

# DeepMend: Learning Occupancy Functions to Represent Shape for Repair

Nikolas Lamb<sup>Ⓛ</sup>, Sean Banerjee<sup>Ⓛ</sup>, and Natasha Kholgade Banerjee<sup>Ⓛ</sup>

Clarkson University, Potsdam, NY 13699, USA  
{lambne,sbanerje,nbanerje}@clarkson.edu

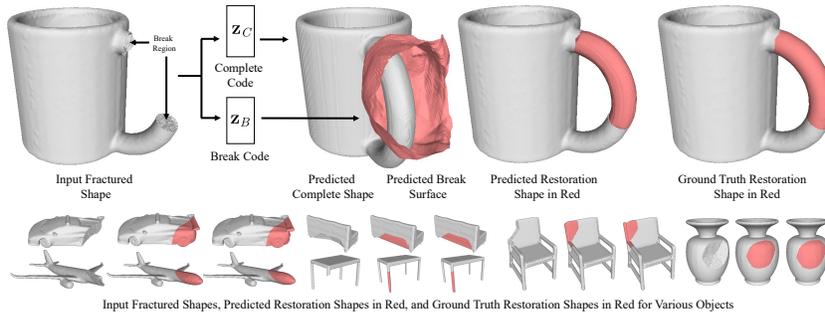
**Abstract.** We present DeepMend, a novel approach to reconstruct restorations to fractured shapes using learned occupancy functions. Existing shape repair approaches predict low-resolution voxelized restorations or smooth restorations, or require symmetries or access to a pre-existing complete oracle. We represent the occupancy of a fractured shape as the conjunction of the occupancy of an underlying complete shape and a break surface, which we model as functions of latent codes using neural networks. Given occupancy samples from a fractured shape, we estimate latent codes using an inference loss augmented with novel penalties to avoid empty or voluminous restorations. We use the estimated codes to reconstruct a restoration shape. We show results with simulated fractures on synthetic and real-world scanned objects, and with scanned real fractured mugs. Compared to existing approaches and to two baseline methods, our work shows state-of-the-art results in accuracy and avoiding restoration artifacts over non-fracture regions of the fractured shape.

**Keywords:** Learned Occupancy, Shape Representation, Repair, Fracture, Implicit Surface, Neural Networks

## 1 Introduction

Automated restoration of fractured shapes is an important area of study, with applications in consumer waste reduction, commercial recycling, cultural heritage object restoration, medical fields such as orthopedics and dentistry, and robot-driven repair. Despite its wide application, automated repair of fractured shapes has received little attention. Most current automated techniques use symmetries to complete fractured shapes [17, 37]. These techniques do not generalize to objects with non-symmetrical damage. The only existing generalizable approaches for repair operate in low-resolution voxel space [22], or infer restoration shapes directly, resulting in smooth restorations with a high rate of failure [25].

In this work, we present DeepMend, a novel deep learning based approach to generate high-fidelity restoration shapes given an input fractured shape. Our approach is inspired by work that learns the signed distance function (SDF) or occupancy function to implicitly represent a shape surface over the continuous 3D domain [7, 14, 21, 38, 45, 51, 51, 57, 62]. These approaches perform partial shape



**Fig. 1.** Given a fractured shape, our approach infers latent codes for an underlying complete shape and a break surface. We use the codes to generate a restoration shape that repairs the input fracture shape.

completion by inferring a latent code using point samples from a partial shape, and reconstruct the complete shape from the latent code. Different from partial shape completion, DeepMend addresses the challenge that, unlike a partial shape that is a subset of a complete shape, a fractured shape contains a novel break region that is missing in the complete shape, as shown on the left of Figure 1. DeepMend leverages the autoencoder architecture from DeepSDF [38] to estimate restorations, by addressing key challenges in the autoencoder that prevent its direct use in restoration generation. DeepSDF is geared toward single shape representation. By itself, DeepSDF cannot predict one shape, i.e., the restoration, from point samples of a different shape, i.e., the fractured shape. To solve this, our first contribution is to represent occupancies for fractured and restoration shapes as logical operations on occupancies for a shape space common to them both, i.e., complete shapes and break surfaces. We use T-norms [19] to relax the logical operations into arithmetic operations. We restructure the autoencoder loss using the fractured shape’s dependence on complete/break codes, and reconstruct the restoration from codes estimated at inference.

Our approach estimates two codes (complete and break) from point samples of a single fractured shape at inference, a more complex problem than DeepSDF, that estimates one code. Using the autoencoder loss alone, trivial codes can be estimated where the break surface lies outside the complete shape or coincides with the fractured shape surface yielding empty or voluminous restorations. Our second contribution is to augment the inference loss with two penalty terms—(i) a non-empty restoration term that penalizes the mean restoration occupancy against being zero to avoid empty restorations, and (ii) a proximity term that encourages the mean distance between the complete and fractured occupancy to be low to prevent voluminous restorations. We compute the restoration occupancy as the conjunction of the complete occupancy and the negation of the break occupancy, and obtain a restoration mesh using Marching Cubes [33].

We train and test our approach on synthetically fractured meshes from 8 classes in the ShapeNet [5] dataset, and on the Google Scanned Objects

dataset [42] which contains 1,032 scanned real-world objects. We use ShapeNet-trained networks to restore synthetically fractured meshes from the QP Cultural Heritage dataset [24], and to generate restorations for physically fractured and scanned real-world mugs. We compare our work to 3D-ORGAN [22] and MendNet [25], the only existing automated fracture restoration approaches, and to two baselines. We show state-of-the-art results in overall accuracy and avoiding inaccurate artifacts over non-fracture regions. Our code is available at: <https://github.com/Terascale-All-sensing-Research-Studio/DeepMend>.

## 2 Related Work

**Restoration of Fractured Shapes.** Most existing approaches to generate restoration shapes from fractured shapes rely on shape symmetry [17, 37]. They restore shapes by reflecting non-fractured regions of the shape onto fractured regions and computing the subtraction. These approaches fail to restore asymmetrical shapes or shapes that have non-symmetric fractures. Lamb et al. [26] perform repair without relying on symmetries. However, they require that a complete counterpart be provided as input alongside the fractured shape. The complete shape may not always be available, e.g., in the case of a rare object. Our work only requires the fractured shape as input. 3D-ORGAN [22] performs shape restoration in voxel space using a generative adversarial network. 3D-ORGAN operates at a resolution of  $32 \times 32 \times 32$ , which is insufficient to accurately represent the geometric complexity of the fracture region. Scaling 3D-ORGAN to a voxel resolution necessary to represent fracture is impractical at current dataset volumes and hardware. MendNet [25] overcomes the low resolution of 3D-ORGAN by representing shapes using the occupancy function, similar to DeepMend. However, MendNet represents overall shape structure and the high-frequency break surface using a single latent code, resulting in overly smooth restoration shapes. DeepMend overcomes the challenges of MendNet by learning separate codes for the complete shape and break surface, yielding high resolution restoration shapes that accurately reconstruct high-frequency break geometry.

