-oriented reviews like 3D semantic segmentation [168, 175], 3D reconstruction [109, 182], 3D object detection [68], etc. Still, no survey seems to exist on this hot SSC topic and navigating the literature is not trivial.

In this article, we propose the first comprehensive and critical review of the Semantic Scene Completion (SSC) literature, focusing on methods and datasets. With this systematic survey, we expose a critical analysis of the SSC knowledge and highlight the missing areas for future research. We aim to provide new insights to informed readers and help new ones navigate in this emerging field which gained significant momentum in the past few years. To the best of our knowledge this is the first SSC survey, which topic has only been briefly covered in recent 3D surveys [53, 91].

To study SSC, this paper is organized as follows. We first introduce and formalize the problem of Semantic Scene Completion in Sec. 2, briefly brushing closely related topics. Ad-hoc datasets employed for the task and introduction to common 3D scene representations are covered in Sec. 3. We study the existing works in Sec. 4, highlighting the different input encodings, deep architectures, design choices, and training strategies employed. The section ends with an analysis of the current performances, followed by a discussion in Sec. 5.

## 2 Problem statement

Let  $x$  be an incomplete 3D representation of a scene, Semantic Scene Completion (SSC) is the function  $f(\cdot)$  inferring a dense semantically labeled scene  $\hat{y}$  such that  $f(x) = \hat{y}$ , best approximates the real 3D scene  $y$ . Most often,  $x$  is significantly sparser than  $y$  and the complexity lies in the inherent ambiguity, especially where large chunks of data are missing, due to sparse sensing or occlusions (see Fig. 2). Subsequently, the problem cannot be addressed by interpolating data in  $x$  and is most often solved by learning priors from  $(x, y)$  pairs of sparse input and dense 3D scenes with semantic labels.

The nature of the sparse 3D input  $x$  greatly affects the task complexity. While 3D data can be obtained from a wide variety of sensors, RGB-D/stereo cameras or LiDARs are commonly employed. The former, for example, provides a dense description of the visible surfaces where missing regions correspond to occluded areas, as shown in Fig. 2a. This reduces the SSC task to estimating semantic completion only in the occluded regions [137]. Conversely, LiDAR data provides considerably sparser sensing, with density decreasing afar and point-wise returns from laser beams cover an infinitesimal portion of the space leading to a high proportion of unknown volume, as shown in Fig. 2b.

The rest of this section provides the reader with a brief overview of related tasks, covering early works and foundations.

### 2.1 Related tasks

Semantic scene completion inspires from closely related tasks like shape completion, semantic segmentation, and more recently semantic instance completion. Thereof, SSC benefits from individual insights of these tasks which we briefly review their literature pointing to the respective surveys.

*Completion.* Completion algorithms initially used interpolation [29] or energy minimization [69, 100, 139] techniques to complete small missing regions. Completion works were first devoted to object shape completion, which infers occlusion-free object representation. For instance, some trends exploit symmetry [14, 108, 134, 143, 148] and are reviewed in [96]. Another common approach is to rely on *prior* 3D models to best fit sparse input [75, 81, 107, 123, 131]. In recent years, model-based techniques and new large-scale datasets enlarged the scope of action by enabling inference of complete occluded parts in both scanned objects [27, 58, 106, 118, 130, 135, 141, 150, 170, 172] and entire scenes [25, 38, 181]. Moreover, contemporary research shows promising results on challenging multi-object reconstruction from as single RGB image [34]. For further analysis, we refer readers to related surveys [57, 96, 170].

Dataset	Year	Type	Nature	Input→Ground truth	3D Sensor	# Classes	Tasks*					#Sequences	#Frames
							SSC	DE	SPE	SS	OC		
<b>NYUv2 [133]<sup>a</sup></b>	2012	<b>Real-world<sup>†</sup></b>	<b>Indoor</b>	<b>RGB-D → Mesh/Voxel</b>	<b>RGB-D</b>	<b>894 (11)</b>	✓	✓	✓	✓	✓	<b>1449</b>	<b>795/654</b>
SUN3D [165]	2013	Real-world	Indoor	RGB-D → Points	RGB-D	-	✓	✓	✓	✓	✓	254	-
<b>NYUCAD [38]<sup>b</sup></b>	2013	<b>Synthetic</b>	<b>Indoor</b>	<b>RGB-D → Mesh/Voxel</b>	<b>RGB-D</b>	<b>894 (11)</b>	✓	✓	✓	✓	✓	<b>1449</b>	<b>795/654</b>
SceneNet [60]	2015	Synthetic	Indoor	RGB-D → Mesh	RGB-D <sup>‡</sup>	11	✓	✓	✓	✓	✓	57	-
SceneNN [64]	2016	Real-world	Indoor	RGB-D → Mesh	RGB-D	22	✓	✓	✓	✓	✓	100	-
<b>SUNCG [137]</b>	2017	<b>Synthetic</b>	<b>Indoor</b>	<b>Depth → Mesh/Voxel</b>	<b>RGB-D<sup>‡</sup></b>	<b>84 (11)</b>	✓			✓		<b>45622</b>	<b>139368/470</b>
Matterport3D [13]	2017	Real-world	Indoor	RGB-D → Mesh	3D Scanner	40 (11)	✓	✓		✓		707	72/18
ScanNet [24]	2017	Real-world	Indoor	RGB-D → Mesh	RGB-D	20 (11)	✓	✓	✓	✓	✓	1513	1201/312
2D-3D-S [3]	2017	Real-world	Indoor	RGB-D → Mesh	3D Scanner	13	✓	✓	✓	✓	✓	270	-
SUNCG-RGBD [86] <sup>c</sup>	2018	Synthetic	Indoor	RGB-D → Mesh/Voxel	RGB-D <sup>‡</sup>	84 (11)	✓			✓		45622	13011/499
<b>SemanticKITTI [6]</b>	2019	<b>Real-world</b>	<b>Outdoor</b>	<b>Points/RGB → Points/Voxel</b>	<b>Lidar-64</b>	<b>28 (19)</b>	✓	✓		✓		<b>22</b>	<b>23201/20351</b>
SynthCity [49]	2019	Synthetic	Outdoor	Points → Points	Lidar <sup>‡</sup>	9	✓			✓		9	-
CompleteScanNet [164] <sup>d</sup>	2020	Real-world <sup>†</sup>	Indoor	RGB-D → Mesh/Voxel	RGB-D	11	✓	✓	✓	✓	✓	1513	45448/11238
SemanticPOSS [105]	2020	Real-world	Outdoor	Points/RGB → Points	Lidar-40	14	✓	✓	✓	✓	✓	2988	-

<sup>‡</sup> Simulated sensor. <sup>†</sup> Synthetically augmented. <sup>a</sup> Mesh annotations from [52]. <sup>b</sup> Derivates from NYUv2 [133] by rendering depth images from mesh annotation.

<sup>c</sup> Derivates from subset of SUNCG [137] where missing RGB images were rendered. <sup>d</sup> Derivates from ScanNet [24] by fitting CAD models to dense mesh.

\* SSC: Semantic Scene Completion; DE: Depth Estimation; SPE: Sensor Pose Estimation; SS: Semantic Segmentation; OC: Object Classification; SNE: Surface Normal Estimation

Table 1: **SSC datasets**. We list here datasets readily usable for the SSC task in chronological order. Popular datasets are **bold** and previewed in Fig. 1. Classes show the total number of semantic classes and when it differs, SSC classes in parenthesis.

*Semantics*. Traditional segmentation techniques reviewed in [102] were based on hand-crafted features, statistical rules, and bottom-up procedures, combined with traditional classifiers. The advances in deep learning have reshuffled the cards [167]. Initial 3D deep techniques relied on multiviews processed by 2D CNNs [8, 9, 142] and were quickly replaced by the use of 3D CNNs which operate on voxels [26, 93, 95, 120, 145, 153], tough suffering from memory and computation shortcomings. Point-based networks [74, 80, 115, 116, 147, 158] remedied this problem by operating on points and quickly became popular for segmentation, though generative tasks are still a challenge. Therefore, if the point space is pre-defined, SSC can be seen as a point segmentation task [180]. We refer readers to dedicated surveys on semantic segmentation [42, 167].

*Semantic Instance Completion*. Different from SSC, semantic instance completion is applied only at the objects instance level and cannot complete scene background like walls, floor, etc. Most existing methods require instance detection and complete the latter with CAD models [4, 99, 129] or a completion head [63, 63, 97, 104, 172]. Notably, [5] detects both objects and the scene layout to output a complete lightweight CAD representation. These approaches are commonly based on 3D object detection [114, 132, 136] or instance segmentation [62]. Furthermore, some recent methods tackle multiple object completion from single images either through individual object detection and reconstruction [47, 67, 72, 172] or joint completion by understanding the scene structure [103, 113].

Song *et al.* [137] were the first to address semantic segmentation and scene completion jointly, showing that both tasks can mutually benefit. Since then, many SSC works

gather ideas from the above described lines of work and are extensively reviewed in Sec. 4.

### 3 Datasets and representations

This section presents existing SSC datasets (Sec. 3.1) and commonly used 3D representations for SSC (Sec. 3.2).

#### 3.1 Datasets

A comprehensive list of all SSC ready datasets is shown in Tab. 1. We denote as *SSC ready* any dataset containing pairs of sparse/dense data with semantics label. Note that while 14 datasets meet these criteria, only half has been used for SSC among which the four most popular are bold in the table and previewed in Fig. 1. Overall, there is a prevalence of indoor stationary datasets [3, 13, 24, 60, 64, 133, 137, 164] as opposed to outdoors [6, 49, 105].

*Datasets creation*. Synthetic datasets can easily be obtained by sampling 3D object meshes [39, 60, 137] or simulating sensors in virtual environments [31, 41, 49, 126]. Their evident advantage is the virtually free labeling of data, though transferability of synthetically learned features is arguable. Real datasets are quite costly to record and annotate, and require a significant processing effort. Indoor datasets [3, 13, 24, 64, 133, 164, 165] are commonly captured with RGB-D or stereo sensors. Conversely, outdoor datasets [6, 10, 105] are often recorded with LiDAR and optional camera. They are dominated by autonomous driving applications, recorded in (peri)-urban environment, and must subsequently account for dynamic agents (e.g. moving objects, ego-motion, etc.).

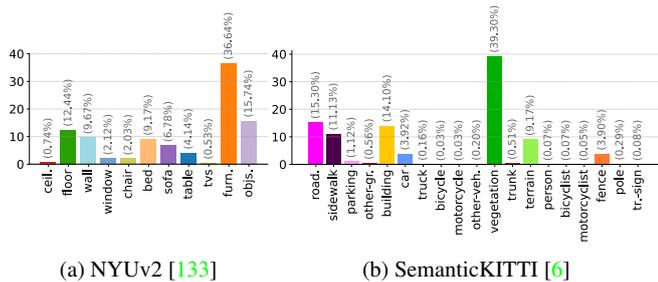


Fig. 3: **Semantics distribution.** Class-wise frequencies of the most popular real datasets prove to be highly imbalanced.

*Ground truth generation.* An evident challenge is the dense annotation of such datasets. Specifically, while indoor stationary scenes can be entirely captured from multiple views or rotating apparatus, 3D outdoor dynamic scenes are virtually impossible to capture *entirely* as it would require ubiquitous scene sensing.

Subsequently, ground truth  $y$  is obtained from the aggregation and labeling of sparse sequential data  $\{y_0, y_1, \dots, y_T\}$  over a small time window  $T$ . Multi-frame registration is usually leveraged to that matters, assuming that consecutive frames have an overlapping field of view. For RGB-D datasets, mostly stationary and indoors, it is commonly achieved from Structure from Motion (SfM) [98, 128] or visual SLAM [128] (vSLAM), which cause holes, missing data, and noisy annotations [40]. Such imperfections are commonly reduced by inferring dense complete geometry of objects with local matching of CAD models [38, 52, 164], or post-processing hole filling techniques [140]. In outdoor settings, point cloud registration techniques [111, 112] enable the co-location of multiple LiDAR measurements into a single reference coordinate system [6].

Interestingly, while frequently referred to as *dense*, the ground truth scenes are often noisy and non-continuous in real datasets, being in fact an *approximation* of the real scene. Firstly, regardless of the number of frames used, some portions of the scene remain occluded, see Fig. 4d, especially in dynamic environments. Secondly, sensors accuracy and density tend to steadily decrease with the distance, as in Fig. 4b. Thirdly, rigid registration can only cope with viewpoint changes, which leads to dynamic objects (e.g. moving cars) producing *traces*, which impact on the learning priors is still being discussed [121, 124, 169], see Fig. 4c. Finally, another limitation lies in the sensors, which only sense the geometrical surfaces and not the volumes, turning all solid objects into shells. To produce semantics labels, the common practice is to observe the aggregated 3D data from multiple virtual viewpoints to minimize the labeling ambiguity. This process is tedious and

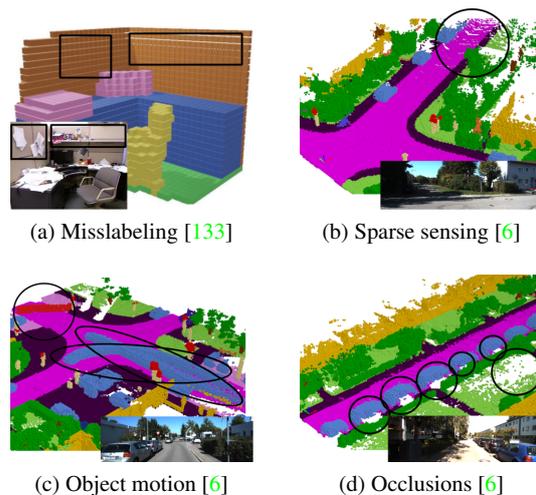


Fig. 4: **Inaccuracies of SSC ground truth.** Providing semantics and geometrics annotation in real-world datasets proves to be complex, and the resulting process is imperfect – which lead to noisy supervision signal. Ground truth mislabeling is a well known bias (a). Sparsity is common in LiDAR-based datasets (b). Object motion causes temporal traces (c), and occlusions are inevitable in dynamic scenes (d).

prone to errors<sup>1</sup>, visible in Fig. 4a. Ultimately, semantics distribution is highly imbalanced as shown in the two most used indoor/outdoor datasets Fig. 3.

*Indoor datasets.* From Tab. 1, NYUv2 [133] (aka NYU-Kinect) is the most popular indoor *real-world* dataset, composed of mainly office and house room scenery. Despite complete 3D ground truth scene volumes not being originally provided, they have been generated in [52] by using 3D models and 3D boxes or planes for producing complete synthetically augmented meshes of the scene, generating 1449 pairs of RGB-D images and 3D semantically annotated volumes. Extension of the mesh volumes to 3D grids has been done in [137]. However, mesh annotations are not always perfectly aligned with the original depth images. To solve this, many methods also report results by using depth maps rendered from the annotations directly as in [38]. This variant is commonly named as NYUCAD and provides simplified data pairs at the expense of geometric detail loss. Additional indoor real-world datasets as Matterport3D [13], SceneNN [64] and ScanNet [24] can be used for completing entire rooms from incomplete 3D meshes [25, 28]. The latter has also been synthetically augmented from 3D object models and referred to as CompleteScanNet [164], providing

<sup>1</sup> Authors of SemanticKITTI report that semantic labeling a hectar of 3D data takes approx. 4.5hr [6].

cleaner annotations. Additionally, smaller **SUN3D** [165] provides RGB-D images along with registered semantic point clouds. Note that datasets providing 3D meshes or point clouds easily be voxelized as detailed in [137]. Additionally, Stanford **2D-3D-S** [3] provides 360° RGB-D images, of interest for completing entire rooms [33]. Due to real datasets small sizes, low scene variability, and annotation ambiguities, *synthetic SUNCG* [137] (aka SUNCG-D) was proposed, being a large scale dataset with pairs of depth images and complete synthetic scene meshes. An extension containing RGB modality was presented in [86] and known as **SUNCG-RGBD**. Despite its popularity, the dataset is no longer available due to copyright infringement<sup>2</sup>, evidencing a lack of synthetic indoor datasets for SSC, that could be addressed using **SceneNet** [60].

*Outdoor datasets.* **SemanticKITTI** [6] is the most popular large-scale dataset and currently the sole providing single sparse and dense semantically annotated point cloud pairs from *real-world* scenes. It derives from the popular odometry benchmark of the **KITTI** dataset [45], which provides careful registration and *untwisted* point clouds considering vehicle’s ego-motion. Furthermore, voxelized dense scenes were later released as part of an evaluation benchmark with a hidden test set<sup>3</sup>. The dataset presents big challenges given the high scene variability and the high class imbalance naturally present in outdoor scenes (Fig. 3b). **SemanticPOSS** [105] also provides single sparse semantic point clouds and sensor poses in same format as the latter to ease its implementation. Furthermore, *synthetic SynthCity* additionally provides dense semantic point clouds and sensor poses. It has the advantage of excluding dynamic objects, which solves the effect of object motion (cf. Fig. 4c), but not occlusions (cf. Fig. 4d).

### 3.2 Scene representations

We now detail the common 3D representation for SSC output, shown in Fig. 5. While voxel grid or point cloud are the most used, other implicit/explicit representations are of interest for applications like rendering, fluid simulation, etc.

