International Journal of Computer Vision (2022) 130:2149–2165
https://doi.org/10.1007/s11263-022-01637-1

Data-Driven Restoration of Digital Archaeological Pottery with Point Cloud Analysis

Ivan Sipiran[1] · Alexis Mendoza[2] · Alexander Apaza[2] · Cristian Lopez[3]

Received: 13 March 2021 / Accepted: 30 May 2022 / Published online: 30 June 2022

Abstract
The Josefina Ramos de Cox museum in Lima, Peru, decided to digitize hundreds of archaeological pieces from pre-Colombian cultures to support further research and create virtual educational environments. However, the 3D scanning procedure led to imperfections in the objects' surface, mainly due to the difficulty of manipulating the fragile objects during the acquisition. The problem was that many of the scanned artifacts do not contain the base because the contact surface during acquisition was not visible to the scanner. This paper proposes a method to repair the digital objects' surface using a data-driven approach. We design and train a point cloud neural network that learns to synthesize the missing geometry in an end-to-end manner. Our model consists of a novel architecture and training protocol that addresses the problem of point cloud completion. We propose an end-to-end neural network architecture that focuses on calculating the missing geometry and merging the known input and the predicted point cloud. Our method is composed of two neural networks: the missing part prediction network and the merging-refinement network. The first module focuses on extracting information from the incomplete input to infer the missing geometry. The second module merges both point clouds and improves the distribution of the points. Our approach is effective in repairing pottery objects with large imperfections during the scanning. Besides, our experiments on ShapeNet and Completion3D datasets show that our method is effective in a general setting for shape completion.

Keywords Cultural heritage restoration · Shape completion · Point cloud analysis · Shape analysis

1 Introduction

Three-dimensional digitization of archaeological objects is gaining more interest in recent times. It enables the use of digital cultural heritage (CH) objects in archaeological research and digital archival. The use of a digital counterpart for CH objects prevents fragile pieces' over-manipulation, which could lead to critical deterioration over time. However, the 3D acquisition process is not exempt from imperfections that need to be addressed in a post-processing step through specialized algorithms. Sometimes, objects are challenging to manipulate; hence the scanner cannot get some portions of the surface, which results in a partial representation of the real shape. The challenge during acquisition can also be attributed to measurement time limitations that makes difficult to scan complete objects. In this context, the problem is the effective restoration of the 3D object's surface that closely reassembles the surface of the real object.

The restoration of cultural heritage objects can be cast as a shape completion problem. In shape analysis, the shape completion problem consists of predicting the geometric shape of an object from a partial observation. The problem is challenging because it can be seen as an inverse problem where the solution space can be quite large. Typically, solutions to this problem need to make assumptions (in the form of geometric prior) to limit the search for feasible outputs. Nev-

Communicated by Rei Kawakami.

✉ Ivan Sipiran
isipiran@dcc.uchile.cl

Alexis Mendoza
amendozavil@unsa.edu.pe

Alexander Apaza
aapazato@unsa.edu.pe

Cristian Lopez
clopez@ulasalle.edu.pe

[1] Department of Computer Science, University of Chile, Santiago, Chile

[2] Universidad Nacional de San Agustin, Arequipa, Peru

[3] Universidad de Ingenieria y Tecnologia, UTEC, Barranco, Peru

ertheless, meaningful progress in shape completion comes from developing established techniques for surface analysis and 3D shape geometry processing.

Two of the main ideas that have led to significant advances are structure analysis and similarity-based approaches for shape completion. To use these assumptions, we need to assume that the underlying structure of a 3D object can be interpreted as an invariant characteristic of the object or a set of objects. Hence, there are two strategies commonly adopted in previous works: the analysis of self-similarity and the data-driven approach. The former states that an automatic method might get enough information for the completion process by analyzing the object's structure. For example, if one can detect an object's symmetries, one can use the symmetries to reproduce the missing information. The latter states that a method can recover the structure of a partial observation from a set of objects with a similar structure. If one can retrieve objects with similar geometry, one can use the external objects to infer the partial observation's missing geometry.

This paper adopts the data-driven approach and proposes a neural network that predicts the missing geometry of cultural heritage objects with scanning defects. We choose the data-driven approach due to several reasons. Although there are significant contributions in the use of self-similarity for shape completion, its applicability is limited to cases where the symmetric structure is not affected by partiality. Besides, the recent success of deep learning architectures in 3D shape analysis tasks such as classification, segmentation, and representation learning opens up a broad range of possibilities to explore the identification of structural geometric patterns in a large collection of objects. Therefore, we can use the learned patterns to repair scanning defects in cultural heritage objects.

Our proposal is motivated by a real-world application: the geometric restoration of archaeological objects with scanning defects. The Josefina Ramos de Cox museum (Lima, Peru) started a program to digitize cultural heritage artifacts to maintain a digital record of the pieces for exhibition purposes. The scanning used a desktop structured-light scanner, where the objects are placed on a turntable to take the views. Nevertheless, the scanning session gave limited results to the museum due to the impossibility of capturing the objects' base, mainly because this surface was in contact with the turntable. The acquisition process mistakes are also related to the fear of over-manipulation of the fragile objects in the scanning setup. Consequently, the scanned objects have a large portion of surface missing at their base, and our goal is to predict the missing geometry with high precision.

We use our completion neural network to restore the missing geometry of archaeological objects. We carefully design and implement a training protocol for this application. This protocol aims to take advantage of collections of objects with a similar structure to that in cultural heritage artifacts. Previous restoration methods assume that human-made objects exhibit some structure and regularity (Sipiran, 2017; Pratikakis et al., 2018). If an algorithm can detect symmetries in the object, we can apply the symmetric transformation to synthesize what is missing. However, the main drawback of this approach is the assumption of self-similarity in the input object. Our neural network directly predicts the missing geometry, which enables the restoration of the input object. Besides, neural network training allows us to introduce invariance to different geometries, which is beneficial for a robust predictor.

Our method uses point clouds as a 3D representation. Although there exist deep learning methods for several 3D data such as multi-view images (Su et al., 2015), volumetric (Maturana and Scherer, 2015), meshes (Hanocka et al., 2019), or implicit representations (Gropp et al., 2020); point cloud is a compact and effective representation to tackle the problem of shape completion. Pioneering approaches such as PointNet (Charles et al., 2017) and PointNet++ (Qi et al., 2017) have opened the possibility of learning good high-level representations from point clouds, which are now used in almost every problem related to 3D (Fan et al., 2017; Liu et al., 2019; Qi et al., 2020). Our work proposes using a PointNet-based encoder-decoder architecture that learns to predict the missing point cloud from the partial observation. Subsequently, a second neural network integrates the missing point cloud and the input point cloud in a single reconstruction that provides a smooth geometric transition between the original object and the prediction. This particular design is devoted to keep the original geometry (the archaeological object) untouched and to predict a good complement (in the geometric sense) for the restoration.

Nevertheless, although our design aims to restore CH objects, our method is also general enough to tackle arbytrary shape completion. We evaluate our method's capability and robustness for shape completion using the ShapeNet dataset. This dataset contains a large variability of classes and objects, and it is one of the standard benchmark used to evaluate shape completion algorithms. Our comparison against state-of-the-art methods shows that our approach has a good generalization and robustness.

The contributions of our paper can be summarized as follows:

- We propose a method that predicts the missing part of the analyzed point cloud with a learning process controlled by a specific loss function.
- We propose a refinement strategy to guarantee a good distribution of the final point cloud.
- We design a simple and effective overall architecture for point cloud completion.

- We perform a set of experiments to show our proposal's effectiveness and robustness using a standard dataset (ShapeNet).
- We perform a set of experiments to show the application of completion to restore cultural heritage objects.
- We also make publicly available the complete implementation and data used in all our experiments.[1]

Our paper is organized as follows. Section 2 shows the related work. Section 3 describes our architecture and the loss function. Section 5 presents our experiments on the Completion3D benchmark to evaluate the ability of our method to complete partial scans. Section 4 shows the experiments conducted on the ShapeNet dataset to evaluate the ability of our method to complete point clouds with missing parts. Section 6 performs an ablation study to analyze the effect of the key components of our proposal. Section 7 shows the results of the application of the completion network in archaeological objects. Finally, Sect. 8 concludes our paper.

2 Related Work

We divide the discussion of the related work to consider methods that specifically address the restoration of CH objects and methods that address the problem of shape completion in a more general sense.

2.1 Restoration of CH Objects

Most algorithms for CH restoration are devoted to repairing fractured or deteriorated objects (Papaioannou et al., 2017). Several proposals assume the existence of a rotational symmetry that can help us to synthesize missing geometry. For example, Son et al. (2013) used the profile curve and the symmetry axis of a shape to reassemble 3D potteries. Similarly, Sipiran (2017) presented a method to detect the symmetry axis of fractured CH objects by analyzing the heat diffusion process over the surface of a shape. Also, Sipiran (2018) described a method to detect the symmetric axis of rotationally symmetric CH objects through the analysis of high-curvature features.

Another proposal assumes the presence of reflective symmetries. For instance, Sipiran et al. (2014) described a method to identify symmetric correspondences across reflective planes in a 3D shape. The symmetric transformation derived from these correspondences helped to restore the missing geometry in CH objects. Likewise, Li et al. (2014) described a method to restore ancient Chinese architecture using global reflective symmetries. Also, Mavridis et al. (2015) formulated the completion problem as an optimization solved by a registration method with sparsity constraints.

