

Point Cloud Completion: A Survey

Keneni W. Tesema* Lyndon Hill† Mark W. Jones* Muneeb I. Ahmad* Gary K.L. Tam*
*Department of Computer Science, Swansea University †Vaarst
Swansea, United Kingdom Bristol, United Kingdom

Abstract—Point cloud completion is the task of producing a complete 3D shape given an input of a partial point cloud. It has become a vital process in 3D computer graphics, vision and applications such as autonomous driving, robotics, and augmented reality. These applications often rely on the presence of a complete 3D representation of the environment. Over the past few years, many completion algorithms have been proposed and a substantial amount of research has been carried out. However, there are not many in-depth surveys that summarise the research progress in such a way that allows users to make an informed choice of what algorithms to employ given the type of data they have, the end result they want, the challenges they may face and the possible strategies they could use. In this study, we present a comprehensive survey and classification of papers on point cloud completion until August 2023 based on the strategies, techniques, inputs, outputs, and network architectures. We will also cover datasets, evaluation methods, and application areas in point cloud completion. Finally, we discuss challenges faced by the research community and future research directions.

Index Terms—Point cloud completion, Deep Learning, Computer Vision

I. INTRODUCTION

Point clouds have become a popular 3D geometrical data representation in computer graphics and computer vision with applications in numerous fields. In augmented reality [1], point clouds help mimic real-world scenes by enabling users to visualise multimedia content in an immersive way. In archaeology [2] [3], point clouds enable archaeologists to simulate reconstruction of antiques and historical sites in their computer before reconstructing them in the real-world. In robotics [4] [5] [6], point clouds provide robots with precisely measured information of their surrounding environments which enables them to do tasks such as grasping and moving. In automated driving [7] [8], accurate and complete point cloud measurements allow automated vehicles to navigate and avoid accidents on the road.

With the advance in sensory technology, scanners nowadays can produce a detailed, high-quality scan of real-world objects. Most 3D scanning devices readily output 3D point clouds offering high resolution at smaller storage space [9]. However, factors such as viewing angles, occlusions, and resolution power affect the acquisition pipeline and result in incomplete or partial point clouds. A complete representation of the target object is often necessary for downstream vision tasks. 3D shape completion thus has an important role in the acquisition pipeline.

Point cloud completion is the task of estimating the missing part of a point cloud from a partial scan [10]. Early completion methods relied on geometric cues such as standard shape geometry, symmetry, and templates that can be aligned with

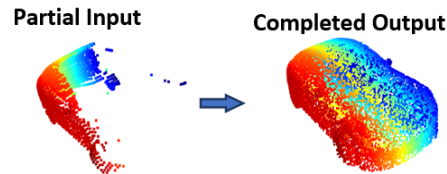


Fig. 1. Point Cloud completion

the partial point clouds to guide the completion process [11]. In contrast, recent works on completion rely on features processed by deep-learning networks. The latest works in completion [12]–[14] integrate techniques from traditional methods and deep-learning methods to achieve the best results.

There are not many surveys on point cloud completion. Existing surveys about point cloud processing focus on downstream tasks such as registration [15] and segmentation [16] or on deep learning based applications like object detection [9]. Surveys covering surface reconstruction [17] include methods similar to those used in completion. A closely related point cloud completion survey by Fei et al. [18] largely focuses on the network architecture of deep-learning-based algorithms. In contrast, our survey aims to summarise the overall progress made in the field including both learning-based and traditional approaches with a focus on important strategies used to overcome challenges in the completion process. We further classify existing techniques into tables covering inputs, outputs, datasets, and evaluation metrics. At the end, we generalise research gaps and discuss possible future research directions.

We summarise the main contributions:

- To our knowledge, this is the first survey that covers both learning-based and traditional approaches in point cloud completion.
- We present a systematic classification of completion algorithms based on their inputs, outputs, completion approaches, datasets, evaluation metrics, and respective strategic techniques to tackle challenges.
- We identify current trends in completion and make suggestions for research directions and summarise the latest progress made in the field.

Our survey is organised as follows. Section II defines the scope of this point cloud completion survey. Section III discusses different inputs and outputs of point cloud completion algorithms. Section IV classifies completion algorithms into traditional and deep learning-based approaches. Public datasets frequently used in completion processes are described in Section V. Section VI discusses evaluation metrics. Section VII surveys the common network architectures used in learning-based algorithms. Section VIII analyses the technical chal-

TABLE I
LITERATURE SOURCES (WITH TWO OR MORE PAPERS)

Journal/Conference	Papers	Journal/Conference	Papers
CVPR	40	ICCV	17
TPAMI	7	ECCV	9
AAAI	8	CGF	6
TVCG	6	ACM SIGGRAPH/TOG	6
IEEE Intell Transp	4	IEEE Robot.autom.lett.	2
IROS	3	3DV	3
NeurIPS	3	ISPRS	4
Sensors	2	ArXiv(preprint)	37

allenges faced both by traditional and learning-based algorithms, and summarises strategies for solving these challenges. Section IX describes the application of completion in other point cloud processing tasks. Section X offers insights from the survey and Section XI discusses current trends, research gaps and possible future works.

II. METHODOLOGY AND SCOPE

Point cloud completion is an area with a long history. It is a part of Shape completion and shares similarities with surface reconstruction and registration. We first discuss the difference between these fields and then define our scope and methodology below.

In this survey, Point cloud completion refers to relevant work in Object completion performed on point cloud data. Object completion is a sub-field of Shape completion. Shape completion is the task of inferring the complete geometric shape from a partial input [13] of any 3D data type (i.e., mesh, voxels, point clouds, etc.). Shape completion is classified into object completion, semantic (whole) scene completion, and semantic instance completion [19]. Object completion aims to complete the missing structure of a single object by using information that is derived from the object or external sources [9]. Semantic (whole) scene completion aims to simultaneously predict the semantics (object labels) and the complete 3D shape of all the objects in the scene from a partial input [20]. Semantic instance completion aims to detect individual instances in a scene or an object and infer their complete object (or object parts) geometry. Compared to Semantic (whole) scene/instance completion, Object completion only predicts the complete shape of a single object and does not estimate the semantics (the label) of that object. However, it is possible that semantic information provided as shape prior may be used to help the completion process. All three subsets of shape completion may work on various 3D data such as meshes, voxels, and point clouds. We focus only on works in object completion that use point cloud data. Additionally, our work excludes works in point cloud generation and 2D-3D estimation problems that construct 3D point clouds from single-view RGB images.

Some works in Surface/Shape reconstruction also contribute to the research in point cloud completion. Surface reconstruction is an estimation of continuous surfaces captured from 3D point clouds [17]. It is a large field on its own and will not be covered in this survey. However, we include a few papers from surface reconstruction that have contributed to completion. In the 3D acquisition pipeline, registration is also

a component that aims to produce a more complete point cloud [15]. Registration however is defined as a transformation estimation problem between two or more point clouds and is not included in this survey. Papers that use completion for the purpose of registration [21] are not included. Interested readers are referred to [15], [22] for a complete discussion.

We collected and analysed papers in the field of point cloud completion. The literature search used the following keywords: “point cloud completion”, “shape completion”, and “object completion”. These keywords are searched in multiple repositories including the Web of Science, Google Scholar, IEEE, ACM digital portals and ArXiv. Our search produced 409 papers, most of which were relevant. We excluded papers relating to Semantic Scene/Instance completion, registration, surface reconstruction and point cloud generation according to the above definition. We further filtered the papers in scope to produce the final selection of 157 papers based on their close relevance to the topic, prominence in the field (citations), and any innovative approaches. A list of literature resources can be found in Table I. Peer-reviewed and published papers, and some recent pre-prints on point cloud completion, up to August 2023, are included in our survey. We paid attention to the strategies used that overcome common challenges in point cloud completion, and offer detailed breakdown and classifications.

III. POINT CLOUD COMPLETION, INPUTS AND OUTPUTS

Early point cloud completion methods took one partial point cloud and produced one predicted complete shape. With the recent development of learning-based completion, methods that take in multiple inputs and produce multiple outputs have been developed. The use of multiple inputs can improve the precision and efficiency of the completion process, whereas multiple outputs can lead to a better understanding of the completed results. We provide a brief classification of the inputs and outputs to help users better understand the problems given the data available.

A. Inputs

The inputs to completion algorithms are mainly partial point clouds (PPC). However, RGB images and object label information on the shape of the object, if available, may be used as input to aid the completion process. Based on the number and type of inputs used at the same time, completion algorithms can be classified into unimodal-input and multimodal-input algorithms. The classification is shown in Table II.

1) **Unimodal Inputs:** Most early works in completion used unimodal inputs of Partial point clouds (PPC). Depending on the completion approach they follow, it may be necessary for the partial point clouds to be scans of objects with standard geometrical and structural features, or for the missing regions not to be too large.

PPC with specific geometric feature: Early works in completion leveraged standard geometrical properties in shapes to guide the completion process. These completion methods are often tailored for primitive shapes such as rectangles, cubes, cylinders, etc. Some methods also explore symmetry cues

where the parts of shape on either side of the symmetry axes (the line or plane of symmetry) is congruent. An early work by Demir et al. [23] used partial point clouds of buildings as input. They segmented the input, identified the repeated structures, and used cues from the repeated segments to complete the missing regions. Similarly, Rumezhak et al. [24] process partial point clouds of symmetric objects with non-critical damage. In their context, non-critical damage refers to damages in the scan of an object where no more than 40 percent of the object is missing, and geometrical characteristics such as repeating patterns and symmetry can be inferred from the scan [24]. Some completion algorithms perform the task on specific objects. For instance, Ren et al. [25] completes scans of vehicles in real traffic scenes by using a vehicle memory bank of point cloud frames created based on symmetry and similarity of vehicles. Completion for applications in specific fields such as autonomous driving [26], [27], medical field [28], [29], agriculture [30] [31] might require point clouds from specific objects. Algorithms that require inputs with specific geometric features are applied to objects such as buildings and vehicles, which often consist of symmetric, repetitive, standard shapes. However, such features may not be guaranteed in real-world environments and geometric cues may be obstructed by noise.

PPC without specific geometric features: The development in learning-based completion algorithms allows completion processes to support a more flexible selection of inputs. Learning-based algorithms [12], [32], [33] have demonstrated successful completion on input point clouds with irregular features and up to 70 percent missing points. Such flexibility is observed in fully supervised methods that are trained on synthetic data. The performance of supervised algorithms, however, would be poor on objects whose categories are not included in the training data. This means, despite the progress made to generalise completion algorithms to different type of inputs, there are still limitations that arise from the training datasets. We discuss more details in Section V.

Multiple point clouds from the same object: Spurek et al. [34] take two partial inputs, transform them to feature vectors separately and use complimentary information from their encodings to produce a complete 3D shape. To the best of our knowledge, [34] is the only work that uniquely uses multiple point clouds for feature extraction without registration.

2) **Multimodal Inputs:** Many researchers consider multimodal inputs (multiple types of information, e.g., single-view images, semantic labels) simultaneously with the point cloud. Imaging devices are less expensive and easy to use compared to 3D-capturing devices. More detailed information can be provided by images. Cues such as geometry, symmetry, semantics, and edge features can be inferred from such data to help predict the missing parts of shapes. Multimodal inputs are often observed in unsupervised and semi-supervised methods.

Partial point clouds (PPC) + Single-view RGB images: Several learning-based completion algorithms make use of images to guide the completion. ViPC [19] uses a single-view image to obtain global structural prior information and

TABLE II
INPUT-BASED CLASSIFICATIONS

Input	Input Types	Papers
Unimodal Input	PPC with specific geometrical features	[24] [23] [25] ([26]–[31], [39])*
	PPC without geometrical features	[40]–[44], [44]–[46]
	Multiple partial input of an object	[47]
Multimodal Input	Single-view images and PPC	[35] [19] [37] [36] [38] [37]
	PPC and Shape priors Information	[48] [49] [50] [25] [51] [52]

TABLE III
OUTPUT-BASED CLASSIFICATIONS

Output	Output Types	Papers
Unimodal	Completed predicted part	[54] [41] [42] [43] [44] [45] [44] [55] [46]
	Predicted missing part	[56] [57] [58] [13] [59] [60]
Multimodal	Multiple completion results	[61] [62] [53] [63] [34]

combine it with information on local details and camera poses (viewing angles) from the partial input. On the contrary, CSDN [35] uses images as a source of intrinsic, fine-grained shape characteristics to fine-tune the generated output. Aiello et al. [36] and Zhang et al. [37] explore complementary information and exploit cross-modal data use in coarse-to-fine completion by using images as a weak supervision signal. Wu et al. [38] use 2D feature information from images combined with 3D feature information from partial point clouds for an unsupervised completion. Supervised, weakly-supervised and unsupervised approaches are discussed in Section IV.

PPC+ Shape priors information: External information such as semantic labels can improve the completion performance. Yang et al. [51] use a semantic segmentation branch that provides semantic information to the completion branch. Shi et al. [52] use temporal information in the form of a sequence of unaligned sparse inputs to enhance completion. Similarly, Ren et al. [25] use symmetry and similarity of vehicles as supplemental information. Katsen et al. [53] use a textual description of an object with PPC.

Multiple point clouds database: Zhang et al. [47] use structural point clouds retrieved from a database of complete point clouds to aid completion. The latent representations from similar structured point clouds is used as complementary features to produce a complete output.

Overall, leveraging external information to improve completion performance [25], [35] is becoming more frequent.

B. Outputs

Some recent works in completion produce other outputs such as multiple predicted completed shapes, prediction uncertainty maps, etc. These can be useful to explain and understand the completion results. We have classified such works into unimodal and multimodal outputs.

1) **Unimodal Outputs:** Most unimodal outputs are either predicted complete-point clouds (PCPC) or predicted missing parts of the partial inputs.

Predicted complete point cloud (PCPC): Early completion algorithms (eg. [40], [42], [64]) predicted the complete shape of the target object. Partial inputs are first encoded to a latent feature vector and then decoded into a complete shape. Original points and local features of the input are often lost in this process. Many of the state-of-the-art algorithms [19],

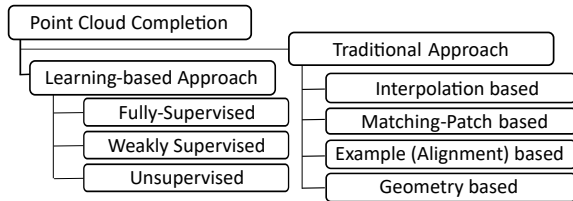


Fig. 2. Classification of Point Cloud Completion Approaches

[41], [65]–[67] that predict complete output have come up with different strategies to preserve the local features of the input which are discussed in Section VIII.