**Completion of Partial Shapes.** Though not directly related to our work, a large body of prior work focuses on completing shapes from partial shape representations, e.g. depth maps or color images. Recent approaches hypothesize complete shapes from partial shapes using deep networks. Approaches that use point clouds as input [1, 11, 20, 32, 36, 43, 47, 61] lack an intrinsic surface representation. Some approaches predict 3D meshes [18, 60] to incorporate surfaces. These approaches are limited in the complexity of meshes reliably predicted [35], and cannot represent arbitrary topologies. Most approaches using voxels [3, 44, 46, 53] struggle to predict high-resolution outputs while being computationally tractable. Some voxel approaches address computational inefficiency by employing hierarchical models [9, 10] or sparse convolutions [9, 59]. However, voxel approaches pre-discretize the domain, making it challenging to use them to represent arbitrarily fine resolutions needed for geometric detail, especially for the problem of fracture surface representation considered in this work.

A large body of recent work focuses on using neural networks to represent point samples of values that implicitly define surfaces, e.g., occupancy [7, 8, 15, 23, 28, 30, 35, 39, 40, 49, 56, 57], SDF [4, 14, 21, 29, 34, 38, 45, 51, 54, 55, 58, 62], unsigned distance [50], or level sets [16]. By representing shapes as continuous functions, these approaches show high reconstruction fidelity while remaining computationally tractable. In contrast to traditional encoder-decoder architectures, approaches based on the autoencoder introduced by DeepSDF [38] use maximum *a posteriori* estimation to obtain a latent code for an input shape. The approach enables reconstruction of a complete shape using a latent code estimated from observations of an incomplete shape. Later approaches provide improvements by using meta-learning and post-training optimization [45, 58], learning increasingly complex shape representations during training [12], deforming implicit shape templates [62], or reconstructing shapes at multiple resolutions [21].

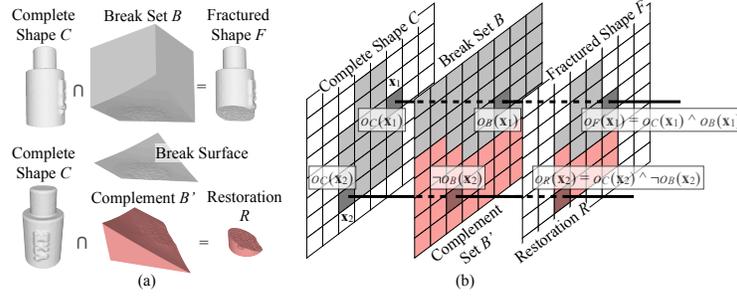
A potential approach for shape restoration is to convert the fractured shape into a partial shape by removing the fracture surface, perform shape completion, and subtract the fractured shape from the complete shape to obtain a restoration. In Section 5 we show that subtraction approaches yield surface artifacts on the non-fracture regions of the fractured shape. Our approach mitigates artifacts by learning the interplay between the complete shape and break surface.

### 3 Representing Fractured Shapes

We represent the complete, fractured, and restoration shapes as point sets  $C$ ,  $F$ , and  $R$ . For  $S \in \{C, F, R\}$  the occupancy  $o_S(\mathbf{x}) \in \{0, 1\}$  of a point  $\mathbf{x}$  is 1 if  $\mathbf{x}$  is inside the shape, and 0 if it is on the boundary or outside the shape. The original shapes are closed surfaces. However, we exclude boundary points from the definitions of the sets  $C$ ,  $F$ , and  $R$  to ensure that a point does not simultaneously belong to two sets, e.g.,  $F$  and  $R$ . Exclusion of boundary points makes the sets  $C$ ,  $F$ , and  $R$  open and bounded. We define the break surface as a 2D surface that intersects the fracture region of  $F$ . Points on the side of the break surface corresponding to the fractured shape receive an occupancy of 1. Points on the side corresponding to the restoration shape have an occupancy of 0. We use the open unbounded set  $B$ , termed the ‘break set’ to represent the set of points with an occupancy  $o_B(\mathbf{x})$  of 1. In principle, the break surface is infinite. In practice, we limit the region containing the break set to be a cube of finite length to make point sampling for network training and inference tractable<sup>1</sup>.

As shown in Figure 2(a), we represent the fractured shape set as the intersection of the sets for the complete shape  $C$  and the break set  $B$ , i.e., as  $F = C \cap B$ . The relationship implies that for a point  $\mathbf{x}$ , occupancy  $o_F(\mathbf{x})$  of the fractured shape  $F$  is the logical conjunction of the occupancies  $o_C(\mathbf{x})$  and  $o_B(\mathbf{x})$  of  $C$  and  $B$ , i.e.,  $o_F(\mathbf{x}) = o_C(\mathbf{x}) \wedge o_B(\mathbf{x})$ . We represent the restoration shape  $R$  as the intersection of  $C$  and the complement of  $B$ , i.e., as  $R = C \cap B'$ . The relationship implies that occupancy  $o_R(\mathbf{x})$  of  $R$  is expressed as the logical conjunction

<sup>1</sup> Hereafter, we drop ‘set’ from references to  $C$ ,  $F$ , and  $R$ , and refer to them as shapes.



**Fig. 2.** (a) We represent fractured and restoration shapes  $F$  and  $R$  as intersections of the complete shape  $C$  with the break set  $B$  and its complement  $B'$ . (b) Logical expressions for occupancy at points  $\mathbf{x}_1$  in  $F$  and  $\mathbf{x}_2$  in  $R$  using  $C$  and  $B$  occupancies.

of  $o_C(\mathbf{x})$  with the negation of  $o_B(\mathbf{x})$ , i.e.,  $o_R(\mathbf{x}) = o_C(\mathbf{x}) \wedge \neg o_B(\mathbf{x})$ . We show the logical relationships in Figure 2(b). For use in neural networks, we relax the logical relationships using the product T-norm [19], as

$$o_F(\mathbf{x}) = o_C(\mathbf{x})o_B(\mathbf{x}) \text{ and} \quad (1)$$

$$o_R(\mathbf{x}) = o_C(\mathbf{x})(1 - o_B(\mathbf{x})). \quad (2)$$

We represent the occupancy functions for the complete shape  $C$  and break set  $B$  respectively using neural networks  $f_{\Theta}$  and  $g_{\Phi}$ , such that  $o_C(\mathbf{x}) = f_{\Theta}(\mathbf{z}_C, \mathbf{x})$  and  $o_B(\mathbf{x}) = g_{\Phi}(\mathbf{z}_B, \mathbf{x})$ .  $\Theta$  and  $\Phi$  are the network weights,  $\mathbf{z}_C \in \mathbb{R}^p$  is a latent code of size  $p$  corresponding to the complete shape, and  $\mathbf{z}_B \in \mathbb{R}^q$  is a latent code of size  $q$  corresponding to the break set. We use the autoencoder architecture introduced by Park et al. [38] for  $f_{\Theta}$  and  $g_{\Phi}$ . Figure 3(a) shows our network structure. We provide network details in the supplementary.