To the best of our knowledge, the first method in applying a data-driven approach for restoration is the predictive scanning technique proposed by Pratikakis et al. The predictive scanning is a computational tool that combines similarity search and registration techniques to predict the overall shape of an input partial scan. Moreover, the first method in exploring a deep learning approach is the generative approach proposed by Hermoza and Sipiran (2018). This method trained a conditional GAN network that learns to map partial observations to complete reconstructions. However, this method has two limitations. First, the voxelized representation prevents the representation of details in shape. And second, there is no guarantee that the reconstructed output preserves the geometry of the partial observation. In contrast, the method proposed in this paper overcomes both limitations with a point cloud neural architecture designed to keep the geometry of the original input while predicting the missing information with high precision. Besides, the data-driven nature of our method ensures that we can learn the structure information from a collection of shapes; hence, we do not need to impose any assumption about symmetries or any other prior knowledge.

2.2 Shape Completion

Several techniques have been proposed to address the problem of shape completion. Pauly et al. (2005) described a method to restore a 3D scan using a similar object from an external dataset. The reference object is registered to fit the scan's geometry and repair its surface. Also, Huang et al. (2012) proposed a registration-based method that extrapolates the missing geometry using feature-conforming fields. The use of surface local descriptor also enabled the search of self-similarities in 3D objects for repair (Harary et al., 2014b, a). More specifically, the study of symmetries to restore missing geometry has been a crucial research path with outstanding results (Thrun and Wegbreit, 2005; Xu et al., 2009; Zheng et al., 2010; Jiang et al., 2013).

Due to the rise of deep learning techniques and the availability of large-scale 3D benchmarks, many efforts devised shape completion algorithms in a data-driven manner. The first proposals took advantage of the progress of convolutional neural networks (CNN) in the computer vision field to manage volumetric representations. Dai et al. (2017) proposed a 3D encoder predictor network that completes an input scan. Similarly, Han et al. (2017) used an LSTM network to provide a high-resolution volumetric completion. The same progress in CNN networks also enabled the image-based representation of 3D shapes. Hu et al. (2019) developed a generative adversarial network(GAN) that completes rendered depth-maps from incomplete point clouds.

[1] https://github.com/ivansipiran/Data-driven-cultural-heritage.

The reprojection of the completed depth-maps allows us to reconstruct the original 3D incomplete shape. Also, Hu et al. (2020) improved how to evaluate the loss of the depth-maps generated by a GAN to introduce geometric consistency.

The success of the point cloud representation for a 3D shape is mainly due to the advent of neural networks that can directly process unordered sets. The core idea for all the current progress in point cloud analysis is the PointNet architecture (Charles et al., 2017) and its hierarchical variant PointNet++ (Qi et al., 2017). A significant result of these architectures is the possibility of computing a high-level feature vector that conveys the geometric information of a point cloud in a condensed manner. Also, Achlioptas et al. (2018) motivated the use of generative models for point clouds. They showed the representational power of encoder-decoder architectures for this kind of data. Yuan et al. (2018) developed a completion algorithm in two stages to cope with coarse and fine point cloud decoding. Similarly, Tchapmi et al. (2019) proposed a tree-based decoder to give the completion task the multi-resolution capacity. Also, Liu et al. (2020) presented a two-stage completion algorithm by mixing a patch-wise folding decoder and a density sampling merging to deal with a good distribution of the object surface. Likewise, Xie et al. (2020) developed a neural network architecture that transform a point cloud into an intermediate 3D grid representation. As a result, the reasoning about the completion task occurs in this intermediate representation.

We have witnessed the rise of more elaborated approaches to tackle point cloud completion. Most of these approaches use modern strategies in deep learning, trying to surpass the problem of generating visually appealing shapes. For instance, Wang et al. (2020) proposed an algorithm to generate a coarse point cloud subsequently refined through a lifting module that upsamples the generated point cloud. Likewise, Huang et al. (2020) preferred to predict the missing region with a multi-resolution point pyramid decoder, inspired by the feature pyramid network in computer vision tasks. On the other hand, the attention mechanism's idea to guide the data generation is an important new idea to point cloud completion. For instance, Sun et al. (2020) developed a conditional generative model that includes an attention mechanism to weight point features according to their importance in reconstructing the point cloud. Similarly, Wen et al. (2020) devised an encoder-decoder architecture with an attention mechanism and hierarchical folding-based decoder to complete point clouds. More recently, Pan et al. (2021) proposed a two-stage completion method that uses probabilistic modeling to generate a coarse shape and refines the solution with a relational structure module. Also, Wen et al. (2021a) described an architecture that combines a partial-to-complete and complete-to-partial mappings to ensure a proper transfer of features between embeddings spaces of complete and partial shapes. Wen et al. (2021b) presented a model that learns displacements of points in the partial view to cover the structure of the complete shape. It is also possible to find very recent proposals which take advantage of transformers architectures for point cloud completion (Yu et al., 2021; Xiang et al., 2021).

3 Method

The typical strategy to address the problem of point cloud completion is the encoder–decoder model. In this model, the encoder transforms the incomplete point cloud into an intermediate representation that conveys the input's main geometric characteristics. Subsequently, the decoder converts the intermediate representation into the completed point cloud. Typically, this approach requires a refinement step to ensure some output properties, such as level of resolution or point distribution. Nevertheless, previous methods try to reconstruct the complete point cloud from the input, which may lead to undesired results such as the loss of the original point cloud distribution.

In CH restoration, an ideal completion algorithm should hold one crucial property: to keep the original incomplete data unchanged as much as possible. This property is fundamental in our scenario, where the data comes from sensing devices, and we could be interested in keeping the main characteristics of the incoming geometry. Precisely, our proposal addresses the point cloud completion problem taking into account the property above. We propose an end-to-end neural network that predicts the missing part from a partial observation of a given object. By predicting only the missing geometry, the input point cloud remains unchanged, and therefore we can exploit its structure to produce a final reconstruction. Our proposal still requires a refinement step to improve the resulting point cloud's overall distribution and close the gaps between the missing part and the original input. Nevertheless, our results show that preserving the original incomplete point cloud leads to superior results compared to previous approaches.

Let $\mathcal{P}$ be the point cloud representing a 3D shape. We assume that $\mathcal{P}$ is split into the incomplete point cloud $\mathcal{I}$ and the missing part $\mathcal{M}$ such that $\mathcal{P} = \mathcal{I} \cup \mathcal{M}$. Figure 1 shows our proposed workflow to reconstruct $\mathcal{P}$ from the observed point cloud $\mathcal{I}$. Our method consists of three modules that operate in an end-to-end manner to complete a given input.

The first module (*missing part prediction*) is a learned parametric function that computes the missing part $\widehat{\mathcal{M}}$. During training, our method controls the learning process by applying a loss function to measure the similarity between the predicted missing part $\widehat{\mathcal{M}}$ and the real missing part $\mathcal{M}$. Recently, the Point Fractal Network(PF-Net) (Huang et al., 2020) also proposed to generate the missing part from the incomplete input. The differences between our method and

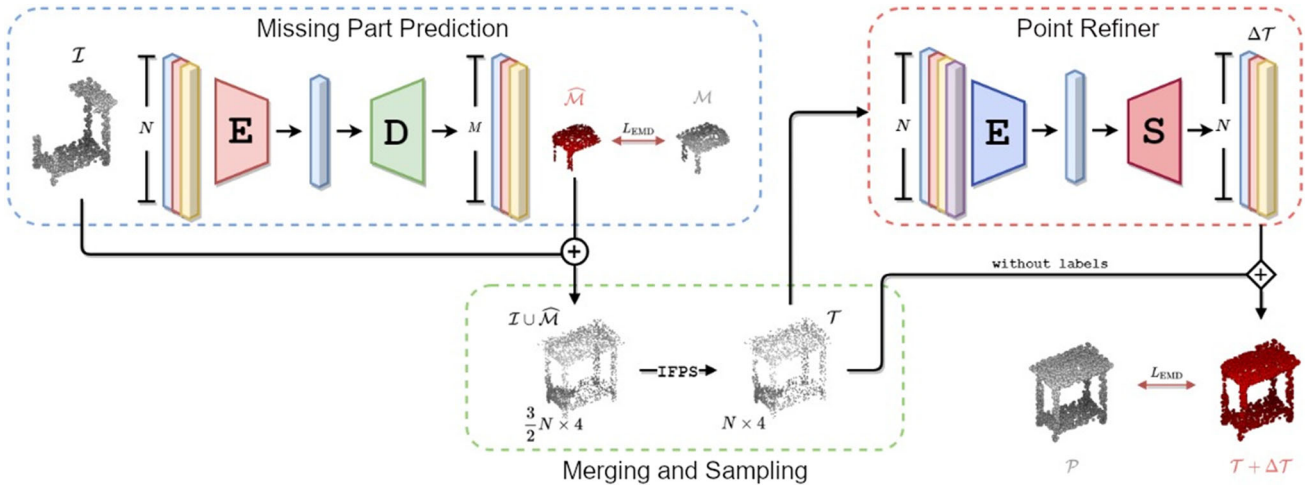

Fig. 1 Workflow of our method. It consists of three main stages. First, the Missing Part Prediction Network (Sect. 3.1) takes an incomplete point cloud $\mathcal{I}$ and outputs the corresponding missing part $\widehat{\mathcal{M}}$. Second, the Merging and Sampling strategy concatenate and sample the incomplete input cloud and the predicted missing part (Sect. 3.2). Third, the Point Refiner Network improves the distribution of the point cloud and closes the gaps generated by the merging (Sect. 3.3)

PF-Net are two-fold: (a) our architecture is simple, and (b) our explicit loss function for the missing part helps guide the end-to-end learning. Our experiments confirm that the design introduced by our method outperforms the performance of PF-Net.