*Point Cloud* is a convenient and memory efficient representation, which expresses scene geometry in the 3D continuous world as a set of points lying on its surface. Different from others, point cloud omits definition of the free space and lacks geometrical structure. Few works have in fact applied point-based SSC. On objects, PCN [172] was the first to address point-based completion followed by [66, 146, 155, 156, 161–163, 166, 178]. However, point generation on scenes is more challenging and [121, 180] circumvent this need by querying semantic classes at given locations.

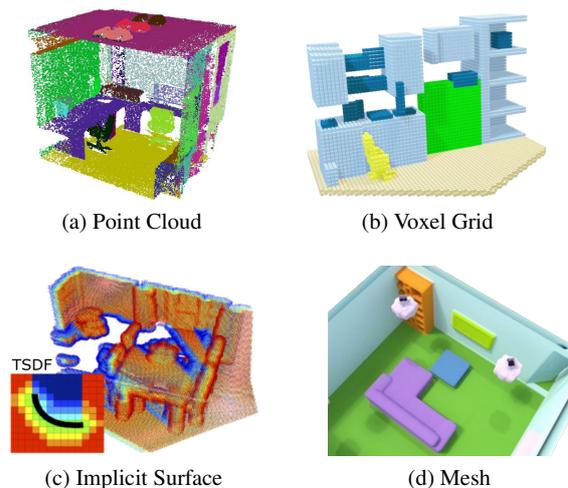


Fig. 5: **SSC representations.** Several 3D representations co-exist in the literature. Its choice has a major impact on the method to use, as well as the memory or the computation needs. (Source: [28, 137, 154, 160])

*Occupancy grid* encodes scene geometry as a 3D grid, in which cells describe semantic occupancy of the space. Opposed to point clouds, grids conveniently define neighborhood with adjacent cells, and thus enable easy application of 3D CNNs, which facilitates to extend deep learning architectures designed for 2D data into 3D [17, 20, 22, 25, 28, 32, 33, 43, 54, 78, 124, 137, 174, 177]. However, the representation suffers from constraining limitations and efficiency drawbacks since it represents both occupied and free regions of the scene, leading to high memory and computation needs. Voxels are also commonly used as a support for implicit surface definition which we now describe.

*Implicit surface* encodes geometry as a gradient field expressing the signed distance to the closest surface, known as the Signed Distance Function (SDF). While fields are continuous by nature, for ease of implementation they are commonly encoded in a discrete manner. The value of the gradient field is estimated at specific locations, typically at the voxel centers [25, 28], or at the point locations for point clouds [121]. Implicit surface may also be used as input [17, 20, 25, 28, 32, 33, 78, 137, 160, 174, 177] to reduce the sparsity of the input data, at the expense of greedy computation. For numerical reason, most works encode in fact a flipped version (cf. f-TSDF, sec. 4.1). Meshes, explained in detail below, can be obtained from the implicit surface, by using meshification algorithms such as marching cubes [90].

*Mesh* enables an explicit surface representation of the scene by a set of polygons. Implementing deep-learning algorithms directly on 3D meshes is challenging and most works obtain

<sup>2</sup> <https://futurism.com/tech-suing-facebook-princeton-data>

<sup>3</sup> <https://competitions.codalab.org/competitions/22037>

the mesh from intermediate implicit voxel-based TSDF representations [25, 28] by minimizing distance values within voxels and applying meshification algorithms [90]. Other alternatives contemplate applying view inpainting as in [59] or using parametric surface elements [50], which are more oriented to object/shape completion. Furthermore, recent learning-based algorithms such as Deep Marching Cubes [83] enable to obtain continuous meshes end-to-end from well sampled point clouds, but similarly have not been applied to fill large areas of missing information or scenes.

#### 4 Semantic Scene Completion

The seminal work of Song *et al.* [137] first addressed Semantic Scene Completion (SSC) with the observation that semantics and geometry are ‘tightly intertwined’. While there have been great progress lately, the best methods still perform little below 30% mIoU on the most challenging SemanticKITTI benchmark [6], advocating that there is a significant margin for improvement.

Inferring a dense 3D scene from 2D or sparse 3D inputs is in fact an ill-posed problem since the input data are not sufficient to resolve all ambiguities. As such, apart from [38, 46, 84, 179], all existing works rely on deep learning to learn semantics and geometric priors from large scale datasets reviewed in Sec. 3.1.

In the following, we provide a comprehensive survey of the SSC literature. Unlike some historical computer vision tasks, for SSC we found little consensus and a wide variety of choices exists. As such we also focus on the remaining avenues of research to foster future works.

The survey is organized in sections that follow a standard SSC pipeline, with each section analyzing the different line of choices. We start in Sec. 4.1 by reviewing the various input encoding strategies, broadly categorized into 2D/3D, and discuss their influence on the global problem framing. Following this, we study SSC deep architectures in Sec. 4.2. While initially, the task was addressed with vanilla 3D CNNs, other architectures have been leveraged such as 2D/3D CNNs, point-based networks, or various hybrid proposals. Sec. 4.3 presents important design choices, such as contextual aggregation which greatly influences any geometrical task like SSC, or lightweight designs to leverage the burden of 3D networks spanning from compact 3D representations to custom convolutions. We discuss the training strategies in Sec. 4.4, along with the losses and their benefits. Finally, a grouped evaluation of the metrics, methods performance and network efficiency is in Sec. 4.5.

We provide the reader with a digest overview of the field, chronologically listing methods in Tab. 2 – where columns follow the paper structure. Because SSC definition may overlap with some *reconstruction methods* that also address semantics, we draw the inclusion line in that SSC must also

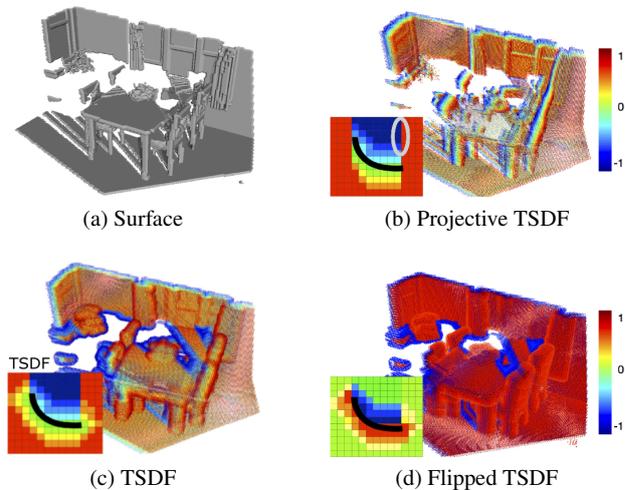


Fig. 6: **TSDF variants.** Projective TSDF (b) is fast to obtain but view-dependent. TSDF (c) [17, 19, 20, 28, 152, 160] solves the view dependency but gradient is stronger at farther areas from the surface, being inadequate for learning-based methods. In contrast, f-TSDF (d) [16, 21, 32, 33, 137, 174, 177] has strongest gradient near the surface. (Source: [137]).

complete semantics *and* geometry of *unseen* regions. We indistinctively report as SSC any method meeting these criteria. Looking at Tab. 2, it illustrates both the growing interest in this task and the current lack of consensus on input encoding, architectures, etc.

##### 4.1 Input Encoding

Given the 3D nature of the task, there is an evident benefit of using 3D inputs as it already withholds geometrical insights. As such, it is easier to leverage sparse 3D surface as input in the form of occupancy grid, distance fields, or point clouds. Another line of work uses RGB data in conjunction with depth data since they are spatially aligned and easily handled by 2D CNNs.

*3D grid-based.* In most works, 3D occupancy grid (aka Voxel Occupancy) is used [43, 124, 164, 169], encoding each cell as either *free* or *occupied*. Such representation is conveniently encoded as binary grids, easily compressed (cf. Sec. 4.3.4), but provides little input signal for the network. An alternative richer encoding is the use of TSDF (Fig. 6c), where the signed distance  $d$  to the closest surface is computed at given 3D locations (usually, voxel centers), as in [17, 22, 25, 28]. Instead of only providing input signal at the measurement locations like occupancy grids or point cloud, TSDF provides a richer supervision signal for the network. The sign for example provides guidance on which part of the scene is occluded in the input. The greedy 3D computation can be

	Input Encoding (Sec. 4.1)		Architecture (Sec. 4.2)	Design choices (Sec. 4.3)					Training (Sec. 4.4)			Evaluation (Sec. 4.5)		Open source				
	2D	3D		Contextual Awareness	Position Awareness	Fusion Strategies	Lightweight Design	Refinement	End-to-end	Coarse-to-fine	Multi-scale	Adversarial	Losses	NYU <sup>b</sup> [38, 133]	SUNCG <sup>c</sup> [86, 137]	SemanticKITTI [6]	Other [3, 13, 24, 86, 164]	Framework
	RGB	Depth/Range/HHA Other (seg., normals, etc.)	Network type															
2017			volume	DC								CE	✓	✓	✓		Caffe	✓
2018	✓		volume	DC		E			✓			CE	✓				-	
			volume				GrpConv		✓	✓		ℓ <sub>1</sub> CE	✓	✓	✓		TF	✓
		✓	view-volume	DC		E			✓			CE	✓	✓	✓		TF	✓
	✓	✓	volume	PDA		E	MSO		✓	✓	✓	CE	✓		✓		-	
			volume	DC					✓			CE	✓	✓			-	
			volume				GrpConv Sparse				✓	CE	✓	✓			PyTorch	✓
	✓	✓	view-volume	Mscale. CAT							✓	CE	✓	✓			TF	
	✓	✓	view-volume	ASPP		M/L			✓			CE	✓	✓			PyTorch	✓
2019	✓	✓	view-volume	LW-ASPP		M	DDR		✓			CE	✓	✓			PyTorch	✓
	✓		hybrid	DC		E			✓			CE	✓	✓	✓		-	
	✓		volume	DC	✓	M			✓			CE	✓	✓	✓		-	
			volume	DC							✓	BCE CE	✓	✓	✓		-	
	✓	✓	hybrid	DC		E			✓			CE	✓	✓	✓		-	
	✓	✓	hybrid	DC		E			✓			CE	✓	✓	✓		-	
			volume	DC							✓	BCE CE	✓	✓	✓		TF	✓
			volume	CCP DC			GrpConv		✓	✓		CE	✓	✓	✓		-	
	✓	✓	hybrid	DC	✓	M			✓	✓		BCE CE	✓	✓	✓		-	
2020	✓	✓	view-volume	LW-ASPP DC		M	DDR		✓			CE	✓	✓			-	
	✓		volume	DC	✓	E			✓	✓		CE	✓	✓			-	
	✓	✓	view-volume	LW-ASPP		L	RAB		✓			CE	✓	✓	✓		-	
	✓	✓	hybrid	FAM DC	✓	M			✓			PA	✓	✓	✓		PyTorch	✓
	✓	✓	hybrid	DC	✓	M	DDR		✓		✓	BCE CE CCY	✓	✓	✓		PyTorch	✓
	✓	✓	view-volume	FAM AIC		M	Anisotropic		✓			CE	✓	✓	✓		PyTorch	✓
			volume				Octree-based		✓			BCE CE	✓	✓	✓		-	
			volume				Octree-based		✓	✓		BCE CE	✓	✓	✓		-	
	✓	✓	hybrid			E			✓			CE	✓	✓	✓		-	
			volume	GA Module					✓			BCE CE	✓	✓	✓		-	
			view-volume	MSFA			2D		✓	✓		CE	✓	✓	✓		PyTorch	✓
			volume	DC					✓		✓	CE	✓	✓	✓		-	
	✓	✓	hybrid			L	Sparse		✓	✓		BCE CE PA	✓	✓	✓		-	
			volume				Sparse		✓			CE	✓	✓	✓		PyTorch	✓
			point-based						✓			BCE CE SCY	✓	✓	✓		-	
2021	✓	✓	hybrid			M	DDR		✓	✓		CE	✓	✓	✓		PyTorch	✓

<sup>a</sup> These original works were significantly extended in [54], [6], or [124].

<sup>b</sup> Includes both NYUv2 [133], NYUCAD [38]. <sup>c</sup> Includes both SUNCG [137], SUNCG-RGBD [86]. <sup>d</sup> Includes both ScanNet [24], CompleteScanNet [164].

Contextual Awareness - DC, Dilated Convolutions. (LW)-ASPP, (Lightweight) Atrous Spatial Pyramid Pooling. CCP, Cascaded Context Pyramid. FAM, Feature Aggregation Module. AIC, Anisotropic Convolutional Module. GA, Global Aggregation. MSFA, Multi-scale Feature Aggregation. PDA, Primal-Dual Algorithm.

Fusion Strategies - E, Early. M, Middle. L, Late.

Lightweight Design - GrpConv, Group Convolution. DDR, Dimensional Decomposition Residual Block. RAB, Residual Attention Block. MSO, MultiScale Optimization.

Losses - **Geometric**: BCE, Binary Cross Entropy. ℓ<sub>1</sub>, L1 norm. **Semantic**: CE, Cross Entropy. PA, Position Awareness. **Consistency**: CCY, Completion Consistency. SCY, Spatial Semantics Consistency.

Table 2: **Semantic Scene Completion (SSC) methods.**

avoided using projective TSDF (p-TSDF, cf. Fig. 6b) which only computes the distance along the sensing path [101], but with the major drawback of being view-dependent. Highlighted by Song *et al.* [137], another limitation of TSDF or p-TSDF lies in the strong gradients being in the free space area rather than close to the surface where the networks need guidance. This is noticeable in Fig. 6b, 6c since the red/blue gradients are afar from the surface. To move strong gradients closer to the surface, they introduced flipped TSDF (f-TSDF, Fig. 6d) such that  $f\text{-TSDF} = \text{sign}(\text{TSDF})(d_{\max} - d)$  with  $d_{\max}$  the occlusion boundary, showing improvement in both network guidance and performance [137]. However, the field yet lacks thorough study on the benefit of TSDF

encoding. While f-TSDF is still commonly used in recent SSC [16,32,33,78,174,176,177] or SC [30], other approaches stick with original TSDF [17,28,160,176]. Furthermore, the benefit of f-TSDF over TSDF is questioned in [43,176] with experiments showing it may slightly harm the performance or bring negligible improvement. Except for the lighter p-TSDF, all TSDF-variants require high computation times and hinder real time implementations.

*3D point cloud.* Despite the benefit of such representation, only three recent SSC works [121,122,180] use point cloud as input encoding. In [180], RGB is used in conjunction to augment point data with RGB features, whereas [121,122]

reproject point features in the top-view space. Despite the few SSC methods using points input, it is commonly used for object completion [65, 155, 166, 172].

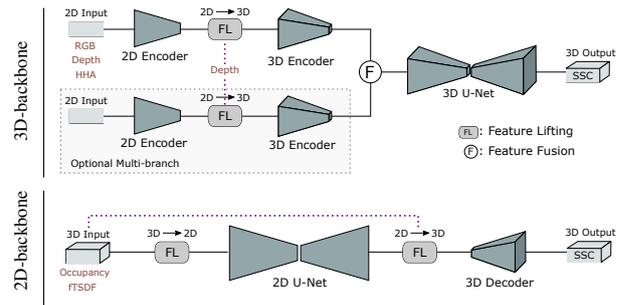
*2D representations.* Alternatively, depth maps or range images (i.e. 2D polar-encoded LiDAR data) are common 2D representations storing geometrical information, therefore suitable candidates for the SSC task. Indeed many works [6, 21, 54, 76–79, 84, 86, 159] used either representation alone or in conjunction with other modalities. Opposite to point cloud but similarly to grid representation, such encoding enables meaningful connexity of the data and cheap processing with 2D CNNs. Some works propose to transform depth into an HHA image [55], which keeps more effective information when compared to the single channel depth encoding [54, 86]. Additional non-geometrical modalities such as RGB or LiDAR refraction intensity provide auxiliary signals specifically useful to infer semantic labels [6, 17, 22, 43, 77, 79, 86, 89, 180]. In practice, some works have shown that good enough semantic labels can be inferred directly from depth/range images [6, 22, 169] to guide SSC.

Interestingly, the vast majority of the literature only accounts for surface information while ignoring any free space data provided by the sensors (see Fig. 2). While *free space* labels might be noisy, we believe it provides an additional signal for the network which was found beneficial in [121, 164]. Conversely, Roldao et al. [124] relate that encoding *unknown* information brought little improvement.

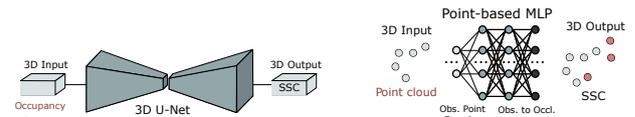
## 4.2 Architecture choices

Directly linked to the choice of input encoding, we broadly categorize architectural choices into 4 groups. In detail: *Volume networks* leveraging 3D CNNs to convolve volumetric grid representations, *Point-based networks* which compute features on points locations, *View-Volume networks* that learn the 2D-3D mapping of data with 2D and 3D CNNs, and *Hybrid networks* that use various networks to combine modalities of different dimension. All architectures output  $N \times C$  data ( $N$  the spatial dimensions and  $C$  the number of semantic classes) where the last dimension is the probability of either semantic class at the given location. In most works, the final prediction is the softmax output with the class probabilities. We refer to Fig. 7 for a visual illustration of either architecture type.