### 3.1 Network Training

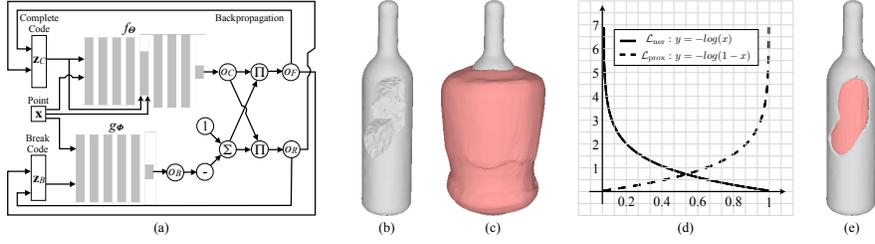
During training, we optimize for network parameters  $\Theta$  and  $\Phi$ , and the latent codes  $\mathbf{z}_B$  and  $\mathbf{z}_C$  over each training sample. Each sample consists of a tuple  $(F, C, R, B)$  representing the fractured, complete, and restoration shapes  $F$ ,  $C$ , and  $R$ , and the break set  $B$  for the sample. We define the training loss as

$$\mathcal{L} = \sum_{\mathbf{z}_C \in Z_C, \mathbf{z}_B \in Z_B} \mathcal{L}_F + \mathcal{L}_C + \mathcal{L}_R + \mathcal{L}_B + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} \quad (3)$$

where  $Z_C$  is the set of all training complete latent codes, and  $Z_B$  is the set of all training break latent codes. The term  $\mathcal{L}_F$ , represented as

$$\mathcal{L}_F = (1/|X|) \sum_{\mathbf{x} \in X} \text{BCE}(f_{\Theta}(\mathbf{z}_C, \mathbf{x})g_{\Phi}(\mathbf{z}_B, \mathbf{x}), o_F(\mathbf{x})), \quad (4)$$

models the reconstruction of the fractured shape occupancy values.  $\text{BCE}$  represents the binary cross-entropy loss function. The first argument to  $\text{BCE}$  represents the occupancy expression from Equation (1), with the expressions for



**Fig. 3.** (a) Networks  $f_{\Theta}$  and  $g_{\Phi}$  represent the complete and break occupancy in terms of an input point  $\mathbf{x}$  and latent codes  $\mathbf{z}_C$  and  $\mathbf{z}_B$ . As shown, the fractured and restoration occupancies,  $o_F$  and  $o_R$ , are obtained using the complete and break occupancies,  $o_C$  and  $o_B$ , where  $\Pi$  and  $\Sigma$  represent the product and sum respectively. (b) Predicted restoration shape with high non-empty penalty and (c) with high proximity penalty. (d) Functions used for the non-empty penalty term  $\mathcal{L}_{\text{ner}}$  and the proximity penalty term  $\mathcal{L}_{\text{prox}}$  during inference. (e) Predicted restoration with balanced penalties. Restoration shapes (red) shown with ground truth fractured shapes (gray).

the complete and break occupancy values in terms of  $f_{\Theta}$  and  $g_{\Phi}$  substituted in. The second argument represents the fractured shape ground truth occupancy values.  $X$  represents the set of point samples used to probe the ground truth occupancy values. We include terms  $\mathcal{L}_C$ ,  $\mathcal{L}_B$ , and  $\mathcal{L}_R$  to improve the representation capability of the network by using ground truth occupancy values from training complete shapes, break sets, and restorations. We define  $\mathcal{L}_C$  and  $\mathcal{L}_B$  as

$$\mathcal{L}_C = (1/|X|) \sum_{\mathbf{x} \in X} BCE(f_{\Theta}(\mathbf{z}_C, \mathbf{x}), o_C(\mathbf{x})) \quad \text{and} \quad (5)$$

$$\mathcal{L}_B = (1/|X|) \sum_{\mathbf{x} \in X} BCE(g_{\Phi}(\mathbf{z}_B, \mathbf{x}), o_B(\mathbf{x})). \quad (6)$$

In Equations (5) and (6), the first argument to  $BCE$  represents the occupancy for the complete shape and break set respectively expressed in terms of  $f_{\Theta}$  and  $g_{\Phi}$ . The second argument represents the ground truth occupancy values for the complete shape  $C$  and break set  $B$ . We define  $\mathcal{L}_R$  as

$$\mathcal{L}_R = (1/|X|) \sum_{\mathbf{x} \in X} BCE(f_{\Theta}(\mathbf{z}_C, \mathbf{x})(1 - g_{\Phi}(\mathbf{z}_B, \mathbf{x})), o_R(\mathbf{x})). \quad (7)$$

The first argument to  $BCE$  in Equation (7) represents the expression for restoration shapes from Equation (2), with expressions for complete and break occupancy in terms of  $f_{\Theta}$  and  $g_{\Phi}$  substituted in. The second argument represents the ground truth restoration occupancy. We train  $f_{\Theta}$  and  $g_{\Phi}$  together to incorporate dependency of gradient descent update of complete parameters on break parameters and vice versa, based on the product rule with terms from Equations (4) and (7). Joint training enables learning of fractured and restoration shapes in terms of complete shapes and break surfaces.  $\mathcal{L}_{\text{reg}} = \|\mathbf{z}_C\|_2^2 + \|\mathbf{z}_B\|_2^2$  regularizes latent code estimation by imposing a zero-mean Gaussian prior on the latent codes. We set the weight  $\lambda_{\text{reg}}$  on  $\mathcal{L}_{\text{reg}}$  to be  $1e - 4$ .

### 3.2 Inference of Latent Codes

During inference, we estimate optimal latent codes  $\mathbf{z}_C$  and  $\mathbf{z}_B$  for point observations of occupancy  $o_F(\mathbf{x})$  from a novel input fractured shape  $F$ . With knowledge of the fractured shape occupancy, the inference loss is given as  $\mathcal{L}_{\text{inf}} = \mathcal{L}_F + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}$ . By itself, the loss does not prevent the break surface from being predicted outside or on the boundary of the complete shape. This may result in an empty restoration shape as shown in Figure 3(b). The loss also does not constrain the restoration shape from growing arbitrarily large. Gradient descent on the loss may generate a locally optimal latent code that yields a plausible complete shape, but a large restoration as shown in Figure 3(c). We introduce two penalty terms that encourage point occupancy values for the restoration that constrain its structure. The non-empty penalty term  $\mathcal{L}_{\text{ner}}$ , given as

$$\mathcal{L}_{\text{ner}} = -\log\left(\frac{1}{|X|}\sum_{\mathbf{x}\in X}f_{\Theta}(\mathbf{z}_C, \mathbf{x})(1 - g_{\Phi}(\mathbf{z}_B, \mathbf{x}))\right), \quad (8)$$

penalizes the mean restoration occupancy against being zero. The term encourages the complete shape to have a non-empty intersection with the break set on the restoration side of the break surface. The proximity loss,  $\mathcal{L}_{\text{prox}}$ , given as

$$\mathcal{L}_{\text{prox}} = -\log\left(1 - \frac{1}{|X|}\sum_{\mathbf{x}\in X}(f_{\Theta}(\mathbf{z}_C, \mathbf{x}) - o_F(\mathbf{x}))^2\right), \quad (9)$$

penalizes the network from predicting complete shapes that are not in close proximity to the fractured shape. The term discourages voluminous restorations. As shown in Figure 3(d), the negative log functions for  $\mathcal{L}_{\text{ner}}$  and  $\mathcal{L}_{\text{prox}}$  strongly penalize the mean occupancy from being too low or the mean complete-fractured occupancy distance from being too high. Using the non-empty and proximity penalties, we express the augmented inference loss  $\mathcal{L}_{\text{infaug}}$  as