The second module (*merging and sampling*) performs the union of the incomplete input $\mathcal{I}$ and the predicted missing part $\widehat{\mathcal{M}}$. The simple concatenation of these two point sets could lead to problems in the point distribution of the object and the transition between both point sets. Therefore, this module is responsible for improving the sampling of the point set $\mathcal{I} \cup \widehat{\mathcal{M}}$ using the recently proposed Minimum Density Sampling technique (Liu et al., 2020). The result of this module is a sampled point set $\mathcal{T}$.

The third module (*point refiner*) is a learned parametric function that refines the sampled point set $\mathcal{T}$. This function computes a displacement field $\Delta\mathcal{T}$ that shifts the points of $\mathcal{T}$ to the desired position to improve the point distribution. The final output of our proposal is a point cloud $\mathcal{T} + \Delta\mathcal{T}$. During training, the learning process of this module is controlled by a loss function that measures the similarity between the predicted point cloud $\mathcal{T} + \Delta\mathcal{T}$ and the ground truth $\mathcal{P}$.

Section 3.1 introduces the architecture of our neural network for the missing part prediction. Section 3.2 describes the merging and sampling method. Section 3.3 presents the architecture of our refinement network. Finally, Sect. 3.4 defines the joint loss function for training our model.

3.1 Missing Part Prediction Network (MPN)

The main goal of this network is to predict the missing part from an incomplete point cloud. This task extracts features from the incomplete input with an encoder network. For efficiency, we adopt a PointNet (Charles et al., 2017) based encoder; nevertheless, any feature extractor network could be used. We then use a decoder network to obtain the missing part from the feature vector. For the decoder, we use a Morphing-based Decoder as proposed by Liu et al. (2020). We also test a Multi-layer Perceptron decoder to see the decoding's impact on the missing part prediction. In our experiments, the model that uses the morphing-based decoder is named *MBD*, and the model with the MLP decoder is named *MLPD*.

3.1.1 Missing Part Network Encoder

We adopt a PointNet based encoder that obtains a feature vector FV from a $N \times 3$ incomplete point cloud. The encoder architecture consists of four block layers. The first (L1) and second (L2) blocks consist of three layers: 1D-convolution, batch-normalization, and ReLU activation function. In contrast, the third block (L3) has only a 1D-convolution layer and a batch-normalization layer. The number of kernels for 1D convolutions in L1, L2, and L3 is 64, 128, and 1024. The kernels have a size of one, a stride of one, and a padding of zero. Finally, the fourth layer is a fully-connected layer with 1024 neurons. The result is a global feature vector of dimension 1024.

3.1.2 Missing Part Network Decoder

The decoder must transform the feature vector FV into the desired missing part. We denote the output missing part as a point cloud of size $M \times 3$. To compare the overall architecture performance, we propose to use two different networks for

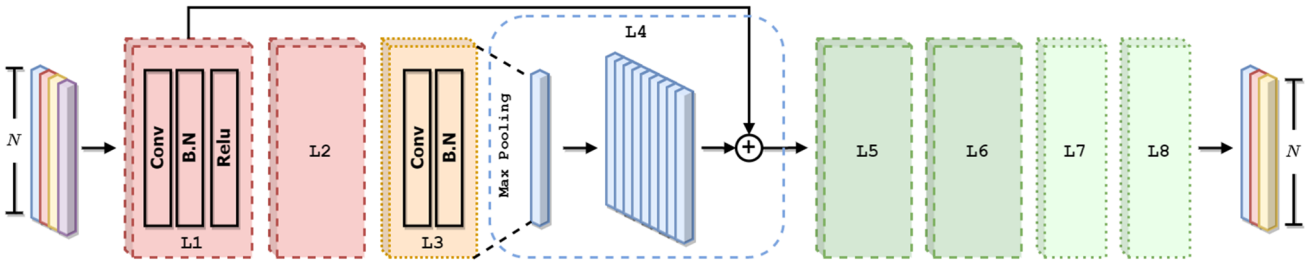

Fig. 2 Point Refiner Network. It takes a labeled point cloud (the purple column represents the label) and passes it through a PointNet Residual Network (L1 expands the point cloud to 64, L2 to 128, and L3 to 1024) to generate a global feature for the point cloud. Then we replicate it 1024 times and concatenate it with the result of L1. Layers L5, L6, L7, and L8 reduce the concatenation and output a three-channel displacement field

the decoder: A Multi-Layer Perceptron (MLP) and a Morphing Based Decoder (MBD) (Liu et al., 2020).

The MLP decoder consists of three building blocks. Each block L_i consists of fully connected layers followed by a ReLU activation except for the third block. The fully connected layers have a number of neurons of 1024, 1024, $M \times 3$). For the Morphing-based Decoder, we use the architecture described in Liu et al. (2020). This decoder consists of K (16 in experiments) morphing networks. Each network maps a 2D surface, sampled from the unit square $[0, 1]^2$, to a 3D surface. In each forward pass, we sample (M/k) points in the unit square. The feature vector FV is then concatenated to each point coordinate before passing through the K morphing networks. Each sampled 2D point is thus mapped back to a 3D surface. The resulting is a point set of M points describing the predicted shape.

Each morphing network has four blocks. The first three blocks consist of a convolution, followed by batch normalization and a ReLU activation function. The last block has a convolution layer and a *tanh* activation function.

3.2 Merging and Sampling

The input (incomplete) point cloud $\mathcal{I}$ is concatenated to the predicted missing part $\widehat{\mathcal{M}}$. The concatenation result is a point cloud of size $\frac{3}{2} \times N \times 3$. We realize that the concatenation has two issues. First, the density of the missing part $\widehat{\mathcal{M}}$ is higher than the density of the input point clouds $\mathcal{I}$. Second, the size of our concatenated point cloud does not match the size of the ground truth. We use a sampling method to adjust point distribution and the size of the merged point cloud to address these problems. We use the iterative farthest point sampling (IFPS), which has been effectively applied in PointNet++ (Qi et al., 2017).

3.3 Point Refiner Network (PRN)

The concatenation of the incomplete input $\mathcal{I}$ and the predicted missing part $\widehat{\mathcal{M}}$ may produce a visible crack in the resulting point cloud. Our refiner network's goal is to solve this issue by improving the transition between the merged point clouds. Besides, the refiner network also can enhance the final distribution of the points in the object. The Point Refiner Network (PRN) architecture is in Fig. 2. The input to the PRN module is the sampled point cloud of size $N \times 3$. Nevertheless, we augment the fourth channel to the input point cloud to assign a label for each point. We give a zero label to points belonging to the original incomplete point cloud $\mathcal{I}$, and one label to points belonging to the predicted missing part $\widehat{\mathcal{M}}$. This augmented channel is used as extra information by the refiner network to decide which points require changes to improve the reconstruction.

The Point Refiner Network has eight layers. Layers L1, L2, and L3 consist of 1D-convolutions, followed by a batch normalization layer. A ReLU activation function also follows L1 and L2. The number of kernels of the first three convolutional layers is 64, 128, and 1024. Layer L4 applies a max-pooling operator to the output of L3 and replicates the output N times. The resulting tensor of size $(1024 \times N)$ is then concatenated to the output of L1, resulting in a tensor of size $(1088 \times N)$, which is then passed through the next 1D-convolutions layers: L5, L6, L7, and L8. These layers are followed by batch normalization and a ReLU activation function, except the last layer L8, which has a Tanh activation function. The number of kernels of the convolutional layers L5, L6, L7, and L8, are 512, 256, 128, and 3.

The Point Refiner Network intends to predict a displacement field $\Delta\mathcal{T}$. The whole network's final output is the point cloud $\mathcal{T} + \Delta\mathcal{T}$. Figure 3 shows the effect of the refinement in our proposal.

3.4 Loss Function

One crucial step for the reconstruction is to define a proper reconstruction loss. In point cloud analysis, there exist predominantly two loss functions: Earth Mover's Distance (EMD) and Chamfer Distance (CD). Given two point clouds S_1 and S_2, CD measures the mean distance between each

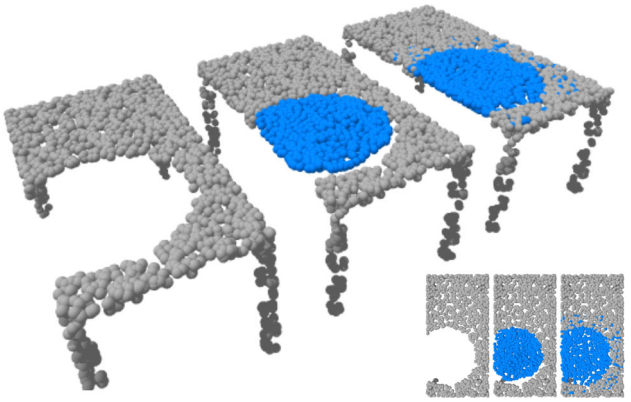

Fig. 3 Our solution takes an incomplete point cloud as input (left). A Missing Part Prediction Network infers the missing part and concatenate it with the input (center). Finally, a Point Refiner Network improves the distribution of points and the geometric transition between the merged sets (right)

point in S_1 to its spatial nearest neighbor in S_2 plus the mean distance between each point in S_2 to its spatial nearest neighbor in S_1.