Input partial point cloud (PPC)+ Predicted missing part:

The necessity to preserve the original local details from the input point clouds led to algorithms that predict only the missing part of the input. The prediction is then combined with the input and refined to achieve more detailed results [12], [54], [56], [60], [67]–[71].

2) **Multimodal Outputs:** Point cloud completion is a challenging task. It is difficult to be certain that the predicted output completes the shape of the target object. Some works [61] recommend generating multiple outputs for explanation purposes. Wu et al. [61] first proposed generating multiple complete shapes for a single partial input. The completion is conditioned on a learned multimodal distribution of possible results. This approach was also followed by [34], [62], [63]. In addition, Cui et al. [63] produce a completed point cloud per object with uncertainty maps. The maps are consistent with human perception and lead to an explainable unsupervised completion method. Not many completion methods produce multiple outputs. However, it is our opinion that multiple outputs could add to the explainability of completion results.

IV. POINT CLOUD COMPLETION APPROACHES

Point cloud completion methods can be classified into learning-based and traditional approaches (Fig. 2) [72]. Learning-based approach uses neural networks to directly predict the complete point cloud from the partial input, whereas the traditional approach optimises the parameters of shape models to fit the partial inputs and produce a complete point cloud. Both completion approaches are discussed in detail below.

A. Traditional Approaches

There are different methods to optimise the parameters of a shape model in a traditional approach. These methods include interpolation-based, matching-patches-based, alignment-based, and geometric-based methods.

1) **Interpolation-based methods:** fill holes locally by smooth interpolation [19]. Interpolation constructs new data points from a shape model built on other known data points in the same data set [89]. Common interpolation techniques include Poisson surface reconstruction [73] and Euler Spiral [74]. Kimia et al. [74] use a curve completion algorithm to reconstruct objects occluded by other objects. Kazhdan et al. [73] proposed Poisson surface reconstruction, which creates smooth surfaces by removing noisy data. It offers a global solution that examines all of the data instead of

segmenting the data for a local fitting. Sometimes referred to as gap-completion methods, interpolation-based methods are frequently used on surface reconstruction problems. This paper does not cover surface reconstruction. Interested readers may refer to a recent survey [17].

2) **Matching-Patch searching methods:** find similar points or patches to the missing parts of the input point cloud from the data that is present, duplicate them, and fill the holes. Sharf et al. [75] propose a context-aware technique where the geometric characteristics of the given missing surface are analysed and holes in the point cloud are iteratively filled by copying patches from valid regions. The technique determines the best matching patch, fits and aligns it with the surrounding surface. 3D-PatchMatch [76], an optimisation-based searching algorithm, handles the occlusion problem. It searches for the best boundary match from the complete region. In a slightly different manner, Doria and Radke [77] fill the holes in 3D point clouds by transforming the point cloud into a depth image. Inspired by image-inpainting methods, Sarkar et al. [78] compute 3D shape parameterisation from surface patches, predict missing vertices and inpaint holes of moderate size. Users can also specify variable patch sizes.

3) **Example-based (Alignment-based) methods:** complete shapes by matching the partial input with template shape models from a large database. They are sometimes known as alignment-based methods [80]. Some retrieve the complete shape directly [80] while some retrieve object parts and then assemble them to obtain the complete shape [81]. Pauly et al. [80] retrieve suitable context shape models from a database, enclose the shape models to fit the input data and iteratively blend the wrapped shape models to obtain the final complete output. Schnabel et al. [81] detect a set of primitive shapes on the input point cloud and extend them into empty regions to guide the hole-filling. Shape primitives are implicit surfaces such as spheres, cylinders, cones, planes, etc. with a possibly infinite extent [81]. Li et al. [82] align and scale hand-modelled shape templates of similar objects from shape databases to the input point cloud. Shen et al. [79] propose a local to global bottom-up structure recovery approach that recovers parts of the scanned object instead of a complete object that matches the input. This allows them to use a small-scale shape repository. Alignment-based methods can achieve highly plausible completions, however, they assume the presence of a suitable template that can cover the missing point clouds.

4) **Geometry-based methods:** complete shapes by using geometric cues, e.g. structural property, symmetry [11] and regularity [87]. Although geometric-based methods produce good results by inferring data from partial input, they assume moderately complete inputs where the geometry of the missing regions can be inferred directly from the observed regions. This assumption may not hold true for real-world data.

Friedman and Stamos [84], Chauve et al. [85] fill holes in point clouds by exploiting the geometric regularity of urban scenes. They detect regularity by performing a Fourier analysis, and combine the results with planarities in the scene. They process a large number of points and preserve the

TABLE IV
TRADITIONAL SHAPE COMPLETION METHODS

Method Type	Advantage	Disadvantage
Interpolation-based [17], [73], [74]	Can fill small holes well	Cannot fill large missing areas, the optimisation process takes a long time.
Matching patch-based [75] [76] [77] [78]	Reconstructs shapes patch by patch based on similar patches found in the object, can cover larger holes than those by interpolation-based methods	Matching based searching can take a long time, the completion can be affected if no similar patches are found and they are sensitive to noise.
Alignment-based [79] [80] [81] [82] [83]	Achieves reasonably successful results on completing large missing parts	Relies on the presence of a dataset with templates to do the completion, which may not exist for every object. If the specific object category and type is not known, it could lead to misleading completion results.
Geometry-based [11] [24] [84] [85] [86] [87] [88]	Can infer global structure, symmetry and geometrical properties of the object and give reasonably precise completion results.	Assumes moderately complete inputs with specific geometric properties and does not perform well on point clouds of irregularly shaped objects.

structural components of the input. Rumezhak et al. [24] carry out completion on symmetric objects with no critical damage (i.e. their shape is discernible even with the missing parts). They first approximate the symmetry plane of the input using Principal Component Analysis (PCA) and construct a mirror reflection plane to the approximated symmetry plane. Then they apply a point-to-point matching registration algorithm to fill in the missing regions. Sung et al. [86] uses a data-driven symmetry-based algorithm that combines alignment and geometric-based methods. It infers global structure, symmetry axes, and planes, and uses a collection of example 3D shapes to build structural part-based priors necessary to complete the shape. Kroemer et al. [87] considered an extrusion-based technique for completion. They first search and detect planar (both linear and rotational) reflection symmetries to determine the initial options for extruded shapes and their parameters. Applicable extrusions are evaluated and chosen for completion. This method requires structured point clouds: point clouds that have one point for each pixel in a 2D grid.

Overall, traditional completion approaches require no training data and produce a reasonably convincing completed output. Their optimisation process could be slow [19] and they require inputs with certain geometry priors or cues which may not always be present. For these reasons, recent works have turned to a learning-based approach. A summary of traditional completion approaches is presented in Table IV.

B. Learning-based approach

Learning-based approaches complete shapes with a data-driven parameterised model (often a deep neural network) that directly maps the partial input to a complete shape while offering fast inference and better generalisation. Deep learning-based models can encode geometric information of point clouds in high dimensions [9] and make no prior assumptions about the shape of the object unless explicitly trained to do so. Learning-based approaches can be divided into fully supervised, semi-supervised, and unsupervised methods [69]. Most works follow a fully supervised manner and require a lot of training data and computation power [90]. During inference, learning-based approaches in general are more efficient than traditional ones as they tend to avoid the slow optimisation process [72].

1) **Supervised methods:** rely on the availability of paired training data for completing tasks. Paired data refers to a collection of partial point cloud scans and their corresponding

completed data in the form of Point Clouds, Voxels, CAD, Signed Distance Fields (SDF) etc. [65]. Collecting paired data from real-world objects is very difficult. Most supervised completion algorithms are trained on synthetic datasets such as ShapeNet [91] and ModelNet [92]. Datasets are further discussed in Section V.

Most supervised methods use generative models [93]. They are statistical models that generate new data instances. These models determine the probabilities that an instance of a certain group in the data is observed and mimic the probability distribution in a way that resembles the original data to generate new instances. In contrast. Discriminative models are usually used in classification tasks to discriminate between different instances. Although generative models are computationally expensive and less accurate than discriminative models, they can work with missing data and are suitable for point cloud completion.

The encoder-decoder is the most common architecture used by supervised methods. It was first introduced in L-GAN [94] for 3D representation learning with an application to completion. PCN then designed the first learning-based algorithm with encoder-decoder architecture tailored for point cloud completion [95] [69]. There are a large number of papers on supervised methods. Common architectures are further summarised in Section VII.

2) **Weakly supervised methods:** (also known as semi-supervised methods) use information from a small set of input-output paired data and apply it to the larger part of the data which only consists of unpaired inputs without labels. Stutz et al. [72] learn prior shape information on synthetic data and predicts the maximum likelihood fitting solutions. A weak supervision signal is provided in the form of a known object category and a known object location in the form of 3D bounding boxes. The likelihood helps derive the object reference shape from the set of ground truth shapes taken from the synthetic data. The latter signal helps zero in on the object of interest in a noisy, real-world input. Saroha et al. [96] learn enhanced shape priors via a generative adversarial network (GAN) and combine them with the prediction of a conditional Deep-SDF (Signed Distance Function) architecture. They encode the point clouds to a global latent code and then complete with signed distance. Gu et al. [97] estimate 3D canonical shapes and 6-Degrees of Freedom (6-DoF) poses for the alignment of multiple real-world point clouds that represent the same instance. These point clouds are captured

TABLE V
COMPARISON BETWEEN COMPLETION APPROACHES

Method Type	Advantage	Disadvantage
Traditional methods	Can process large, dense point clouds, does not require training data, good results for small holes and regular-shaped objects	Require the input to have specific geometrical property (symmetry, repetition etc), can be sensitive to noise [19], usually qualitative evaluations only
Learning-based	Better performance for large missing parts, do not require specific geometric shapes, quantitative evaluation available	Cannot handle dense, large point clouds due to lack of computational power and memory cost, requires a lot training data
Supervised	Better evaluation performance than weakly-supervised and unsupervised methods as ground-truth is available	Requires paired-training data (with ground truth (GT)) so usually trained on synthetic datasets
Weakly Supervised	Requires a lesser amount of paired-data than supervised methods, can make use of other information like images and shape priors	Hard to train, requires complex architecture to accommodate multi-modality
Unsupervised	Do not require paired training data	Take longer time to train, evaluation is approximated as there is no GT

from different viewpoints by the sensor with no assumption of knowledge of ground truth shape. Multiple observations of the object from several viewpoints is used as a weak supervision signal. Aiello et al. [35] fuse complementary information from weak supervision signals of an auxiliary image, a shape-prior, and a partial point cloud by using a cross-modal transformer. Fan et al. [98] learns deep semantic prior from unpaired complete and partial point clouds through a reconstruction-aware pretraining process and then apply selected learned priors to improve learning on a small number of the paired training samples.

3) **Unsupervised methods:** A complete point cloud scan of real-world objects that can serve as ground truth for training purposes is hard to collect. Researchers have turned to unsupervised methods that can perform completion without requiring paired training data [63]. Large-scale 3D scans and virtual 3D object repositories are used for training unsupervised completion algorithms.

Zhang et al. [99] use a generative adversarial network (GAN) inversion approach for 3D shapes. A GAN pre-trained on complete shapes searches for latent codes that are similar to the latent codes of the input. Similarly, Chen et al. [65] transforms the input (a real scan) to a latent representation that is indistinguishable from representations derived from the training data (synthetic) via GAN and maps the representations to a complete point cloud by setting up a min-max game where the generator fools the discriminator. Ma et al. [100] use the assistance of artificial CAD models to complete partial point clouds of real objects that consist of noise and outliers. The auto-encoding learns the basic shape of the object and uses it to fool the discriminator by minimising the gap between real-scene and artificial data. PointPnCNet [101] uses inpainting, where a portion of the input data is removed and the network trains to complete the point cloud including the missing region. PointPnCNet is trained only on partial inputs. It relies on large datasets to infer domain-specific priors. OptDE [102] disentangle partial scans into domain, shape and occlusion factors in consideration of the domain gap between synthetic data used for training and real-world data used for testing.

Cai et al. [10] create partial point clouds by occluding complete shapes at varying degrees so that incomplete shapes are represented by an occlusion code and a complete shape code in a unified latent space. They provide supervision in the form of ranking constraints assigned to the series of partial point clouds derived from one complete point cloud with varying degrees of occlusion. Wen et al. [66] use two simul-

taneous cycles of transformations between the latent space of complete shapes and partial shapes for multidirectional learning of geometric correspondences. Tang et al. [103] use an unsupervised multi-scale key point detector and a complete point cloud generator to localise aligned key points from partial inputs and complete point clouds. The key points are used to generate a surface skeleton based on geometric priors which are then refined for the final output. Wu et al. [38] use 2D images of the objects to extract 2D features and combine them with the 3D features extracted from the partial point clouds. ACL-SPC [104] uses a self-supervised adaptive control-loop framework that only uses a single partial input and no prior information. Ren et al. [25] use self-supervised completion for vehicles in real traffic scenes. All unsupervised methods mentioned so far assume a one-to-one deterministic approach that maps a partial shape to a latent code and then the code to a complete shape [63]. Cui et al. [63] consider one-to-many mapping to find prediction uncertainties by using shared parameters in the encoder-decoder for partial and complete point clouds. They use a energy-based model (EBM) in the latent space to transform the partial shape encoding into a complete one. Completion approaches have their advantages and shortcomings which are summarised in Table V.

V. COMPLETION DATASETS

The development and research in computer vision depends on public datasets. These datasets are essential for training learning-based algorithms as well as evaluation. We survey the type of datasets available for point cloud completion below.

A. Datasets for traditional completion approach

There are only a limited number of datasets used by traditional completion approaches as only example-based methods use datasets for completion. These datasets are often a collection of template shapes. Pauly et al. [80] used an annotated 3D shapes data-set consisting of scans of a lion, camel, and horse to complete a scan of a giraffe. They also used a dataset of acquired 3D shapes from the Galleria dell'Accademia museum. Shen et al. [79] use a dataset of four categories (table chair, bicycle, and airplane) to help complete scans of everyday objects. Each category of objects is segmented into 5 to 12 semantically meaningful parts. For example, a chair can be segmented into a leg, seat, arm, etc. The completion method then uses a part assembly approach where different models of a given object provide parts for the final result. To our knowledge, these are the datasets used for traditional 3D

point cloud completion. There are, however, several datasets used in surface reconstruction which may be found at [17].

B. Datasets for learning-based completion methods

Several 3D shape datasets have been used to train completion algorithms. The most commonly used ones are ShapeNet [91] and KITTI [105]. Most completion algorithms use subsets of these datasets. Synthetic datasets consist of clean point clouds of distinct objects without noise and outliers [9]. This limits the application areas of most supervised algorithms.