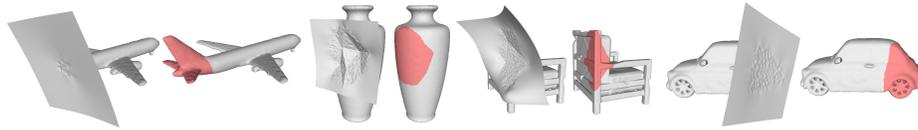
$$\mathcal{L}_{\text{infaug}} = \mathcal{L}_{\text{inf}} + \lambda_{\text{ner}}\mathcal{L}_{\text{ner}} + \lambda_{\text{prox}}\mathcal{L}_{\text{prox}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} \quad (10)$$

where  $\lambda_{\text{ner}}$  and  $\lambda_{\text{prox}}$  are weights on the non-empty and proximity penalties. During code estimation, we use  $\lambda_{\text{ner}} = 1e-5$ ,  $\lambda_{\text{prox}} = 5e-3$ , and  $\lambda_{\text{reg}} = 1e-4$ . We optimize  $\mathcal{L}_{\text{infaug}}$  to estimate the complete and break codes  $\mathbf{z}_C$  and  $\mathbf{z}_B$ . We use the estimated codes to reconstruct restoration occupancy values using Equation (2), and obtain the restoration shape as a 3D mesh using Marching Cubes [33].

## 4 Datasets and Data Preparation

We evaluate our work using 3D meshes from four datasets.

1. **ShapeNet.** We use meshes from 8 classes in the ShapeNet dataset [5]: airplanes, bottles, cars, chairs, jars, mugs, sofas, and tables. The classes have 1,534 to 5,614 shapes, with an average of 3,019 shapes. We train one network per class, and use an 80%/20% train/test split within each class.
2. **Google Scanned Objects Dataset.** The dataset [42] contains 1,032 digitally scanned common objects such as cups, bowls, plates, baskets, and shoes. We train a network with an 80%/20% train/test split of the dataset.



**Fig. 4.** Ground truth fractured shapes shown with fitted thin-plate splines (TPS) corresponding to the break surface, and with ground truth restoration shapes. Fractured shapes and TPS are shown in gray, restoration shapes are shown in red.

3. **QP Cultural Heritage Object Dataset.** The QP dataset [24] contains 317 meshes computer-modeled in the style of ancient Greek pottery. We use all models for testing using a network trained on ShapeNet jars.
4. **Real-World Fractured Mugs.** We perform in-house fractures of 4 real-world mugs, and scan the fractured mugs for testing. We use all mug models for testing using a network trained on ShapeNet mugs.

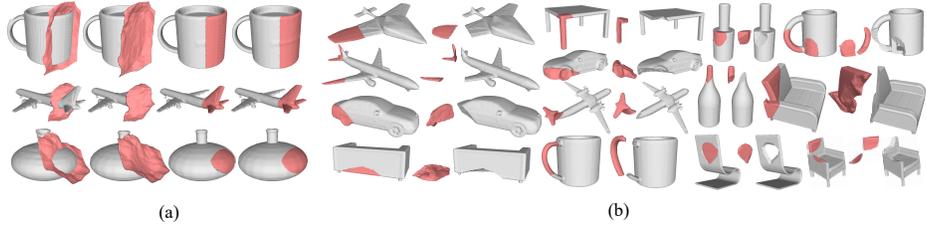
**Data Preparation.** We center meshes and scale them to lie within a unit cube. ShapeNet and QP models come pre-oriented to be consistent. Though meshes from the Google Scanned Objects dataset have a common ground plane, they are not oriented in a consistent direction. We augment the training set for Google Scanned Objects by randomly rotating meshes by  $90^\circ$  around the ground plane normal. We orient all real-world mugs to line up with ShapeNet mugs. We waterproof all meshes using the approach of Stutz and Geiger [48].

The ShapeNet, Google Scanned Objects, and QP datasets lack fractures. We synthetically fracture meshes in these datasets by repeatedly subtracting a randomized geometric primitive from each mesh. We adopt the fracture approach from Lamb et al. [27] to remove 5-20% of the mesh surface area. We show example fractured shapes, restorations, and break surfaces in Figure 4. The mugs, jars, and bottles classes from ShapeNet have fewer than 600 meshes. We fracture meshes belonging to these classes 10, 3, and 3 times respectively. We augment the mugs set by requiring that 3 fractures for each complete mesh only remove parts of the handle. We fracture meshes in the remaining ShapeNet classes and the QP and Google Scanned Objects datasets once.

We obtain point samples for the set  $X$  by randomly sampling a unit cube around the object and sampling on the surface of the mesh, as described in the supplementary. To generate the break surface for each training sample, we fit a thin-plate spline (TPS) [13] to the fracture vertices, such that the spline domain corresponds to the closest fitting plane to the fracture region vertices. We use the spline to partition sample points in the interior of the fractured and restoration meshes into two groups. We denote the side of the spline that contains the most fractured shape sample points as belonging to the break set  $B$ .

## 5 Results

**Metrics.** For evaluation we use the chamfer distance (CD), non-empty restoration percent (NE%), and non-fracture region error (NFRE). NFRE measures



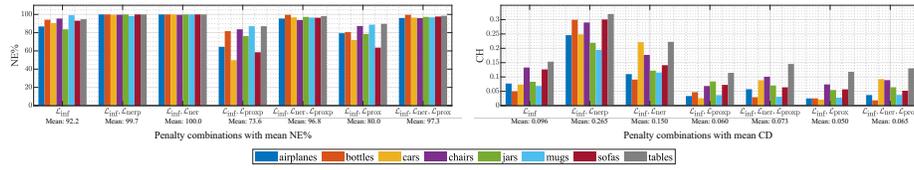
**Fig. 5.** (a) Predicted break surface and predicted complete shape, predicted break surface and fractured shape, predicted restoration shape and fractured shape, and ground truth fractured and restoration shape for three objects. (b) Predicted restoration shapes, shown with ground truth fractured shapes joined and opened to show the fracture surface. Restoration shapes and break surfaces in red, all other shapes in gray.

the presence of incorrect geometric protrusions on the restoration near the non-fracture region of the fractured shape. Bulk metrics, e.g. precision/recall using point occupancies over the whole shape, do not model thin protrusions which contain very few points. To capture protrusions using the NFRE, we sample  $n$  points on the surfaces of the predicted restoration, ground truth restoration, and non-fractured region of the fractured shape. We count protruding points on the predicted restoration shape as points that have a nearest neighbor in the non-fracture region of the fractured shape closer than  $\eta$  and a nearest neighbor in the ground truth restoration farther than  $\eta$ . We normalize the count by  $n$ . We use  $\eta = 0.02$  and  $n = 30,000$ .