In contrast, EMD is a metric between two distributions based on the minimal cost of transforming one distribution into another. Given two point clouds of the same size, the loss EMD is defined as follows:

$$L_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \frac{1}{|S_1|} \sum_{x \in S_1} \|x - \phi(x)\|_2 \quad (1)$$

where ϕ is a bijection. Many works use the Chamfer distance because it is less computationally intensive than EMD. Nevertheless, EMD provides better reconstruction results due to its one-to-one point mapping. In this work, we used the EMD implementation presented in Liu et al. (2020), which has a $O(n)$ memory footprint.

In the proposed architecture, we have two losses calculated for both the **MPN** and the **PRN** networks. The first loss estimates the EMD between the predicted missing point cloud $\widehat{\mathcal{M}}$ and the missing ground truth point cloud $\mathcal{M}$. The second loss compares the refined point cloud $\widehat{\mathcal{P}}$ with the complete ground truth point cloud $\mathcal{P}$. Our joint loss is calculated as

$$L = L_{\text{EMD}}(\widehat{\mathcal{M}},\ \mathcal{M}) + L_{\text{EMD}}(\widehat{\mathcal{P}},\ \mathcal{P}). \quad (2)$$

4 Experiments on ShapeNet

4.1 Data Generation and Training

We evaluate our proposal using the same protocol defined in Huang et al. (2020). The dataset comprises 13 classes from the ShapeNet-Part dataset (Yi et al., 2016): airplane, bag, cap, car, chair, guitar, lamp, laptop, motorbike, mug, pistol, skateboard, and table. The dataset contains 14,473 models (11,705 for training and 2768 for testing). Before generating the point clouds from the models, we normalize the input shapes' position and scale. We translate the objects so that the object's centroid is in origin. To normalize the scale, we inscribe the input object inside a sphere of radius one. We finally sample 8192 points from the surface for each CAD model.

During training, for each epoch, we generate ten partial point clouds for each model. To produce a partial point cloud, we randomly choose a model's point and a radius of $r = 0.35$. Subsequently, we split the complete point cloud into two sets: the points outside the sphere (partial point cloud) and the points within the sphere (missing part ground-truth). Finally, the complete point cloud is sampled to 2048 points, the partial point cloud is sampled to 2048 points, and the missing part is sampled to 1024 points.

The implementation of our models is in PyTorch. We use the ADAM optimizer with a learning rate of 0.001. We use the $O(n)$ variant of the Earth Mover Distance as loss function as proposed in Liu et al. (2020). The models were trained in an NVIDIA GeForce RTX 2080 GPU for 200 epochs with a batch size of 64.

4.2 Comparison with Previous Methods

We compare our methods with state-of-the-art techniques and present quantitative and qualitative comparisons on the ShapeNet-Part test data. In this comparison, *MBD* denotes the our proposal that uses a morphing-based decoder for the missing part prediction, and *MLPD* denotes our proposal that uses an MLP as the decoder. We trained all the methods with the same data partition and in the same setup for a fair comparison. The methods used in our comparison are:

- Fully Convolutional Autoencoder (FCAE): We trained an FCAE with the same encoder used in our methods and a decoder resembling our vanilla decoder. The main difference with our method is that the FCAE predicts the complete point cloud (2048 points) rather than the missing part. EMD is used as a loss function.
- Morphing and Sampling Network (MSN) (Liu et al., 2020): We trained the MSN method to predict a coarse complete point cloud (2048 points). This coarse point cloud is further enhanced using a residual network. EMD is used as a loss function.
- Point Fractal Network (PF-Net) (Huang et al., 2020): PF-Net predicts the missing part rather than the overall shape of the point cloud. To do so, it uses a multi-stage completion loss and adversarial loss, whose input is the incomplete point cloud (2048 points), and outputs are the missing point cloud (1024 points). To compare against

Table 1 Quantitative comparison on ShapeNet dataset using average Chamfer distance

Categories	MSN	PF-Net	FCAE	VRCNet	MBD	MLPD
Airplane	6.613	7.601	14.027	3.697	4.675	5.105
Bag	23.336	28.534	63.820	9.044	16.760	19.980
Cap	22.666	32.703	90.008	7.521	11.096	19.543
Car	15.435	12.849	35.532	6.055	9.466	11.199
Chair	10.368	10.577	30.141	4.247	6.629	8.100
Guitar	8.592	6.263	6.461	2.492	3.813	4.203
Lamp	29.692	34.701	60.401	16.246	23.257	25.064
Laptop	7.453	7.141	18.875	2.970	3.790	4.262
Motorbike	13.980	11.502	25.686	6.931	10.843	10.359
Mug	14.539	14.312	47.342	6.395	7.430	10.518
Pistol	11.441	12.557	21.013	5.514	8.676	9.350
Skateboard	7.207	8.350	17.976	5.104	4.545	4.687
Table	14.099	15.420	36.053	5.533	8.333	9.459
Average	14.099	15.420	36.053	6.288	9.174	10.910

Results are scaled by 10,000

the other methods, we merged the input and the predicted point clouds and sampled 2048 points using farthest point sampling. The multi-stage completion loss is equal to the CD of the point clouds for each stage.

– Variational Relational Point Cloud Network (VRCNet) (Pan et al., 2021): VRCNet uses a two-stage process to predict the completion of a partial input. First, the method uses probabilistic modeling to predict a coarse point cloud. Subsequently, a self-attention method computes relational structures in the shape to reproduce small details and generate the fine point cloud.

To compare our proposal with other state-of-the-art methods, we use the Chamfer distance between the ground truth and the predicted point cloud. The results are presented in Table 1. *MBD* outperforms all the methods in the comparison except from VRCNet. MBD computes a point cloud similar to the ground truth, in terms of the average distances of points in the predicted shape. Also, MLPD also outperforms previous methods, but it is slightly worse than MBD. The election of the morphing-based decoder helps to keep the details of the original shape better than a MLP decoder. Our method performs well even for the harder classes in the dataset, namely Bag, Cap, and Lamp. The problem with classes Bag and Cap is the unbalanced number of shapes in the dataset (Bag contains 54 training shapes and Cap contains 39 training shapes), and the problem with Lamp is the high intra-class variability. We believe that these factors prevent automatic shape completion algorithms from generalizing and performing well. Nevertheless, our method gets a notorious improvement concerning other techniques in such classes. Our method's advantage is the combination of missing part prediction with the refinement: in hard examples, the missing part prediction computes a coarse output, subsequently corrected by the refinement. In Sect. 6.2, we present an ablation study with quantitative results about the importance of the sampling and the refinement.

Figure 4 shows qualitative results. Our methods (MBD and MLPD) compute better completions for the shown examples. By definition, our methods preserve the original input geometry while focusing on the missing part's computation and refinement. FCAE produces a fair coarse completion, but many of the details of the shape are lost. Similarly, MSN (Liu et al., 2020) focuses on predicting the entire shape, but it still struggles to reproduce the geometric details of the shape. In contrast, PF-Net (Huang et al., 2020) directly computes the missing part; however, there is no good integration between the partial input and the predicted missing part. Our method's main result is producing good missing parts that are effectively integrated through refinement.

4.3 Robustness Test

In this experiment, we aim at evaluating the ability to perform the completion with a varying size of removed geometry. The models used in this experiment are the same trained models from the comparison in Sect. 4.2. Recall that each input shape undergoes a splitting operation where a partial shape and a missing part are computed during training. The missing part's location is random, and the size depends on a given radius around the selected point. In the previous comparison, we used a radius of 0.35. We now measure the robustness of the trained models to perform the completion when the test shape has a missing part with different sizes. We vary the radius from 0.25 to 0.55 and compute the average Chamfer distance for every algorithm. Figure 5 shows the results.