*Volume networks.* As they are convenient for processing grid data 3D CNNs (Fig. 7b) are the most popular for SSC [19, 20, 22, 28, 32, 33, 51, 137, 152, 157, 160, 164, 169, 174, 176, 177]. Since completion heavily requires contextual information it is common practice to use a U-Net architecture [125] (see Fig. 7), i.e. encoder-decoder with skip connections. The benefit of the latter is not only to provide contextual information

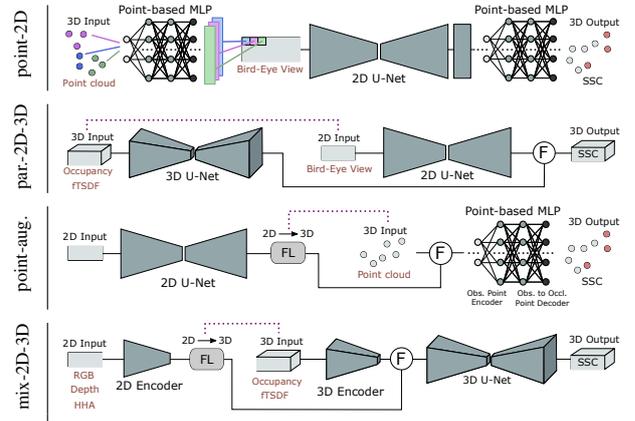


(a) View-Volume Nets., 3D-backbone: [54, 76, 77, 79, 86, 89], 2D-backbone: [124]



(b) Volume Nets. [19, 20, 22, 28, 32, 33, 51, 137, 152, 157, 160, 164, 169, 174, 176, 177]

(c) Point-based Nets. [180]



(d) Hybrid Nets., point-2D: [121, 122], parallel-2D-3D: [21], point-augmented: [180], mix-2D-3D: [6, 11, 17, 43, 78]

Fig. 7: **Architectures for SSC.** For compactness, we do not display all connections but rather focus on the global architectures and information exchanges between the different networks.  $\oplus$  stands for any type of fusion.

but also to enable meaningful coarser scene representation, used in SSC for outputting multi-scale predictions [124, 174] or for enabling coarse-to-fine refinement [28].

There are two important limitations of 3D CNNs: their cubically growing memory need, and the dilation of the sparse input manifold due to convolutions. To circumvent both, one can use sparse 3D networks like SparseConvNet [48, 174] or Minkowski Network [23] which conveniently operates only on input geometry, thus allowing high grid resolution where each cell contains typically a single point. While they were found highly beneficial for most 3D tasks, they show limited interest for SSC since the output is expected to be denser than the network input. Considering the output to be *sparse*,

Dai *et al.* [25] used a sparse encoder and a custom sparse generative decoder to restrict the manifold dilation, applied for SC rather than SSC. This is beneficial but cannot cope with large chunks of missing data. An alternative is [169] that first performs pointwise semantic labeling using a sparse network. To enable a *dense* SSC output in [174], authors merge the output of multiple *shared* SparseConvNets applied on sub-sampled non-overlapping sparse grids. Both [25, 174] are a clever use of sparse convolutions but somehow limit the memory and computational benefit of the latter. In [22], variational optimization is used to regularize the model and avoid the need for greedy high-capacity 3D CNN.

*View-volume networks.* To take advantage of 2D CNNs efficiency, a common strategy is to use them in conjunction with 3D CNNs as in [54, 76, 77, 79, 86, 89, 124], see Fig. 7a. Two different schemes have been identified from the literature. The most common, as in [54, 76, 77, 79, 86, 89], employs a 2D CNN encoder to extract 2-dimensional features from 2D texture/geometry inputs (RGB, depth, etc.), which are then lifted to 3D and processed by 3D CNN (Fig. 7a, 3D-backbone). Optional modalities may be added with additional branches and mid-fusion scheme [76, 77, 86, 89]. In brief, the use of sequential 2D-3D CNNs conveniently benefits of different neighboring definitions, since 2D neighboring pixels might be far away in 3D and vice versa, thus providing rich feature representation, at the cost of increased processing. To address this limitation, the second scheme (Fig. 7a, 2D-backbone) projects 3D input data into 2D, then processed with normal 2D CNN [124] significantly lighter than its 3D counterpart. The resulting 2D features are then lifted back to 3D and decoded with 3D convolutions, retrieving the third dimension before the final prediction. This latter scheme is irrefutably lighter in computation and memory footprint but better suited for outdoor scenes (cf. Sec. 4.5), as the data exhibits main variance along two axes (i.e. longitudinal and lateral).

*Point-based networks.* To ultimately prevent discretization of the input data, a few recent works [121, 122, 180] employ point-based networks, see Fig. 7c. In 2018, [172] first proposed to apply permutation-invariant architecture [116] to object completion with promising results. However, its use for generative SSC is hindered by the limited points generation capacity, the need for fixed size output, and the use of global features extraction. To date, only SPC-Net [180] relies solely on a point-based network –  $\mathcal{X}$ Conv [80] – to predict the semantics of observed and unobserved points. The fixed size limitation is circumvented by assuming a regular distribution of unobserved points, addressing the problem as a point segmentation task. Overall, we believe point-based SSC has yet attracted too few works and is a viable avenue of research.

*Hybrid networks.* Several other works combine architectures already mentioned, which we refer to as *hybrid networks* [6, 17, 21, 43, 78, 121, 122, 180], see Fig. 7d. A common 2D-3D scheme combines 2D and 3D features (e.g. RGB and f-TSDF) through expert modality networks in a common latent space decoded via a 3D CNN [17, 43] (Fig. 7d, mix-2D-3D). This enables additional benefit with the combined use of texture and geometrical features. Similarly, IPF-SPCNet [180] performs semantic segmentation from RGB image on an initial 2D CNN and lifts image labels to augment ad-hoc 3D points (Fig. 7d, point-augmented). In [121, 122], PointNet [116] encodes geometrical features from sub-set of points later convolved in a bird eye view (BEV) projection with 2D CNN in a hybrid architecture manner (Fig. 7d, point-2D). Another strategy employs parallel 2D-3D branches to process the same data with different neighborhoods definition contained in the 2D projected image and 3D grid as in [78]. Recently, S3CNet [21] combines 3D voxelized f-TSDF and normal features with a 2D BEV [18], passing both modalities through sparse encoder-decoder networks for late fusion (Fig. 7d, parallel-2D-3D), achieving impressive results in outdoor scenes. A similar architecture is proposed by [1] to perform what they refer to as *scene extrapolation* [138], by performing extrapolation of a half-known scene into a complete one.

#### 4.3 Design choices

The significant sparsity difference between the input data and the expected *dense* output imposes special design choices to be made, specifically to ensure efficient flow of information. In the following, we elaborate on the most important ones such as contextual awareness (Sec. 4.3.1), position awareness (Sec. 4.3.2), and fusion strategies (Sec. 4.3.3). Finally, we detail lightweight designs (Sec. 4.3.4) to efficiently process 3D large extent of sparse data, and the common refinement processes (Sec. 4.3.5).

##### 4.3.1 Contextual awareness

To correctly complete the missing information in the scene, it is necessary to gather contextual information from multiple scales, which enables to disambiguate between similar objects present in the scene. This makes it possible to capture both local geometric details and high-level contextual information [137]. A common strategy to accomplish this is to add skip connections between different convolutional layers to aggregate features from different receptive fields [17, 20, 25, 32, 33, 43, 54, 86, 124, 137, 174, 177]. Additionally, serial context aggregation with multi-scale feature fusion can be used as proposed in [78], Fig. 8a. In [137], the use of dilated convolutions (aka ‘atrous’) [171] are proposed to increase receptive fields and gather context information at

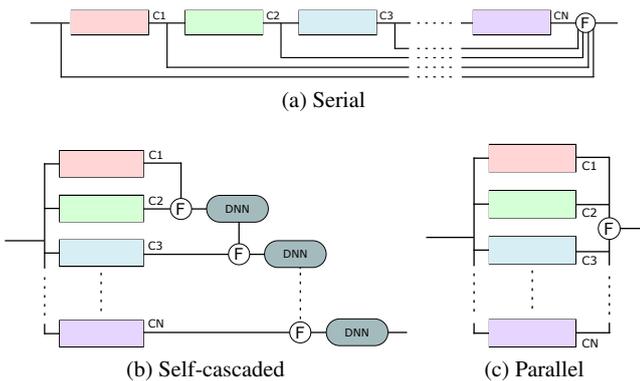


Fig. 8: **Multi-scale contextual aggregation.** While context is indubitably important for SSC, different strategies are used to aggregate features from various spatial/scale contexts. Color blocks stand for convolutions with different dilation rates.  $\textcircled{F}$  stands for any type of fusion.

low computational cost. The strategy became popular among most works [17, 32, 33, 43, 54, 78, 79, 86, 124, 137, 160, 164, 177]. Such convolutions are only suitable for dense networks (as opposed to sparse networks), and even then should only be applied in deeper layers of the network after dilation of the input manifold. In [86], a feature aggregation module is introduced by using Atrous Spatial Pyramid Pooling blocks (ASPP) [15], which exploits multi-scale features by employing multiple parallel filters with different dilation rates, Fig. 8c. A lightweight ASPP is presented in [77]. Dilated convolutions in the ASPP module can be replaced by Residual dilated blocks [61] to increase spatial context and improve gradient flow. A Guided Residual Block (GRB) to amplify fused features and a Global Aggregation module to aggregate global context through 3D global pooling are proposed in [19]. An additional feature aggregation strategy is presented in [177], where multi-context aggregation is achieved by a cascade pyramid architecture, Fig. 8b. In [22] multi-scale features are aggregated together following a Primal-Dual optimization algorithm [110], which ensures semantically stable predictions and further acts as a regularizer for the learning.

#### 4.3.2 Position awareness

Geometric information contained in voxels at different positions has high variability, i.e. *Local Geometric Anisotropy*. In particular, voxels inside an object are homogeneous and likely to belong to the same semantic category as their neighbors. Conversely, voxels at the surface, edges, and vertices of the scene provide richer geometric information due to the higher variance of their surroundings. To deal with this, PALNet [78] proposes a Position Aware loss (cf. Sec. 4.4.1), which consists of a cross entropy loss with individual voxel weights assigned according to their geometric anisotropy, providing slightly faster convergence and improving results.

Likewise, AM<sup>2</sup>FNet [16] supervises contour information by an additional cross entropy loss as a supplementary cue for segmentation.

In the same line of work, EdgeNet [32] calculates Canny edges [12] then fused with an f-TSDF obtained from the depth image. Hence, it increases the gradient along the geometrical edges of the scene. Additionally, detection of RGB edges enables the identification of objects lacking geometrical saliency. The same network is used in [33] to predict complete scenes from panoramic RGB-D images.

Similarly, [17] introduces an explicit and compact geometric embedding from depth information by predicting a 3D sketch containing scene edges from an input TSDF. They show that this feature embedding is resolution-insensitive, which brings high benefit, even from partial low-resolution observations.

#### 4.3.3 Fusion strategies

SSC requires outputting both geometry and semantics. Though highly coupled – geometry helping semantics and vice-versa –, there is a natural benefit to use inputs of different natures for example to provide additional texture or geometry insights. We found that about two-thirds of the literature use multi-modal inputs though it appears less trendy in most recent works (see Tab. 2 col ‘Input’). For the vast majority of multi-input works, RGB is used alongside various geometrical input [6, 16, 17, 32, 33, 43, 51, 76, 77, 79, 86, 89, 180] as it is a natural candidate for semantics. Even without color, the fusing of 2D and 3D modalities is often employed because it enables richer feature estimation. This is because 2D and 3D neighborhoods differ, since 2D data results of planar projection along the optical axis of the sensor. Subsequently, a common strategy consists of fusing geometrical features processed with different 2D / 3D encoding to obtain richer local scene descriptors. In [6] depth and occupancy are fused while [78] uses depth along with TSDF-like data. As mentioned earlier (cf. Sec. 4.1), TSDF provides a gradient field easing the network convergence. Finally, application-oriented fusion is also found such as fusing bird eye view along with geometrical inputs as in [21] – which is better suited for outdoor SSC.

We group the type of fusion in three categories, shown in Fig. 9. Fusion applied at the input level (*Early fusion*), at the mid-level features (*Middle fusion*), or at the late/output level (*Late fusion*). They are respectively referred to as *E*, *M*, and *L* in column ‘Fusion strategies’ Tab. 2.

*Early fusion.* The most trivial approach is to concatenate input modalities [6, 22, 33, 43, 51, 54, 180] before any further processing, see Fig. 9a. There are two strategies here: when spatially aligned (e.g. RGB/Depth) inputs can be concatenated channel-wise; alternatively, inputs can be projected

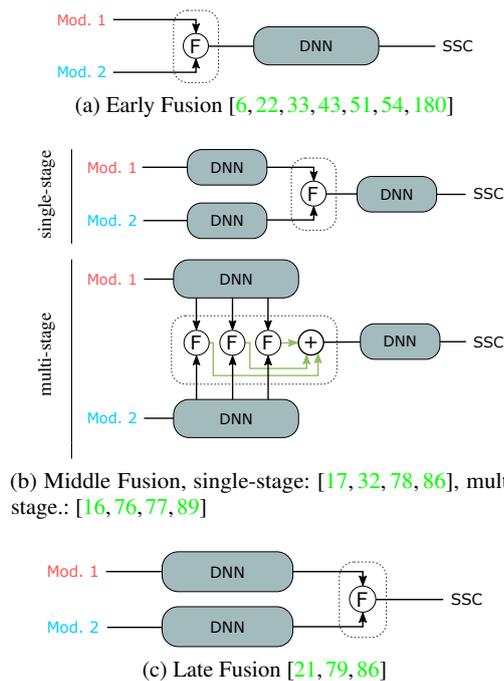


Fig. 9: **Fusion Strategies.** To accommodate for multiple input modalities (Mod. 1, Mod. 2), several fusion strategies are found in the literature. Here,  $\textcircled{F}$  stands for fusion and might be any type of fusion like concat  $\textcircled{C}$ , sum  $\textcircled{+}$ , multiply  $\textcircled{\times}$ , convolutions, etc.

to a shared 3D common space (*aka* features lifting). For spatially aligned modalities, it is common to use pairs of normals/depth [54] or RGB/semantics [6] and to process them with 2D CNNs. The second strategy lifts any 2D inputs to 3D – assuming depth information and accurate inter-sensors calibration – and processes it with 3D networks. This has been done with RGB/depth [51], depth/semantics [22, 43], points/semantics [180]. Except when using points, this second strategy leads to a sparse tensor since not all 3D cells have features. Noteworthy, [6, 22, 43, 180] use semantics, which is first estimated either from RGB or depth-like data. A 2D or 3D network processes the concatenated tensor, and while it logically outperforms single-modality input [32, 43, 54] there seems to be little benefit to apply early fusion.

*Middle fusion.* To exploit all modalities, middle fusion uses expert networks that learn modality-centric features. A straightforward fusion strategy is employed in [17, 32, 78, 86] where the features are simply concatenated and processed with a U-Net style architecture (cf. Fig. 9b, single-stage), which improves over early fusion but still limits the exchange of information between modalities. The information flow is improved in [16, 76, 77, 89] by fusing modality-centric features in a multi-stage manner (cf. Fig. 9b, multi-stage); meaning that low-level features from different modalities are fused

together and aggregated with fused mid/high level features gathered similarly. While ultimately the number of fusion stages shall be a function of the input/output size, 3 stages are often used [16, 76, 77], though [89] claims 4 stages boost performances with similar input/output. The fused mechanism can be a simple summation [77] or concatenation [76], but [16, 89] benefit from smarter selective fusion schemes using respectively RefineNet [85] and Gated Recurrent Fusion. Overall, the literature consensus is that middle fusion is highly efficient for SSC. The ablation studies of Liu *et al.* [89] reports that any selective fusion schemes bring at least a 20% performance boost over simple sum/concat/max schemes.

*Late fusion.* Few works use late fusion for SSC [21, 79, 86], see Fig. 9c. The straightforward strategy in [79] is to apply fusion – namely, element-wise multiplication – of two SSC branches (a 3D guidance branch, and a semantic completion branch), followed by a softmax. The benefit still appears little (5 to 10%) given the extra computational effort. Similarly, color and geometry branches are concatenated and shallowly convolved before softmax in [86], also providing a small benefit (less than 3%). A unique strategy was proposed in the recent S3CNet [21] where the output of parallel 2D top-view and 3D SSC are fused together in a semantic-wise manner. While it was only evaluated on outdoor scenes – which setup naturally minimizes vertically overlapping semantic labels – ablation reports an overall 20% boost.

Summarizing the different strategies, *Middle fusion* appears to be the best general SSC practice, though *Late fusion* was found beneficial in some specific settings. On fused modalities, RGB/geometry fusion boosts performance but at the cost of an additional sensor need, but even using fusion of geometrical input with different encodings is highly beneficial. An interesting insight from [17, 32] advocates that RGB *or* geometry can be fused with edge features as they provide additional boundaries guidance for the SSC network.

#### 4.3.4 Lightweight designs

A few techniques for lightweight designs are often applied for SSC with the aim of addressing two separate problems: how to improve the memory or computation efficiency, and how to design meaningful convolutions to improve the information flow. We detail either problem and its solutions below.