## 5.1 Results using ShapeNet

Figure 5(a) shows DeepMend-generated complete shapes and predicted break surfaces, and restoration shapes joined to input fractured shapes using shapes from ShapeNet. The break surfaces predicted by DeepMend mimic the fracture region at the join, resulting in accurate connections between fractured and restoration shapes. Figure 5(b) shows restorations joined to corresponding ground truth fractured shapes and opened to show the fracture. DeepMend restoration shapes match closely to the fracture, and avoid surface artifacts that may otherwise prevent the restoration shape from being joined to the fractured shape. DeepMend regenerates complex missing geometry, such as the tail of the plane in Figure 5(b) and the car spoiler in Figure 1. It also restores multi-component fractures such as the center airplane, top right mug, and armchair in Figure 5(b). In contrast to symmetric approaches [17, 37], DeepMend repairs shapes with asymmetrical fractures such as the sofa, plane with broken tail, bottle, and chairs in Figure 5, and the car, mug, sofa, and airplane in Figure 1.

**Effect of Penalties on Restoration Shape During Inference.** We evaluate the impact of our augmented inference loss  $\mathcal{L}_{\text{infaug}}$  in Equation (10) versus adding no penalty terms, adding solely the non-empty term  $\mathcal{L}_{\text{ner}}$ , and adding solely the proximity term  $\mathcal{L}_{\text{prox}}$ . We also evaluate alternate penalties given as



**Fig. 6.** Percentage of non-empty restorations (NE%, left) and chamfer distance (CD, right) for our approach with various combinations of penalties for the inference loss.



**Fig. 7.** Predicted restoration shapes (red) shown with ground truth fractured shapes (gray) using using various combinations of penalty terms for the inference loss.

$$\mathcal{L}_{\text{nerp}} = -(1/|X|) \sum_{\mathbf{x} \in X} \log(f_{\Theta}(\mathbf{z}_C, \mathbf{x})(1 - g_{\Phi}(\mathbf{z}_B, \mathbf{x}))) \quad \text{and} \quad (11)$$

$$\mathcal{L}_{\text{proxp}} = (1/|X|) \sum_{\mathbf{x} \in X} BCE(f_{\Theta}(\mathbf{z}_C, \mathbf{x}), o_F(\mathbf{x})), \quad (12)$$

that penalize individual distance rather than mean distance.  $\mathcal{L}_{\text{nerp}}$  penalizes 0 values for  $R$ , and is the same as the non-zero penalty used in MendNet [25] restorations.  $\mathcal{L}_{\text{proxp}}$  encourages individual values in  $C$  to be similar to  $F$ .

We summarize results in the bar plots in Figure 6. Using no penalties predicts large restoration shapes, or generates multiple empty restorations as shown by the jar in Figure 7(a). Mean NE% and CD with  $\mathcal{L}_{\text{inf}}$  are 92.2% and 0.096. Including  $\mathcal{L}_{\text{nerp}}$  raises mean NE% to 99.7%. However, since the penalty is applied to individual points, restorations appear splayed out and non-smooth as shown by the jar in Figure 7(b). Mean CD is the highest with  $\mathcal{L}_{\text{nerp}}$  at 0.265. By penalizing mean occupancy,  $\mathcal{L}_{\text{ner}}$  remedies the splaying by keeping occupancy values concentrated, improves restoration quality, and lowers CD to a mean of 0.150. However, restorations may now appear bulkier as shown by the mug in Figure 7(c). Mean CD is higher than when no penalty term is used.

When comparing the effect of the proximity penalties on inference, we find that including  $\mathcal{L}_{\text{proxp}}$  improves mean CD to 0.060. However, it drops the percentage of non-empty restorations to 73.6%. The per-point penalty induces individual points to approach the fracture surface, causing sparse and non-smooth restorations as shown in Figure 7(d). Including the non-empty restoration term  $\mathcal{L}_{\text{ner}}$  improves NE% to 96.8% with minimal impact on CD. However, shapes are still non-smooth, as shown by the jar in Figure 7(e). By using the penalty on the mean complete-fractured proximity  $\mathcal{L}_{\text{prox}}$ , per-point distances remain concentrated and NE% is higher than with  $\mathcal{L}_{\text{proxp}}$ . Combining it with  $\mathcal{L}_{\text{ner}}$  improves NE% over solely using  $\mathcal{L}_{\text{prox}}$  from 80.0% to 97.3%, without compromising on the mean CD at 0.065. As shown by Figure 7(g), penalties on the mean occupancy and proximity values provides balanced, concentrated, and smooth restorations.

## 5.2 Comparing DeepMend to Existing Approaches and Baselines

We compare DeepMend restorations to MendNet [25] and 3D-ORGAN [22], the only existing approaches to restore fractured shapes, and to two baselines.

**MendNet.** MendNet takes a fractured shape as input and predicts a restoration shape directly using the occupancy function. We train MendNet on fractured shapes from ShapeNet. As recommended by the authors, we use a latent code of size 256 for bottles, jars, and mugs, and of size 400 for all other classes.

**3D-ORGAN.** 3D-ORGAN takes a voxelized fractured shape at a resolution of  $32^3$  as input and predicts a corresponding complete voxelized shape. We obtain the restoration shape for 3D-ORGAN as the element-wise difference between the predicted complete voxels and the input fractured voxels. We train 3D-ORGAN on fractured shapes from ShapeNet. During training, we generate random fractured shapes by removing voxel regions from each complete shape according to the original implementation. During testing, we input voxelized fractured shapes generated using the approach described in Section 4. As recommended by the authors, we use a two-step approach to predict complete shapes by feeding the output of the first iteration to 3D-ORGAN again for a second iteration.

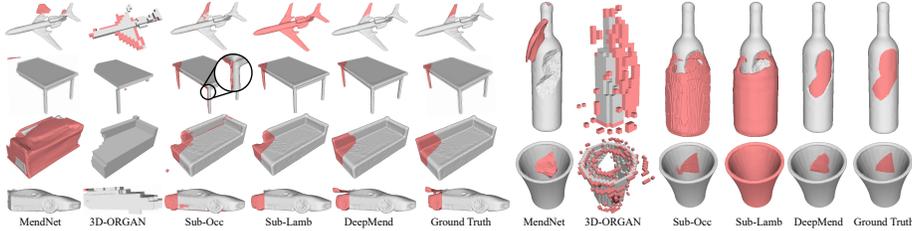
**Baselines of Performing Subtraction from Complete Shape.** We adapt the partial shape completion approach of DeepSDF [38] to generate restoration shapes using subtraction. We generate a partial shape from the fractured shape by removing fracture region points detected by a fracture/non-fracture classifier. We train a PointNet [41] classifier to classify points as fracture versus non-fracture. The classifier provides a test accuracy of 81.3%. We remove detected fracture points to generate the partial shape. We train DeepSDF on complete shapes for the 8 ShapeNet classes studied in this work. We use DeepSDF to complete the partial shape using the shape completion method discussed by the authors. We use two approaches for subtraction as baselines. For the first approach, **Sub-Occ**, we convert SDF values for the input fractured shape and DeepSDF-predicted complete shape into occupancy values. We take the difference of the complete and fractured occupancy, and extract the 0-level isosurface. To remove artifacts, we discard closed surfaces with a volume less than  $\eta = 0.01$ . If this step removes all closed surfaces, we retain the largest surface. For the second method, **Sub-Lamb**, we use Lamb et al. [26] to perform subtraction. Lamb et al. additionally requires a complete counterpart to be provided as input. We use the complete shape from DeepSDF as the complete counterpart. We repair self-intersections at the fracture-restoration join using MeshFix [2].