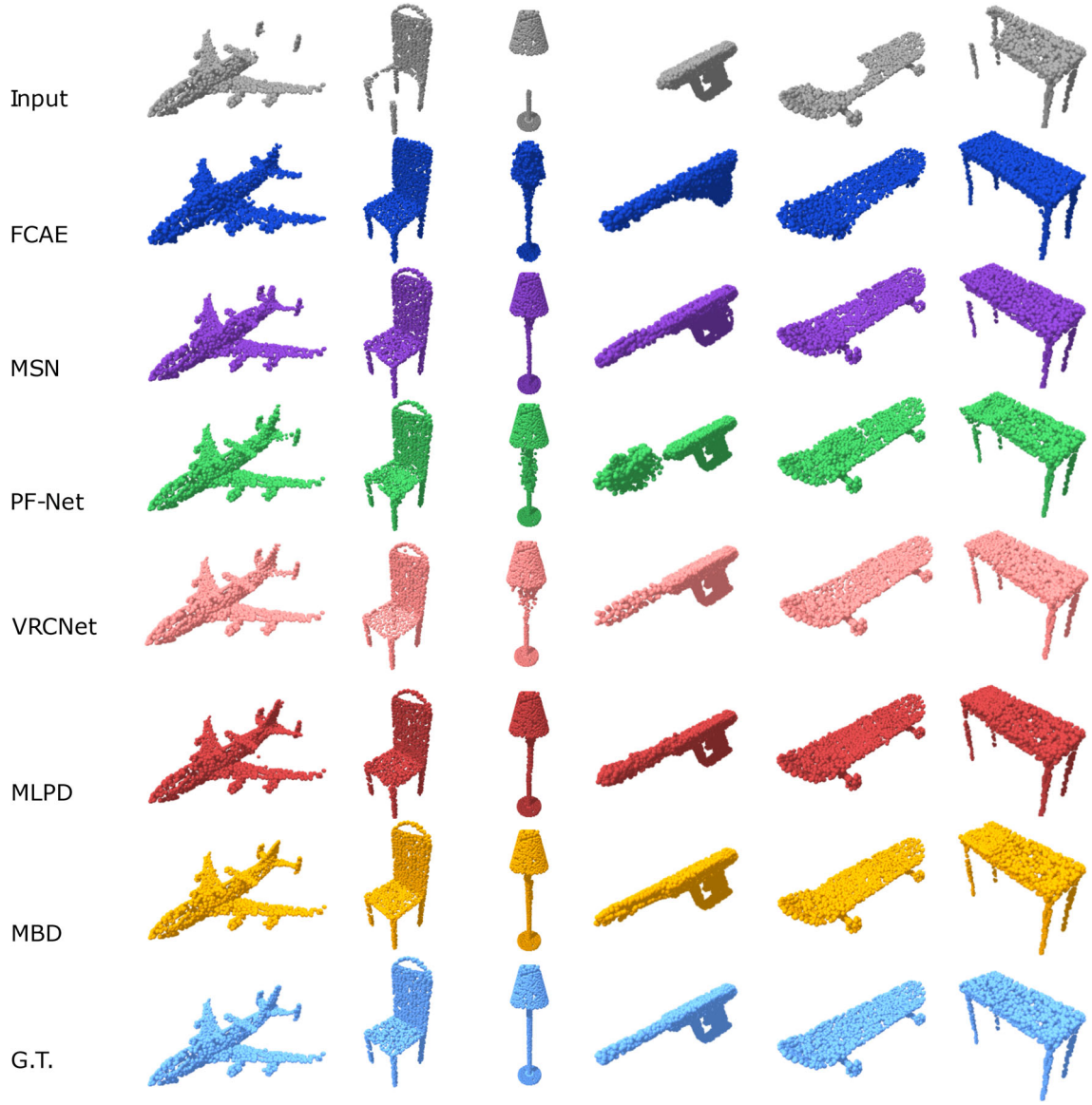

Fig. 4 Comparison of completion results between our method and state-of-the-art methods. FCAE predicts reasonable coarse shapes but fails to complete details. MSN (Liu et al., 2020) produces the entire shape but struggles to find some details. PF-Net (Huang et al., 2020) predicts the coarse missing part, but it cannot incorporate the prediction with the input. Our methods (MBD and MLPD) successfully predict the missing part and integrate it to produce a good result. In these examples, our method and VRCNet (Pan et al., 2021) are the methods that produces the best visual results

Our methods (MBD and MLPD) consistently outperform the other methods when the missing part's size varies from 0.25 to 0.45. Nevertheless, the performance of our methods degrades when the radius is above 0.5. In our opinion, the reason for this behavior is the extent of the missing part that is not well covered by our fixed number of points to represent the missing part. In this case, methods that predict the complete point cloud show better robustness. One of our methods' main limitations is the lack of adaptativeness to represent the missing part according to its extent. The computation of adaptive varying-size point clouds via neural networks is a promising idea to incorporate in our approach in future researches to increase robustness.

5 Experiments on Completion3D

Completion3D (Tchapmi et al., 2019) is an online benchmark to evaluate the performance of point cloud completion algorithms. Participants must train their models on the train-

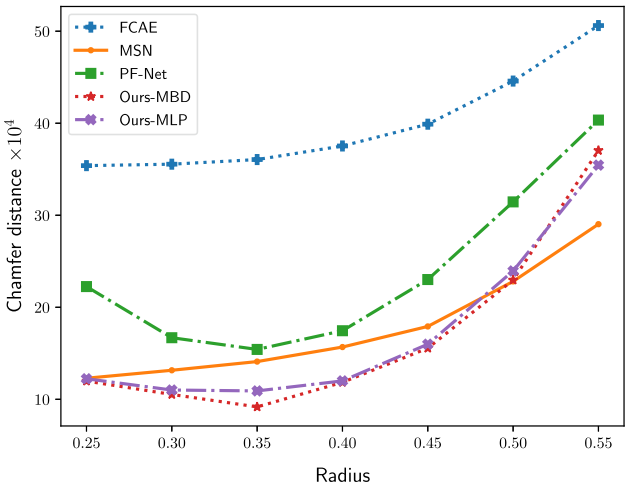

Fig. 5 Robustness against variable size of missing parts. Our methods are robust when the size of missing part is similar to the size of the trained missing part

ing set and upload the inferred completion on the test set. The platform automatically evaluates the results and delivers the performance using the Chamfer Distance. The Completion3D dataset is built on partial views of objects from ShapeNet dataset.

The training set contains the set of partial views and target complete point clouds. However, our method also requires the missing part since the first module in our architecture predicts the missing part and compares the result with the real missing part. Therefore, we have to preprocess the Completion3D training data to obtain the missing part for each partial view. Let P be the partial point cloud and T be the corresponding complete point cloud, we compute the missing part H as follows:

$$H = \left\{ p \in T \text{ such that } \min_{q \in P} \| p - q \| > 10^{-4} \right\} \quad (3)$$

In other words, the missing part H contains the target points that do not overlap with the partial point cloud. The threshold (10^{-4}) was found empirically. Even with the small threshold, we note that this procedure creates a discontinuity between the partial view and the missing part; nevertheless, this problem is fixed in our model with the refinement step which tends to close the gap between the partial point cloud and the predicted missing part.

Table 2 shows the results on this benchmark. We compare our method with previous methods such as FoldingNet (Yang et al., 2018), AtlasNet (Groueix et al., 2018), PCN (Yuan et al., 2018), TopNet (Tchapmi et al., 2019) and VRCNet (Pan et al., 2021). Our method obtains competitive results, outperforming methods such as FoldingNet, AtlasNet, and PCN. Tecniques such as TopNet and VRCNet use hierarchical approaches to learn intrincate details in shapes, and we believe this is one of the reason of their superior performance. Even when our method is simple in nature, the average Chamfer distance is close to the result from TopNet.

6 Ablation Study

In this section, we evaluate the impact of two important components in our proposal: the marging and sampling module (block B in our architecture in Fig. 1) and the refinement module (block C in our architecture in Fig. 1).

6.1 The Merging and Sampling Module

The part prediction module (block A in our architecture) generates the missing point cloud and concatenate it with the partial input. As a result, the point cloud in this stage has $1.5 \times N$ points, where N is the size of the input point cloud. Since our method uses the EMD loss function, we apply a sampling strategy to select only N points which are fed to the refinement step. In this experiment, we evaluate the performance of our proposal if we remove the sampling and feed the refinement step directly with the concatenated point cloud of size $1.5 \times N$. It means the predicted point cloud has

Table 2 Quantitative comparison on Completion3D dataset using average Chamfer distance

Categories	FoldingNet	AtlasNet	PCN	TopNet	VRCNet	MBD
Airplane	12.83	10.36	9.79	7.32	3.94	7.42
Cabinet	23.01	23.40	22.70	18.77	10.93	25.89
Car	14.88	13.40	12.43	12.88	6.44	11.35
Chair	25.69	24.16	25.14	19.82	9.32	22.91
Lamp	21.79	20.24	22.72	14.60	8.32	24.16
Sofa	21.31	20.82	20.26	16.29	11.35	20.52
Table	20.71	17.52	20.27	14.89	8.60	16.10
Watercraft	11.51	11.62	11.73	8.82	5.78	7.95
Average	19.07	17.77	18.22	14.25	8.12	17.16

Results are scaled by 10,000

Table 3 Merging and sampling is important to improve the quality of the completion

Categories	MBD	MBDw/o Sampling
Airplane	**4.675**	29.038
Bag	**16.706**	34.488
Cap	**11.096**	115.648
Car	**9.466**	28.954
Chair	**6.629**	24.784
Guitar	**3.813**	47.081
Lamp	**23.257**	66.946
Laptop	**3.790**	45.070
Motorbike	**10.843**	21.181
Mug	**7.430**	167.613
Pistol	**8.676**	34.128
Skateboard	**4.545**	30.072
Table	**8.333**	32.572
Average	**9.174**	52.121

Our methods consistently outperform the results against not using merging and sampling. Results are scaled by 10,000
Best values in bold

$1.5 \times N$ points while the ground truth has N points. Therefore, we use the Chamfer distance for this specific experiment to cope with the different point cloud sizes in the final loss.

For the evaluation, note that the Chamfer distance is proportional to the number of points in the comparison. Therefore, to make a fair comparison, we sample 2048 points using farthest point sampling to compute the metrics during test. Table 3 presents the results of this experiment.