1) 3D Shapes Datasets:

ShapeNet [91]: ShapeNet is a large-scale repository for 3D CAD models developed by researchers from Stanford and Princeton Universities and the Toyota Technological Institute at Chicago, USA. It contains over 300M models with 220,000 classified into 3,135 classes arranged using WordNet hypernym-hyponym relationships. There are several subsets of ShapeNet dataset.

- **PCN dataset [42]** contains 30,974 shapes from 8 categories: airplane, cabinet, car, chair, lamp, sofa, table, and vessel.
- **ShapeNet-55 (ShapeNetCore) [12]** contains clean 3D shapes and manually verified categories. It consists of 51,300 shapes of 55 common object categories.
- **ShapeNetSem [106]** is a well-annotated subset of ShapeNet with 12,000 shapes of 270 object categories. The annotations consist of data on physical sizes, attachment surfaces, material compositions, weights, etc.
- **Completion3D dataset [64]** consists of 8 classes (Plane, Cabinet, Car, Chair, Lamp, Couch, Table, and Watercraft). It comprises of both partial and complete shapes, unlike the datasets mentioned above that consist of only complete 3D shapes.

ModelNet [92]: contains synthetic object point clouds. ModelNet contains 151,128 3D CAD models belonging to 660 unique object categories. It has a smaller subset ModelNet40 which was used in [92]. ModelNet40 originally consisted of 12,311 CAD-generated meshes in 40 object categories.

Multi-View Partial Point cloud (MVP) [107]: contains over 800,000 diverse partial point clouds rendered by uniformly distributed camera poses.

2) Other Datasets:

Some non-point-cloud datasets have been used by supervised and unsupervised methods to assist the completion problem.

KITTI [105]: contains traffic scenarios collected using 3D laser scanners, high-resolution RGB cameras, and grayscale stereo cameras. While the original KITTI dataset does not contain annotations, various authors have annotated subsets of the data and generated ground truth labels to suit their needs.

Matterport3D [108]: is a large-scale RGB-D dataset containing 10,800 panoramic views from 194,400 RGB-D images of 90 building-scale scenes. It consists of annotations with surface reconstructions, camera poses, and 2D and 3D semantic segmentation.

ScanNet [109]: is an RGB-D video dataset containing 2.5 million crowd-sourced views and more than 1500 scans. They

TABLE VI
DATASETS AND COMPLETION ALGORITHMS

Dataset	Papers (training)	Papers (testing)	Evaluation Metrics
ShapeNetCore	[111] [42] [64] [41]	[42] [64] [56] [12] [47]	CD, EMD
ModelNet40	[72] [71] [33] [112]	[72] [33] [112]	CD, EMD
PCN	[42] [57] [45] [58]	[58]	CD, FPD
Completion3D	[32] [113] [54] [47]	[32] [54] [113] [47]	CD
ScanNet	[65] [114] [54]	[104]	CD, EMD, F-Score
MVP	[115] [116] [59] [60] [34] [107]	[115] [116] [59] [60] [34] [107]	CD,EMD,F-Score,DCD
KITTI	[72] [69]	[117] [42] [65] [56] [66] [70] [69] [67] [104]	MMD,Fidelity,F-score,Consistency
Matterport3D	[65] [57] [118]	[65] [57] [118] [104]	CD,
ShapeNetViPC	[19] [36]	[19]	CD, F-score
Some other datasets	Pandar40 [119],3D-EPN [66],Trees [30]	BuildingNet [120], nuScenes [25], 3D-future [102]	CD
Own dataset	[11]		CD

are annotated with 3D camera poses, surface reconstructions, and instance-level semantic segmentations.

Scan2CAD [110]: is an alignment dataset based on 1506 ScanNet scans with 97,607 annotated key points pairs between 14,225 (3049 unique) CAD models from ShapeNet and their counterpart objects in the scans. These CADs are annotated with either none, 2-fold, 4-fold or infinite rotational symmetries around a canonical axis of the objects. All datasets are summarised in Table VI.

VI. EVALUATION METRICS FOR COMPLETION

A. Evaluation Metrics for Traditional Methods

Traditional methods often use a qualitative visual evaluation to assess the results of their completion. The visual evaluation compares the partial input to the completed output and, if available, outputs from other traditional methods for completeness and accuracy. The evaluation is mostly subjective and can rely on human perception wherever quantitative measures are not present. The evaluations can also be designed to observe the effects of occlusions, symmetry, completeness of the input, etc.

Among interpolation-based methods, Kazhdan et al. [73] evaluate their construction results with other methods by reconstructing the same piece of input with different methods and visually comparing them. They compare memory usage, processing time, and resolution (number of triangles) in the reconstruction. Matching patch methods evaluate how well the selected patch can fill the hole until the completion has reached a level of detail of the area surrounding the hole. Sharf et al. [75] evaluates local shape approximation by comparing the set of transformed cells to other cells in a local symmetry group. Cai et al. [76] evaluate their completion results by comparing the time cost (the time required to find similar patches/points in the missing region) to those in [77]. They use visual evaluation to compare completeness and uniformity of the results. Among example-based methods, Li et al. [82] evaluate their work by completing objects from a synthetic data set and comparing them with the ground truth. Geometric-based methods use a variety of methods for evaluation. Friedman and Stamos [84] evaluate the time of processing in relation to the scan size. Kroemer et al. [87] measure the errors of the extrusions

compared to the measurements directly taken from the objects.

B. Evaluation Metrics for Learning-Based Methods

Compared to traditional completion approaches, learning-based methods have common evaluation metrics. The evaluation metrics for completion are calculated by comparing the results of the completion and the ground-truth data.

1) Metrics for Synthetic Datasets:

Earth Mover's Distance (EMD) [121]: evaluates the dissimilarity between two point clouds by finding a bijection $\phi : D_1 \rightarrow D_2$ (where D_1 and D_2 are point clouds) that minimises the average distance between corresponding points between the partial and complete point clouds. EMD is defined as:

$$EMD(D_1, D_2) = \min_{\phi: D_1 \rightarrow D_2} \frac{1}{|D_1|} \sum_{x \in D_2} \|x - \phi(x)\|_2 \quad (1)$$

EMD requires the number of points in D_1 ($|D_1|$) and D_2 be the same. Though EMD is a popular metric for completion, it can be affected by global distribution and may overlook the fidelity of structural details [116].

Chamfer Distance (CD) [122]: also known as Chamfer Discrepancy, represents the average distance of closest point between two point clouds. Given two sets of points D_1 and D_2 , the Chamfer Distance between the prediction set D_1 and the ground truth D_2 is defined as:

$$CD(D_1, D_2) = \frac{1}{|D_1|} \sum_{x \in D_1} \min_{y \in D_2} \|x - y\|_2 + \frac{1}{|D_2|} \sum_{y \in D_2} \min_{x \in D_1} \|y - x\|_2 \quad (2)$$

where $\|\cdot\|_2$ is the Euclidean distance. Chamfer Distance is used both as an evaluation measure and a loss function for optimizing learning-based algorithms. It is the most popular metric used in completion. However, CD can be insensitive to the disparity in local density distribution and the square operation makes it sensitive to outliers [116].

Density-aware Chamfer Distance (DCD) [116]: DCD is derived from CD and considers disparity of density distributions. DCD is claimed to be stricter with structural details and more computationally efficient than EMD. Unlike CD and EMD which have unbounded value ranges, DCD has bounded value range and is not easily influenced by outliers [116].

DCD is defined as:

$$DCD(D_1, D_2) = \frac{1}{2} \left(\frac{1}{|D_1|} \sum_{x \in D_1} \left(1 - \frac{1}{n_{y'}} e^{-\alpha \|x - y'\|_2} \right) + \frac{1}{|D_2|} \sum_{x \in D_2} \left(1 - \frac{1}{n_{x'}} e^{-\alpha \|x' - y\|_2} \right) \right) \quad (3)$$

where each point $x \in D_1$ finds its nearest neighbors $y' \in D_2$ and vice versa. α is a temperature scalar. $n_{y'}$ was introduced for cases where y' is shared by multiple x s. More about DCD is discussed at [116].

F-score [123]: evaluates the accuracy of completion by means of precision and recall. Given the predicted point cloud (D_1) and ground truth (D_2), the F-Score is defined as:

$$F\text{-Score}(\delta) = \frac{2G(\delta)H(\delta)}{G(\delta) + H(\delta)} \quad (4)$$

where $G(\delta)$ and $H(\delta)$ denote point-wise precision and recall for a threshold δ . The higher the F-score the better.

TABLE VII
COMPLETION EVALUATION METRICS

Architecture	Advantage	Disadvantage
EMD [121]	Popular, sensitive to capturing detailed shape symmetry [124]	Overlooks fidelity, affected by density distribution, requires the point clouds to have same number of points [116]
CD [122], Fidelity [42]	Popular, less computationally complex than EMD [122]	Not aware of incompleteness [125], insensitive to density distribution [116], computationally inefficient [116]
DCD [116]	Density-aware, more computationally efficient than CD [116]	Not aware of incompleteness [125]
RCD [125]	Incompleteness-aware, considers outliers [125]	Computationally inefficient [124]
F-Score [123]	Measures precision and recall [126]	Lacks mutual information and ignores true negatives [126]
MMD [62]	Measures quality	Sensitive to choice of kernel

2) **Metrics for Real-world Datasets:** The above metrics are computed between predicted output and complete ground truth data. However, real-world scans often do not have ground truth. Researchers use the following alternative evaluation metrics.

Maximum Mean Discrepancy (MMD) [62]: is the Chamfer Distance between the output and the shape from a synthetic dataset that most resembles the real-world input.

Fidelity [42]: measures the preservation of input data. It calculates the average distance from each point in the input and the corresponding nearest neighbor in the output.

Consistency [42]: aims to estimate how consistent the model's outputs are against variations in the inputs. Consistency is the average CD between the completion outputs of the same instance in consecutive frames.

Region-Aware Chamfer Distance(RCD) [125]: is a variation of CD proposed for self-supervised algorithms. It is designed to be aware of both observed and unseen regions and evaluate CD only for observed regions.

MMD, Fidelity, and Consistency are variations of CD. The formula above can be used for computing them [42]. There are some evaluation metrics that are not covered in this survey in detail because they are not commonly used. Readers are referred to [127] for further details. The pros and cons of common evaluation metrics can be found in table VII.

VII. LEARNING-BASED COMPLETION NETWORK ARCHITECTURES

Network architectures are the core parts of learning-based completion approaches. Fei et al. [18] presented a recent survey on point cloud completion that classifies algorithms based on network architectures into point-based, view-based, convolution-based, graph-based, GAN-based, transformer-based, and Variational Autoencoder (VAE)-based methods. This survey follows a similar classification but introduces a few rearrangements. First, we believe that view-based methods refer to the use of additional input but not a new architectural design. So we exclude them from our classification. Second, we group GAN-based, Diffusion model-based and Variational Autoencoder-based (VAEs) networks under one subsection as they are types of generative models. Finally, our survey includes more recent publications from 2022 to August 2023 that have not been included in the survey by Fei

TABLE VIII
COMPLETION NETWORK ARCHITECTURES

Architecture	Advantage	Disadvantage
Convolution-based	Allows the use of 3D CNNs on point clouds by structuring unordered points to 3D grids or voxels	The voxelisation of the point clouds to 3D grid or Voxel format, has a high memory requirement for high-resolution output, can be time-consuming and computationally expensive
Point-based	Enables the processing of raw point clouds treats points on an individual basis, and maintains permutation invariance property	Leads to loss of local features because it overlooks geometric relations between neighbouring points
Graph-based	Are better at considering local geometric structural details	It is difficult to come up with dynamic operator that can deal with variable sizes of graphs
Generative-model based	Can be trained on a limited amount of paired data	Are hard to train because both the generator model and the discriminator model are trained simultaneously in a game
Transformer-based	Allows better capturing of local and global features due to the attention mechanism and its ability to extract long-range information	Has too many parameters and massive linear operations that incur high memory consumption and computational cost
Other-methods	Non-generative approach that does not require the encoding of point clouds and does not rely on feature extraction	Not many constraints that can be used for optimisation and measuring

et al. [18]. We briefly summarise their common architectures. Interested readers can also refer to [18] for more details. In Table VIII, we summarise the advantages and disadvantages of these architectures. Table IX further survey and classify these works according to their architecture types and learning methods.

A. Convolution-Networks (3D CNNs)

Convolution-Networks were mostly used by early completion algorithms. Due to the unordered nature of point clouds, convolution-based methods usually require pre-processing to transform point clouds into voxels or 3D-grids (called *Voxelisation* [128]). Voxelisation leads to a loss of local geometric information. The voxel size also affects the resolutions. High resolution (with smaller voxel size) requires additional memory and compute time [18].

B. Point-based Networks

Point-based networks directly process raw point inputs whilst maintaining the permutation invariance property of point clouds. Point clouds are processed by Multi-Layer Perceptrons. Max-pooling is applied to aggregate the global feature. Local geometric information among neighboring points is lost during max pooling. Several methods capture local and global features simultaneously (discussed in Section VIII). Point-based Networks tend to follow an encoder-decoder based, end-to-end, coarse-to-fine generation process. Point-based Networks with folding-based decoder [129], [130] are also included in this group.

C. Graph-based Networks

Graph-based networks treat points in point clouds as vertices of graphs and generate edges between neighboring points. The graphs are then processed by graph convolutions [131] and gather spatial information from neighboring points. Graph-based methods are good at capturing local structural details and relationships. Graphs however are hard to analyse as they have variable sizes and dynamic forms. Developing pooling operators that can maintain CNN's characteristics of weight sharing while dealing with dynamic forms can be difficult [18].

D. Generative-Model based Networks

Generative models generate data based on the data they are trained on. Generative models include Generative adversarial network (GAN) [132], Variational-autoencoders (VAEs), and diffusion models.

1) *GAN-based networks*: GANs are often used in unsupervised completion algorithms. GAN-based networks generate new data with the same properties as the input data with a generator network and try to fool the discriminator. GANs are hard to train because both the generator and discriminator have to be trained simultaneously [132]. The training process can be affected by the unordered nature of point clouds as it will be difficult for the discriminator to compare the predicted output and input.

2) *Variational-Autoencoders (VAEs)*: are autoencoders whose training is regulated to avoid overfitting and optimize the generative capability of the encodings. VAEs are probabilistic generative models that encode the distribution of the given data instead of encoding the input as a single point. The generated shapes in VAEs can be controlled by manipulating the latent representations. There are many design types of VAE-based models. However, the quality of the output may be affected by noise injections and imperfect measurements. The outputs are not always smooth and are of a lesser quality compared to GANs [9].