*Memory and computation efficiency.* Voxel grids are often used as input/output encoding of the 3D data since current datasets provide ground truth in such a format. However, only a tiny portion of the voxels are occupied which makes the naive *dense* grid inefficient in memory and computation. Memory wise, a few works use compact hierarchical

3D representation inspired from pre-deep learning, like Kd-Tree [7] and Octree [94]. Octree-based deep networks are often used for learning object reconstruction [119,120,151,153] though little applied on real semantic scene completion problem [22,152,157]. Meanwhile, deep Kd-Networks [71] proposal seems less appropriate and has not yet been applied to SSC. Computation-wise, [22] proposed a custom network architecture with adjustable multi-scale branches in which inference and backpropagation can be run in parallel, subsequently enabling faster training and good performance with low-capacity dense 3D CNNs. Alternatively, few SSC or SC works [25,174] use sparse networks like SparseConvNet [48] or Minkowski [23] which operate only in active locations through a hash table. While sparse convolutions are very memory/computation efficient, they are less suitable for completion, since they deliberately avoid filling empty voxels to prevent dilation of the input domain. To remedy this for the SSC task, dense convolutions are still applied in the decoder, which subsequently reduces sparse networks efficiency. Overall, while Kd-/Octree- networks are highly memory efficient, the complexity of their implementation has restricted a wider application. Contrastingly, sparse networks [23,48] are more used [21,25,169,174].

*Efficient convolutions.* A key observation is the spatial redundancy of data since neighboring voxels contain similar information. To exploit such redundancy, [174] proposes Spatial Group Convolutions (SGC) to divide input volume into different sparse tensors along the spatial dimensions which are then convolved with shared sparse networks. A similar strategy is followed by [28], dividing the volumetric space into a set of eight interleaved voxel groups and performing an auto-regressive prediction [117]. Dilated convolutions are also widely used for semantic completion methods [17,19,20,32,33,43,54,77–79,86,89,124,137,160,177], since they increase receptive fields at small cost, providing large context, which is crucial for scene understanding as discussed in Sec. 4.3.1. Dilated convolutions with *separated kernels* are proposed in [177] by separating the input tensor into subvolumes. This enables to reduce the number of parameters and consider depth profiles in which depth values are continuous only in neighboring regions. DDRNet [77] also introduces Dimensional Decomposition Residual (DDR) block, decomposing 3D convolutions into three consecutive layers along each dimension, subsequently reducing the network parameters. In [76], this concept is extended with the use of anisotropic convolutions, where the kernel size of each 1D convolution is adaptively learned during training to model the dimensional anisotropy.

#### 4.3.5 Refinement

Refinement is commonly used in many vision tasks, but little applied in SSC. VD-CRF [176] extends SSCNet [137]

by applying Conditional Random Field (CRF) to refine output consistency, achieving little over 4% gain. Additionally, S3CNet [21] presents a 3D spatial propagation network [87] to refine segmentation results after fusion of 2D semantically completed bird eye view image and 3D grid. Additional partial refinement is applied in [33,164] to fuse SSC predictions from different viewpoints, by either softmax applied to overlapping partitions [33,164] or an occupancy-based fusion policy [164]. Though few works address the refinement problem, some notable performance boosts are found in the literature, thus being an encouraging topic to explore.

## 4.4 Training

We now detail the SSC training process, starting with the SSC losses (Sec. 4.4.1), and subsequently the implemented training strategies (Sec. 4.4.2).

### 4.4.1 Losses

We classify the SSC losses found in the literature in 3 broad categories: *geometric losses* which optimize geometrical accuracy, *semantics losses* which optimize semantics prediction, and *consistency losses* which guide the overall completion consistency. Note that other non-SSC losses are often added and that the type of SSC losses are commonly mixed – specifically, geometric+semantics [19,21,28,152] or all three types [17,20,121,122,160]. We refer to Tab. 2 for a quick overview of the losses used by each method. In this section, we also refer to  $\hat{y}$  as SSC prediction and  $y$  as ground truth, though for clarity we add a subscript notation to distinguish between SSC encoding. For example,  $y^{\text{mesh}}$  corresponds to the ground truth mesh.

*Geometric losses.* These losses penalize the geometrical distance of the output  $\hat{y}$  to ground truth  $y$ , in a self-supervised manner.

On occupancy grids outputs ( $\hat{y}^{\text{occ}}$ ), **Binary Cross-Entropy** loss (BCE) is most often used [19,21,121,122,152] to discriminate *free* voxels from *occupied*. Assuming a binary class mapping where all non-free semantic classes map to ‘occupy’. It writes:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=0}^N \hat{y}_i^{\text{occ}} \log(y_i^{\text{occ}}) - (1 - \hat{y}_i^{\text{occ}}) \log(1 - y_i^{\text{occ}}), \quad (1)$$

with  $N$  the number of voxels. The drawback of such loss is that it provides little guidance to the network due to its sparsity. Smoother guidance can be provided by outputting an implicit surface ( $\hat{y}^{\text{SDF}}$ ) through minimization of the predicted signed distance values in  $\hat{y}^{\text{SDF}}$  and corresponding SDF-encoded mesh ( $y^{\text{SDF}}$ ) – using  $\ell_1$  or  $\ell_2$  norms.

On points outputs ( $\hat{y}^{\text{pts}}$ ), if SSC is approached as a generative task, the above losses could also be used to penalize distance to a ground truth mesh, though it might be more suitable to apply points-to-points distances, thus assuming a ground truth point cloud ( $y^{\text{pts}}$ ). To that end, permutation invariant metrics as the **Chamfer Distance** (CD) [36] or the **Earth Mover’s Distance** (EMD) [36, 73] have been employed for object completion tasks [36, 172] but have not been explored yet for SSC because of their computational greediness [36]. We highlight that such losses could provide an additional geometric supervision signal when used in conjunction with semantic losses described below.

*Semantic losses.* Such losses are suitable for occupancy grids or points and can accommodate for either  $C$  classes (considering only semantics classes of occupied voxels or points) or  $C + 1$  classes (considering all voxels/points and ‘free space’ being the additional class). Note that only the second case ( $C + 1$  classes) enforce reconstruction, so the first one ( $C$  classes) would require additional *geometric losses*. **Cross-Entropy** loss (CE) is the preferred loss for SSC [20, 22, 32, 43, 54, 76, 77, 79, 86, 124, 137, 174, 177], it models classes as independent thus considering the latter to be equidistant in the semantic space. Formally, supposing  $(y, \hat{y})$  it writes:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{C} \sum_{i=0}^N \sum_{c=0}^N w_c \hat{y}_{i,c} \log \left( \frac{e^{y_{i,c}}}{\sum_{c'}^C e^{y_{i,c'}}} \right), \quad (2)$$

assuming here that  $y$  is the one-hot-encoding of the classes (i.e.  $y_{i,c} = 1$  if  $y_i$  label is  $c$  and otherwise  $y_{i,c} = 0$ ). In practice,  $(y, \hat{y})$  can be either occupancy grids ( $y^{\text{occ}}, \hat{y}^{\text{occ}}$ ) or points ( $y^{\text{pts}}, \hat{y}^{\text{pts}}$ ). A rare practice from [160] is to address classification with BCE (Eq. 1) through the sum of  $C$  binary classification problems between each semantic class and the free class. However, such a practice is unusual and arguably beneficial.

Recently, PALNet [78] proposed the **Position Aware** loss (PA), a weighted cross-entropy accounting for the local semantics entropy to encourage sharper semantics/geometric gradients in the completion (cf. Sec 4.3.2). The loss writes:

$$\mathcal{L}_{\text{PA}} = -\frac{1}{N} \sum_{i=0}^N \sum_{c=0}^C (\lambda + \alpha W_{\text{LGA}_i}) \hat{y}_{i,c}^{\text{occ}} \log \left( \frac{e^{y_{i,c}^{\text{occ}}}}{\sum_{c'}^C e^{y_{i,c'}^{\text{occ}}}} \right), \quad (3)$$

with  $\lambda$  and  $\alpha$  being simple base and weight terms, and  $W_{\text{LGA}_i}$  being the *Local Geometric Anisotropy* of  $i$  that scales accordingly to the semantic entropy in its direct vicinity (i.e.  $W_{\text{LGA}}$  lowers in locally smooth semantics areas). We refer to [78] for an in-depth explanation. From the latter,  $\mathcal{L}_{\text{PA}}$  leads to a small performance gain of 1–3%. Noteworthy, this loss could easily accommodate point clouds as well.

Note that *geometric* or *semantics* losses can only be computed on *known* ground truth location, due to the ground truth sparsity. Additionally, because SSC is a highly imbalanced problem (cf. Fig. 3), class-balancing strategy is often used.

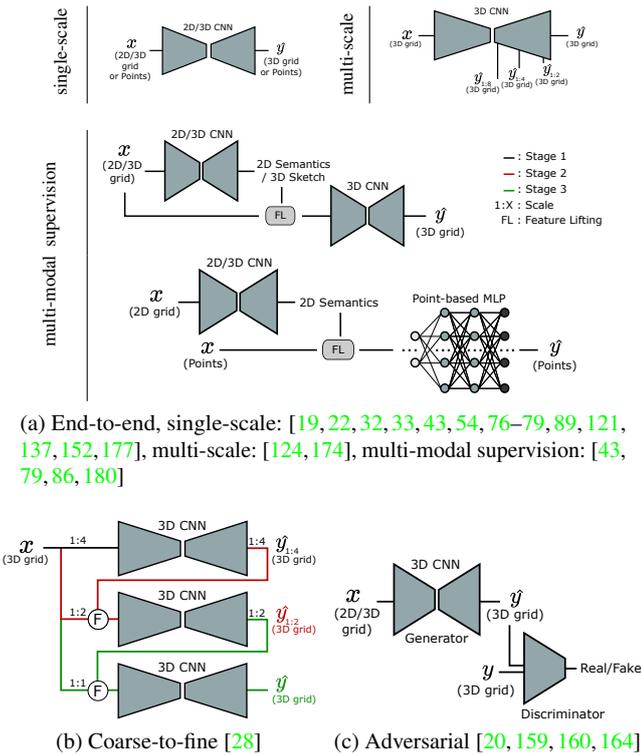
*Consistency losses.* Different from most *semantics* losses, these losses [17, 121, 122] provide a self-supervised semantic signal. In [17] the **completion consistency** (CCY) of predictions from multiple partitioned sparse inputs is enforced via a Kullback-Leibler divergence. Differently, [121, 122] enforces **spatial semantics consistency** (SCY) by minimizing the Jensen-Shannon divergence of semantic inference between a given spatial point and some given *support points*. This self-supervision signal is available at any position within the scene. However, the strategy for *support points* is highly application dependent and while suitable for outdoor scenes which have repetitive semantic patterns, we conjecture it might not scale as efficiently to cluttered indoor scenes.

Overall, few self-supervised or even unsupervised strategies exist and we believe that such type of new losses [173] should be encouraged.

#### 4.4.2 Training strategies

The vast majority of SSC works are trained end-to-end for *single-scale* reconstruction, although others prefer *multi-scale* supervision to output SSC at different resolutions. Few works also employ *multi-modal supervision* commonly relying on auxiliary 2D semantic segmentation. It is also possible to train  $n$ -stage networks with *coarse-to-fine* strategies, or even train with *adversarial* learning to enforce realism. Strategies are illustrated in Fig. 10 (with link color indicating the stage) and reviewed below.

*End-to-end.* Most architectures (Fig. 10a, top left) are trained end-to-end and output a *single scale* SSC [19, 22, 32, 33, 43, 54, 76–78, 89, 121, 137, 152, 177] – often similar to the input size. Training that way is straightforward and often offers minimal memory footprint. Noteworthy, [25] – which does geometric completion only – gradually increases sparsity during training to ease the completion of large missing chunks. To guide the training, multi-scale SSC outputs can also be supervised, typically from the early layers of a U-Net decoder. A simple, yet efficient *multi-scale* strategy [124, 174] is to minimize the sum of SSC losses at different resolutions (Fig. 10a, top right), thus also enforcing coarse SSC representations in the network. In [174], two different scales are predicted, versus four in [124] providing down to 1:8 (1 over 8) downscaled SSC. In the latter, authors also report that the decoder can be ablated to provide very fast inference at coarsest resolution (370FPS at 1:8 scale). When available, some



(a) End-to-end, single-scale: [19, 22, 32, 33, 43, 54, 76–79, 89, 121, 137, 152, 177], multi-scale: [124, 174], multi-modal supervision: [43, 79, 86, 180]

Fig. 10: **Training strategies.** Most SSC architectures are trained end-to-end (a) outputting single or multi-scale SSC. Additionally, multi-modal supervision training commonly lift semantic features calculated on sparse input to a second stage network. Coarse-to-fine (b), similarly to multi-scale relies on multiple size predictions, but trains in a multi-stage coarse to fine manner. Finally, Adversarial training (c) discriminates between ground truth and predicted scenes.  $\oplus$  stands for fusion of any type.

works leverage multi-modal supervision relying on intermediate auxiliary tasks, typically 2D or 3D semantics, later used along original input data to infer the final SSC (Fig. 10a, bottom), as in [43, 79, 86, 180]. The latter could also be trained in a two-stage manner. In general, end-to-end training is conducted from scratch, though some [32, 54, 86, 137, 177] report pretraining on the synthetic SUNCG dataset.

*Coarse-to-fine.* ScanComplete [28] also follows a multi-scale strategy somehow close to [124, 174], though training in a coarse-to-fine manner (Fig. 10b). In detail, three sequential training are achieved at increasingly higher resolutions, with each stage network taking as input the ad-hoc sparse input and the previous stage SSC prediction (for stage > 1). Interestingly, no one explored a continuous curriculum learning setting, which could yield stabler training and performance improvement. Still, [22] (intentionally omitted Fig. 10b) applies a unique coarse-to-fine proposal in a fully end-to-end manner, via parallel backpropagations in all scales. Of simi-

lar spirit, [25] proposes an iteration-based progressive refinement during training for scene completion, but insights of such strategy are not deeply discussed.

*Adversarial.* Even SSC ground truth has large missing chunks of data, leading to ambiguous supervision. To address this, [20, 159, 160, 164] use adversarial training (Fig. 10c), since the discriminator provides an additional supervision signal. This is straightforwardly implemented in [20, 164], where the discriminator classifies ground truth from generated SSC (aka real/fake). In [159, 160] of same authors, 2 discriminators are used in a somehow similar fashion to discriminate both the SSC output and the latent depth or semantics features to enforce deep shared representation. Additionally, [17] employs a Conditional Variational Autoencoder (CVAE) to generate completed border sketches to be fed to the main SSC branch. Despite few works on the matter, adversarial appears a logical choice to improve SSC consistency and provide additional self-supervision. Both [159] and [164] report a 10%-15% boost on several datasets.

Finally, on implementation – where mentioned – only [28, 124, 159, 160, 180] train with Adam optimizer, [20] with a mix of Adam/SGD, and all others use only SGD with momentum 0.9 and  $10^{-4}$  weight decay, except for [19, 20, 32, 79, 151, 152, 177] using  $5 \times 10^{-4}$ . The training most often uses standard learning rate scheduler [19, 43, 76–78, 89, 151, 152, 177, 180] though sophisticated scheduling [32] or fixed learning rate [79] are also used. Because of 3D greediness, the common practice is to train with small batch size of 1 [79], 2 [77], 3 [32], 4 [20, 22, 32, 54, 76, 78, 89, 124, 177, 180], 8 [151, 152, 159, 160] or 16 [19] to fit in standard 12GB GPUs.

## 4.5 Evaluation

We now provide an in-depth evaluation of the field, reviewing first the common metrics (Sec. 4.5.1), the qualitative and quantitative performance of the literature (Sec. 4.5.2), and the networks’ efficiency (Sec. 4.5.3).

### 4.5.1 Metrics

*Joint Semantics-Geometry.* The preferred metric for SSC is the mean **Jaccard Index** or **mean Intersection over Union** (mIoU) [35], which considers IoU of all semantic classes for prediction, without considering free space. It writes

$$\text{mIoU} = \frac{1}{C} \sum_{c=1}^C \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c}, \quad (4)$$

where  $\text{TP}_c$ ,  $\text{FP}_c$  and  $\text{FN}_c$  are the true positives, false positives and false negatives predictions for class  $c$ , respectively. Since

ground truth is commonly semi-dense for real-world datasets, evaluation is performed in known space only.

*Geometry only.* Because mIoU considers semantic classes, the pure geometrical reconstruction quality is not encompassed. Therefore **Intersection over Union (IoU)**, along with **Precision** and **Recall** are commonly used on the binary free/occupy scene representation, obtained by mapping all semantic classes to *occupy*.

Alternatively, any distance metrics from Sec. 4.4.1 (i.e.  $\ell_1$ ,  $\ell_2$ , **EMD** or **CD**) may be used as in [25, 28] though less used in real datasets, due to their lower precision when sparsity increases.

On common practice, we highlight that evaluation on real indoor or outdoor datasets is usually performed differently. This results of the common sensors setup, respectively RGB-D (indoor) and LiDAR (outdoor), providing significantly different density information. Referring to Fig. 2, in real indoor [137] the geometrical IoU is evaluated on input occluded regions while the mIoU is evaluated on input occluded (blue) and observed (red) surfaces. In real outdoor [6] the IoU and mIoU are commonly evaluated on the entire known space, regardless of whether regions were observed or occluded in the input. Obviously, synthetic datasets can cope with either practice. In the following, we describe the common practices and report semantics metrics (mIoU) along with geometrical ones (Precision, Recall, IoU).