6.2 The Refinement Module

An essential contribution of our methods is the combination of missing part prediction plus the final refinement. In this experiment, we show how important refinement is in the completion result. We take the networks trained in the experiment of Sect. 4.2 and evaluate the completion performance with and without the refinement module. Table 4 shows the effect of using the Point Refiner Network in our models.

The results show the importance of refinement in the final output. The improvement using the refinement is consistent in our two MBD and MLPD. Note that the more significant gain is in the challenging classes: Bag, Cap, and Lamp. In effect, these results demonstrate that the refinement can be a useful complement to the missing part prediction, to the point that it can help deal with unbalanced data and high intra-class variability. Figure 6 shows two examples of the effect of the refinement in our proposal.

7 Completion of Archeological Pottery

7.1 Data Generation and Training

Any data-driven approach for shape completion requires a considerable amount of data to learn the geometry's characteristics and the mapping between partial input and completed outputs. Unfortunately, in our application, there is no large existing dataset of cultural heritage objects to learn the specific characteristics of this kind of object. Nevertheless, we address this challenging issue by making a key

Table 4 Refinement is important to improve the quality of the completion

Categories	MBD	MBDw/o Ref	MLPD	MLPDw/o Ref
Airplane	**4.675**	5.638	**5.105**	6.225
Bag	**16.706**	20.325	**19.980**	25.433
Cap	**11.096**	18.701	**19.543**	31.026
Car	**9.466**	10.263	**11.199**	12.571
Chair	**6.629**	7.967	**8.100**	9.680
Guitar	**3.813**	4.194	**4.203**	4.721
Lamp	**23.257**	28.289	**25.064**	38.002
Laptop	**3.790**	5.145	**4.262**	5.972
Motorbike	**10.843**	11.734	**10.359**	13.120
Mug	**7.430**	9.624	**10.518**	13.897
Pistol	**8.676**	10.330	**9.350**	12.122
Skateboard	**4.545**	5.377	**4.687**	6.516
Table	**8.333**	10.706	**9.459**	12.272
Average	**9.174**	11.415	**10.910**	14.735

Our methods consistently outperform the results against not using refinement. The improvement is even more evident in challenging classes such as Cap and Lamp. Results are scaled by 10,000
Best values in bold

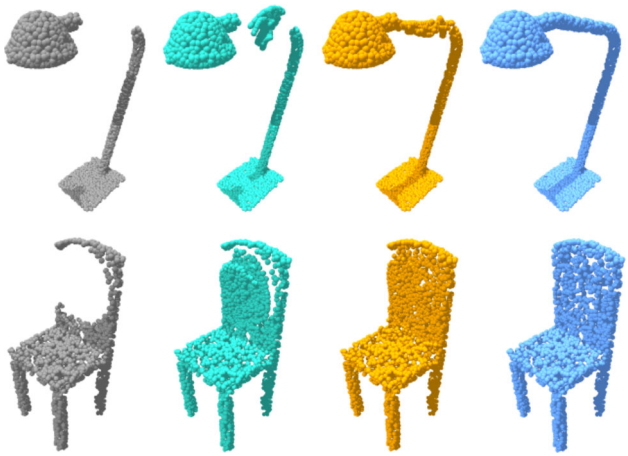

Fig. 6 The refinement is a key ingredient in our proposal. The predicted missing part is often a coarse representation which is considerably enhanced by the point refiner network. From left to right: input, input+predicted missing part, input+refined missing part, ground-truth

observation about the typical structure of cultural heritage objects (and human-made objects, in general).

We collected a dataset of objects with a structure similar to those commonly found in archaeological objects. We used the 3D Pottery Benchmark (Koutsoudis et al., 2010), which contains 1012 pottery models, as a starting point. To increase the number of models for training, we added two categories from the Shapenet dataset (Chang et al., 2015), namely: Bowl (containing 186 models) and Jar (containing 596 models). Also, we added 84 models from the Bowl category from the ModelNet40 dataset (Zhirong et al., 2015). In total, we collected 1890 objects. Subsequently, we manually removed objects with a dissimilar structure to the scanned set of archaeological objects. We ended up with a total of 1458 objects for this experiment. Figure 7 shows some examples from our training set for our application in restoration of pottery objects.

We apply a pre-processing stage in the collected objects: We manually corrected the objects' orientation so that the orientation up is in the positive Z-axis. This step guarantees that the training objects have the same orientation as the scanned objects. For every model in the dataset, we sample 8192 points from the surface. The point cloud is normalized in scale to fit a sphere of radius one. We also translate the point cloud so the center of the objects' mass coincides with the origin in 3D space.

During training, for each epoch we generate 50 partial point clouds with their corresponding missing point clouds for each training object. However, the protocol to generate the partial point clouds has a crucial difference with respect to the procedure applied in ShapeNet experiments. In Sect. 4, the normalization of position and scale assumes that we already know the complete object (ground truth). However, in our CH application, the input is an incomplete object and therefore, there is no way to know the real center of mass or scale of the object we want to predict. To deal with this problem, we look for introducing the invariance to transformations during training. Since we want to add invariance to affine transformations (translation, scaling and rotation), we design a procedure to perform the splitting as follows:

- Scaling by a random factor in the range [0.5, 1.5]. This operation introduces invariance to the object size, which is important since we do not know the scanned object's scale in advance.
- Rotation around Z-axis with a random angle in the range $[0, 2\pi]$. We assume that input objects are approximately well oriented concerning the Z-axis. However, the object could be oriented at any angle around the Z-axis. This operation introduces invariance to the orientation in the XY plane.
- Rotation around the X-axis with a random angle in the range $[0, \pi/12]$. Our assumption of orientation regarding the Z-axis only is approximated. Hence, this operation introduces invariance to a certain degree of misalignment of the object.
- Translation in the Z direction by a random amount in the range $[-0.3, 0.3]$. This operation introduces invariance to the original location of the object. In particular, the position normalization of the ground truth is based on the center of the mass. However, it is impossible to know the center of mass of a damaged object because of the missing

Fig. 7 Example shapes in our training set for restoration of pottery objects

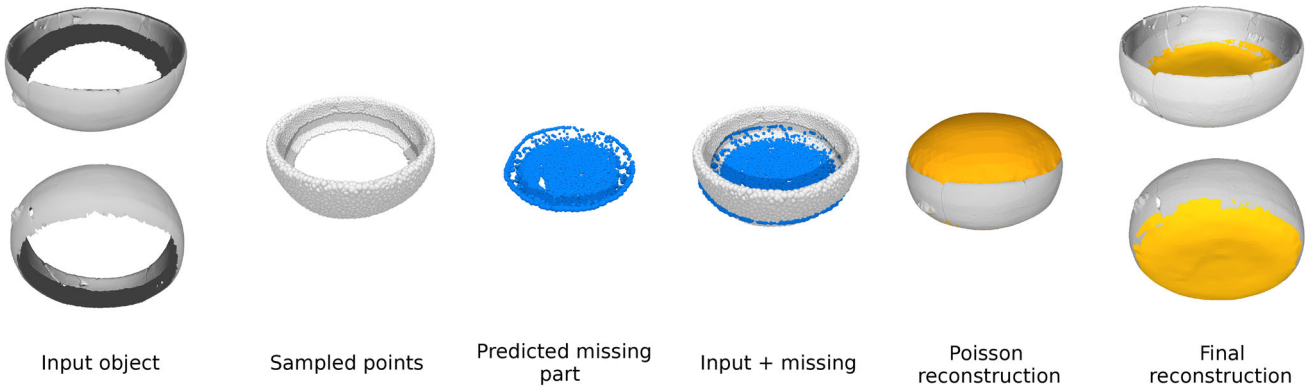

Fig. 8 Reconstruction of cultural heritage objects. From left to right: The input object lacks the base (top and bottom views). The input object is sampled many times. For each sampling, the neural network predicts the missing point cloud. The resulting reconstruction is the concatenation of the input point cloud with the predicted missing point cloud. Poisson reconstruction method creates a triangular mesh that approximates the point cloud. A post-processing step removes the unnecessary geometry and keeps the reconstruction of the input object

information. This operation introduces invariance to the location of the object.

- Translation in X and Y directions by a random amount in the range $[-0.03, 0.03]$. This operation introduces invariance to small shifts in the location of the object with respect to axes X and Y.
- Object splitting into partial and missing point cloud. Let z_{min} and z_{max} be the minimum and maximum Z coordinate of the object, respectively. Let h_z be the height of the object, where $h_z = z_{max} - z_{min}$. We select a random number r in the range $[0.05, 0.175]$ and compute the partial and missing point cloud as follows:

$$partial(P) = \{p \in P \| \; p_z \geq z_{min} + r \times h_z\} \quad (4)$$

$$missing(P) = \{p \in P \| \; p_z < z_{min} + r \times h_z\} \quad (5)$$

The value $z_{min} + r \times h_z$ denotes the Z coordinate of a plane that is parallel to the XY plane. Points above this plane (in the Z+ direction) go in the partial shape and points below this plane (in the Z− direction) go in the missing part.

7.2 Reconstruction Procedure

We use the model trained in the previous section to reconstruct a collection of damaged objects scanned from the Josefina Ramos de Cox museum in Lima, Peru. Algorithm 1 details the procedure to reconstruct an input object. Figure 8 also shows an overview of the procedure.