Table 1 summarizes results of CD and NFRE using MendNet, 3D-ORGAN, Sub-Occ, Sub-Lamb, and DeepMend. 100% of restorations are generated by Sub-Occ and Sub-Lamb for all classes, and by 3D-ORGAN for all classes except cars and mugs. We report results over all shapes where DeepMend and MendNet return non-empty restorations. For 3D-ORGAN, we exclude metrics for cars and mugs, as 3D-ORGAN only produces restorations for 6 out of 349 or 1.7% of mugs and 159 out of 661 or 24.1% of cars. Figure 8 shows qualitative results.

DeepMend shows state-of-the-art results compared to MendNet, 3D-ORGAN, and the baselines in terms of overall CD and NFRE and per-class NFRE. Deep-

**Table 1.** Chamfer (CD) distance and NFRE for MendNet, 3D-ORGAN, Sub-Occ, Sub-Lamb, and DeepMend. For 3D-ORGAN, we do not report results for cars and mugs, as it only restores 21.7% cars and 1.8% mugs. Overall mean values are provided over reported classes. Bold values correspond to the lowest value within a class.

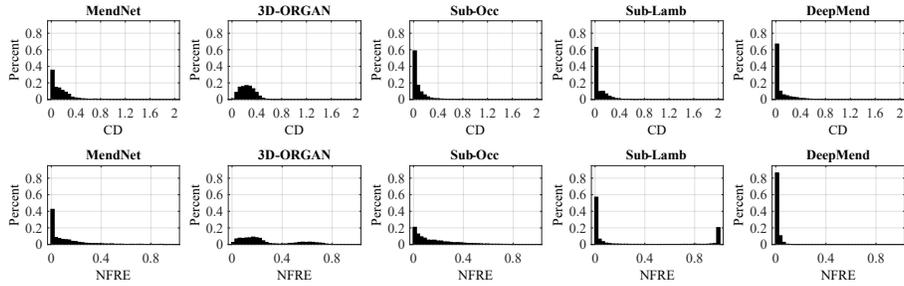
	Metric	airplanes	bottles	cars	chairs	jars	mugs	sofas	tables	Mean
MendNet	CD	0.091	0.080	0.025	0.171	0.129	0.109	0.190	0.208	0.126
	NFRE	0.070	0.045	<b>0.017</b>	0.143	0.028	<b>0.008</b>	0.085	0.203	0.075
3D-ORGAN	CD	0.173	0.146	-	0.184	0.262	-	0.320	0.333	0.237
	NFRE	0.192	0.070	-	0.588	0.041	-	0.200	0.138	0.205
Sub-Occ	CD	0.050	0.041	<b>0.024</b>	0.112	0.119	<b>0.035</b>	0.066	0.122	0.071
	NFRE	0.099	0.076	0.142	0.262	0.183	0.070	0.170	0.175	0.147
Sub-Lamb	CD	0.075	0.039	0.050	<b>0.086</b>	0.082	0.100	<b>0.053</b>	<b>0.093</b>	0.072
	NFRE	0.302	0.120	0.272	0.330	0.289	0.452	0.192	0.204	0.270
DeepMend (Ours)	CD	<b>0.037</b>	<b>0.022</b>	0.108	0.088	<b>0.065</b>	<b>0.035</b>	0.057	0.129	<b>0.068</b>
	NFRE	<b>0.009</b>	<b>0.012</b>	<b>0.017</b>	<b>0.009</b>	<b>0.007</b>	<b>0.008</b>	<b>0.012</b>	<b>0.012</b>	<b>0.011</b>



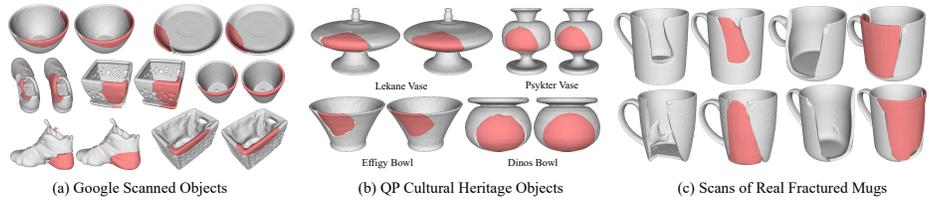
**Fig. 8.** Pictorial results of restorations using MendNet, 3D-ORGAN, baselines Sub-Occ and Sub-Lamb, and DeepMend, together with ground truth restorations.

Mend also shows lower CD for all classes compared to 3D-ORGAN, 7 out of 8 classes compared to MendNet, 5 out of 8 classes compared to Sub-Occ, and 4 out of 8 classes compared to Sub-Lamb. MendNet predicts smooth restoration shapes that may not join completely to the fractured shape. While MendNet generates lower CD than DeepMend for cars common to both methods, MendNet only restores 57.9% cars, and fails for 42.1% cars as shown in Figure 8. On average, MendNet generates 84.8% restorations compared to 97.3% generated by DeepMend. As shown in Figure 8, 3D-ORGAN predicts small restoration shapes, e.g. the table, sofa, and car. The histogram on the left of Figure 9 shows that MendNet and 3D-ORGAN predict 36.3% and 1.0% restorations with a chamfer distance less than 0.05 respectively, compared to 67.9% with DeepMend.

Restoration shapes generated using Sub-Occ exhibit artifacts on the surface of the fractured shape as shown in Figure 8. The histogram of the NFRE values for Sub-Occ in Figure 9 shows that 21.8% of restorations have NFRE lower than 0.025, as opposed to 87.8% by DeepMend. The fracture classifier may not reliably remove the entire fracture region to create a partial shape that is a precise subset of the complete shape. As such, Sub-Occ generates restorations that exhibit elements of the fracture, e.g., the sofa, car, and bottle in Figure 8. As shown in Figure 9, Sub-Lamb is effective at removing artifacts for some objects, as 58.0% of restoration shapes have NFRE lower than 0.025. However, Sub-Lamb



**Fig. 9.** Histograms of chamfer distance (CD, top) and non-fracture region error (NFRE, bottom) for MendNet, 3D-ORGAN, Sub-Occ, Sub-Lamb, and DeepMend.



**Fig. 10.** Results using (a) synthetic breaks on Greek pottery from QP [24], (b) synthetic fractures on 3D scans from Google Scanned Objects [42], and (c) 3D scans for real-world fractured mugs. Ground truth restoration shapes shown on right when applicable.

may incorrectly mark the exterior region of the fractured shape as belonging to the fracture, causing the entirety of the fractured shape to be merged with the restoration, e.g., for the airplane and jar in Figure 8. Figure 9 shows that for Sub-Lamb, 20.4% of restoration shapes have a high NFRE between 0.975 and 1.