#### 4.5.2 Performance

We report the available mIoU and IoU performance on the most popular SSC datasets in Tab. 3, which are all obtained from voxelized ground truth. For non-voxel methods the output is voxelized beforehand. Additionally, detailed classwise performance of top five methods for SemanticKITTI, NYUv2 and SUNCG are presented in Tabs. 4, 5 and 6, respectively. From the performance Tab. 3, the mIoU of the best methods plateaus around 75 – 85% on synthetic indoor dataset, 52% on real indoor, and 30% on real outdoor. Importantly, note that *most* indoor datasets performance are evaluated at 1:4 of the original ground truth resolution – that is  $60 \times 36 \times 60$  – for historical reasons<sup>4</sup>. Additionally, some methods refine parameters on NYUv2 and NYUCAD after SUNCG pre-training and are shown separately in Tab. 7. This makes indoor / outdoor performance comparison tricky. It is interesting to note that IoU – geometrical completion (i.e. ignoring semantics) – is way higher than best mIoU. In detail, best IoU are 78% on

real indoor, and 57% on real outdoor.

Qualitative results of a dozen of methods are shown in Fig. 11 for indoor datasets, and Fig. 12 for outdoor datasets.

Overall, one may note the synthetic to real best performance gap of indoor datasets, which is approx. 10 – 35% mIoU and 10 – 18% IoU. While a difference is expected, once again it highlights that geometry has a smaller synthetic/real domain gap compared to semantics. On a general note also, most methods perform significantly better on IoU than on mIoU, demonstrating the complexity of the *semantics* scene completion. In fact, the ranking of methods differs depending on the metric. For example, on NYUv2 (indoor) the recently presented SISNet has best mIoU (52.4%) and IoU (78.2%) by a large margin thanks to their semantic instance completion block and iterative SSC architecture. By relying solely on SSC, CCPNet [177] gets best indoor mIoU (41.3%), although achieved through SUNCG pre-training. Conversely, 3DSketch [17] achieves similar mIoU (41.1%) by training solely on NYUv2 and Chen et al. [19] is second overall on IoU (73.4%) with the same setup. On SemanticKITTI (outdoor) S3CNet [21] has best mIoU (29.5%) and Local-DIFs [121] best IoU (57.7%). Note also the large difference between best indoor/outdoor metrics. While only a handful of methods [86, 124, 137, 174] are evaluated in both setups, they indeed perform significantly worse on outdoor data – though indoor/outdoor performance should be carefully compared given the different resolution. This is partially explained by the higher sparsity in outdoor datasets, visible in ‘input’ of Fig. 12. Another explanation is the higher number of classes in SemanticKITTI versus NYU and the extreme class-imbalance setup given that minor classes are very rarely observed, see Fig. 3. On general qualitative results, either indoor (Fig. 11) or outdoor (Fig. 12) results show that predictions are accurate in large homogeneous areas (walls/ground, floor, buildings) and most errors occur at object boundaries. This is evident in Tab. 4, where most methods achieve high performance in the largest classes of SemanticKITTI, but struggle with predictions in less represented ones (e.g. bicycle, motorcycle, person). Worth mentioning, S3CNet [21] achieves considerably larger scores in rare classes (+25%, +37%, +38% respectively), more than twice when compared to next best classed scores. The reason for such behavior is regrettably not deeply explored in their work.

*Inputs.* To ease interpretation, col ‘Input’ in Tab. 3 shows the nature of input used, where ‘G’ is Geometry of any type (depth, TSDF, points, etc.) – possibly several – and ‘T’ is Texture (RGB). From Tab. 3 using both geometry *and* texture (G+T) performs among the best indoor, such as SISNet [11] using both RGB and TSDF, and 3DSketch [17] which relies on textural edges and depth. Generally speaking, G+T enables the prediction of non-salient geometric objects (i.e.

<sup>4</sup> In their seminal work, for memory reason Song *et al.* [137] evaluated SSC only at the 1:4 scale. Subsequently, to provide fair comparisons between indoor datasets and methods, most other indoor SSC have been using the same resolution despite the fact that higher resolution ground truth is available. Recent experiments in [17] advocate that using higher input/output resolution boosts the SSC performance significantly.

Method	Input	Indoor												Outdoor				
		Real-world NYUv2 [133] 60 × 36 × 60				Synthetic NYUCAD [38] 60 × 36 × 60				Synthetic SUNCG [137] 60 × 36 × 60				Real-world SemanticKITTI [6] 256 × 32 × 256				
		Prec.	Recall	IoU	mIoU	Prec.	Recall	IoU	mIoU	Prec.	Recall	IoU	mIoU	Prec.	Recall	IoU	mIoU	
2017	SSCNet <sup>†</sup> [137] <sup>a</sup>	G	57.0	<b>94.5</b>	55.1	24.7	75.0	<b>92.3</b>	70.3	-	76.3	<b>95.2</b>	73.5	46.4	31.7	<b>83.4</b>	29.8	9.5
	SSCNet-full [124]	G	-	-	-	-	-	-	-	-	-	-	-	-	59.6	<b>75.5</b>	50.0	16.1
2018	Guedes et al. [51]	G+T	62.5	82.3	54.3	27.5	-	-	-	-	-	-	-	-	-	-	-	-
	VVNet <sup>†</sup> [54]	G	<b>74.8</b> <sup>‡</sup>	<b>74.5</b> <sup>‡</sup>	<b>59.5</b> <sup>‡</sup>	<b>29.3</b> <sup>‡</sup>	<b>91.1</b> <sup>‡</sup>	<b>86.6</b> <sup>‡</sup>	<b>79.8</b> <sup>‡</sup>	<b>40.9</b> <sup>‡</sup>	<b>90.8</b>	91.7	84.0	66.7	-	-	-	-
	VD-CRF <sup>†</sup> [176]	G	-	-	-	-	-	-	-	-	-	-	48.8	-	-	-	-	-
	ESSCNet [174] <sup>b</sup>	G	71.9	71.9	56.2	26.7	-	-	-	-	<b>92.6</b>	90.4	84.5	<b>70.5</b>	62.6	55.6	41.8	17.5
	SATNet <sup>†</sup> [86]	G+T	-	-	-	-	-	-	-	-	80.7*	<b>96.5*</b>	<b>78.5*</b>	<b>64.3*</b>	-	-	-	-
2019	DDRNet [77]	G+T	71.5	80.8	61.0	30.4	<b>88.7</b>	<b>88.5</b>	79.4	42.8	-	-	-	-	-	-	-	-
	TS3D [43] <sup>c</sup>	G+T	-	-	60.0	34.1	-	-	76.1	46.2	-	-	-	-	31.6	<b>84.2</b>	29.8	9.5
	TS3D+DNet [6]	G	-	-	-	-	-	-	-	-	-	-	-	-	25.9	<b>88.3</b>	25.0	10.2
	TS3D+DNet+SATNet [6]	G	-	-	-	-	-	-	-	-	-	-	-	-	<b>80.5</b>	57.7	50.6	<b>17.7</b>
	EdgeNet <sup>†</sup> [32]	G+T	<b>76.0</b>	68.3	56.1	27.8	-	-	-	-	<b>93.1*</b>	<b>90.4*</b>	<b>84.8*</b>	<b>69.5*</b>	-	-	-	-
	SSC-GAN [20]	G	63.1	<b>87.8</b>	57.8	22.7	80.7	91.1	74.8	42.0	83.4	92.4	78.1	55.6	-	-	-	-
	ForkNet <sup>†</sup> [160]	G	-	-	-	-	-	-	-	-	-	-	86.9 <sup>^</sup>	63.4 <sup>^</sup>	-	-	-	-
	CCPNet <sup>†</sup> [177]	G	74.2 <sup>^</sup>	<b>90.8</b> <sup>^</sup>	<b>63.5</b> <sup>^</sup>	<b>38.5</b> <sup>^</sup>	<b>91.3</b> <sup>^</sup>	<b>92.6</b> <sup>^</sup>	<b>82.4</b> <sup>^</sup>	<b>53.2</b> <sup>^</sup>	<b>98.2</b> <sup>^</sup>	<b>96.8</b> <sup>^</sup>	<b>91.4</b> <sup>^</sup>	<b>74.2</b> <sup>^</sup>	-	-	-	-
	AM <sup>2</sup> FNet [16]	G+T	72.1	80.4	61.3	31.7	87.2	91.0	80.2	44.6	-	-	-	-	-	-	-	-
2020	GRFNet [89]	G+T	68.4	<b>85.4</b>	61.2	32.9	87.2	91.0	80.1	45.3	-	-	-	-	-	-	-	-
	AMFNet [79]	G+T	67.9	82.3	59.0	33.0	-	-	-	-	-	-	-	-	-	-	-	-
	PALNet [78]	G	68.7	85.0	61.3	34.1	87.2	91.7	80.8	46.6	-	-	-	-	-	-	-	-
	3DSketch [17]	G+T	<b>85.0</b>	<b>81.6</b>	<b>71.3</b>	<b>41.1</b>	<b>90.6</b>	<b>92.2</b>	<b>84.2</b>	<b>55.2</b>	-	-	<b>88.2*</b>	<b>76.5*</b>	-	-	-	-
	AIC-Net [76]	G+T	62.4	<b>91.8</b>	59.2	33.3	88.2	90.3	80.5	45.8	-	-	-	-	-	-	-	-
	Wang et al. [152]	G	-	-	-	-	-	-	-	-	<b>92.1</b>	<b>95.5</b>	<b>88.1</b>	<b>74.8</b>	-	-	-	-
	IPF-SPCNet [180]	G+T	70.5	46.7	39.0	<b>35.1</b>	83.3	72.7	63.5	50.7	-	-	-	-	-	-	-	-
	Chen et al. [19]	G	-	-	<b>73.4</b>	34.4	-	-	<b>82.2</b>	44.5	-	-	84.8	63.5	-	-	-	-
	LMSCNet [124]	G	-	-	-	-	-	-	-	-	-	-	-	-	77.1	66.2	55.3	17.0
	LMSCNet-SS [124]	G	-	-	<b>62.2</b> <sup>‡</sup>	<b>28.4</b> <sup>‡</sup>	-	-	-	-	-	-	-	-	<b>81.6</b>	65.1	<b>56.7</b>	17.6
	S3CNet [21]	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	45.6	<b>29.5</b>
	JS3C-Net [169]	G	-	-	-	-	-	-	-	-	-	-	-	-	71.5	73.5	56.6	23.8
	Local-DIFs [121]	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	<b>57.7</b>	22.7
2021	SISNet [11]	G+T	<b>92.1</b>	83.8	<b>78.2</b>	<b>52.4</b>	<b>94.1</b>	91.2	<b>86.3</b>	<b>63.5</b>	-	-	-	-	-	-	-	-

<sup>a</sup> Results in SemanticKITTI reported on [124]. <sup>b</sup> Results in SemanticKITTI reported on [169]. <sup>c</sup> Results in SemanticKITTI reported on [6].

Input: Geometry (depth, range, points, etc.), Texture (RGB). \* Texture input not used due to absence in SUNCG.

<sup>†</sup> Results with SUNCG pre-training on NYUv2 and NYUCAD presented in Tab. 7. <sup>‡</sup> Results provided by authors.

<sup>^</sup> Results reported at a different resolution. CCPNet: (240 × 144 × 240). ForkNet: (80 × 48 × 80).

Table 3: **SSC performance on the most popular datasets without pretraining.** The relatively low best mIoU scores on the challenging real outdoor SemanticKITTI [6] (29.5%) and real indoor NYUv2 [133] (41.1%) shows the complexity of the task. In the ‘method’ column, we indicate variants with an offset. To better interpret the performance, column ‘Input’ shows the type of input modality used where ‘G’ is Geometry (depth, range, points, etc.) and ‘T’ is Texture (RGB). Note that all indoor datasets commonly report performance for 60 × 36 × 60 grids for historical reasons though 4x bigger input is commonly treated, cf. Sec. 4.5.2. Top 5 methods are highlighted in each column from red to white.

paints, windows, doors) as shown in Fig. 11a by the door predicted 3DSketch [17] and missed by SSCNet [137]. Noteworthy, among the best mIoU methods [17, 21, 177] all use TSDF-encoding as geometrical input. On outdoor datasets, only TS3D [43] uses texture without significant improvement. More works are required to evaluate the benefit of RGB modality on outdoor data.

*Architecture and design choices.* One may notice the good performance of hybrid networks [11, 17, 21, 78, 180] (Fig. 7d), which we believe results from richer input signal due to the fusion of multiple modalities. We also argue that multiple neighboring definitions (2D and 3D) provide beneficial complementary signals. For instance, S3CNet combines 2D BEV and 3D f-TSDF for late fusion through post-processing refinement, achieving the best semantic completion performance on SemanticKITTI [6] by a considerable margin (+5.7% mIoU). Qualitative results of the approach

are shown in Fig. 12c, 12d. Similarly, JS3CNet [169] ranks second in the same dataset (23.8% mIoU and 56.6% IoU) with point-wise semantic labeling through SparseConvNet architecture [48] and dense semantic completion using a point-voxel interaction module, enabling to better infer small vehicles as shown in circled areas of Fig. 12e, 12f. Analogously, PALNet [78] middle fuses depth image and f-TSDF features, achieving good performance on NYUv2 (34.1% mIoU and 61.3% mIoU) and NYUCAD (46.6% mIoU and 80.8% mIoU) datasets, such performance can also be attributed to its position-aware loss, to be discussed next.

Contextual awareness (Sec. 4.3.1) seems also to play an important role for the task. This is noticeable with CCPNet [177] encouraging results given the use of a single geometric input (see Fig. 11d, 11e). Note however that in addition to its lightweight design, the output of CCPNet is higher in resolution (240 × 144 × 240) which was proved to

Method	Input	Class																	mIoU		
		road (15.30%)	sidewalk (11.13%)	parking (1.12%)	other-gr. (0.56%)	building (14.1%)	car (3.92%)	truck (0.16%)	bicycle (0.03%)	motorcycle (0.03%)	other-veh. (0.20%)	vegetation (9.30%)	trunk (0.51%)	terrain (9.17%)	person (0.07%)	bicyclist (0.07%)	motorcyclist (0.05%)	fence (3.90%)		pole (0.29%)	tr. sign (0.08%)
S3CNet [169]	G	42.0	22.5	17.0	7.9	<b>50.2</b>	31.2	6.7	<b>41.5</b>	<b>45.0</b>	<b>16.1</b>	39.5	<b>34.0</b>	21.2	<b>45.9</b>	<b>35.8</b>	<b>16.0</b>	<b>31.3</b>	<b>31.0</b>	<b>24.3</b>	<b>29.5</b>
JS3CNet [169]	G	64.7	39.9	34.9	<b>14.1</b>	39.4	33.3	<b>7.2</b>	14.4	8.8	12.7	<b>43.1</b>	19.6	<b>40.5</b>	8.0	5.1	0.4	30.4	18.9	15.9	23.8
Local-DIFs [121]	G	<b>67.9</b>	<b>42.9</b>	<b>40.1</b>	11.4	40.4	<b>34.8</b>	4.4	3.6	2.4	4.8	42.2	26.5	39.1	2.5	1.1	0	29.0	21.3	17.5	22.7
TS3D+DNet+SATNet [6]	G	62.2	31.6	23.3	6.5	34.1	30.7	4.9	0	0	0.1	40.1	21.9	33.1	0	0	0	24.1	16.9	6.9	17.7
LMSCNet-SS [124]	G	64.8	34.7	29.0	4.6	38.1	30.9	1.5	0	0	0.8	41.3	19.9	32.1	0	0	0	21.3	15.0	0.8	17.6

Table 4: Detailed SSC class performance on SemanticKITTI [6] dataset. Best 5 methods are presented and ordered in decreasing mIoU performance from top to bottom.

Method	Input	Class											mIoU
		ceiling (0.74%)	floor (12.44%)	wall (9.67%)	win. (2.12%)	chair (2.03%)	bed (9.17%)	sofa (6.78%)	table (4.14%)	tv (0.53%)	furn. (36.64%)	objs. (15.74%)	
SISNet [11]	G+T	<b>54.7</b>	93.8	<b>53.2</b>	<b>41.9</b>	<b>43.6</b>	<b>66.2</b>	<b>61.4</b>	<b>38.1</b>	29.8	<b>53.9</b>	<b>40.3</b>	<b>52.4</b>
CCPNet <sup>†</sup> [177]	G	25.5	<b>98.5</b>	38.8	27.1	27.3	64.8	58.4	21.5	<b>30.1</b>	38.4	23.8	41.3
3DSketch [17]	G+T	43.1	93.6	40.5	24.3	30.0	57.1	49.3	29.2	14.3	42.5	28.6	41.1
ForkNet <sup>†</sup> [160]	G	36.2	93.8	29.2	18.9	17.7	61.6	52.9	23.3	19.5	45.4	20.0	37.1
IPF-SPCNet [180]	G+T	32.7	66.0	41.2	17.2	34.7	55.3	47.0	21.7	12.5	38.4	19.2	35.1

<sup>†</sup> Results reported on resolution different to (60 × 36 × 60).  
CCPNet: (240 × 144 × 240). ForkNet: (80 × 48 × 80).  
<sup>†</sup> Pretraining on SUNCG.

Table 5: Detailed SSC class performance on NYUv2 [133] dataset. Best 5 methods from Tabs. 3 and 7 are presented and ordered in decreasing mIoU performance from top to bottom.