The input to the algorithm is the scanned triangular mesh and the trained neural network. The first stage (line 1) is to normalize the scale of the input object. This step is necessary since we perform the training with normalized data. The second stage (lines 2–6) predicts the missing part of the damaged object. Recall that our neural network receives 2048 points and produces a missing part of 1024 points. Our strategy to obtain a reconstruction with more points is to apply the neural network many times with several point samplings. In this application, we apply the neural network eight times (lines 2–6) and accumulate the input point clouds and missing part predictions in sets P and H, respectively. In this second stage, the output point cloud is the union of sets P and H (line 7). The third stage (lines 8–10) reconstructs the final point cloud into a triangular mesh that completes the original triangular mesh. We show that a Poisson reconstruction is suitable to get a triangular mesh from the point cloud. However, the Poisson reconstruction has problems dealing with the open structure of the top of the objects. Subsequently, we simply remove the reconstructed geometry above a certain level in the Z-axis. The user could prefer to remove or keep more geometry depending on their needs and the final visual result.

Algorithm 1: Shape reconstruction algorithm

Input: Triangular mesh S (the input object), neural network for completion NN
Output: Triangular mesh S' (the missing surface)

1 Scale S to fit into a unit cube
2 **for** i = *1 to 8* **do**
3 pcd = sample_points(S, 2048)
4 missing = NN.predict(pcd)
5 $P = P \cup$ pcd
6 $H = H \cup$ missing
7 $M = P \cup H$ `// The final point cloud is the union of partial and missing parts.`
8 Scale M to the original size of S
9 Apply Poisson reconstruction to M to obtain a triangular surface representation of the point cloud
10 Post-process the surface to preserve only valid geometry.
11 Return the final surface S'

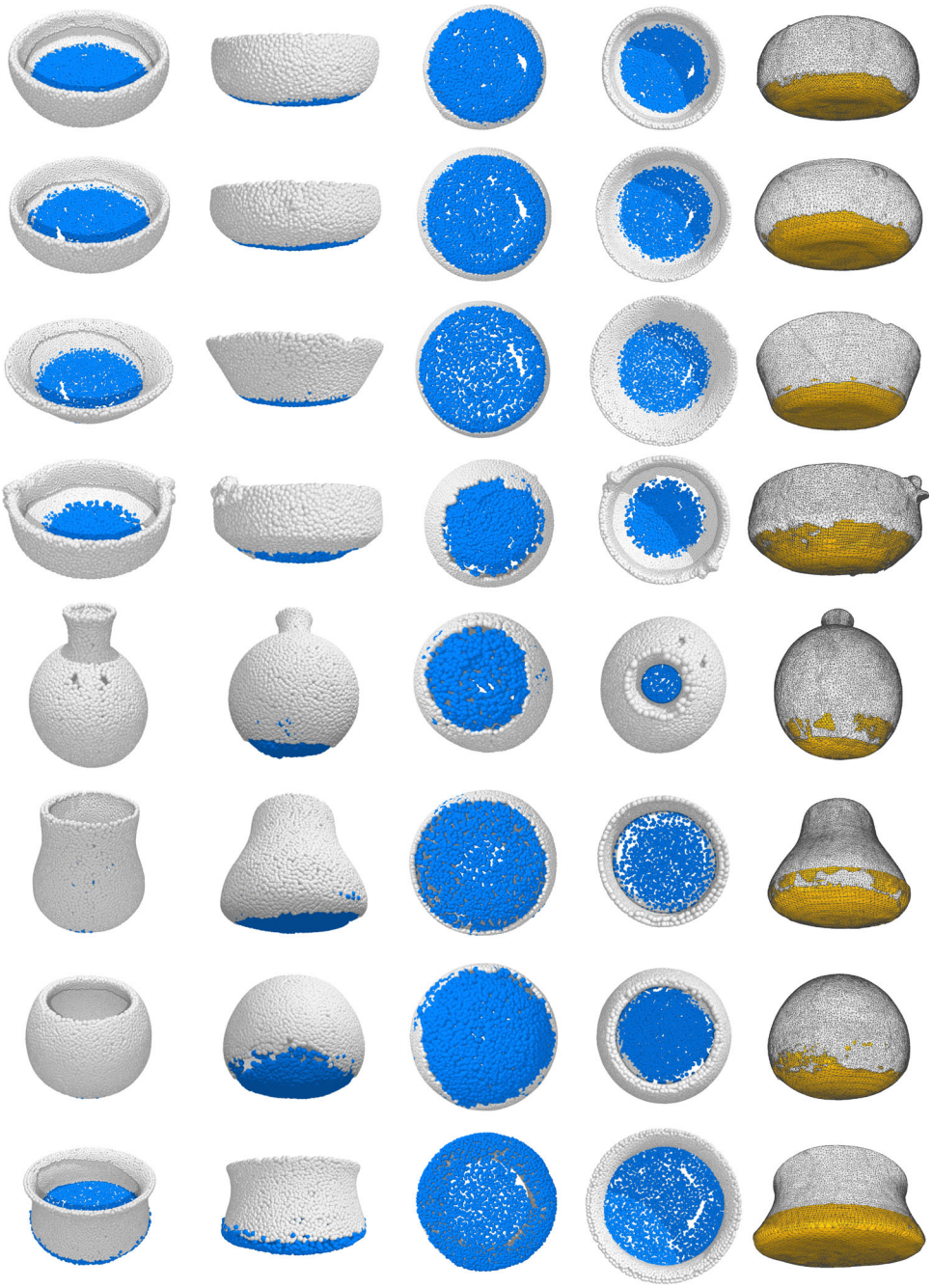

Fig. 9 Results of our method to repair the surface of CH objects in the Josefina Ramos de Cox museum, Lima, Peru. The first four columns depict the input point cloud and the predicted point cloud in several views (perspective, lateral, bottom, top, respectively). The fifth column shows the reconstructed surface for the final repair

Figure 9 shows the results of applying our model to repair objects. Our method recovers the geometry of the object's base despite the high variability of input shapes. This is a result of building our training dataset by collecting objects with different shapes but still similar to objects that one would find in a CH application. Our method is able to learn to synthesize the missing geometry and make smooth transitions between the original scanned surface and the prediction.

We also train the MSN (Liu et al., 2020) and VRCNet (Pan et al., 2021) methods with the pottery dataset to make a qualitative comparison with our method. Figure 10 shows the results of the compared method with two point clouds from real pottery objects. The advantage of our method is that the partial input (which is reliable geometry because it was scanned from a real object) remains unchanged in the final result, since our method focus on generating the missing part.

Fig. 10 Results on point clouds from real pottery objects. From left to right: the first column shows the partial input, the second column shows the results of MSN (Liu et al., 2020), the third column shows the results from VRCNet (Pan et al., 2021), and the fourth column shows the results from our method

8 Conclusion

We describe a data-driven method to repair large imperfections during scanning in the Josefina Ramos de Cox museum in Lima, Peru. To address this problem, we presented a novel architecture for point cloud completion that emphasizes the combination of missing part prediction and point cloud refinement to improve the completion. Unlike other architectures that predict the overall shape, our method retains the existing geometry and refined details while focusing on predicting and integrating the missing part. Our experiments in the ShapeNet dataset, under a standard protocol to evaluate completion algorithms, show that our method is robust and effective. Moreover, the refinement of the merge between the incomplete input and the predicted missing part is crucial to reduce the reconstruction error in challenging cases.

When trained in a more specific scenario for a cultural heritage domain, our proposal has several advantages. First, our method preserves the original geometry, and it only predicts the missing surface. This property is essential in archaeological research where the original object's characteristics should not be changed at all. Second, our method and the repair application are parameter-free. Once the neural network has been trained with a specific protocol for the problem to solve, the inference does not require user intervention or final users' modeling capabilities. Third, even when the representation is a point cloud, the quality of the prediction enables the use of well-established methods for surface reconstruction. The result is a surface with smooth transitions between the input and the prediction. Therefore, all these advantages make our proposal suitable to use in the cultural heritage domain.

The main limitation of our method is the dependency on the problem to solve. In this paper, we focused on repair a specific imperfection that occurred during scanning. Nevertheless, if one wants to apply a similar methodology to a more general problem, such as repair or reconstruction of damaged CH objects, one would require a different training protocol. In this regard, we envision two potential developments that could lead to a significant impact in applying computer vision to cultural heritage. First, we need large datasets of CH objects to build more specific deep learning models. Although we have shown that a deep learning model can somehow transfer knowledge from a general-purpose dataset to CH objects, more specific datasets could benefit the learning process, thereby also adopting these models in more complex problems in cultural heritage. Second, we believe that the introduction of geometric reasoning (such as structure knowledge, invariances, and so on) in the learning process could help to improve the explainability of the trained model. In the eyes of expert archaeologists and curators, an automatic repair tool that explains the result would help to facilitate the incorporation of cutting-edge technology in their work.

Acknowledgements The work of Ivan Sipiran has been partially supported by ANID Chile—Research Initiation Program—Grant N° 11220211. The work of Cristian Lopez has been supported by Proyecto de Mejoramiento y Ampliacion de los Servicios del Sistema Nacional de Ciencia Tecnologia e Innovacion Tecnologica(Banco Mundial, Concytec-Peru), Nr. Grant 062-2018-FONDECYT-BM-IADT-AV. Thanks to the Josefina Ramos de Cox Museum for the continuous support during the realization of this project.