### 5.3 Results with Google Scanned Objects, QP, and Fractured Mugs

We show results of training and testing DeepMend with the Google Scanned Objects dataset in Figure 10(a). We obtain a chamfer distance of 0.112. DeepMend generates closely fitting restorations for objects that are prone to fracture such as the plate, the pot, and the bowls in Figure 10(a), and reasonable restoration shapes for objects with high intra-class variety such as shoes.

We demonstrate the generalizability of our approach to novel datasets by using ShapeNet-trained jars and mugs networks to restore synthetically fractured shapes for objects from the QP dataset, shown in Figure 10(b), and for 3D scans of 4 real-world fractured mugs, shown in Figure 10(c). We achieve a mean chamfer distance of 0.047 for QP objects. DeepMend generates plausible restorations for shapes that resemble modern bowls, such as the effigy bowl in Figure 10(b), and for uncommon shapes, such as the psykter vase in Figure 10(b).

The restoration process for real-world mugs is particularly challenging. For synthetic breaks, the fractured and non-fracture regions in synthetic breaks have

a clear edge. In real-world mugs, the break structure is far more complex with sharp curvature and smoothed out edges as seen in Figure 10(c). Scanning limitations may cause the fracture surface geometry to be less precisely captured in comparison to the roughness of simulated fractures. Despite the challenges, our approach shows the capability to reconstruct the restoration by generating break surfaces that approach the fracture surface of real broken mugs.

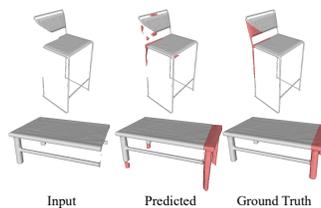
## 6 Conclusion

We provide DeepMend, an approach to automatically restore fractured shapes by learning to represent complete shapes, break surfaces, and their interplay. We contribute penalty functions for inference that penalize mean occupancy values against being too high or low, thereby ensuring well-structured restorations. DeepMend does not require ground truth knowledge of the fracture region, making it amenable for rapid repair.

One **limitation** of our work is that it may predict multiple unnecessary components, especially for thin structures, e.g., the chair in Figure 11, which are often problematic for learned volumetric functions. Since the components are on the restoration side of the surface, NFRE remains lower than with the baseline methods and the components do not effect joinability. In many cases, e.g. the table in Figure 11, multiple components yield plausible restorations. For the table class, these components together with the high intra-class variance contribute to an increased CD, as shown in Table 1. Multiple component prediction may arise as the break surface can adopt arbitrarily complex geometries during inference. As part of future work, data-driven priors can be incorporated on the structure of break surfaces. Future work can use datasets to learn prior probability distributions of occupancy of 3D objects. The learned shape representation can be strengthened with structural constraints, e.g., symmetries and planarity.

While not explored in this work, DeepMend may be combined with approaches to perform fractured object reassembly [6, 31, 52] to enable automatic restoration of fractured objects with multiple components. Novel scanned datasets of objects with diverse materials and diverse damage, e.g., chipping, shearing, splintering, and ductile versus brittle fractures can benefit the study of the damage process and impact on fracture surface geometry. Our work contributes a geometric foundation for the study of fractured shape repair using closed 3D surfaces. The work opens the scope for future research on automated shape repair using depth and color images to facilitate democratization of the repair process.

**Acknowledgments.** This work is supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant 1945954. We thank the Clarkson University Office of Information Technology for the use of the ACRES GPU server sponsored by NSF Grant 1925596.



**Fig. 11.** Restorations with multiple components.

## References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International Conference On Machine Learning. PMLR (2018)
2. Attene, M.: A lightweight approach to repairing digitized polygon meshes. *The visual computer* **26**(11), 1393–1406 (2010)
3. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks. arXiv preprint arXiv:1608.04236 **1**(1), 1–9 (2016)
4. Chabra, R., Lenssen, J.E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., Newcombe, R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In: ECCV. pp. 608–625. Springer, Berlin, Germany (2020)
5. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)
6. Chen, Y.C., Li, H., Turpin, D., Jacobson, A., Garg, A.: Neural shape mating: Self-supervised object assembly with adversarial shape priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12724–12733 (2022)
7. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proc. CVPR. pp. 5939–5948. IEEE, Piscataway, NJ (2019)
8. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: Proc. CVPR. pp. 6970–6981. IEEE, Piscataway, NJ (2020)
9. Dai, A., Diller, C., Nießner, M.: Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 849–858 (2020)
10. Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., Nießner, M.: Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In: Proc. CVPR. pp. 4578–4587. IEEE, Piscataway, NJ (2018)
11. Dai, A., Ruizhongtai Qi, C., Nießner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5868–5877 (2017)
12. Duan, Y., Zhu, H., Wang, H., Yi, L., Nevatia, R., Guibas, L.J.: Curriculum deepsdf. In: European Conference on Computer Vision. pp. 51–67. Springer (2020)
13. Duchon, J.: Splines minimizing rotation-invariant semi-norms in sobolev spaces. In: Constructive theory of functions of several variables, pp. 85–100. Springer (1977)
14. Duggal, S., Wang, Z., Ma, W.C., Manivasagam, S., Liang, J., Wang, S., Urtasun, R.: Mending neural implicit modeling for 3d vehicle reconstruction in the wild. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1900–1909 (2022)
15. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local deep implicit functions for 3d shape. In: Proc. CVPR. pp. 4857–4866. IEEE, Piscataway, NJ (2020)
16. Genova, K., Cole, F., Vlastic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. In: Proc. CVPR. pp. 7154–7164. IEEE, Piscataway, NJ (2019)