Method	Input	Class											mIoU
		ceiling (2.68%)	floor (12.27%)	wall (33.55%)	win. (5.79%)	chair (1.80%)	bed (5.95%)	sofa (4.94%)	table (2.90%)	tv (0.36%)	furn. (15.04%)	objs. (14.73%)	
3DSketch <sup>‡</sup> [17]	G*	97.8	91.9	<b>84.1</b>	<b>72.6</b>	60.8	86.8	81.7	<b>68.7</b>	52.6	75.7	<b>68.2</b>	<b>76.5</b>
Wang et al. [152]	G	98.2	92.8	76.3	61.9	<b>62.4</b>	<b>87.5</b>	80.5	66.3	<b>55.2</b>	74.6	67.8	74.8
CCPNet <sup>†</sup> [177]	G	<b>99.2</b>	89.3	76.2	63.3	58.2	86.1	<b>82.6</b>	65.6	53.2	<b>76.8</b>	65.2	74.2
ESSCNet [174]	G	96.6	83.7	74.9	59.0	55.1	83.3	78.0	61.5	47.4	73.5	62.9	70.5
EdgeNet [32]	G*	97.2	<b>94.4</b>	78.4	56.1	50.4	80.5	73.8	54.5	49.8	69.5	59.2	69.5

<sup>†</sup> Results reported on resolution different to (60 × 36 × 60).  
CCPNet: (240 × 144 × 240).  
\* Texture input not used due to absence in SUNCG.  
<sup>‡</sup> Results provided by authors.

Table 6: Detailed SSC class performance on SUNCG [137] dataset. Best 5 methods are presented and ordered in decreasing mIoU performance from top to bottom.

boost performance [177]. The performance of SISNet [11] is also remarkable thanks to instance-wise completion at high resolution and iterative SSC.

On position awareness (Sec. 4.3.2) it seems to boost intra-class consistency together with inter-class distinction. For example 3DSketch [17] and PALNet [78], both use position awareness and achieve high performances in indoor

Method	Input	Indoor							
		Real-world NYUv2 [133] 60 × 36 × 60				Synthetic NYUCAD [38] 60 × 36 × 60			
		Prec.	Recall	IoU	mIoU	Prec.	Recall	IoU	mIoU
2017 SSCNet [137]	G	59.3	92.9	56.6	30.5	75.0	<b>96.0</b>	73.0	-
2018 VVNet [54]	G	69.8	83.1	61.1	32.9	86.4	92.0	80.3	-
VD-CRF [176]	G	-	-	60.0	31.8	-	-	78.4	43.0
SATNet [86]	G+T	67.3	85.8	60.6	34.4	-	-	-	-
2019 EdgeNet [32]	G+T	<b>79.1</b>	66.6	56.7	33.7	-	-	-	-
ForkNet [160]	G	-	-	63.4 <sup>^</sup>	37.1 <sup>^</sup>	-	-	-	-
CCPNet [177]	G	78.8 <sup>^</sup>	<b>94.3<sup>^</sup></b>	<b>67.1<sup>^</sup></b>	<b>41.3<sup>^</sup></b>	<b>93.4<sup>^</sup></b>	91.2 <sup>^</sup>	<b>85.1<sup>^</sup></b>	<b>55.0<sup>^</sup></b>

Input: Geometry (depth, range, points, etc.), Texture (RGB).  
<sup>^</sup> Results reported at a different resolution. CCPNet: (240 × 144 × 240).  
ForkNet: (80 × 48 × 80).

Table 7: SSC performance on indoor datasets with SUNCG pre-training. Synthetic data pre-training results on slight performance gains for all methods.

scenes with 3DSketch ranking second in mIoU among non-pre-trained methods and third overall (Tabs. 3 and 7), visible in Figs. 11a, 11b, 11c. Similarly, S3CNet dominates performance in SemanticKITTI as already mentioned, which performance is noticeable in Figs. 12c, 12d.

An interesting observation is the high density of the completion even regarding the ground truth, visible in Figs. 12b, 12g, 12h. This relationship is studied in [25], where sparsity is exploited by removing input data to impulse unknown space completion.

*Synthetic data pre-training.* Pre-training on large SUNCG is a common workaround to improve performance on the smaller NYUv2 and NYUCAD datasets. Comparison between Tabs. 3 and 7 show that the technique always brings performance gains although the gap becomes less important for most recent methods (i.e. +5.8% mIoU for SSCNet vs. +2.8% mIoU for CCPNet). Since SUNCG is no longer legally available, this practice is less common in recent works.

#### 4.5.3 Network efficiency

In Tab. 8, network parameters and floating-point operations (FLOPs) are listed – where possible – with separation of in-

	Method	Params (M)	FLOPs (G)
2017	SSCNet [137] <sup>a</sup>	0.93	163.8
	VVNet [54] <sup>a</sup>	0.69	119.2
	ESSCNet [174] <sup>a</sup>	0.16	22
	SATNet [86] <sup>a</sup>	1.2	187.5
2018	DDRNet [77]	0.20	27.2
	CCPNet [177]	0.08	6.5
	GRFNet [89]	0.82	713
2020	PALNet [78]	0.22	78.8
	AIC-Net [76]	0.72	96.77
	Chen et al. [19]	<b>0.07</b>	<b>1.6</b>

<sup>a</sup> Reported in [177].

(a)  $60 \times 36 \times 60$  prediction (indoor)

	Method	Params (M)	FLOPs (G)
2017	SSCNet [137] <sup>b</sup>	0.93	82.5
	SSCNet-full [137] <sup>b</sup>	1.09	769.6
2019	TS3D [43] <sup>b</sup>	43.77	2016.7
	TS3D+DNet [6] <sup>b</sup>	51.31	847.1
	TS3D+DNet+SATNet [6] <sup>b</sup>	50.57	905.2
2020	LMSCNet [124]	<b>0.35</b>	<b>72.6</b>
	JS3C-Net [169]	3.1	-
	Local-DIFs [121]	9.9	-

<sup>b</sup> Reported in [124].

(b)  $256 \times 32 \times 256$  prediction (outdoor)

**Table 8: Network statistics.** Number of parameters and FLOPs are reported per method, grouped by resolution output:  $60 \times 36 \times 60$  for typical indoor datasets [38, 133, 137] or  $256 \times 32 \times 256$  for outdoor dataset [6].

door and outdoor networks because they have different output resolutions. Notice the extreme variations between networks, which scale from 1:144 in number of parameters and 1:1260 in FLOPs. Chen *et al.* [19] and LMSCNet [124] are by far the lightest networks with the fewest parameters and lower FLOPs, in indoor and outdoor settings respectively. They also account for the lower number of operations, which can – though not necessarily [92] – contribute to faster inference times. Furthermore, the use of sparse convolutions [48] is commonly applied as a strategy to reduce memory overhead in [21, 25, 169, 174].

## 5 Discussion

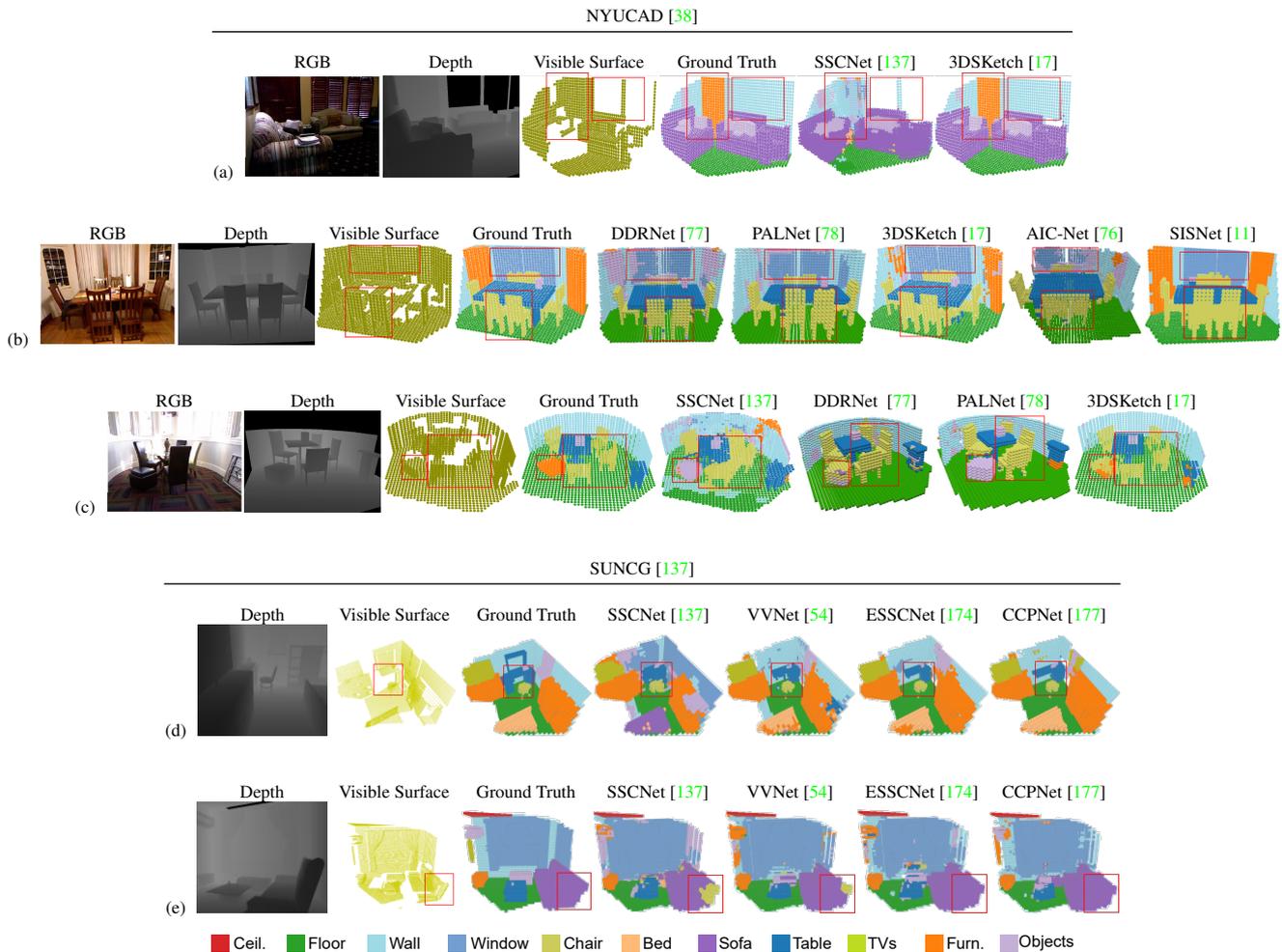
Despite growing interest, there are still major challenges to solve SSC as the best methods still perform poorly on real datasets (see Tabs. 5, 4). In this section, we wish to highlight important remaining issues and provide future research directions.

*Best practices for SSC.* Among the various viable choices for SSC, some were proven highly beneficial. For instance, contextual aggregation (Sec. 4.3.1) improves the information flow. In that sense, fusion of low and high level features with different receptive fields provides a contextual signal for the network, benefiting both the identification of fine-grained details and the overall scene understanding. Like

for semantic segmentation, features aggregation significantly boosts performance, typically with UNet [21, 32, 124, 151, 169, 174], Cascade Pyramid [177] or Feature Aggregation Modules [76, 78, 79, 89]. Further geometrical cues often boost SSC, whether if it is multiple geometrical representations (e.g. depth + voxel, Sec. 4.1) or boundaries (e.g. edges, Sec. 4.3.2). Multi-task and auxiliary tasks can also bring a performance boost. This is commonly done in the form of separated semantic segmentation [169] or instance completion [11] to improve input to SSC. Sketch supervision [17] – which comes at virtually no cost – was shown particularly helpful to boost indoor performance but might be less beneficial for outdoor scenarios given larger semantic geometrical variance (i.e. vegetation). On networks, sparse convolutions [21, 25, 169] can reduce memory needs and therefore enable higher resolution, although they must be combined with dense convolutions. Similar for dilated convolutions that have been used in the wide majority of SSC works [17, 32, 33, 43, 54, 78, 79, 86, 124, 137, 160, 164, 177]. On training, [121] shows that free space supervision close to the geometry can provide sharper inference, and we believe adversarial training (Sec. 4.4.2) is key to cope with the ground truth ambiguities. Another way to have an additional signal is the use of position aware losses [78] which provides additional spatial supervision and was shown to bring performance improvements in both indoor [78] and outdoor scenarios [21]. On evaluation, we encourage authors to evaluate on both indoor *and* outdoor datasets which exhibit different challenges. Finally, for real-time applications, more works like [19, 124] should account for lightweight and fast inference architectures (Sec. 4.5.3).

*Supervision bias.* An important challenge for completion results from the big imbalance ratio between free and occupied space (9:1 in both NYUv2 [133, 137] and SemanticKITTI [6]) which biases the networks towards free space predictions. To deal with this problem, random undersampling of the major free class is often applied [137] to reach an acceptable 2:1 ratio. The strategy reportedly improves completion performance (i.e. +4% IoU [137]) and is widely employed [32, 43, 86, 174, 177]. Similarly, the loss can be balanced to favor occupy predictions [77, 89]. Again, few works like [121] efficiently benefit from free space information.

*Semantic class balancing.* Imbalance is also present in the semantic labels, especially in outdoor datasets, where there is a prevalence of road or vegetation (see Fig. 1 and 3). Class-balancing can be applied to mitigate imbalanced distribution, usually weighting each class according to the inverse of its frequency [124], though prediction of under-represented classes still suffers (e.g. pedestrian or motorcycle in [6]). This may have a catastrophic impact on robotics applications. An approach worth mentioning is S3CNet [21], where



**Fig. 11: Performance of indoor Semantic Scene Completion** on NYUCAD [38] and SUNCG [137]. Methods with RGB modalities (i.e. 3DSketch) enable detection of color salient objects as the highlighted door in row (a). Position awareness also contributes to better reconstruction consistency and inter-class distinction as seen in rows (b), (c) by PALNet [78] and 3DSketch [17]. Furthermore, SISNet [11] overcomes all other methods through their scene-instance-scene loop architecture, seen in row (b). Multi-scale aggregation also improves reconstruction performance as seen on rows (d), (e), where CCPNet [177] achieves the best performance on SUNCG [137].

combined weighted cross entropy and position aware loss (cf. Sec. 4.4.1) achieve impressive improvements in under-represented classes of SemanticKITTI. We believe SSC could benefit from smarter balancing strategies.

*Object motion.* As mentioned in Sec. 3.1, real-world ground truth is obtained by the *rigid* registration of contiguous frames. While this corrects for ego-motion, it doesn't account for scene motion and moving objects produce temporal tubes in the ground truth, as visible in SemanticKITTI [6] (Fig. 4c). As such, to maximize performance, the SSC network must additionally predict the motion of any moving objects. To evaluate the influence of such imperfections for SSC, some works reconstruct target scenes by accounting only for a few future scans [121, 169]. Results show marginal comple-

tion improvement from the application of such a strategy. An alternative proposal [70], is to remove dynamic objects from the detection of spatial singularities after frames registration. On the challenging SemanticKITTI [6], because there are few insights to classify dynamic objects, all methods tend to predict vehicles as stationary (cf. Fig. 12) – producing appealing results but being punished by dataset metrics. This obviously results from the dataset bias, given the abundance of parked vehicles.

The introduction of larger synthetic datasets [31, 126] could be an interesting solution to fight ground truth inaccuracies.

*Datasets extendable for SSC.* Because semantic labeling is the most complex and costly, we denote that a large amount of

SemanticKITTI [6]

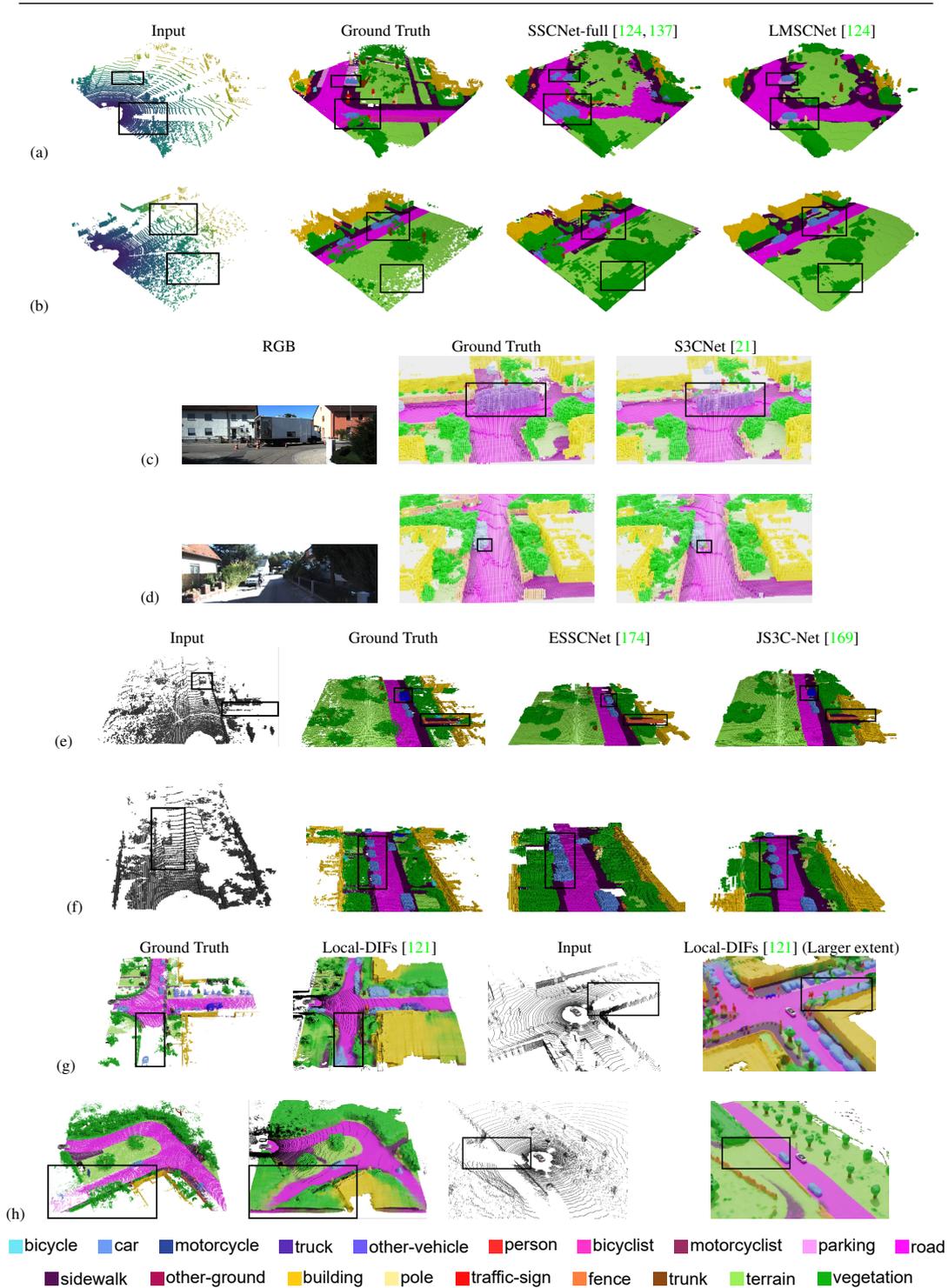


Fig. 12: **Performance of outdoor Semantic Scene Completion** on SemanticKITTI [6]. LMSCNet [124] proposes a lightweight architecture with small performance decrease, rows (a), (b). S3CNet [21] achieves SoA performance by their sparse bird's eye view and 3D f-TSDF feature encoders, rows (c), (d). Two-stage JS3CNet [169] performs point-wise semantic segmentation and semantic scene completion sequentially, enabling better completion as seen in rows (e), (f). Finally, Local-DIFs [121] enables continuous surface prediction, thanks to deep implicit functions, which enable predictions of considerably larger spatial extent, rows (g), (h).