3) *Generative Diffusion Models*: are models that learn by deconstructing the training data, adding Gaussian noise and recovering the data by reversing the noising process [133]. After the training, diffusion models generate new data from a random noise input through the learned denoising method. Diffusion models have been found to perform better than GANs in image processing tasks [134]. There is a growing interest in the use of diffusion models for point cloud completion. However, the unordered nature of point clouds presents a difficulty for diffusion models [133]. Zhou et al. [133] overcomes this problem by combining point-voxel representation of 3D shapes with diffusion models. The resulting Point-Voxel Diffusion Model (PVD) can perform both generation and multimodal completion.

E. Transformer-based Networks

Transformers were applied to computer vision tasks after their success in the Natural Language Processing (NLP) field. They are very good for processing irregular data [135] and have advantages like representation learning and attention mechanism. It is the trending choice of architecture among the latest completion algorithms. However, the high number of parameters and steps for comparison often require large computational resources to train.

TABLE IX
COMPLETION APPROACHES AND NETWORK STRUCTURES

Architecture Type	Learning Methods		
	<i>Supervised</i>	<i>Weakly-supervised</i>	<i>Unsupervised</i>
Point-based	[13], [40]–[44], [44]–[46], [55], [64], [136], [137]	[63], [72], [97]	[101] [102], [104], [125]
Convolution	[19], [33], [58], [138]	[36] [139], [140]	[10] [35], [141]
Graph-based	[57], [119], [137], [142]	[96]	[118]
Generative model based	[10], [56], [57], [118] [53], [143]–[146]*, [28], [140], [147]	[96]	[65], [66], [99], [100], [148], [149]
Transformer	[12], [54], [112], [150] [14], [29], [151]–[153]		[50]
Other	[32], [113] [50]		

F. Other Networks

There are a few papers in completion that do not use any of the network architectures mentioned above. They usually redefine the completion as a different type of problem. One approach is treating completion as a point displacement problem [32], [113]. Another approach is using symmetry shape completion. Details about these architectures are discussed in Section VIII.

VIII. TECHNICAL CHALLENGES IN COMPLETION

Regardless of the completion approaches, researchers often face challenges that arise from the properties of point clouds. Understanding the source of these challenges can help researchers to develop a better solution. We group these challenges into *Noise and Outliers*, *Unordered structure*, *Lack of Paired Data*, *Density Issue* and *Effective Feature Representation*. We then discuss existing solutions for these challenges.

A. Noise and outliers

Point cloud data of real-world objects acquired from sensors and cameras is often scattered, contains noise, and outliers [154]. This is due to sensor quality, lighting conditions, measurement noise of the sensors, and other environmental factors. Moreover, the characteristics of the measured object, measurement method and condition can affect the result. In our context, noise refers to points in the point cloud that do not belong to the target object. For example, LIDAR data from autonomous vehicles captures surrounding scenes like the road, other cars, and moving objects that may occlude the view. In semantic scene completion, this extra information is useful, since the goal is to complete all objects in the scene. However, in object completion, where only a single car is being completed, points belonging to other objects would be considered noise.

Outliers refer to the dataset points in the scan that deviate (are placed away) from measured objects [9]. They often arise due to environmental conditions and device properties. For instance, instability of a scanner - such as its mechanical structure, rotation, and movement (eg. LIDAR sensors mounted on autonomous vehicles) can cause a difference between the location of the initial signal targeted at the actual object and the echo/reflection at the collection point [18]. The captured data will have outlier points that belong to the given object but have the wrong measurements. Noise and

outliers directly influence point cloud processing results such as registration, feature extraction, and completion. They can skew the optimisation processes in traditional approaches and affect feature extraction in learning-based approaches.

Many existing works assume clean point clouds and avoid the noise issue altogether. There are limited works in completion that process noisy data. Agrawal et al. [155] pass noisy inputs through denoising autoencoder and then process with a network trained on clean data. Arora et al. [62] perform noise tolerance analysis by adding Gaussian noise to the inputs. PDCNet [156] uses density-based clustering algorithms to reduce noise. Ma et al. [114] uses a segmentation task to define the boundary of the object and its parts. Any point outside the boundary can be removed by Farthest Point Sampling (FPS) and thus a cleaner output is made. Li et al. [117] use an outlier removal process to suppress outlier points from sparse point clouds during the patch-training process.

B. Unordered structure

A point cloud is a collection of 3D points. Each point is a 3D coordinate measurement, which is put together with other individual point measurements to make the complete cloud [154]. Point clouds differ from other 3D data representation formats such as RGB-D images, meshes, and volumetric grids because Point clouds are irregular and unordered [157]. This means regardless of the order in which the points in the point cloud are captured, the final point cloud remains the same. This property is known as *Permutation Invariance*. The same X number of points in a point cloud can be represented by $X!$ different number of permutations (or arrangements) in the storage [158]. Permutation Invariance does not have much effect on traditional methods as they do not deal with point clouds on a point-to-point basis. Permutation-invariant inputs can only be processed by permutation-invariant networks with symmetric functions. Thus, learning-based methods are affected by it. Some deep-learning systems like convolutional neural networks (CNNs) are unable to adapt to sudden re-ordering of sensory inputs unless the model is retrained [159]. Thus, despite their success in 2D tasks, CNNs cannot be used on point clouds directly as they are not permutation invariant. To handle this challenge, two common ways are used in the literature. The first is to transform the point clouds into structured data like 3D grids or voxels. The second method is to use permutation invariant networks.

1) **Restructuring point clouds:** Pre-processing point clouds into 3D grids or voxels can give order to point clouds. Early learning-based methods often follow this direction. Dai et al. [160] introduced a 3D-Encoder-Predictor network (3D-EPN) that encodes both known and unknown spaces in the data into a latent representation to predict global structure at unknown areas with high accuracy. The result is matched to a shape in a shape database that serve as constraints. The authors do not explicitly state the input to be partial point cloud data, but it can be assumed that partial point clouds that have been preprocessed (voxelised) can be fed to 3D-EPN's convolutional networks and be completed. Han et al.

[161] use a global structure inference network with a 3D fully convolutional module that uses volumetric information in the input to further enrich the global structure representation.

Voxelisation often leads to a loss of information as individual points are quantized to 3D boxes with a lower resolution. Voxelisation does not consider local spatial relationships before clustering the points into voxels. Xie et al. [58] use 3D grids as intermediate representations to regularise unordered point clouds. They try to lower the loss of information by designing a novel Gridding Network that learns context-aware and spatially-aware features. Similarly, Wang et al. [69] use a voxel-based network to recover realistic structure details and avoid over-smoothing of fine-grained details. They first embed the point clouds into regular voxel grids and use hallucinated shape edges to help complete output. Guo et al. [70] propose carving 3D blocks that contained uniformly distributed 3D blocks modeled after the input point cloud. Wei et al. [162] proposed a sampling model that can represent a 3D shape as a compact 1D array of depth values giving structure to the unordered set of point clouds. Wang et al. [163] proposed octree-based CNNs with U-Net-like structures for completing incomplete shapes. They use output guided skip-connection to preserve the input geometry.

2) **Permutation invariant networks:** are networks that can process order-less inputs where the sequence of the input does not matter. The first permutation invariant network for point cloud was PointNet [164]. Although PointNet is designed for classification and segmentation tasks, it has become a very important architecture in many point cloud processing tasks. PointNet applies MLP (MultiLayer Perceptron) module and a symmetric function to each point in the point cloud to extract global features. However, the max-pooling operation disassembles the point-wise features and ignores the local neighborhood in 3D space. PointNet also assumes that points are uniformly distributed which is not the case with real world data [69]. To alleviate this problem, PointNet++ [165] introduced a hierarchical neural network that samples the local subset of points with farthest point sampling (FPS) and feeds it into PointNet. DGCNN [166] extends PointNet by applying the edge convolution neural network practical operation (EdgeConv) and uses it to get edge features which better aggregate each point and matching edges connected to adjacent pairs. The number of points in point-cloud in PointNet and DGCNN remain fixed throughout the process. PointNet++ downsamples point clouds using FPS. It can be slow when generating overlapping points by using K-nearest neighbour search (KNN). Regardless of their shortcomings, PointNet, PointNet++, and DGCNN now serve as backbones for various completion networks [42], [64], [129] etc. Several researchers use different techniques to deal with these shortcomings that are discussed in Subsection VIII-E.

C. Lack of paired data

One of the most challenging issues in learning-based completion is the lack of paired datasets of real-world objects to be used for training. It is almost impossible to capture clean,

complete data of real-world objects to be used as ground truth [9]. We observe that there are two common solutions for this issue so far. The first is to manipulate synthetic data to simulate real-world data in a supervised setting. The second is to use the assistance of external information in a weakly-supervised and unsupervised setting.

1) **Simulating real-world data:** The following methods are used to produce inputs that resemble real-world datasets.

Back-projecting depth Images: PCN [42], TopNet [64] create the ground truth by sampling 16,384 points uniformly on the mesh surfaces. The partial point clouds are generated by back-projecting 2.5D depth images into 3D. This is done in order to make the distribution of the input close to real-world data. Liu et al. [41] take CAD models from ShapeNet, normalise, and uniformly sample 8,192 points on the surface to form the complete point cloud. The partial point clouds are created by randomly sampling 50 camera poses and lifting the 2.5D captured images into 3D partial point clouds. It mimics a 3D acquisition pipeline by obtaining different views in real-world applications. The partial point clouds are unified with a set of 5,000 points which is obtained by randomly dropping and replicating points after denoising the complete ones. Back-projecting is a method used by many benchmarks such as the PCN [42] and Completion3D [42] etc. Back Projection is a method of how well the pixels of a given image fit the distribution of pixels in a histogram model.

Removing points from a random centered point cloud: Huang et al. [56] take different objects from 13 categories of Shapenet-Part, center and normalise the coordinates of the shapes. They make the ground-truth points by sampling 2,048 points uniformly from each shape. The partial point clouds are created by a random selection of a point on the point cloud as a center and by removing points with a certain radius from the center. The partial point clouds are created to contain 25 percent less points than the original points.

The methods above do a reasonable job of simulating partial point clouds from real-world scenes. However, the performance of learning-based algorithms is still quite poor [12] on real-world datasets. Synthetic data cannot fully represent real-world data. Firstly, real-world data often contain noise, which is challenging to properly simulate. Additionally, the data can exhibit varying densities within a single scan, resulted from the use of different scanners. Sparsity may also arise due to occlusions during acquisition. Unlike synthetic datasets where the world origin is well-defined, the same assumption cannot be extended to real world scans, which complicates computation. Lastly, different training datasets may exhibit varying data distributions. This affects completion performance. There is a large domain gap between synthetic data and real data. Most current methods often overlook these challenges when attempting to replicate the probability distribution of real-world data in training datasets. Some works [63], [99], [167] fine-tune the decoder trained on synthetic shapes according to real scans, while others [102] introduce special representation for the inputs. Some recent techniques are beginning to solve the noise and varying density problems. These techniques are

discussed below (in VIII-D **Density**).

2) *Using limited paired data with external information:*

Not all learning-based methods need paired data for training. Many weakly-supervised and unsupervised methods use a small amount of paired data and external information such as single-view RGB images, shape priors etc. to assist the completion. Readers are referred to earlier discussions on weakly supervised (IV-B2) and unsupervised techniques (IV-B3).

D. *Density*

Point cloud density refers to the number of point coordinates available per unit area. It can be a measurement of the acquisition resolution of the scanner. Point clouds are often captured at different densities based on the sensors and the environment. Real-world scans of objects often have a high density of points. The density distribution of the points may not be uniform. We look at the density problem from two perspectives. The first is the challenge that comes with having high/varying-density inputs. The second is the expectation to have a uniformly dense output.

1) *Input point cloud density:* Real-world scans often consist of millions/billions of points that represent object surfaces. This poses more problems for learning-based completion approach than it does for traditional one.

Traditional approaches can benefit from dense inputs. For instance, interpolation-based methods benefit from a dense input as the neighbouring points are used to assist the filling of the small holes from the input. Matching-patch searching methods could be slowed down by dense point clouds if the search is on a point-by-point basis [77] instead of a patch-by-patch [76] search. For geometric methods, dense inputs help the inference of geometric characteristics from the inputs. We do not observe much discussion about the effect of dense inputs for example-based methods so far.

Density can have a negative effect on learning-based approaches. The higher the resolution, the greater the number of points to be processed, leading to higher computational cost and time. Most state-of-the-art algorithms use downsampling as it can reduce the need for subsequent convolution layers, reduce memory consumption, and sometimes increase robustness. There are several ways to downsample point cloud data. For instance, PointNet++ [165] downsamples points using farthest point sampling (FPS), whereas folding net [129] uses graph-based max-pooling that filters maximum features over every node's locality using a pre-built KNN graph. Downsampling can be a challenging problem as important features could be lost.

2) *Output point cloud density:* Uniformly dense point cloud data is beneficial in downstream applications such as automatic grasping in Robotics [4], supporting maintenance in urban structures [2] etc. Non-uniform point clouds can lead to misinformation [4]. For instance, surface texture can not be properly captured without proper density distribution of the point clouds on the surface of the given shape. In order to avoid this, researchers have come up with a refinement method to generate a uniformly dense output point cloud.

Iterative Refinement: Most completion algorithms use a coarse-to-fine generation approach where a coarse skeleton of the missing point cloud is predicted first and the details are filled in later for a dense and uniform point cloud. By Iterative Refinement, we are referring to the last step in most generative completion approaches that produce fine-grained output from the predicted output point clouds. The refinement processes can be built in as part of the architecture design, or as a separate step at the end of a completion task.

There are several refinement methods. For instance, PF-Net uses multi-scale generation by using a Point Pyramid Decoder (PPD) that predicts the skeleton center points followed by fine-grained details from different layers. Wang et al. [57] use a conditional iterative refinement sub-network to generate dense point clouds with the help of a feature contraction-expansion unit that progressively refines point position by up-sampling the point sizes by a factor of two. Lyu et al. [67] produce a uniformly dense output by a dual-path architecture network that extracts multilevel features from the input and manipulates spatial locations to produce sharp and smooth details. It also consists of a Conditional Generation Network (CGNet) and a Refinement Network (RFNet) that establish one-to-one mapping with the ground truth for an optimised completion.