17. Gregor, R., Sipiran, I., Papaioannou, G., Schreck, T., Andreadis, A., Mavridis, P.: Towards automated 3d reconstruction of defective cultural heritage objects. In: GCH. pp. 135–144. EUROGRAPHICS, Geneva, Switzerland (2014)
18. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proc. CVPR. pp. 216–224. IEEE, Piscataway, NJ (2018)
19. Gupta, M.M., Qi, J.: Theory of t-norms and fuzzy inference methods. *Fuzzy sets and systems* **40**(3), 431–450 (1991)
20. Han, X., Li, Z., Huang, H., Kalogerakis, E., Yu, Y.: High-resolution shape completion using deep neural networks for global structure and local geometry inference. In: Proceedings of the IEEE international conference on computer vision. pp. 85–93 (2017)
21. Hao, Z., Averbuch-Elor, H., Snavely, N., Belongie, S.: Dualsdf: Semantic shape manipulation using a two-level representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7631–7641 (2020)
22. Hermoza, R., Sipiran, I.: 3d reconstruction of incomplete archaeological objects using a generative adversarial network. In: Proceedings of Computer Graphics International, pp. 5–11. ACM, New York, NY (2018)
23. Jia, M., Kyan, M.: Learning occupancy function from point clouds for surface reconstruction. *arXiv preprint arXiv:2010.11378* **1**(1), 1–11 (2020)
24. Koutsoudis, A., Pavlidis, G., Arnaoutoglou, F., Tsiafakis, D., Chamzas, C.: Qp: A tool for generating 3d models of ancient greek pottery. *Journal of Cultural Heritage* **10**(2), 281–295 (2009)
25. Lamb, N., Banerjee, S., Banerjee, N.K.: MendNet: Restoration of Fractured Shapes Using Learned Occupancy Functions. *Computer Graphics Forum* (2022)
26. Lamb, N., Banerjee, S., Banerjee, N.K.: Automated reconstruction of smoothly joining 3d printed restorations to fix broken objects. In: Proc. SCF. pp. 1–12. ACM, New York, NY (2019)
27. Lamb, N., Wiederhold, N., Lamb, B., Banerjee, S., Banerjee, N.K.: Using learned visual and geometric features to retrieve complete 3d proxies for broken objects. In: Proc. SCF. pp. 1–15. ACM, New York, NY (2021)
28. Liao, Y., Donne, S., Geiger, A.: Deep marching cubes: Learning explicit surface representations. In: Proc. CVPR. pp. 2916–2925. IEEE, Piscataway, NJ (2018)
29. Lin, C.H., Wang, C., Lucey, S.: Sdf-srn: Learning signed distance 3d object reconstruction from static images. *arXiv preprint arXiv:2010.10505* **1**(1), 1–17 (2020)
30. Lionar, S., Emtsev, D., Svilarkovic, D., Peng, S.: Dynamic plane convolutional occupancy networks. In: Proc.WACV. pp. 1829–1838. IEEE, Piscataway, NJ (2021)
31. Liu, B., Wang, M., Niu, X., Wang, S., Zhang, S., Zhang, J.: A fragment fracture surface segmentation method based on learning of local geometric features on margins used for automatic utensil reassembly. *Computer-Aided Design* **132**, 102963 (2021)
32. Liu, M., Sheng, L., Yang, S., Shao, J., Hu, S.M.: Morphing and sampling network for dense point cloud completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11596–11603. AAAI, Menlo Park, CA (2020)
33. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics* **21**(4), 163–169 (1987)
34. Ma, B., Han, Z., Liu, Y.S., Zwicker, M.: Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *arXiv preprint arXiv:2011.13495* **1**(1), 1–12 (2020)

35. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proc. CVPR. pp. 4460–4470. IEEE, Piscataway, NJ (2019)
36. Pan, L., Chen, X., Cai, Z., Zhang, J., Zhao, H., Yi, S., Liu, Z.: Variational relational point completion network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8524–8533 (2021)
37. Papaioannou, G., Schreck, T., Andreadis, A., Mavridis, P., Gregor, R., Sipiran, I., Vardis, K.: From reassembly to object completion: A complete systems pipeline. *Journal on Computing and Cultural Heritage* **10**(2), 1–22 (2017)
38. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proc. CVPR. pp. 165–174. IEEE, Piscataway, NJ (2019)
39. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 523–540. Springer, Berlin, Germany (2020)
40. Poursaeed, O., Fisher, M., Aigerman, N., Kim, V.G.: Coupling explicit and implicit surface representations for generative 3d modeling. In: European Conference on Computer Vision. pp. 667–683. Springer (2020)
41. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proc. CVPR. pp. 652–660. IEEE, Piscataway, NJ (2017)
42. Research, G.: Google scanned objects. [https://fuel.ignitionrobotics.org/1.0/Google Research/fuel/collections/Google Scanned Objects](https://fuel.ignitionrobotics.org/1.0/GoogleResearch/fuel/collections/Google%20Scanned%20Objects) (August 2021)
43. Sarmad, M., Lee, H.J., Kim, Y.M.: Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In: Proc. CVPR. pp. 5898–5907. IEEE, Piscataway, NJ (2019)
44. Sharma, A., Grau, O., Fritz, M.: Vconv-dae: Deep volumetric shape learning without object labels. In: ECCV. pp. 236–250. Springer, Berlin, Germany (2016)
45. Sitzmann, V., Chan, E.R., Tucker, R., Snavely, N., Wetzstein, G.: Metasdf: Meta-learning signed distance functions. *arXiv preprint arXiv:2006.09662* **1**(1), 1–17 (2020)
46. Smith, E.J., Meger, D.: Improved adversarial systems for 3d object generation and reconstruction. In: Conference on Robot Learning. pp. 87–96. PMLR, Cambridge, UK (2017)
47. Son, H., Kim, Y.M.: Saum: Symmetry-aware upsampling module for consistent point cloud completion. In: Proc. ACCV. pp. 1–17. Springer, Berlin, Germany (2020)
48. Stutz, D., Geiger, A.: Learning 3d shape completion under weak supervision. *CoRR* **abs/1805.07290** (2018), <http://arxiv.org/abs/1805.07290>
49. Sulzer, R., Landrieu, L., Boulch, A., Marlet, R., Vallet, B.: Deep surface reconstruction from point clouds with visibility information. *arXiv preprint arXiv:2202.01810* (2022)
50. Tang, J., Lei, J., Xu, D., Ma, F., Jia, K., Zhang, L.: Sign-agnostic conet: Learning implicit surface reconstructions by sign-agnostic optimization of convolutional occupancy networks. *arXiv preprint arXiv:2105.03582* **1**(1), 1–16 (2021)
51. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Stoll, C., Theobalt, C.: Patchnets: Patch-based generalizable deep implicit 3d shape representations. In: Proc. ECCV. pp. 293–309. Springer, Berlin, Germany (2020)

52. Wang, H., Zang, Y., Liang, F., Dong, Z., Fan, H., Yang, B.: A probabilistic method for fractured cultural relics automatic reassembly. *Journal on Computing and Cultural Heritage (JOCCH)* **14**(1), 1–25 (2021)
53. Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: *Proc. NeurIPS*. pp. 82–90. Neural Information Processing Systems, San Diego, CA (2016)
54. Xiao, Y., Xu, J., Gao, S.: Taylorimnet for fast 3d shape reconstruction based on implicit surface function. *arXiv preprint arXiv:2201.06845* (2022)
55. Xu, Y., Fan, T., Yuan, Y., Singh, G.: Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry. In: *European Conference on Computer Vision*. pp. 248–263. Springer (2020)
56. Yan, S., Yang, Z., Li, H., Guan, L., Kang, H., Hua, G., Huang, Q.: Implicit auto-encoder for point cloud self-supervised representation learning. *arXiv preprint arXiv:2201.00785* (2022)
57. Yan, X., Lin, L., Mitra, N.J., Lischinski, D., Cohen-Or, D., Huang, H.: Shapeformer: Transformer-based shape completion via sparse representation. *arXiv preprint arXiv:2201.10326* (2022)
58. Yang, M., Wen, Y., Chen, W., Chen, Y., Jia, K.: Deep optimized priors for 3d shape modeling and reconstruction. In: *Proc. CVPR*. pp. 3269–3278. IEEE, Piscataway, NJ (2021)
59. Yi, L., Gong, B., Funkhouser, T.: Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 15363–15373 (2021)
60. Yu, Q., Yang, C., Wei, H.: Part-wise atlasnet for 3d point cloud reconstruction from a single image. *Knowledge-Based Systems* p. 108395 (2022)
61. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: *2018 International Conference on 3D Vision (3DV)*. pp. 728–737. IEEE (2018)
62. Zheng, Z., Yu, T., Dai, Q., Liu, Y.: Deep implicit templates for 3d shape representation. In: *Proc. CVPR*. pp. 1429–1439. IEEE, Piscataway, NJ (2021)