Dataset	Year	Type	Nature	Data	3D Sensor	# Classes	Extension	#Sequences
IQMulus [149]	2015	Real-world	Outdoor	→ Points	Lidar	-	Sparse input scene subsampling	-
Semantic3D [56]	2017	Real-world	Outdoor	→ Points	3D Scanner	8	Sparse input scene subsampling	30
Paris-Lille-3D [127]	2018	Real-world	Outdoor	→ Points	Lidar-32	50	Sparse input scene subsampling	4
Replica [140]	2019	Real-world <sup>†</sup>	Indoor	→ 3D Mesh	RGB-D	88	Sparse input from virtual RGB-D	35
nuScenes [10]	2020	Real-world	Outdoor	Points/RGB →	Lidar-32	32	Dense scenes registration	1000
Toronto-3D [144]	2020	Real-world	Outdoor	→ Points	Lidar-32	8	Sparse input scene subsampling	4
3D-FRONT [39]	2020	Synthetic	Indoor	→ Mesh	-	-	Sparse input from virtual RGB-D	-

<sup>†</sup> Synthetically augmented.

Table 9: **SSC-extendable datasets.** To promote research on SSC we highlight that existing 3D semantic datasets could be extended for SSC, at the cost of processing work (cf. col. ‘Extension’). While some extensions could be obtained with little processing (e.g. Replica [140], 3D-Front [39]), others are significantly more complex (e.g. nuScenes [10]).

existing 3D semantics datasets [10, 39, 56, 127, 140, 144, 149] could also be extended to SSC at the cost of some processing effort. A selective list of these *SSC-extendable* datasets is in Tab. 9 and we believe that their use should be encouraged to serve the interest of research on SSC. Interestingly, most need little processing for SSC (e.g. sparse input generation from 3D meshes or point clouds, virtual sensor configurations) [56, 127, 140, 144, 149], though some require more complex processing (e.g. aggregation of sparse inputs [10]). We also encourage the use of autonomous driving simulators such as CARLA [31], SYNTHIA [126] for *synthetic* dataset generation, devoid of dynamic objects and subsequent registration problems. More extensive surveys on RGB-D and Lidar datasets are provided in [37, 42].

## 6 Conclusion

This paper provided a comprehensive survey on contemporary state-of-the-art methods for 3D Semantic Scene Completion. We reviewed, and critically analyzed major aspects of proposed approaches, including important design choices to be considered, and compared their performance in popular SSC datasets. We believe that this survey will support further development in the field, aiming to provide new insights and help inexperienced readers to navigate the field.

## References

- Abbasi, A., Kalkan, S., Sahillioglu, Y.: Deep 3D semantic scene extrapolation. *The Visual Computer* (2018) 9
- Ahmed, E., Saint, A., Shabayek, A.E.R., Cherenkova, K., Das, R., Gusev, G., Aouada, D., Ottersten, B.: A survey on deep learning advances on different 3d data representations. *arXiv* (2018) 2
- Armeni, I., Sax, S., Zamir, A., Savarese, S.: Joint 2D-3D-semantic data for indoor scene understanding. *ArXiv* (2017) 3, 5, 7
- Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A.X., Nießner, M.: Scan2CAD: Learning CAD model alignment in RGB-D scans. In: *CVPR* (2019) 2, 3
- Avetisyan, A., Khanova, T., Choy, C., Dash, D., Dai, A., Nießner, M.: SceneCAD: Predicting object alignments and layouts in RGB-D scans. In: *ECCV* (2020) 3
- 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) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 15, 16, 17, 18, 19, 20
- Bentley, J.L.: Multidimensional binary search trees used for associative searching. *Communications of the ACM* (1975) 12
- Boulch, A., Guerry, J., Saux, B.L., Audebert, N.: SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Comput. Graph.* (2018) 3
- Boulch, A., Saux, B.L., Audebert, N.: Unstructured point cloud semantic labeling using deep segmentation networks. In: *3DOR@Eurographics* (2017) 3
- Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: *CVPR* (2020) 3, 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) 7, 8, 15, 16, 17, 18, 19
- Canny, J.: A computational approach to edge detection. *PAMI* (1986) 10
- Chang, A.X., Dai, A., Funkhouser, T.A., Halber, M., Nießner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. In: *3DV* (2017) 3, 4, 7
- de Charette, R., Manitsaris, S.: 3D reconstruction of deformable revolving object under heavy hand interaction. *arXiv* (2019) 2
- Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic image segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *PAMI* (2018) 10
- Chen, R., Huang, Z., Yu, Y.: Am2fnet: Attention-based multiscale & multi-modality fused network. *ROBIO* (2019) 6, 7, 10, 11, 16
- 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) 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
- Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3D object detection network for autonomous driving. In: *CVPR* (2017) 9
- Chen, X., Xing, Y., Zeng, G.: Real-time semantic scene completion via feature aggregation and conditioned prediction. In: *ICIP* (2020) 6, 7, 8, 10, 12, 13, 14, 15, 16, 18
- Chen, Y., Garbade, M., Gall, J.: 3D semantic scene completion from a single depth image using adversarial training. In: *ICIP* (2019) 5, 6, 7, 8, 9, 12, 13, 14, 16
- Cheng, R., Agia, C., Ren, Y., Li, X., Bingbing, L.: S3CNet: A sparse semantic scene completion network for LiDAR point clouds. In: *CoRL* (2020) 6, 7, 8, 9, 10, 11, 12, 15, 16, 18, 20
- Cherabier, I., Schönberger, J.L., Oswald, M., Pollefeys, M., Geiger, A.: Learning priors for semantic 3D reconstruction. In: *ECCV* (2018) 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
- Choy, C., Gwak, J., Savarese, S.: 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In: *CVPR* (2019) 8, 12

24. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T.A., Nießner, M.: ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: CVPR (2017) [3](#), [4](#), [7](#)
25. Dai, A., Diller, C., Nießner, M.: SG-NN: Sparse generative neural networks for self-supervised scene completion of RGB-D scans. In: CVPR (2020) [2](#), [4](#), [5](#), [6](#), [9](#), [12](#), [13](#), [14](#), [15](#), [17](#), [18](#)
26. Dai, A., Nießner, M.: 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In: ECCV (2018) [3](#)
27. Dai, A., Qi, C.R., Nießner, M.: Shape completion using 3D-encoder-predictor CNNs and shape synthesis. In: CVPR (2017) [2](#)
28. Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., Nießner, M.: ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans. In: CVPR (2018) [4](#), [5](#), [6](#), [7](#), [8](#), [12](#), [14](#), [15](#)
29. Davis, J., Marschner, S., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission (2002) [2](#)
30. Denninger, M., Triebel, R.: 3D scene reconstruction from a single viewport. In: ECCV. Springer (2020) [7](#)
31. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: CoRL (2017) [3](#), [19](#), [21](#)
32. Dourado, A., de Campos, T.E., Kim, H.S., Hilton, A.: EdgeNet: Semantic scene completion from RGB-D images. ICPR (2020) [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [16](#), [17](#), [18](#)
33. Dourado, A., Kim, H., de Campos, T.E., Hilton, A.: Semantic scene completion from a single 360-Degree image and depth map. In: VISIGRAPP (2020) [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [18](#)
34. Engelmann, F., Rematas, K., Leibe, B., Ferrari, V.: From points to multi-object 3D reconstruction. In: CVPR (2021) [2](#)
35. Everingham, M., Eslami, S., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal visual object classes challenge: A retrospective. International Journal of Computer Vision (2014) [14](#)
36. Fan, H., Su, H., Guibas, L.: A point set generation network for 3D object reconstruction from a single image. In: CVPR (2017) [13](#)
37. Firman, M.: RGBD datasets: Past, present and future. In: CVPRW (2016) [21](#)
38. Firman, M., Aodha, O.M., Julier, S.J., Brostow, G.J.: Structured prediction of unobserved voxels from a single depth image. In: CVPR (2016) [2](#), [3](#), [4](#), [6](#), [7](#), [16](#), [17](#), [18](#), [19](#)
39. Fu, H., Cai, B., Gao, L., Zhang, L.X., Li, C., Xun, Z., Sun, C., Fei, Y., Zheng, Y., Li, Y., Liu, Y., Liu, P., Ma, L., Weng, L., Hu, X., Ma, X., Qian, Q., Jia, R., Zhao, B., Zhang, H.: 3D-FRONT: 3D furnished rooms with layouts and semantics. ArXiv (2020) [3](#), [21](#)
40. Fuentes-Pacheco, J., Ascencio, J.R., Rendón-Mancha, J.M.: Visual simultaneous localization and mapping: a survey. Artificial Intelligence Review (2012) [4](#)
41. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: VirtualWorlds as proxy for multi-object tracking analysis. In: CVPR (2016) [3](#)
42. Gao, B., Pan, Y., Li, C., Geng, S., Zhao, H.: Are we hungry for 3D LiDAR data for semantic segmentation? a survey of datasets and methods. T-ITS (2021) [3](#), [21](#)
43. Garbade, M., Sawatzky, J., Richard, A., Gall, J.: Two stream 3D semantic scene completion. In: CVPR Workshops (2019) [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [16](#), [18](#)
44. Garg, S., Sünderhauf, N., Dayoub, F., Morrison, D., Cosgun, A., Carneiro, G., Wu, Q., Chin, T., Reid, I.D., Gould, S., Corke, P., Milford, M.: Semantics for robotic mapping, perception and interaction: A survey. Foundations and Trends in Robotics (2020) [1](#)
45. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. IJRR (2013) [5](#)
46. Geiger, A., Wang, C.: Joint 3D object and layout inference from a single RGB-D image. In: GCPR (2015) [6](#)
47. Gkioxari, G., Malik, J., Johnson, J.J.: Mesh R-CNN. In: ICCV (2019) [3](#)
48. Graham, B., Engelcke, M., van der Maaten, L.: 3D semantic segmentation with submanifold sparse convolutional networks. In: CVPR (2018) [8](#), [12](#), [16](#), [18](#)
49. Griffiths, D., Boehm, J.: Synthcity: A large scale synthetic point cloud. ArXiv (2019) [3](#)
50. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: AtlasNet: A papier-mâché approach to learning 3D surface generation. In: CVPR (2018) [6](#)
51. Guedes, A.B.S., de Campos, T.E., Hilton, A.: Semantic scene completion combining colour and depth: preliminary experiments. ArXiv (2018) [7](#), [8](#), [10](#), [11](#), [16](#)
52. Guo, R., Hoiem, D.: Support surface prediction in indoor scenes. ICCV (2013) [3](#), [4](#)
53. Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M.: Deep learning for 3D point clouds: A survey. PAMI (2020) [1](#), [2](#)
54. Guo, Y.X., Tong, X.: View-volume network for semantic scene completion from a single depth image. In: IJCAI (2018) [1](#), [5](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [16](#), [17](#), [18](#), [19](#)
55. Gupta, S., Girshick, R.B., Arbeláez, P., Malik, J.: Learning rich features from RGB-D images for object detection and segmentation. In: ECCV (2014) [8](#)
56. Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M.: Semantic3D.net: A new large-scale point cloud classification benchmark. ISPRS Annals (2017) [21](#)
57. Han, X., Laga, H., Bennamoun, M.: Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era. PAMI (2019) [2](#)
58. Han, X., Li, Z., Huang, H., Kalogerakis, E., Yu, Y.: High-resolution shape completion using deep neural networks for global structure and local geometry inference. In: ICCV (2017) [2](#)
59. Han, X., Zhang, Z., Du, D., Yang, M., Yu, J., Pan, P., Yang, X., Liu, L., Xiong, Z., Cui, S.: Deep reinforcement learning of volume-guided progressive view inpainting for 3D point scene completion from a single depth image. In: CVPR (2019) [6](#)
60. Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., Cipolla, R.: SceneNet: Understanding real world indoor scenes with synthetic data. In: CVPR (2016) [3](#), [5](#)
61. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) [10](#)
62. Hou, J., Dai, A., Nießner, M.: 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In: CVPR (2019) [3](#)
63. Hou, J., Dai, A., Nießner, M.: RevealNet: Seeing behind objects RGB-D scans. In: CVPR (2020) [3](#)
64. Hua, B.S., Pham, Q.H., Nguyen, D., Tran, M., Yu, L.F., Yeung, S.: SceneNN: A scene meshes dataset with annotations. In: 3DV (2016) [3](#), [4](#)
65. Huang, H., Chen, H., Li, J.: Deep neural network for 3D point cloud completion with multistage loss function. Chinese Control And Decision Conference (CCDC) (2019) [8](#)
66. Huang, Z., Yu, Y., Xu, J., Ni, F., Le, X.: PF-Net: Point fractal network for 3D point cloud completion. In: CVPR (2020) [5](#)
67. Izadinia, H., Shan, Q., Seitz, S.M.: IM2CAD. In: CVPR (2017) [3](#)
68. Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., Qu, R.: A survey of deep learning-based object detection. Access (2019) [2](#)
69. Kazhdan, M.M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: SGP (2006) [2](#)
70. Kim, G., Kim, A.: Remove, then revert: Static point cloud map construction using multiresolution range images. In: IROS. IEEE (2020) [19](#)
71. Klokov, R., Lempitsky, V.: Escape from cells: Deep Kd-networks for the recognition of 3D point cloud models. In: ICCV (2017) [12](#)
72. Kundu, A., Li, Y., Rehg, J.M.: 3D-RCNN: Instance-level 3D object reconstruction via render-and-compare. In: CVPR (2018) [3](#)