SnowflakeNet [54] uses point deconvolution which is a snowflake-like growth of points where parent points progressively generate child points by splitting patterns. The process is supported by a skip transformer that ensures the deconvolution is constrained within the required shape. FBNet [168] refines the output by rerouting high-level information from the coarse output predicted by a Hierarchical Graph-based Network. It feeds the completion result back to the network so that the network can further enhance fine-grained features. Choe et al. refinement network [169] denoises, densifies, and completes point clouds by using a reconstruction network that takes in voxel inputs. It identifies outliers by using a sparse convolution layer transforming the voxels back to point clouds. Shi et al. [119] use a graph-guided deformation network. It generates new points as intermediate controlling and supporting points and for later refinement.

E. *Effective Feature Representation*

Feature extraction is a crucial step in learning-based point cloud completion. It is often the first step where the point clouds are encoded into latent features, and later decoded to complete point clouds. Effective features have a direct effect on the quality of the completion. In this section, we discuss techniques in feature extraction. Broadly, we classify them into *Detailed local features Extraction*, and *Local and Global features Extraction*.

1) *Extracting detailed local features:* Local features refer to features that can be inferred from different parts of a whole object such as texture, edge information, curvature, orientation, etc. Global features refer to features that are observed from the overall regions of the object such as shape descriptors, arbitrary scale, pose, etc.

Parametric surface features: To better represent the underlying surface, researchers capture more descriptive features in the form of surface parameters. AtlasNet [40] map a set of squares to the surface of the 3D shape and encode them along with point clouds to a latent representation. It represents and completes a shape as a collection of surface parameters in terms of local square planes (one square one point). Liu et al. [41] predict complete but coarse point clouds with a collection of parametric elements. These elements characterise the underlying geometry and include roughness, curvature, surface area, tangent planes, normal vectors etc. [170].

Intermediate representations: The global latent vector encoded by point-net-based networks often lacks explicit encoding for local features and thus leads to a lack of local structural details when decoded. Researchers come up with different intermediate representations (patches, seeds, proxies, skeletons, spots etc.) and encoding that contain both global and local features. SeedFormer [95] uses new shape representations known as Patch Seeds instead of using a global feature vector. Patch seeds contain both general structures and local patterns. Spatial and semantic relationships among neighboring points are leveraged with the help of an Upsampling transformer. PoinTr [12] changes unordered point clouds to point proxies (feature vectors encode both the features and local regions of point clouds). It uses geometry-aware transformer blocks to learn structural knowledge and preserve the details. Jiang et al. [171] splits the incomplete input into patches and mask them to learn both local features and high-level contextual relationships between the patches and their local representations. Patchnets [60] uses a mid-level patch-based representation for better generalization capability and feature extraction. LakeNet [103] uses a Surface-skeleton generated from key points as an intermediate representation. The skeleton helps leveraging information in key points to construct a finer reconstruction. AXformNet [172] first generates points in an intermediate space with a fully connected layer and then aggregates them to form the objective shape. This allows for faster processing as points, already generated, can be moved to their proper positions. CompleteDT [173] takes several point clouds of varying resolution sampled from the same input point cloud and changes them into "spots". The spots are fed to a multi-resolution point fusion module that transforms them into a complete point cloud. Anchorformer [151] uses pattern-aware discriminative nodes for to capture regional information. ProxyFormer [153] uses point proxies with feature and position information to encode both the partial and missing regions.

Blending the input with predicted missing part: Processed point clouds often lose local details. Some researchers preserve the local details by directly using the input and only predict the missing part. The input and predicted point clouds are then stringed together [56], [174]. Medoza et al. [175] combine the input and the predicted output with a merging refinement network. Liu et al. [41] create a coarse-grained output with parametric surface elements via a morphing-based decoder. They then merge the coarse output with the input by a novel sampling algorithm. DeCo [176] blends the predicted missing

partial point cloud with the input by paying attention to its surrounding region and reconstructing the frame of the completed shape. ME-pcn [177] preserves topology consistency and surface details by leveraging emptiness in 3D shape space, unlike all other methods that encode only the occupied regions.

Completion as an optimization problem: Some researchers reformulated completion as an optimisation problem as the reliance of encoding-decoding process may not give the best results. PMPNet [32] [113] reformulate completion as a point cloud deformation problem where each point in the incomplete point is moved to the completed output in the path where the total distance of Point Moving Path (PMP) is the shortest. It is modelled after the Earth Mover Distance (EMD) [121]. Each moving path is unique for every point, creating a point-to-point correspondence. This preserves the local features. The authors later [113] improved the ideas by using transformers instead of the convolution-based encoder-decoder network. Zhang et al. [178] reformulate completion as a point displacement optimisation problem. They make use of the camera model and cast rays to the object as lines of sight. Points are moved to their goal location along the line of sight and under the constraint of local movement for a refined shape.

2) *Simultaneous Extraction of Local and Global features:*

Network Design-instigated Feature Extraction: Unlike most works that rely on max pooling to extract features from unordered inputs, some algorithms have come up with different pooling strategies to capture both global and local features. SoftPoolNet [33] extracts features based on their activation. Instead of pooling only the features with the highest activation, they take into account multiple high activation features. Soft-Pooling process, in combination with regional convolutions and refinement leads to more accurate, dense outputs. Wang et al. [179] design a neighborhood pooling process that adopts feature vectors with the highest activation. It minimises the loss of individual feature descriptors. SA-net [13] uses a skip-attention mechanism that selectively conveys geometric information of local regions. The features are then decoded by a hierarchical, structure-preserving decoder that uses the geometric information to progressively detail the local regions. DCTR [152] uses skip-attention to bridge local region features in the encoder with point features in the decoder.

Multi-level features extraction: PF-net [56] uses a multi-resolution encoder with Combined Multi-Layer Perception (CMLP) to extract multi-layer features from partial point clouds. Zhang et al. [59] use two separate feature extractors, global and local feature aggregation (GLFA) and residual feature aggregation (RFA), to express the two kinds of features and reconstruct coordinates from their combination via a dual-path architecture network. The residual features are calculated by computing the difference between the global and local features. CP3 [50] takes an NLP-inspired Completion by a generic Pretrain-Prompt-Predict paradigm (CP3). They use a self-supervised pretraining process followed by a prediction stage that makes use of a Semantic Conditional Refinement (SCR) unit and is finalised by the multistage refinement stage.

Multi-path network architectures (completion and recon-

struction): Han et al. [161] propose an architecture with two sub-networks; a global structure inference and a local geometry refinement network. The global inference network extracts global feature information from multi-view depth information provided as part of the input. In addition, it guides the local geometric refinement network which takes local 3D patches around missing regions for effective completion. The training of both sub-networks is done simultaneously in an end-to-end manner.

VRCNet [107] use a variational relational completion network that performs probabilistic modeling from two separate network paths. One path reconstructs complete point clouds from a complete input, and the other generates complete point cloud from a partial input. Hyperpocket [34] splits the point cloud processing into two disjoint data streams of partial input and “pockets”: empty spaces left by the missing part of the objects. Li et al. [117] propose to decode a low-resolution point cloud first and perform path-wise noise-aware up-sampling and recover the details patch by patch. More works that use parallel, dual-path networks include [180] [181].

Attention-assisted: Attention mechanism was proposed to improve the performance of encoder-decoder architectures and allows the decoder to utilise the most relevant parts of the input sequence in a flexible manner. It compares the weighted combination of all the encoded input vectors, and uses the most relevant vectors being attributed the highest weights. Attention is an important component in transformers for completion. Most recent completion algorithms use transformers to improve the feature extraction. Traditional CNN and other graph techniques focus on short-ranged relationships and require deep architecture to handle long-ranged relationships. Guo et al. [90] uses cross-attention and self-attention mechanisms in a neural network to treat point clouds in a per-point manner and establish short-range and structural relationships among points. VRCNet [107] uses self-attention to exploit relational point features and refine local shape details on a coarse output. Wen et al. [13] use skip-attentions to selectively convey information from local regions of a point cloud. Zhang et al. [182] use cross attention and self-attention layers to explore relationships between global shapes and local patterns.

IX. POINT CLOUD COMPLETION AND OTHER TASKS IN POINT CLOUD PROCESSING

During our survey, we observe that many point cloud processing tasks benefit from point cloud completion.

Scene Completion networks use completion and segmentation networks jointly to improve performance [183] [184] [185]. Similar idea is observed in semantic-instance completion [186] [187]. **Object Detection**, e.g. [188] [189] incorporate completion mechanism in their network to enhance spatial information. Tsai et al. [190] use completion to neutralize scan pattern discrepancy in traffic-scene lidar scans between different datasets that cause detection algorithms to overfit on the data they are trained on. **Segmentation** tasks [191] [100] can benefit from shape priors learned from completion networks and vice versa [51], [114].

Point Cloud Pre-training Wang et al. [192] generate masked point clouds with view-point occlusions and use completion to reconstruct them. Yu et al. [193] also mask out patches of the input point cloud and allow the backbone transformer to complete them. Completion was found useful in learning representations as demonstrated in [56] [194].

X. DISCUSSION

Point cloud completion is an essential task in computer graphics and vision. We discuss the many factors to consider when choosing, using, and making a completion algorithm. From our survey, we observed that a good completion method should show good qualities in Generalisation, Accuracy, Resolution and Robustness.

Generalisation refers to the ability of the completion algorithms to complete real-world or synthetic point cloud scans of a large category of objects. It also refers to the ability to complete a large range of missing parts in the input point clouds. Both traditional and learning-based completion approaches require inputs with certain characteristics (see Section IV). Traditional methods often require inputs with only small holes [73], [74] or with standard geometrical shapes [11], [23], [87] etc. Learning-based algorithms are limited by the datasets they are trained on.

The type of output from completion processes can vary according to the downstream applications it is required for. For example, autonomous driving and augmented reality require different precision and resolution in the output. Researchers should think about preprocessing tasks (such as denoising, voxelisation, etc), resolution power (how much local-structural detail is necessary), preservation of the inputs, uniform density in the output, etc. The quality of outputs in completion can be measured by Accuracy and Resolution.

Accuracy: refers to the correctness of the completed result. This may be seen through qualitative visual results of the completion output or through quantitative evaluation measures such as F-score, CD etc. Current research in point cloud completion is heavily focused on this area. Most papers [12], [19], [33] are working on improving accuracy levels on completion on both synthetic and real-world datasets. Users are recommended to consider the best performing algorithm in the given input area when selecting completion algorithms.

Resolution: The required output density of the completion result depends on the downstream task (section VIII-D). Traditional approaches usually strive to fill the missing regions with similar density to the rest of the point cloud. Existing learning-based approaches often produce a point cloud up to 16,384 points [12], [42] (so far) due to computationally manageable training and evaluation process. While this may be sufficient for synthetic input data with smooth surfaces and regular geometry, it does not represent real-world scans of objects with features like surface texture, multiple-edge information and irregular shape geometry well.

Robustness: It is important to take into account the computation power and time needed to train the models. Faster computation can make possible several applications such as

live scanning and completion, navigation, etc. The algorithms we choose should be able to complete noisy and sparse inputs that may be captured in hazardous environments such as deep seas and unknown environments.

XI. FUTURE WORK

In this section, we discuss some observations and summarise possible future research opportunities.

Approaches: We observe that both traditional and learning-based approaches have their own pros and cons (Section IV). However, the two approaches could be used simultaneously and complement one another. For instance, for high-density point clouds of objects without standard geometric features and a large missing part, a smaller model of the input with lower density may be extracted and processed via a deep-learning-based method. The result can be re-scaled and made denser with the assistance of traditional methods for a fine-grained finish. In addition, learning-based methods could be integrated with works such as [195] [83] from traditional methods and techniques from deep-learning like feature learning [71], [196] and acquisition techniques [197]. So far, there are only a few [12]–[14] learning-based methods that try to use features from traditional approaches. Adapting advantages from both approaches and tailoring them for specific scenarios would be an important direction.

Datasets: State-of-the-art algorithms often improve and evaluate completion performance on common benchmarks which are derived from existing 3D shape datasets (see V). While this offers an objective comparison ground among algorithms, it reduces the generalisation potential of the algorithms for real-world and unseen data. There are a limited variety of objects in the current benchmark datasets. In addition, the ground truth data in benchmark datasets [42] have a maximum of 16,384 points. This does not encourage researchers to attempt completion on denser point clouds. It is necessary for the community to create new completion benchmark datasets with more variation of objects [198], [199] and denser ground truth data. We encourage works like [200] that evaluate the suitability of point cloud datasets for completion tasks. We observe that recent algorithms train on synthetic datasets and perform tests on the real-world datasets. There is limitation in the current ability to provide large real-world training datasets as they are costly to acquire. However, it is beneficial to have as large a training dataset as possible. We recommend researchers to consider datasets in Table V and seek ways to augment them with noisy samples, missing points, density variations etc. This can aid in model training and reduce the acquisition cost of new data. We also recommend researchers to use radar scans [25], temporal data [52] etc. Datasets from different real-world environments like aerial scans from drones, 3D scans of living things [80], [201], and subsea data [202], [203] from underwater vehicles could be useful for completion. Specifically, there are no real-world underwater datasets publicly available, indicating a largely unexplored research area. Subsea exploration is an important area that supports subsea infrastructure maintenance (e.g., offshore wind turbines, oil platform, undersea cables,

shipwreck exploration) with different kind of challenges (e.g., low-light, marine biological growth).

Real-world datasets may encounter the issues mentioned in Section VIII-A. When handling noisy datasets, researchers have two options. The first is to apply denoising algorithms [204] [205] and complete it with completion algorithms. The second option is to feed the noisy data directly to the completion algorithms that have the ability to deal with noisy data such as [114] [117]. We find that there is limited research comparing the effect of the two approaches on completion results. Complementing the advantages of the two would also be an interesting future work. In addition, real-world datasets could be dense and may not be able to be processed by learning-based algorithms unless downsampled. The down-sampling algorithms and their effect on completion may be studied by performing multiple tests. Learning-based point cloud downsampling strategies (e.g. [206]) would also be considered [204] [207]. Evaluation metrics that can measure the effect of noise and density distribution could be useful to analyse the completion results with respect to the inputs.

Assistive Tasks and Feature Representation: During our survey, we found a few works that take assistance from other point clouds tasks such as segmentation [51], [98], [114], generation [167], upsampling [150], [208], classification [72], and object detection [7], [209]. Completion also benefits from the use of multitask point generation models [143], [145] and multitask feature learning [210]. There is not much work in multitask-learning in point cloud completion. Zhang et al. [211] claim that existing completion methods lack the ability to accomplish completion jointly with other tasks. In Section VIII-E, we observe that early works focus on local geometric properties while more recent works consider global, local, spatial, structural and contextual information. Leveraging external information from images and semantic information [35], [97], [212] etc. is also becoming more frequent. Contextual cues and insights from video processing and multi-modal search databases, and gaze and saliency information obtained from expert users may be useful in making the completion process more context aware. The understanding of semantics in complex noisy scenes and the modelling of distractors [213] for better completion can be benefited from other computer vision domain. The modelling is closely related to the architecture design.