References

Achlioptas, P., Diamanti, O., Mitliagkas, I., & Guibas, L. J. (2018). Learning representations and generative models for 3D point clouds. In J. G. Dy, & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018, PMLR, Proceedings of machine learning research* (Vol. 80, pp. 40–49).

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., & Xiao, J. (2015). Shapenet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012

Charles R. Q., Su, H., Kaichun, M., & Guibas L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 77–85).

Dai, A., Qi, C. R., & Nießner, M. (2017). Shape completion using 3D-encoder-predictor CNNs and shape synthesis. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 6545–6554).

Fan, H., Su, H., & Guibas, L. (2017) A point set generation network for 3D object reconstruction from a single image. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2463–2471).

Gropp, A., Yariv, L., Haim, N., Atzmon, M., & Lipman, Y. (2020). Implicit geometric regularization for learning shapes. In *Proceedings of the 37th international conference on machine learning, ICML 2020, 13–18 July 2020, Virtual Event, PMLR, proceedings of machine learning research* (Vol. 119, pp. 3789–3799).

Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., & Aubry, M. (2018). A Papier-Mache approach to learning 3d surface generation. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 216–224).

Han, X., Li, Z., Huang, H., Kalogerakis, E., & Yu, Y. (2017). High-resolution shape completion using deep neural networks for global

structure and local geometry inference. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 85–93).

Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., & Cohen-Or, D. (2019) MeshCNN: A network with an edge. *ACM Transactions on Graphics, 38*(4).

Harary, G., Tal, A., & Grinspun, E. (2014a) Context-based coherent surface completion. *ACM Transactions on Graphics, 33*(1), 5:1–5:12.

Harary, G., Tal, A., & Grinspun, E. (2014b). Feature-preserving surface completion using four points. *Computer Graphics Forum, 33*(5), 45–54.

Hermoza, R., & Sipiran, I. (2018). 3D reconstruction of incomplete archaeological objects using a generative adversarial network. In *Proceedings of computer graphics international 2018, CGI 2018* (pp. 5–11). Association for Computing Machinery.

Hu, T., Han, Z., Shrivastava, A., & Zwicker, M. (2019). Render4completion: Synthesizing multi-view depth maps for 3D shape completion. In *2019 IEEE/CVF international conference on computer vision workshop (ICCVW)* (pp. 4114–4122).

Hu, T., Han, Z., & Zwicker, M. (2020). 3D shape completion with multi-view consistent inference. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, The tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020* (pp. 10997–11004). AAAI Press.

Huang, H., Gong, M., Cohen-Or, D., Ouyang, Y., Tan, F., & Zhang, H. (2012). Field-guided registration for feature-conforming shape composition. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2012), 31*, 171:1–171:11.

Huang, Z., Yu, Y., Xu, J., Ni, F., & Le, X. (2020). PF-Net: Point fractal network for 3D point cloud completion. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 7659–7667).

Jiang, W., Xu, K., Cheng, Z. Q., & Zhang, H. (2013). Skeleton-based intrinsic symmetry detection on point clouds. *Graphical Models, 75*(4), 177–188.

Koutsoudis, A., Pavlidis, G., Liami, V., Tsiafakis, D., & Chamzas, C. (2010). 3D pottery content-based retrieval based on pose normalisation and segmentation. *Journal of Cultural Heritage, 11*(3), 329–338.

Li, E., Zhang, X., & Chen, Y. (2014). Symmetry based Chinese ancient architecture reconstruction from incomplete point cloud. In *2014 5th International conference on digital home* (pp. 157–161).

Liu, F., Tran, L., & Liu, X. (2019). 3D face modeling from diverse raw scan data. In *2019 IEEE/CVF international conference on computer vision (ICCV)* (pp. 9407–9417).

Liu, M., Sheng, L., Yang, S., Shao, J., & Hu, S. (2020). Morphing and sampling network for dense point cloud completion. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020* (pp. 11596–11603). AAAI Press.

Maturana, D., & Scherer, S. (2015). VoxNet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 922–928).

Mavridis, P., Sipiran, I., Andreadis, A., & Papaioannou, G. (2015). Object completion using k-sparse optimization. *Computer Graphics Forum, 34*(7), 13–21.

Pan, L., Chen, X., Cai, Z., Zhang, J., Zhao, H., Yi, S., & Liu, Z. (2021). Variational relational point completion network. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 8520–8529). IEEE Computer Society.

Papaioannou, G., Schreck, T., Andreadis, A., Mavridis, P., Gregor, R., Sipiran, I., & Vardis, K. (2017). From reassembly to object completion: A complete systems pipeline. *Journal on Computing and Cultural Heritage (JOCCH), 10*(2).

Pauly, M., Mitra, N. J., Giesen, J., Gross, M., & Guibas, L. J. (2005). Example-based 3D scan completion. In *Proceedings of the third eurographics symposium on geometry processing, SGP'05*. Eurographics Association.

Pratikakis, I., Savelonas, M., Mavridis, P., Papaioannou, G., Sfikas, K., Arnaoutoglou, F., & Rieke-Zapp, D. (2018). Predictive digitisation of cultural heritage objects. *Multimedia Tools and Applications, 77*(10), 12991–13021.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30, pp. 5099–5108). Curran Associates, Inc.

Qi, C. R., Chen, X., Litany, O., & Guibas, L. J. (2020). Imvotenet: Boosting 3D object detection in point clouds with image votes. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 4403–4412).

Sipiran, I. (2017). Analysis of partial axial symmetry on 3D surfaces and its application in the restoration of cultural heritage objects. In *2017 IEEE international conference on computer vision workshops (ICCVW)* (pp. 2925–2933).

Sipiran, I. (2018). Completion of cultural heritage objects with rotational symmetry. In *Proceedings of the 11th eurographics workshop on 3D object retrieval, eurographics association, Goslar, DEU, 3DOR'18* (pp. 87–93).

Sipiran, I., Gregor, R., & Schreck, T. (2014). Approximate symmetry detection in partial 3D meshes. *Computer Graphics Forum, 33*(7), 131–140.

Son, K., Almeida, E. B., & Cooper, D. B. (2013). Axially symmetric 3D pots configuration system using axis of symmetry and break curve. In *2013 IEEE conference on computer vision and pattern recognition* (pp. 257–264).

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3D shape recognition. In *2015 IEEE international conference on computer vision (ICCV)* (pp. 945–953).

Sun, Y., Wang, Y., Liu, Z., Siegel, J. E., & Sarma, S. E. (2020). Pointgrow: Autoregressively learned point cloud generation with self-attention. In *2020 IEEE winter conference on applications of computer vision (WACV)* (pp 61–70).

Tchapmi, L. P., Kosaraju, V., Rezatofighi, H., Reid, I., & Savarese, S. (2019). Topnet: Structural point cloud decoder. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 383–392).

Thrun, S., & Wegbreit, B. (2005). Shape from symmetry. In *Tenth IEEE international conference on computer vision, 2005. ICCV 2005* (Vol. 2, pp. 1824–1831).

Wang, X., Ang, M. H., & Lee, G. H. (2020). Cascaded refinement network for point cloud completion. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 787–796).

Wen, X., Li, T., Han, Z., & Liu, Y. S. (2020). Point cloud completion by skip-attention network with hierarchical folding. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 1936–1945).

Wen, X., Han, Z., Cao, Y. P., Wan, P., Zheng, W., & Liu, Y. S. (2021a). Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 13075–13084).

Wen, X., Xiang, P., Han, Z., Cao, Y. P., Wan, P., Zheng, W., & Liu, Y. S. (2021b). PMP-Net: Point cloud completion by learning multi-step

point moving paths. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 7439–7448).

Xiang, P., Wen, X., Liu, Y. S., Cao, Y. P., Wan, P., Zheng, W., & Han, Z. (2021). Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *2021 IEEE/CVF international conference on computer vision (ICCV)* (pp. 5479–5489).

Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., & Sun, W. (2020). Grnet: Gridding residual network for dense point cloud completion. In A. Vedaldi, H. Bischof, T. Brox, & J. M. Frahm (Eds.), *Computer vision—ECCV 2020* (pp. 365–381). Springer.

Xu, K., Zhang, H., Tagliasacchi, A., Liu, L., Li, G., Meng, M., & Xiong, Y. (2009). Partial intrinsic reflectional symmetry of 3D shapes. *ACM Transactions on Graphics, 28*(5), 138:1–138:10.

Yang, Y., Feng, C., Shen, Y., & Tian, D. (2018). Foldingnet: Point cloud auto-encoder via deep grid deformation. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 206–215).

Yi, L., Kim, V. G., Ceylan, D., Shen, I. C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., & Guibas, L. (2016). A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics, 35*(6).

Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., & Zhou, J. (2021). Pointr: Diverse point cloud completion with geometry-aware transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)* (pp. 12478–12487).

Yuan, W., Khot, T., Held, D., Mertz, C., & Hebert, M. (2018). PCN: Point completion network. In *2018 International conference on 3D vision (3DV)* (pp. 728–737).

Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N. J., Cohen-Or, D., & Chen, B. (2010). Non-local scan consolidation for 3D urban scenes. *ACM Transactions on Graphics, 29*(4), 94:1–94:9.

Zhirong, W., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). 3D shapenets: A deep representation for volumetric shapes. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1912–1920).