73. Kurenkov, A., Ji, J., Garg, A., Mehta, V., Gwak, J., Choy, C.B., Savarese, S.: DeformNet: Free-form deformation network for 3D shape reconstruction from a single image. In: WACV (2018) [13](#)
74. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. In: CVPR (2018) [3](#)
75. Li, D., Shao, T., Wu, H., Zhou, K.: Shape completion from a single RGBD image. *IEEE Transactions on Visualization and Computer Graphics* (2017) [2](#)
76. Li, J., Han, K., Wang, P., Liu, Y., Yuan, X.: Anisotropic convolutional networks for 3D semantic scene completion. In: CVPR (2020) [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [16](#), [18](#), [19](#)
77. Li, J., Liu, Y., Gong, D., Shi, Q., Yuan, X., Zhao, C., Reid, I.D.: RGBD based dimensional decomposition residual network for 3D semantic scene completion. In: CVPR (2019) [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [16](#), [18](#), [19](#)
78. Li, J., Liu, Y.W., Yuan, X., Zhao, C., Siegwart, R., Reid, I., Cadena, C.: Depth based semantic scene completion with position importance aware loss. *Robotics and Automation Letters (RA-L)* (2020) [5](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [16](#), [17](#), [18](#), [19](#)
79. Li, S., Zou, C., Li, Y., Zhao, X., Gao, Y.: Attention-based multi-modal fusion network for semantic scene completion. In: AAAI (2020) [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [16](#), [18](#)
80. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: PointCNN: convolution on X-transformed points. In: NIPS (2018) [3](#), [9](#)
81. Li, Y., Dai, A., Guibas, L., Nießner, M.: Database-assisted object retrieval for real-time 3D reconstruction. *Comput. Graph. Forum* (2015) [2](#)
82. Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M.A., Cao, D., Li, J.: Deep learning for LiDAR point clouds in autonomous driving: A review. *IEEE Trans. on Neural Networks and Learning Systems* (2020) [2](#)
83. Liao, Y., Donné, S., Geiger, A.: Deep marching cubes: Learning explicit surface representations. In: CVPR (2018) [6](#)
84. Lin, D., Fidler, S., Urtasun, R.: Holistic scene understanding for 3D object detection with RGBD cameras. In: ICCV (2013) [6](#), [8](#)
85. Lin, G., Milan, A., Shen, C., Reid, I.: RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In: CVPR (2017) [11](#)
86. Liu, S., Hu, Y., Zeng, Y., Tang, Q., Jin, B., Han, Y., Li, X.: See and think: Disentangling semantic scene completion. In: NeurIPS (2018) [3](#), [5](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#)
87. Liu, S., Mello, S.D., Gu, J., Zhong, G., Yang, M.H., Kautz, J.: Learning affinity via spatial propagation networks. In: NIPS (2017) [12](#)
88. Liu, W., Sun, J., Li, W., Hu, T., Wang, P.: Deep learning on point clouds and its application: A survey. *Sensors* (2019) [2](#)
89. Liu, Y.W., Li, J., Yan, Q., Yuan, X., Zhao, C.X., Reid, I., Cadena, C.: 3D gated recurrent fusion for semantic scene completion. *ArXiv* (2020) [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [16](#), [18](#)
90. Lorensen, W., Cline, H.: Marching cubes: A high resolution 3D surface construction algorithm. In: SIGGRAPH (1987) [5](#), [6](#)
91. Lu, H., Shi, H.: Deep learning for 3d point cloud understanding: A survey. *arXiv* (2020) [2](#)
92. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: ECCV (2018) [18](#)
93. Maturana, D., Scherer, S.A.: VoxNet: A 3D convolutional neural network for real-time object recognition. In: IROS (2015) [3](#)
94. Meagher, D.: Geometric modeling using octree encoding. *Computer graphics and image processing* (1982) [12](#)
95. Meng, H.Y., Gao, L., Lai, Y., Manocha, D.: VV-Net: Voxel VAE net with group convolutions for point cloud segmentation. In: ICCV (2019) [3](#)
96. Mitra, N., Pauly, M., Wand, M., Ceylan, D.: Symmetry in 3D geometry: Extraction and applications. *Comput. Graph. Forum* (2013) [2](#)
97. Müller, N., Wong, Y.S., Mitra, N., Dai, A., Nießner, M.: Seeing behind objects for 3D multi-object tracking in RGB-D sequences. In: CVPR (2021) [3](#)
98. Nair, R., Lenzen, F., Meister, S., Schäfer, H., Garbe, C., Kondermann, D.: High accuracy TOF and stereo sensor fusion at interactive rates. In: ECCV Workshops (2012) [4](#)
99. Nan, L., Xie, K., Sharf, A.: A search-classify approach for cluttered indoor scene understanding. *ACM Trans. Graph.* (2012) [3](#)
100. Nealen, A., Igarashi, T., Sorkine-Hornung, O., Alexa, M.: Laplacian mesh optimization. In: GRAPHITE '06 (2006) [2](#)
101. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.W.: KinectFusion: Real-time dense surface mapping and tracking. *International Symposium on Mixed and Augmented Reality* (2011) [7](#)
102. Nguyen, A., Le, H.: 3D point cloud segmentation: A survey. *Conference on Robotics, Automation and Mechatronics (RAM)* (2013) [3](#)
103. Nie, Y., Han, X.G., Guo, S., Zheng, Y., Chang, J., Zhang, J.: Total3DUnderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In: CVPR (2020) [3](#)
104. Nie, Y., Hou, J., Han, X., Nießner, M.: RfD-Net: Point scene understanding by semantic instance reconstruction. In: CVPR (2021) [3](#)
105. Pan, Y., Gao, B., Mei, J., Geng, S., Li, C., Zhao, H.: SemanticPOSS: A point cloud dataset with large quantity of dynamic instances. In: IV (2020) [3](#), [5](#)
106. Park, J.J., Florence, P., Straub, J., Newcombe, R.A., Lovegrove, S.: DeepSDF: Learning continuous signed distance functions for shape representation. In: CVPR (2019) [2](#)
107. Pauly, M., Mitra, N., Giesen, J., Groß, M., Guibas, L.: Example-based 3D scan completion. In: SGP (2005) [2](#)
108. Pauly, M., Mitra, N., Wallner, J., Pottmann, H., Guibas, L.: Discovering structural regularity in 3D geometry. In: SIGGRAPH 2008 (2008) [2](#)
109. Pintore, G., Mura, C., Ganovelli, F., Perez, L.J.F., Pajarola, R., Gobbetti, E.: State-of-the-art in automatic 3D reconstruction of structured indoor environments. *CGF* (2020) [2](#)
110. Pock, T., Chambolle, A.: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: ICCV. *IEEE* (2011) [10](#)
111. Pomerleau, F., Colas, F., Siegwart, R.: A survey of rigid 3D point-cloud registration algorithms. *International Conference on Ambient Computing, Applications, Services and Technologies* (2014) [4](#)
112. Pomerleau, F., Colas, F., Siegwart, R.: A review of point cloud registration algorithms for mobile robotics. *Found. Trends Robotics* (2015) [4](#)
113. Popov, S., Bauszat, P., Ferrari, V.: CoReNet: Coherent 3D scene reconstruction from a single RGB image. In: ECCV (2020) [3](#)
114. Qi, C., Litany, O., He, K., Guibas, L.: Deep hough voting for 3d object detection in point clouds. In: ICCV (2019) [3](#)
115. Qi, C.R., Su, H., Mo, K., Guibas, L.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: CVPR (2017) [3](#)
116. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017) [3](#), [9](#)
117. Reed, S., Oord, A., Kalchbrenner, N., Colmenarejo, S.G., Wang, Z., Chen, Y., Belov, D., Freitas, N.D.: Parallel multiscale autoregressive density estimation. In: ICML (2017) [12](#)
118. Rezende, D.J., Eslami, S., Mohamed, S., Battaglia, P., Jaderberg, M., Heess, N.: Unsupervised learning of 3D structure from images. In: NIPS (2016) [2](#)

119. Riegler, G., Ulusoy, A.O., Bischof, H., Geiger, A.: OctNetFusion: Learning depth fusion from data. In: 3DV (2017) [12](#)
120. Riegler, G., Ulusoy, A.O., Geiger, A.: OctNet: Learning deep 3D representations at high resolutions. In: CVPR (2017) [3](#), [12](#)
121. Rist, C.B., Emmerichs, D., Enzweiler, M., Gavrilu, D.M.: Semantic scene completion using local deep implicit functions on LiDAR data. ArXiv (2020) [4](#), [5](#), [7](#), [8](#), [9](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#), [19](#), [20](#)
122. Rist, C.B., Schmidt, D., Enzweiler, M., Gavrilu, D.M.: SCSSnet: Learning spatially-conditioned scene segmentation on LiDAR point clouds. In: IV (2020) [7](#), [8](#), [9](#), [12](#), [13](#)
123. Rock, J., Gupta, T., Thorsen, J., Gwak, J., Shin, D., Hoiem, D.: Completing 3D object shape from one depth image. In: CVPR (2015) [2](#)
124. Roldao, L., de Charette, R., Verroust-Blondet, A.: LMSCNet: Lightweight multiscale 3D semantic completion. In: 3DV (2020) [1](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#), [20](#)
125. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015) [8](#)
126. Ros, G., Sellart, L., Materzynska, J., Vázquez, D., López, A.M.: The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR (2016) [3](#), [19](#), [21](#)
127. Roynard, X., Deschaud, J.E., Goulette, F.: Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. The International Journal of Robotics Research (2018) [21](#)
128. Saputra, M.R.U., Markham, A., Trigoni, A.: Visual SLAM and structure from motion in dynamic environments. ACM Computing Surveys (CSUR) (2018) [4](#)
129. Shao, T., Xu, W., Zhou, K., Wang, J., Li, D., Guo, B.: An interactive approach to semantic modeling of indoor scenes with an RGBD camera. ACM Trans. Graph. (2012) [3](#)
130. Sharma, A., Grau, O., Fritz, M.: VConv-DAE: Deep volumetric shape learning without object labels. In: ECCV Workshops (2016) [2](#)
131. Shen, C.H., Fu, H., Chen, K., Hu, S.: Structure recovery by part assembly. ACM Trans. Graph. (2012) [2](#)
132. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. TPAMI (2020) [3](#)
133. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: ECCV (2012) [1](#), [2](#), [3](#), [4](#), [7](#), [16](#), [17](#), [18](#)
134. Sipiran, I., Gregor, R., Schreck, T.: Approximate symmetry detection in partial 3D meshes. Comput. Graph. Forum (2014) [2](#)
135. Smith, E., Meger, D.: Improved adversarial systems for 3D object generation and reconstruction. In: CoRL (2017) [2](#)
136. Song, S., Xiao, J.: Deep sliding shapes for amodal 3d object detection in rgb-d images. CVPR (2016) [3](#)
137. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.A.: Semantic scene completion from a single depth image. In: CVPR (2017) [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#), [19](#), [20](#)
138. Song, S., Zeng, A., Chang, A.X., Savva, M., Savarese, S., Funkhouser, T.: Im2pano3d: Extrapolating 360° structure and semantics beyond the field of view. In: CVPR (2018) [9](#)
139. Sorkine-Hornung, O., Cohen-Or, D.: Least-squares meshes. Proceedings Shape Modeling Applications, 2004. (2004) [2](#)
140. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.A.: The Replica dataset: A digital replica of indoor spaces. ArXiv (2019) [2](#), [4](#), [21](#)
141. Stutz, D., Geiger, A.: Learning 3D shape completion from laser scan data with weak supervision. In: CVPR (2018) [2](#)
142. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3D shape recognition. In: ICCV (2015) [3](#)
143. Sung, M., Kim, V.G., Angst, R., Guibas, L.: Data-driven structural priors for shape completion. ACM Trans. Graph. (2015) [2](#)
144. Tan, W., Qin, N., Ma, L., Li, Y., Du, J., Cai, G., Yang, K., Li, J.: Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways. In: CVPR Workshops (2020) [21](#)
145. Tchapmi, L.P., Choy, C., Armeni, I., Gwak, J., Savarese, S.: SEG-Cloud: Semantic segmentation of 3D point clouds. In: 3DV (2017) [3](#)
146. Tchapmi, L.P., Kosaraju, V., Rezatofighi, H., Reid, I., Savarese, S.: TopNet: Structural point cloud decoder. In: CVPR (2019) [5](#)
147. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.: KPConv: Flexible and deformable convolution for point clouds. In: ICCV (2019) [3](#)
148. Thrun, S., Wegbreit, B.: Shape from symmetry. In: ICCV (2005) [2](#)
149. Vallet, B., Brédif, M., Serna, A., Marcotegui, B., Paparoditis, N.: TerraMobilita/iQmulus urban point cloud analysis benchmark. Comput. Graph. (2015) [21](#)
150. Varley, J., DeChant, C., Richardson, A., Ruales, J., Allen, P.: Shape completion enabled robotic grasping. In: IROS (2017) [2](#)
151. Wang, P., Liu, Y., Guo, Y., Sun, C., Tong, X.: O-CNN: octree-based convolutional neural networks for 3d shape analysis. ACM Trans. Graph. (2017) [12](#), [14](#), [18](#)
152. Wang, P.S., Liu, Y., Tong, X.: Deep Octree-based CNNs with output-guided skip connections for 3D shape and scene completion. In: CVPR Workshops (2020) [6](#), [7](#), [8](#), [12](#), [13](#), [14](#), [16](#), [17](#)
153. Wang, P.S., Sun, C., Liu, Y., Tong, X.: Adaptive O-CNN: A patch-based deep representation of 3D shapes. TOG (2018) [3](#), [12](#)
154. Wang, W., Yu, R., Huang, Q., Neumann, U.: SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In: CVPR (2018) [5](#)
155. Wang, X., Ang, M., Lee, G.H.: Cascaded refinement network for point cloud completion. In: CVPR (2020) [5](#), [8](#)
156. Wang, X., Ang, M., Lee, G.H.: Point cloud completion by learning shape priors. In: IROS (2020) [5](#)
157. Wang, X., Oswald, M., Cherabier, I., Pollefeys, M.: Learning 3D semantic reconstruction on octrees. In: German Conference on Pattern Recognition (2019) [7](#), [8](#), [12](#)
158. Wang, Y., Sun, Y., Liu, Z., Sarma, S., Bronstein, M., Solomon, J.: Dynamic graph CNN for learning on point clouds. TOG (2019) [3](#)
159. Wang, Y., Tan, D.J., Navab, N., Tombari, F.: Adversarial semantic scene completion from a single depth image. In: 3DV (2018) [7](#), [8](#), [14](#)
160. Wang, Y., Tan, D.J., Navab, N., Tombari, F.: ForkNet: Multi-branch volumetric semantic completion from a single depth image. In: ICCV (2019) [5](#), [6](#), [7](#), [8](#), [10](#), [12](#), [13](#), [14](#), [16](#), [17](#), [18](#)
161. Wang, Y., Tan, D.J., Navab, N., Tombari, F.: SoftPoolNet: Shape descriptor for point cloud completion and classification. In: ECCV (2020) [5](#)
162. Wang, Y., Wu, S., Huang, H., Cohen-Or, D., Sorkine-Hornung, O.: Patch-based progressive 3D point set upsampling. In: CVPR (2019) [5](#)
163. Wen, X., Li, T., Han, Z., Liu, Y.S.: Point cloud completion by skip-attention network with hierarchical folding. In: CVPR (2020) [5](#)
164. Wu, S.C., Tateno, K., Navab, N., Tombari, F.: SCFusion: Real-time incremental scene reconstruction with semantic completion. In: 3DV (2020) [3](#), [4](#), [6](#), [7](#), [8](#), [10](#), [12](#), [14](#), [18](#)
165. Xiao, J., Owens, A., Torralba, A.: SUN3D: A database of big spaces reconstructed using SfM and object labels. In: ICCV (2013) [3](#), [5](#)

- 
166. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: GRNet: Gridding residual network for dense point cloud completion. In: ECCV (2020) [5](#), [8](#)
  167. Xie, Y., Tian, J., Zhu, X.: Linking points with labels in 3D: A review of point cloud semantic segmentation. *Geoscience and Remote Sensing Magazine* (2020) [3](#)
  168. Xie, Y., Tian, J., Zhu, X.X.: Linking points with labels in 3d: A review of point cloud semantic segmentation. *Geoscience and Remote Sensing Magazine* (2020) [2](#)
  169. Yan, X., Gao, J., Li, J., Zhang, R., Li, Z., Huang, R., Cui, S.: Sparse single sweep LiDAR point cloud segmentation via learning contextual shape priors from scene completion. In: AAAI (2021) [4](#), [6](#), [7](#), [8](#), [9](#), [12](#), [16](#), [17](#), [18](#), [19](#), [20](#)
  170. Yang, B., Rosa, S., Markham, A., Trigoni, N., Wen, H.: Dense 3D object reconstruction from a single depth view. *PAMI* (2019) [2](#)
  171. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2016) [9](#)
  172. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: PCN: Point completion network. In: 3DV (2018) [2](#), [3](#), [5](#), [8](#), [9](#), [13](#)
  173. Zhang, G., Chen, Y.: A metric for evaluating 3D reconstruction and mapping performance with no ground truthing. *ArXiv* (2021) [13](#)
  174. Zhang, J., Zhao, H., Yao, A., Chen, Y., Zhang, L., Liao, H.: Efficient semantic scene completion network with spatial group convolution. In: ECCV (2018) [5](#), [6](#), [7](#), [8](#), [9](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#), [19](#), [20](#)
  175. Zhang, J., Zhao, X., Chen, Z., Lu, Z.: A review of deep learning-based semantic segmentation for point cloud. *IEEE Access* (2019) [2](#)
  176. Zhang, L., Wang, L., Zhang, X., Shen, P., Bennamoun, M., Zhu, G., Shah, S.A.A., Song, J.: Semantic scene completion with dense crf from a single depth image. *Neurocomputing* (2018) [7](#), [8](#), [12](#), [16](#), [17](#)
  177. Zhang, P., Liu, W., Lei, Y., Lu, H., Yang, X.: Cascaded context pyramid for full-resolution 3D semantic scene completion. In: ICCV (2019) [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#), [19](#)
  178. Zhang, W., Yan, Q., Xiao, C.: Detail preserved point cloud completion via separated feature aggregation. In: ECCV (2020) [5](#)
  179. Zheng, B., Zhao, Y., Yu, J.C., Ikeuchi, K., Zhu, S.C.: Beyond point clouds: Scene understanding by reasoning geometry and physics. In: CVPR (2013) [6](#)
  180. Zhong, M., Zeng, G.: Semantic point completion network for 3D semantic scene completion. In: ECAI (2020) [3](#), [5](#), [7](#), [8](#), [9](#), [10](#), [11](#), [14](#), [16](#), [17](#)
  181. Zimmermann, K., Petríček, T., Salanský, V., Svoboda, T.: Learning for active 3D mapping. In: ICCV (2017) [2](#)
  182. Zollhöfer, M., Stotko, P., Görlitz, A., Theobalt, C., Nießner, M., Klein, R., Kolb, A.: State of the art on 3D reconstruction with RGB-D cameras. *CGF* (2018) [2](#)