Architecture: There are a lot of possibilities with regard to network architectures. We see a growing use of transformers at different places in the completion process [13] [113] [12]. Although transformers are quite successful in capturing both long- and short-range spatial relations between points, they are slow to train. Techniques to speed up the transformers [214] could improve the completion results. Following the use of multimodal inputs such as images with point clouds [19], [35], the use of self-attention and cross-attention had become more frequent. Multipath networks that use complementary intermediate results to assist the completion are also becoming popular [13], [161]. These networks enable the intelligent integration of additional knowledge to assist completion.

Apart from GANs, new generative models (e.g., denoising diffusion [53], [133], [143]–[146], probabilistic VAEs [34], [107]) are showing promising results in other domains. They may inspire new point cloud completion research. A fair and comprehensive performance comparison of different architectures and learning-based methods (supervised, weakly-supervised and unsupervised) under the same testing conditions could be an interesting future work.

Human-in-the-loop: When the completion results are critical to downstream application or analysis (e.g. infrastructure maintenance, path planning), understanding the reliability of the completed regions becomes essential. In the literature, we observe only a few works [34], [61]–[63] (Section III Multi-modal outputs) that use extra outputs to explain the completion results. Perhaps more research can be developed for model explainability. Most existing techniques evaluate on public datasets. There is little research that integrate humans in the loop or takes human input on the fly. Some future research questions may include: *How to visualise the uncertainty in completion algorithms? How to integrate human's specific inputs or expert judgement? How to offer fast corrective completion through iterative, interactive or real-time update (e.g. through remote acquisition by drones/autonomous vehicles)?*

Hardware and Training Issue: We observed that most publications specify the type of equipment (e.g., platforms, graphic cards) used during training. However, the training times are usually not specified. Only a few works [139], [152] compare the size of the model (number of parameters) and the inference time. Finding a range of preferable model size and inference time for different network architectures and completion techniques in the field of completion would be an interesting future work direction. Knowing the time and memory cost of a completion algorithm during training and inference would support users and developers to target/deploy techniques without wasting testing efforts. Intelligent vehicles are often equipped with limited computing and memory resources. Applications that rely on real-time iterative completion processes (e.g., Simultaneous Localisation and Mapping (SLAM)) could benefit from algorithms with lower memory and computing requirements. We appeal to the community to include these timing and resources information in future publications.

XII. CONCLUSION

This paper offers a systematic review of point cloud completion. We have presented both traditional and learning-based approaches, types of inputs/outputs, datasets, learning architectures, evaluation metrics, existing challenges faced in completion processes, and solution strategies followed by previous works. We have discussed issues concerning users of completion techniques. At the end, we put forward possible future research directions.

ACKNOWLEDGMENTS

This work was funded by EPSRC grant number EP/S021892/1 and Vaarst (vaarst.com). Tam is partially supported by a Royal Society fund IEC/NSFC/211159. For the

purpose of open access the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

REFERENCES

- [1] E. Alexiou, E. Upenik, and T. Ebrahimi, "Towards subjective quality assessment of point cloud imaging in augmented reality," in *MMSP*, 2017, pp. 1–6.
- [2] A. Jaklič, M. Erič, I. Mihajlovič, Ž. Stopinšek, and F. Solina, "Volumetric models from 3D point clouds: The case study of sarcophagi cargo from a 2nd/3rd century ad roman shipwreck near sutivan on island brač, croatia," *J. Archaeol. Sci.*, vol. 62, pp. 143–152, 2015.
- [3] S. B. Williams, O. Pizarro, and B. Foley, "Return to antikythera: Multi-session SLAM based AUV mapping of a first century BC wreck site," in *FSR*, 2016, pp. 45–59.
- [4] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *IROS*, 2017, pp. 2442–2447.
- [5] A. H. Quispe, B. Milville, M. A. Gutiérrez, C. Erdogan, M. Stilman, H. Christensen, and H. B. Amor, "Exploiting symmetries and extrusions for grasping household objects," in *ICRA*, 2015, pp. 3702–3708.
- [6] S. S. Mohammadi, N. F. Duarte, D. Dimou, Y. Wang, M. Taiana, P. Morerio, A. Dehban, P. Moreno, A. Bernardino, A. Del Bue *et al.*, "3dsgrasp: 3d shape-completion for robotic grasp," in *ICRA*, 2023, pp. 3815–3822.
- [7] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *ICCV*, 2019, pp. 9277–9286.
- [8] Y. Zheng, Y. Li, S. Yang, and H. Lu, "Global-pbnet: a novel point cloud registration for autonomous driving," *T-ITS*, vol. 23, no. 11, pp. 22 312–22 319, 2022.
- [9] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *PAMI*, vol. 43, no. 12, pp. 4338–4364, 2020.
- [10] Y. Cai, K.-Y. Lin, C. Zhang, Q. Wang, X. Wang, and H. Li, "Learning a structured latent space for unsupervised point cloud completion," in *CVPR*, 2022, pp. 5543–5553.
- [11] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Symmetry descriptors and 3D shape matching," in *SCA*, 2004, pp. 115–123.
- [12] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *ICCV*, 2021, pp. 12 498–12 507.
- [13] N. Soans, E. Asali, Y. Hong, and P. Doshi, "SA-net: Robust state-action recognition for learning from observations," in *ICRA*, 2020.
- [14] S. Zhang, X. Liu, H. Xie, L. Nie, H. Zhou, D. Tao, and X. Li, "Learning geometric transformation for point cloud completion," *IJCV*, pp. 1–21, 2023.
- [15] B. Bellekens, V. Spruyt, R. Berkvens, R. Penne, and M. Weyn, "A benchmark survey of rigid 3D point cloud registration algorithms," *IJAIN*, vol. 1, 2015.
- [16] A. Nguyen and B. Le, "3D point cloud segmentation: A survey," in *RAM*, 2013, pp. 225–230.
- [17] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," in *CGF*, vol. 36, no. 1, 2017, pp. 301–329.
- [18] B. Fei, W. Yang, W. Chen, Z. Li, Y. Li, T. Ma, X. Hu, and L. Ma, "Comprehensive review of deep learning-based 3D point cloud completion processing and analysis," *T-ITS*, vol. 23, pp. 22 862–22 883, 2022.
- [19] X. Zhang, Y. Feng, S. Li, C. Zou, H. Wan, X. Zhao, Y. Guo, and Y. Gao, "View-guided point cloud completion," in *CVPR*, 2021, pp. 15 890–15 899.
- [20] L. Roldao, R. De Charette, and A. Verroust-Blondet, "3D semantic scene completion: A survey," *IJCV*, vol. 130, no. 8, pp. 1978–2005, 2022.
- [21] X. Li, L. Wang, and Y. Fang, "Unsupervised category-specific partial point set registration via joint shape completion and registration," *TVCG*, 2022.
- [22] G. K. L. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin, "Registration of 3D point clouds and meshes: A survey from rigid to nonrigid," *TVCG*, vol. 19, no. 7, pp. 1199–1217, 2013.
- [23] I. Demir, D. G. Aliaga, and B. Benes, "Procedural editing of 3D building point clouds," in *ICCV*, 2015, pp. 2147–2155.

- [24] T. Rumezhak, O. Dobosevych, R. Hryniv, V. Selotkin, V. Karpiv, and M. Maksymenko, "Towards realistic symmetry-based completion of previously unseen point clouds," in *ICCV*, 2021, pp. 2542–2550.
- [25] Y. Ren, P. Cong, X. Zhu, and Y. Ma, "Self-supervised point cloud completion on real traffic scenes via scene-concerned bottom-up mechanism," in *ICME*, 2022, pp. 1–6.
- [26] Y. Ibrahim, B. Nagy, and C. Benedek, "Multi-view based 3d point cloud completion algorithm for vehicles," in *ICPR*, 2022, pp. 2121–2127.
- [27] X. Ma, X. Li, and J. Song, "Point cloud completion network applied to vehicle data," *Sensors*, vol. 22, no. 19, p. 7346, 2022.
- [28] J. D. Toscano, C. Zuniga-Navarrete, W. D. J. Siu, L. J. Segura, and H. Sun, "Teeth mold point cloud completion via data augmentation and hybrid rl-gan," *J. Comput.*, vol. 23, no. 4, p. 041008, 2023.
- [29] G. Hosseinianesh, F. Ghadiri, A. Alshegri, Y. Zhang, J. Keren, F. Cheriet, and F. Guibault, "Improving the quality of dental crown using a transformer-based method," in *FPHY*, vol. 12463, 2023, pp. 802–809.
- [30] D. Xu, G. Chen, and W. Jing, "A single-tree point cloud completion approach of feature fusion for agricultural robots," *Electronics*, vol. 12, no. 6, p. 1296, 2023.
- [31] M. Lou, J. Lu, L. Wang, H. Jiang, and M. Zhou, "Growth parameter acquisition and geometric point cloud completion of lettuce," *Front. Plant Sci.*, vol. 13, p. 947690, 2022.
- [32] X. Wen, P. Xiang, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "PMP-Net: Point cloud completion by learning multi-step point moving paths," in *CVPR*, 2021, pp. 7443–7452.
- [33] Y. Wang, D. J. Tan, N. Navab, and F. Tombari, "Softpoolnet: Shape descriptor for point cloud completion and classification," in *ECCV*, 2020, pp. 70–85.
- [34] P. Spurek, A. Kasymov, M. Mazur, D. Janik, S. K. Tadeja, J. Tabor, T. Trzcinski *et al.*, "Hyperpocket: Generative point cloud completion," in *IROS*, 2022, pp. 6848–6853.
- [35] Z. Zhu, L. Nan, H. Xie, H. Chen, J. Wang, M. Wei, and J. Qin, "Csdn: Cross-modal shape-transfer dual-refinement network for point cloud completion," *TVCG*, 2023.
- [36] E. Aiello, D. Valsesia, and E. Magli, "Cross-modal learning for image-guided point cloud shape completion," *arXiv:2209.09552*, 2022.
- [37] Y. Zhang, Z. Liu, X. Li, and Y. Zang, "Data-driven point cloud objects completion," *Sensors*, vol. 19, no. 7, p. 1514, 2019.
- [38] L. Wu, Q. Zhang, J. Hou, and Y. Xu, "Leveraging single-view images for unsupervised 3D point cloud completion," *arXiv:2212.00564*, 2022.
- [39] Y. Xia, Y. Xu, C. Wang, and U. Stilla, "Vpc-net: Completion of 3d vehicles from mls point clouds," *ISPRS*, vol. 174, pp. 166–181, 2021.
- [40] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3D surface generation," in *CVPR*, 2018, pp. 216–224.
- [41] M. Liu, L. Sheng, S. Yang, J. Shao, and S.-M. Hu, "Morphing and sampling network for dense point cloud completion," in *AAAI*, vol. 34, no. 07, 2020, pp. 11 596–11 603.
- [42] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *3DV*, 2018, pp. 728–737.
- [43] Y. Nie, Y. Lin, X. Han, S. Guo, J. Chang, S. Cui, J. Zhang *et al.*, "Skeleton-bridged point completion: From global inference to local adjustment," *NIPS*, vol. 33, pp. 16 119–16 130, 2020.
- [44] W. Zhang, C. Long, Q. Yan, and C. Xiao, "Multi-stage point completion network with critical set supervision," *Comput. Aided Geom. Des.*, vol. 82, p. 101925, 2020.
- [45] Y. Xia, Y. Xia, W. Li, R. Song, K. Cao, and U. Stilla, "ASFM-Net: Asymmetrical siamese feature matching network for point completion," *ACM MM*, 2021.
- [46] Y. Peng, M. Chang, Q. Wang, Y. Qian, Y. Zhang, M. Wei, and X. Liao, "Sparse-to-dense multi-encoder shape completion of unstructured point cloud," *IEEE Access*, vol. 8, pp. 30 969–30 978, 2020.
- [47] K. Zhang, X. Yang, Y. Wu, and C. Jin, "Srpcn: Structure retrieval based point completion network," *arXiv:2202.02669*, 2022.
- [48] J. Tang, J. Xu, J. Gong, H. Song, Y. Xie, and L. Ma, "Prototype-aware heterogeneous task for point cloud completion," *arXiv:2209.01733*, 2022.
- [49] A. Saroha, M. Eisenberger, T. Yenamandra, and D. Cremers, "Implicit shape completion via adversarial shape priors," *arXiv:2204.10060*, 2022.
- [50] M. Xu, Y. Wang, Y. Liu, T. He, and Y. Qiao, "Cp3: Unifying point cloud completion by pretrain-prompt-predict paradigm," *PAMI*, 2023.
- [51] X. Yang, H. Zou, X. Kong, T. Huang, Y. Liu, W. Li, F. Wen, and H. Zhang, "Semantic segmentation-assisted scene completion for lidar point clouds," in *IROS*, 2021, pp. 3555–3562.
- [52] J. Shi, L. Xu, P. Li, X. Chen, and S. Shen, "Temporal point cloud completion with pose disturbance," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4165–4172, 2022.
- [53] Y. Kasten, O. Rahamim, and G. Chechik, "Point-cloud completion with pretrained text-to-image diffusion models," *arXiv:2306.10533*, 2023.
- [54] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, and Z. Han, "Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *ICCV*, 2021, pp. 5499–5509.
- [55] Y. Chang, C. Jung, and Y. Xu, "Finerpcn: High fidelity point cloud completion network using pointwise convolution," *Neurocomputing*, vol. 460, pp. 266–276, 2021.
- [56] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "Pf-net: Point fractal network for 3D point cloud completion," in *CVPR*, 2020, pp. 7662–7670.
- [57] X. Wang, M. H. Ang Jr, and G. H. Lee, "Cascaded refinement network for point cloud completion," in *CVPR*, 2020, pp. 790–799.
- [58] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "Grnet: Gridding residual network for dense point cloud completion," in *EECV*, 2020, pp. 365–381.
- [59] W. Zhang, Q. Yan, and C. Xiao, "Detail preserved point cloud completion via separated feature aggregation," in *ECCV*, 2020, pp. 512–528.
- [60] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Stoll, and C. Theobalt, "Patchnets: Patch-based generalizable deep implicit 3D shape representations," in *ECCV*, 2020, pp. 293–309.
- [61] R. Wu, X. Chen, Y. Zhuang, and B. Chen, "Multimodal shape completion via conditional generative adversarial networks," in *ECCV*, 2020, pp. 281–296.
- [62] H. Arora, S. Mishra, S. Peng, K. Li, and A. Mahdavi-Amiri, "Multimodal shape completion via imle," *arXiv:2106.16237*, 2021.
- [63] R. Cui, S. Qiu, S. Anwar, J. Zhang, and N. Barnes, "Energy-based residual latent transport for unsupervised point cloud completion," *arXiv:2211.06820*, 2022.
- [64] L. P. Tchapmi, V. Kosaraju, H. Rezafofighi, I. Reid, and S. Savarese, "Topnet: Structural point cloud decoder," in *CVPR*, 2019, pp. 383–392.
- [65] X. Chen, B. Chen, and N. J. Mitra, "Unpaired point cloud completion on real scans using adversarial training," *arXiv:1904.00069*, 2019.
- [66] X. Wen, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding," in *CVPR*, 2021, pp. 13 080–13 089.
- [67] Z. Lyu, Z. Kong, X. Xu, L. Pan, and D. Lin, "A conditional point diffusion-refinement paradigm for 3D point cloud completion," *arXiv:2112.03530*, 2021.
- [68] F. Hu, H. Chen, X. Lu, Z. Zhu, J. Wang, W. Wang, F. L. Wang, and M. Wei, "Spnet: Stepwise point cloud completion network," *arXiv:2209.01746*, 2022.
- [69] X. Wang, M. H. Ang, and G. H. Lee, "Voxel-based network for shape completion by leveraging edge generation," in *ICCV*, 2021, pp. 13 189–13 198.
- [70] Q. Guo, Z. Wang, F. Juefei-Xu, D. Lin, L. Ma, W. Feng, and Y. Liu, "Carvenet: Carving point-block for complex 3D shape completion," *arXiv:2107.13452*, 2021.
- [71] J. Zhang, W. Chen, Y. Wang, R. Vasudevan, and M. Johnson-Roberson, "Point set voting for partial point cloud analysis," *RA-L*, vol. 6, no. 2, pp. 596–603, 2021.
- [72] D. Stutz and A. Geiger, "Learning 3d shape completion from laser scan data with weak supervision," in *CVPR*, 2018, pp. 1955–1964.
- [73] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *SGP*, vol. 7, 2006.
- [74] B. B. Kimia, I. Frankel, and A.-M. Popescu, "Euler spiral for shape completion," *IJCV*, vol. 54, no. 1-3, pp. 159–182, 2003.
- [75] A. Sharf, M. Alexa, and D. Cohen-Or, "Context-based surface completion," in *SIGGRAPH*, 2004, pp. 878–887.
- [76] Z. Cai, C. Wang, C. Wen, and J. Li, "3d-patchmatch: An optimization algorithm for point cloud completion," in *ICSDM*, 2015, pp. 157–161.
- [77] D. Doria and R. J. Radke, "Filling large holes in lidar data by inpainting depth gradients," in *CVPR Workshops*, 2012, pp. 65–72.
- [78] K. Sarkar, K. Varanasi, and D. Stricker, "Learning quadrangulated patches for 3d shape parameterization and completion," in *3DV*, 2017, pp. 383–392.
- [79] C.-H. Shen, H. Fu, K. Chen, and S.-M. Hu, "Structure recovery by part assembly," *TOG*, vol. 31, no. 6, pp. 1–11, 2012.

- [80] M. Pauly, N. J. Mitra, J. Giesen, M. H. Gross, and L. J. Guibas, "Example-based 3D scan completion," in *SGP*, no. CONF, 2005, pp. 23–32.
- [81] R. Schnabel, P. Degener, and R. Klein, "Completion and reconstruction with primitive shapes," in *CGF*, vol. 28, no. 2, 2009, pp. 503–512.
- [82] Y. Li, A. Dai, L. Guibas, and M. Nießner, "Database-assisted object retrieval for real-time 3D reconstruction," in *CGF*, vol. 34, no. 2, 2015, pp. 435–446.
- [83] V. G. Kim, W. Li, N. J. Mitra, S. Chaudhuri, S. DiVerdi, and T. Funkhouser, "Learning part-based templates from large collections of 3D shapes," *SIGGRAPH*, vol. 32, no. 4, 2013.
- [84] S. Friedman and I. Stamos, "Online facade reconstruction from dominant frequencies in structured point clouds," *CVPR Workshops*, pp. 1–8, 2012.
- [85] A.-L. Chauve, P. Labatut, and J.-P. Pons, "Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data," in *CVPR*, 2010, pp. 1261–1268.
- [86] M. Sung, V. G. Kim, R. Angst, and L. Guibas, "Data-driven structural priors for shape completion," *TOG*, vol. 34, no. 6, pp. 1–11, 2015.
- [87] O. Kroemer, H. B. Amor, M. Ewerton, and J. Peters, "Point cloud completion using extrusions," in *Humanoids*, 2012, pp. 680–685.
- [88] M. Mirbauer, M. Krabec, J. Křivánek, and E. Šikudová, "Survey and evaluation of neural 3D shape classification approaches," *PAMI*, vol. 44, no. 11, pp. 8635–8656, 2021.
- [89] C. Caruso and F. Quarta, "Interpolation methods comparison," *Comput.*, vol. 35, no. 12, pp. 109–126, 1998.
- [90] J. Wang, Y. Cui, D. Guo, J. Li, Q. Liu, and C. Shen, "Pointattn: You only need attention for point cloud completion," *arXiv:2203.08485*, 2022.
- [91] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3D model repository," *arXiv:1512.03012*, 2015.
- [92] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015, pp. 1912–1920.
- [93] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," *arXiv:1511.01844*, 2015.
- [94] P. Achlioptas, O. Diamanti, I. Mitiagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *ICML*, 2018, pp. 40–49.
- [95] H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, and C. Wang, "Seedformer: Patch seeds based point cloud completion with upsample transformer," in *ECCV*, 2022, pp. 416–432.
- [96] P. Wang, L. Ren, S. Wu, J. Yang, E. Yu, H. Yu, and X. Li, "Implicit and efficient point cloud completion for 3d single object tracking," *RA-L*, vol. 8, no. 4, pp. 1935–1942, 2023.
- [97] J. Gu, W.-C. Ma, S. Manivasagam, W. Zeng, Z. Wang, Y. Xiong, H. Su, and R. Urtasun, "Weakly-supervised 3D shape completion in the wild," in *ECCV*, 2020, pp. 283–299.
- [98] Z. Fan, Y. He, Z. Wang, K. Wu, H. Liu, and J. He, "Reconstruction-aware prior distillation for semi-supervised point cloud completion," *arXiv:2204.09186*, 2022.
- [99] J. Zhang, X. Chen, Z. Cai, L. Pan, H. Zhao, S. Yi, C. K. Yeo, B. Dai, and C. C. Loy, "Unsupervised 3d shape completion through gan inversion," in *CVPR*, 2021, pp. 1768–1777.
- [100] C. Ma, Y. Yang, J. Guo, F. Pan, C. Wang, and Y. Guo, "Unsupervised point cloud completion and segmentation by generative adversarial autoencoding network," in *NeurIPS*, 2022.
- [101] H. Mittal, B. Okorn, A. Jangid, and D. Held, "Self-supervised point cloud completion via inpainting," *arXiv:2111.10701*, 2021.
- [102] J. Gong, F. Liu, J. Xu, M. Wang, X. Tan, Z. Zhang, R. Yi, H. Song, Y. Xie, and L. Ma, "Optimization over disentangled encoding: Un-supervised cross-domain point cloud completion via occlusion factor manipulation," in *ECCV*, 2022.
- [103] J. Tang, Z. Gong, R. Yi, Y. Xie, and L. Ma, "Lake-net: topology-aware point cloud completion by localizing aligned keypoints," in *CVPR*, 2022, pp. 1726–1735.
- [104] S. Hong, M. Yavartanoo, R. Neshatavar, and K. M. Lee, "Acl-spc: Adaptive closed-loop system for self-supervised point cloud completion," in *CVPR*, 2023, pp. 9435–9444.
- [105] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *IJRR*, 2013.
- [106] M. Savva, A. X. Chang, and P. Hanrahan, "Semantically-enriched 3d models for common-sense knowledge," in *CVPR Workshops*, 2015, pp. 24–31.
- [107] L. Pan, X. Chen, Z. Cai, J. Zhang, H. Zhao, S. Yi, and Z. Liu, "Variational relational point completion network," in *CVPR*, 2021, pp. 8524–8533.
- [108] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv:1709.06158*, 2017.
- [109] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *CVPR*, 2017, pp. 5828–5839.
- [110] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner, "Scan2cad: Learning cad model alignment in rgb-d scans," in *CVPR*, 2019, pp. 2614–2623.
- [111] K. Zhang, X. Yang, Y. Wu, and C. Jin, "Attention-based transformation from latent features to point clouds," in *AAAI*, vol. 36, no. 3, 2022, pp. 3291–3299.
- [112] Y. Wang, D. J. Tan, N. Navab, and F. Tombari, "Softpool++: An encoder–decoder network for point cloud completion," *IJCV*, vol. 130, no. 5, pp. 1145–1164, 2022.
- [113] X. Wen, P. Xiang, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "PMP-Net++: Point cloud completion by transformer-enhanced multi-step point moving paths," *PAMI*, vol. 45, no. 1, pp. 852–867, 2022.
- [114] C. Ma, Y. Yang, J. Guo, C. Wang, and Y. Guo, "Completing partial point clouds with outliers by collaborative completion and segmentation," *arXiv:2203.09772*, 2022.
- [115] C. Hernández and G. Vogiatzis, "Shape from photographs: A multi-view stereo pipeline," *CVDRR*, pp. 281–311, 2010.
- [116] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Density-aware chamfer distance as a comprehensive metric for point cloud completion," *arXiv:2111.12702*, 2021.
- [117] R.-W. Li, B. Wang, C.-P. Li, L.-X. Zhang, and L. Gao, "High-fidelity point cloud completion with low-resolution recovery and noise-aware upsampling," *arXiv:2112.11271*, 2021.
- [118] X. Wang, M. H. Ang, and G. H. Lee, "Cascaded refinement network for point cloud completion with self-supervision," *PAMI*, vol. 44, no. 11, pp. 8139–8150, 2021.
- [119] J. Shi, L. Xu, L. Heng, and S. Shen, "Graph-guided deformation for point cloud completion," *RA-L*, vol. 6, no. 4, pp. 7081–7088, 2021.
- [120] P. Selvaraju, M. Nabail, M. Loizou, M. Maslioukova, M. Averkiou, A. Andreou, S. Chaudhuri, and E. Kalogerakis, "Buildingnet: Learning to label 3d buildings," in *ICCV*, 2021, pp. 10397–10407.
- [121] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *IJCV*, vol. 40, no. 2, p. 99, 2000.
- [122] T. Nguyen, Q.-H. Pham, T. Le, T. Pham, N. Ho, and B.-S. Hua, "Point-set distances for learning representations of 3d point clouds," in *ICCV*, 2021, pp. 10478–10487.
- [123] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation," in *Adv. in Artif. Intell.*, 2006, pp. 1015–1021.
- [124] H. Son and Y. M. Kim, "Saum: Symmetry-aware upsampling module for consistent point cloud completion," in *ACCV*, 2020.
- [125] R. Cui, S. Qiu, S. Anwar, J. Liu, C. Xing, J. Zhang, and N. Barnes, "P2c: Self-supervised point cloud completion from single partial clouds," *arXiv:2307.14726*, 2023.
- [126] P. Flach and M. Kull, "Precision-recall-gain curves: Pr analysis done right," *Adv. neural inf. process. syst.*, vol. 28, 2015.
- [127] H. Huang, H. Chen, and J. Li, "Deep neural network for 3d point cloud completion with multistage loss function," in *CCDC*, 2019, pp. 4604–4609.
- [128] M. W. Jones, "The production of volume data from triangular meshes using voxelisation," *CGF*, vol. 15, no. 5, pp. 311–318, 1996.
- [129] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *CVPR*, 2018, pp. 206–215.
- [130] X. Wen, T. Li, Z. Han, and Y.-S. Liu, "Point cloud completion by skip-attention network with hierarchical folding," in *CVPR*, 2020, pp. 1939–1948.
- [131] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 128–138, 2020.

- [132] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, 2018.
- [133] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," in *ICCV*, 2021, pp. 5826–5835.
- [134] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NIPS*, vol. 33, pp. 6840–6851, 2020.
- [135] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.
- [136] D. Zong, S. Sun, and J. Zhao, "Ashf-net: Adaptive sampling and hierarchical folding network for robust point cloud completion," in *AAAI*, vol. 35, no. 4, 2021, pp. 3625–3632.
- [137] H. Xiao, Y. Li, W. Kang, and Q. Wu, "Distinguishing and matching-aware unsupervised point cloud completion," *TCSVT*, 2023.
- [138] L. Xi, W. Tang, and T. Wan, "Treenet: Structure preserving multi-class 3d point cloud completion," *Pattern Recognit.*, vol. 139, p. 109476, 2023.
- [139] Z. Zhu, H. Chen, X. He, W. Wang, J. Qin, and M. Wei, "Svdformer: Complementing point cloud via self-view augmentation and self-structure dual-generator," *arXiv:2307.08492*, 2023.
- [140] Y. Ibrahim and C. Benedek, "Mvpc-net: Multi-view based point cloud completion network for mls data," *Image Vis. Comput.*, vol. 134, p. 104675, 2023.
- [141] T. T. P. Nguyen, S. L. Phung, V. Gopaldasani, and J. Whitelaw, "Bilateral coarse-to-fine network for point cloud completion," in *ICASSP*, 2023, pp. 1–5.
- [142] L. Zhu, B. Wang, G. Tian, W. Wang, and C. Li, "Towards point cloud completion: Point rank sampling and cross-cascade graph cnn," *Neurocomputing*, vol. 461, pp. 1–16, 2021.
- [143] Y.-C. Cheng, H.-Y. Lee, S. Tulyakov, A. G. Schwing, and L.-Y. Gui, "Sdfusion: Multimodal 3d shape completion, reconstruction, and generation," in *CVPR*, 2023, pp. 4456–4465.
- [144] X. Cheng, N. Zhang, J. Yu, Y. Wang, G. Li, and J. Zhang, "Null-space diffusion sampling for zero-shot point cloud completion," *IJCAI*, 2023.
- [145] Y. Li, Y. Dou, X. Chen, B. Ni, Y. Sun, Y. Liu, and F. Wang, "3dq: Generalized deep 3d shape prior via part-discretized diffusion process," *arXiv:2303.10406*, 2023.
- [146] L. Wu, D. Wang, C. Gong, X. Liu, Y. Xiong, R. Ranjan, R. Krishnamoorthi, V. Chandra, and Q. Liu, "Fast point cloud generation with straight flows," in *CVPR*, 2023, pp. 9445–9454.
- [147] J. Long, Q. Zhu, H. He, Z. Yu, Q. Zhang, and Z. Zhang, "3d point cloud completion based on multi-scale degradation," in *ICASSP*, 2023, pp. 1–5.
- [148] G. Qian, A. Abualshour, G. Li, A. Thabet, and B. Ghanem, "Pu-gcn: Point cloud upsampling using graph convolutional networks," in *CVPR*, 2021, pp. 11 683–11 692.
- [149] C. Ma, Y. Chen, P. Guo, J. Guo, C. Wang, and Y. Guo, "Symmetric shape-preserving autoencoder for unsupervised real scene point cloud completion," in *CVPR*, 2023, pp. 13 560–13 569.
- [150] Y. Zhang, W. Zhao, B. Sun, Y. Zhang, and W. Wen, "Point cloud upsampling algorithm: A systematic review," *Algorithms*, vol. 15, no. 4, 2022.
- [151] Z. Chen, F. Long, Z. Qiu, T. Yao, W. Zhou, J. Luo, and T. Mei, "Anchorformer: Point cloud completion from discriminative nodes," in *CVPR*, 2023, pp. 13 581–13 590.
- [152] B. Fei, W. Yang, L. Ma, and W.-M. Chen, "Dctr: Noise-robust point cloud completion by dual-channel transformer with cross-attention," *Pattern recognit.*, vol. 133, p. 109051, 2023.
- [153] S. Li, P. Gao, X. Tan, and M. Wei, "Proxyformer: Proxy alignment assisted point cloud completion with missing part sensitive transformer," in *CVPR*, 2023, pp. 9466–9475.
- [154] Q. Wang, Y. Tan, and Z. Mei, "Computational methods of acquisition and processing of 3D point cloud data for construction applications," *Arch. Computat. Methods Eng.*, vol. 27, pp. 479–499, 2019.
- [155] S. Gurumurthy and S. Agrawal, "High fidelity semantic shape completion for point clouds using latent optimization," in *WACV*, 2019, pp. 1099–1108.
- [156] J. Li, G. Si, X. Liang, Z. An, P. Tian, and F. Zhou, "Partition-based point cloud completion network with density refinement," *Entropy*, vol. 25, no. 7, p. 1018, 2023.
- [157] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Pu-net: Point cloud upsampling network," in *CVPR*, 2018, pp. 2790–2799.
- [158] C. Wen, X. Sun, J. Li, C. Wang, Y. Guo, and A. Habib, "A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds," *ISPRS*, vol. 147, pp. 178–192, 2019.
- [159] Y. Tang and D. Ha, "The sensory neuron as a transformer: Permutation-invariant neural networks for reinforcement learning," *NIPS*, vol. 34, pp. 22 574–22 587, 2021.
- [160] P.-S. Wang, Y. Liu, and X. Tong, "Deep octree-based cnns with output-guided skip connections for 3d shape and scene completion," in *CVPR Workshops*, 2020, pp. 266–267.
- [161] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu, "High-resolution shape completion using deep neural networks for global structure and local geometry inference," in *ICCV*, 2017, pp. 85–93.
- [162] J. Wei, L. Liu, R. Cheng, W. Jiang, M. Xu, X. Jiang, T. Sun, S. Schwertfeger, and L. Kneip, "Spotlights: Probing shapes from spherical viewpoints," in *ACCV*, 2022, pp. 4310–4326.
- [163] P.-S. Wang, Y. Liu, and X. Tong, "Deep octree-based cnns with output-guided skip connections for 3D shape and scene completion," in *CVPR Workshops*, 2020.
- [164] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *CVPR*, 2017, pp. 652–660.
- [165] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *NIPS*, vol. 30, 2017.
- [166] A. V. Phan, M. Le Nguyen, Y. L. H. Nguyen, and L. T. Bui, "DGCNN: A convolutional neural network over large-scale labeled graphs," *Neural Networks*, vol. 108, pp. 533–543, 2018.
- [167] D. W. Shu, S. W. Park, and J. Kwon, "3d point cloud generative adversarial network based on tree structured graph convolutions," in *ICCV*, 2019, pp. 3859–3868.
- [168] X. Yan, H. Yan, J. Wang, H. Du, Z. Wu, D. Xie, S. Pu, and L. Lu, "Fbnet: Feedback network for point cloud completion," in *ECCV*, 2022, pp. 676–693.
- [169] J. Choe, B. Joung, F. Rameau, J. Park, and I. S. Kweon, "Deep point cloud reconstruction," *arXiv:2111.11704*, 2021.
- [170] J. Minár, I. S. Evans, and M. Jenčo, "A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modelling and prediction," *Earth-Sci Rev.*, vol. 211, p. 103414, 2020.
- [171] J. Jiang, X. Lu, L. Zhao, R. Dazeley, and M. Wang, "Masked autoencoders in 3D point cloud representation learning," *arXiv:2207.01545*, 2022.
- [172] K. Zhang, X. Yang, Y. Wu, and C. Jin, "Attention-based transformation from latent features to point clouds," in *AAAI*, vol. 36, no. 3, 2022, pp. 3291–3299.
- [173] J. Li, S. Guo, Z. Lai, X. Meng, and S. Han, "Completedt: Point cloud completion with dense augment inference transformers," *arXiv:2205.14999*, 2022.
- [174] T. Hu, Z. Han, and M. Zwicker, "3D shape completion with multi-view consistent inference," in *AAAI*, vol. 34, no. 07, 2020, pp. 10997–11 004.
- [175] A. Mendoza, A. Apaza, I. Sipiran, and C. Lopez, "Refinement of predicted missing parts enhance point cloud completion," *arXiv:2010.04278*, 2020.
- [176] A. Alliegro, D. Valsesia, G. Fracastoro, E. Magli, and T. Tommasi, "Denoise and contrast for category agnostic shape completion," in *CVPR*, 2021, pp. 4629–4638.
- [177] B. Gong, Y. Nie, Y. Lin, X. Han, and Y. Yu, "Me-pcn: Point completion conditioned on mask emptiness," in *ICCV*, 2021, pp. 12 488–12 497.
- [178] B. Zhang, X. Zhao, H. Wang, and R. Hu, "Shape completion with points in the shadow," in *SIGGRAPH*, 2022, pp. 1–9.
- [179] Y. Wang, D. J. Tan, N. Navab, and F. Tombari, "Learning local displacements for point cloud completion," in *CVPR*, 2022, pp. 1568–1577.
- [180] M. Xiang, H. Ye, B. Yang, and F. Cao, "Multi-space and detail-supplemented attention network for point cloud completion," *Appl. Intell.*, vol. 53, no. 12, pp. 14 971–14 985, 2023.
- [181] P. Shi, H. Cheng, X. Han, Y. Zhou, and J. Zhu, "Dualgenerator: Information interaction-based generative network for point cloud completion," *arXiv:2305.09132*, 2023.
- [182] W. Zhang, Z. Dong, J. Liu, Q. Yan, C. Xiao *et al.*, "Point cloud completion via skeleton-detail transformer," *TVCG*, 2022.
- [183] Z. Xia, Y. Liu, X. Li, X. Zhu, Y. Ma, Y. Li, Y. Hou, and Y. Qiao, "Scpnet: Semantic scene completion on point cloud," in *CVPR*, 2023, pp. 17 642–17 651.

[184] S. Zhang, S. Li, A. Hao, and H. Qin, "Point cloud semantic scene completion from rgb-d images," in *AAAI*, vol. 35, no. 4, 2021, pp. 3385–3393.

[185] R. Cheng, C. Agia, Y. Ren, X. Li, and L. Bingbing, "S3cnet: A sparse semantic scene completion network for lidar point clouds," in *PMLR*, vol. 155, 2021, pp. 2148–2161.

[186] J. Hou, A. Dai, and M. Nießner, "Revealnet: Seeing behind objects in rgb-d scans," in *CVPR*, 2020, pp. 2098–2107.

[187] A. Dai, C. Diller, and M. Nießner, "Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans," in *CVPR*, 2020, pp. 849–858.

[188] Z. Li, Y. Yao, Z. Quan, J. Xie, and W. Yang, "Spatial information enhancement network for 3d object detection from point cloud," *Pattern recognit*, vol. 128, p. 108684, 2022.

[189] J. Zhou, J. Zhou, T. X. Lin, and Z. X. Gong, "Object detection of occlusion point cloud based on transformer," in *IJCNN*, 2023, pp. 1–7.

[190] D. Tsai, J. S. Berrio, M. Shan, E. Nebot, and S. Worrall, "Viewer-centred surface completion for unsupervised domain adaptation in 3d object detection," in *ICRA*, 2023, pp. 9346–9353.

[191] X. Yan, J. Gao, J. Li, R. Zhang, Z. Li, R. Huang, and S. Cui, "Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion," in *AAAI*, vol. 35, no. 4, 2021, pp. 3101–3109.

[192] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *ICCV*, 2021, pp. 9782–9792.

[193] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *CVPR*, 2022, pp. 19 313–19 322.

[194] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *CVPR*, 2017, pp. 1746–1754.

[195] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, "Discovering structural regularity in 3D geometry," *SIGGRAPH*, 2008.

[196] G. Zhong, L.-N. Wang, X. Ling, and J. Dong, "An overview on data representation learning: From traditional feature learning to recent deep learning," *JFDS*, vol. 2, no. 4, pp. 265–278, 2016.

[197] I. Brilakis, H. Fathi, and A. Rashidi, "Progressive 3D reconstruction of infrastructure with videogrammetry," *AIC*, vol. 20, no. 7, pp. 884–895, 2011.

[198] Q. Liu, H. Su, Z. Duanmu, W. Liu, and Z. Wang, "Perceptual quality assessment of colored 3d point clouds," *TVCG*, 2022.

[199] X. Zhao, B. Zhang, J. Wu, R. Hu, and T. Komura, "Relationship-based point cloud completion," *TVCG*, vol. 28, no. 12, pp. 4940–4950, 2021.

[200] J. Shi, P. Li, X. Chen, and S. Shen, "Are all point clouds suitable for completion? weakly supervised quality evaluation network for point cloud completion," *arXiv:2303.01804*, 2023.

[201] P. Singh, K. Sadekar, and S. Raman, "Treegcn-ed: encoding point cloud using a tree-structured graph network," *arXiv:2110.03170*, 2021.

[202] K. Ichimaru, R. Furukawa, and H. Kawasaki, "Cnn based dense underwater 3D scene reconstruction by transfer learning using bubble database," in *WACV*, 2019, pp. 1543–1552.

[203] A. Burguera Burguera and F. Bonin-Font, "A trajectory-based approach to multi-session underwater visual slam using global image signatures," *J. Mar. Sci. Eng.*, vol. 7, no. 8, p. 278, 2019.

[204] C. Dinesh, G. Cheung, I. V. Bajic, and C. Yang, "Fast 3D point cloud denoising via bipartite graph approximation & total variation," *arXiv:1804.10831*, 2018.

[205] A. Carrilho, M. Galo, and R. C. Dos Santos, "Statistical outlier detection method for airborne lidar data," *ISPRS*, vol. 42, no. 1, 2018.

[206] I. Lang, A. Manor, and S. Avidan, "Samplenet: Differentiable point cloud sampling," in *CVPR*, 2020, pp. 7578–7588.

[207] O. Litany, T. Remez, and A. Bronstein, "Cloud dictionary: Sparse coding and modeling for point clouds," *arXiv:1612.04956*, 2016.

[208] A. Mao, Z. Du, J. Hou, Y. Duan, Y.-j. Liu, and Y. He, "Pu-flow: A point cloud upsampling network with normalizing flows," *TVCG*, 2022.

[209] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *CVPR*, 2017, pp. 1907–1915.

[210] K. Hassani and M. Haley, "Unsupervised multi-task feature learning on point clouds," in *ICCV*, 2019, pp. 8160–8171.

[211] J. Zhang, H. Zhang, R. Vasudevan, and M. Johnson-Roberson, "Hyperspherical embedding for point cloud completion," in *CVPR*, 2023, pp. 5323–5332.

[212] Y. Cui, R. Chen, W. Chu, L. Chen, D. Tian, Y. Li, and D. Cao, "Deep learning for image and point cloud fusion in autonomous driving: A review," *T-ITS*, vol. 23, no. 2, pp. 722–739, 2021.

[213] A. Siris, J. Jiao, G. K. Tam, X. Xie, and R. W. Lau, "Scene context-aware salient object detection," in *ICCV*, 2021, pp. 4156–4166.

[214] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM*, vol. 55, no. 6, 2022.



Keneni W. Tesema received B.Eng degree in Electrical Engineering and Automation from China Three Gorges University in 2021, MSc in Computer Science from Swansea University in 2022. She is currently doing her Ph.D at Swansea University. Her research interests include point cloud completion, computer vision systems, machine learning, and computer graphics.



Lyndon Hill (Senior Member, IEEE) received the B.Eng(Hons) degree in electronic engineering from the University of Reading, UK in 1994 and the M.Sc and Ph.D degrees from the University of Surrey, UK in 1995 and 2001 respectively. His work has included photography and video content creation for 3D autostereoscopic displays, 3D video compression, frame rate conversion, video stabilisation and 3D subsea survey. He is an inventor on 9 patents with several applications pending. He is currently Head of Research at Vaarst Limited. His research interests include visual SLAM, 3D reconstruction, video completion, video quality improvement and underwater computer vision systems.



Mark W. Jones received the B.Sc. and Ph.D. degrees from Swansea University. He is a Professor in the Department of Computer Science at Swansea University, where he leads the Visual Computing Research group. His research interests include global illumination, visualisation, data science, and associated algorithms and data structures.



Muneeb Ahmad Muneeb Ahmad is a Senior Lecturer in Computer Science at Swansea University. His main research focuses on employing methods from the field of human-centred computing to create intelligent interactive systems such as social robots, and user interfaces that can adapt to users through real-time analysis of their social signals and physiological behaviours.



Gary K.L. Tam received his Ph.D. degree from the Durham University. In 2009, he worked as a research associate at the Department of Computer Science and Informatics, Cardiff University. He joined the Department of Computer Science, Swansea University, since 2012. He is now a senior lecturer. He served as the Guest Editor of the journal special issues of *IJCV* (2017), *Computers* (2018, 2019) and *Applied Sciences* (2023). He has also served in the committee of a number of conferences, including Program Co-Chair of *CGVC* 2018, 2019, *BMVW* 2015, and Organisation Co-Chair of *BMVC* 2015 and *EuroVis* 2014. He is as an Executive Officer of the *EGUK* since 2018. His research interests include computer vision, 3D graphics and visualisation.